# Entorn Multiagent de Poker Texas Hold'em per a l'Aprenentatge per Reforç

Joaquin Flores
Grau de Matemàtiques Computacionals i Analitica de Dades

10 de juny de 2025

# Índex

# 1 Abstract

*Limit Hold'em 4-max* is a benchmark for imperfect-information, multi-agent decision making. Each player must act under private information, shifting probabilities and a constantly evolving strategic landscape. While systems such as DeepStack and Libratus have shown that combining deep reinforcement learning with search can beat top professionals in heads-up *no-limit*, the community still lacks an open, replicable environment for the six-handed *limit* variant. This project presents an **open-source PettingZoo environment** that faithfully models betting rules, positional dynamics and showdown resolution for six players with fixed betting structure. The simulator is paired with utilities for large-scale self-play, hand-history logging and metric collection, enabling rapid experimentation with any DRL library.

We benchmark two representative algorithms DQN and APPO across millions of simulated hands. Evaluation focuses on convergence speed, exploitability via counterfactual-regret analysis and head-to-head win-rates against random opponents. Early results indicate that actor–critic methods with centralised critics are markedly more sample-efficient and less exploitable than value-based baselines, yet still fall short of theoretical Nash equilibria.

Limit Hold'em 4-max embodies the challenges of imperfect-information multi-agent learning: private cards, stochastic dynamics and strategic deception. We present an *open-source PettingZoo environment* that faithfully simulates four-handed Limit Hold'em and exposes a unified API for single- and multi-agent DRL. Benchmarking DQN and APPO.

# 2 Introducció

En els últims anys he anat veient com l'Aprenentatge per Reforç Profund (DRL) s'ha convertit en una eina molt potent per aprendre comportaments complexos sense necessitat de supervisió humana directa. Tot i així, els èxits que més han sonat com AlphaGo, AlphaZero, MuZero o els agents d'Atari, s'han donat en escenaris *d'informació perfecta*. Quan passem a jocs d'informació incompleta la cosa es complica: hi ha incertesa sobre l'estat real de la partida, cal mantenir creences probabilístiques i, a més, ens enfrontem a rivals racionals que també s'adapten.

Per això he escollit el **Texas Hold'em Limit 4-max** com a laboratori de proves. El format limit, on les apostes estan discretitzades en múltiples del big blind, redueix l'espai d'accions i fa més assumible l'anàlisi teòrica, però sense perdre la profunditat estratègica del joc. Tot i els avenços de DeepStack, Libratus o Pluribus en variants heads-up i no-limit, el codi i els entorns que fan servir no són oberts o són difícils de reproduir. Davant d'aquest panorama, el propòsit principal del meu Treball de Fi de Grau (TFG) és construir un entorn obert i extensible per a Limit Hold'em 4-max basat en la interfície estàndard *PettingZoo*. Aquesta decisió garanteix compatibilitat immediata amb biblioteques habituals de DRL (Stable-Baselines3, RLlib, CleanRL, etc.) i facilita la integració d'observadors, enregistraments de mans i panells de monitoratge.

Ara bé, el projecte no es limita a construir l'entorn. També vull mesurar fins on poden arribar diversos algoritmes de DRL multiagent a l'hora d'acostar-se a l'equilibri en una taula de quatre jugadors amb informació incompleta. Per fer-ho, faré competir enfocaments *off-policy* basats en valor, com Independent DQN, amb mètodes de la família APPO i analitzaré en quines situacions cadascun s'hi apropa més.

Tota la implementació desenvolupada durant aquest treball incloent-hi el codi de l'entorn, els entrenaments i els scripts d'anàlisi està disponible públicament al següent repositori GitHub: https://github.com/JOAKOF/PokerEntorn.

# 3 Objectius

L'objectiu principal d'aquest treball és dissenyar, implementar i analitzar un entorn de pòquer Texas Hold'em amb límit fix en modalitat 4-max, orientat a l'estudi de tècniques d'aprenentatge per reforç multiagent (MARL) en escenaris d'informació incompleta. A diferència d'altres jocs plenament observables, el pòquer presenta un desafiament únic per a la intel·ligència artificial: els agents han de prendre decisions estratègiques basant-se en informació parcial, múltiples rondes d'interacció i un entorn altament estocàstic.

Amb aquest propòsit, el projecte es divideix en tres grans blocs. En primer lloc, es pretén establir una base teòrica sòlida que permeti seleccionar algoritmes i mètriques adequades. En segon lloc, es desenvoluparà un entorn personalitzat compatible amb la llibreria PettingZoo per simular partides 4-max de pòquer amb límit. Finalment, s'entrenaran i compararan diversos agents, amb l'objectiu d'avaluar-ne el rendiment i la capacitat d'adaptació en un entorn competitiu i parcialment observable.

Els objectius concrets del treball són els següents:

1. **Assimilar els fonaments teòrics de l'Aprenentatge per Reforç en jocs d'informació incompleta**
   - Analitzar algoritmes basats en valor, política i actor–critic, i seleccionar candidats per al benchmarking.
   - Revisar MDP, POMDP, MARL i teoria de jocs aplicada al pòquer Limit 4-max.
   - Estudiar mètriques de rendiment apropiades (exploitability, win-rate, regret, ...) i establir els criteris finals.
2. **Dissenyar i implementar l'entorn Limit Hold'em 4-max sobre PettingZoo**
   - Definir espais d'observacions, accions i recompenses adherits a l'API de PettingZoo.
   - Implementar la lògica completa del joc: baralla, rondes d'aposta de límit fix, gestió del pot i showdown.
   - Permetre paràmetres configurables (blinds, estructura de bets, stacks inicials) per a experiments variats.
   - Desenvolupar eines de logging i reproducció ( *hand history*, visualitzadors) per a anàlisi qualitatiu.
3. **Entrenar agents i comparar algoritmes de MARL**
   - Implementar Independent DQN i APPO.
   - Estudiar l'estil de joc i l'agressivitat (raise/all-in vs call) al llarg de l'entrenament
   - Comparació i evaluacio vs agents randoms

# 4 Poker Texas Hold'em com a joc d'informació incompleta

El Texas Hold'em és una variant de pòquer de cartes comunitàries i *informació incompleta* que s'ha convertit en la més popular tant en casinos com en competició professional. Es tracta d'un joc **competitiu** per excel·lència: diversos jugadors competeixen per guanyar els pots (apostes acumulades) basant-se en combinacions de cartes privades i cartes comunes a la taula, amb la incertesa de no conèixer les cartes dels oponents. En aquesta secció s'explica el funcionament bàsic del joc i es discutiran conceptes estratègics clau: l'òptim teòric del joc sota la teoria de jocs, les *hand ranges* (rang de mans jugables) segons context, i la terminologia i estratègies emprades en partides de pòquer, incloent-hi situacions de torneig.

## 4.1 Regles i dinàmica del joc

En Texas Hold'em, cada **mà (hand)** comença amb cada jugador rebent dues cartes privades (cartes tapades que només veu cadascú). El joc es juga amb una baralla estàndard de 52 cartes, i normalment participen entre 2 i 10 jugadors. Abans de repartir les cartes, dos jugadors han de posar apostes obligatòries anomenades *blinds*: el *small blind* (cec petit) i el *big blind* (cec gran), situats immediatament a l'esquerra del *dealer* (attemptador) i que serveixen per assegurar acció en cada mà. El big blind sol ser el doble del small blind. Un cop posats els blinds, es reparteixen dues cartes privades a cada jugador.

El joc progressa en diverses **fases de aposta**:

- **Pre-flop:** Amb només les cartes privades repartides, comença la primera ronda d'apostes. El jugador immediatament a l'esquerra del big blind (anomenat *under the gun*) és el primer a parlar. Cada jugador en torn pot **foldejar** (retirar-se, abandonant la mà i perdent les apostes posades), **veure** (call, igualar l'aposta actual més alta a la taula) o **pujar** (raise, augmentar l'aposta). En aquesta ronda inicial, l'aposta mínima a igualar és el valor del big blind (si cap jugador puja més, això es considera "veure la cega"). La ronda continua en cercle fins que tots els jugadors actius hagin apostat el mateix import. Si en qualsevol moment tots els jugadors excepte un es retiren, el jugador restant guanya la mà immediatament (no es mostren cartes). Si almenys dos jugadors arriben a igualar totes les apostes, es passa a la següent fase.

- **Flop:** Es col·loquen **tres cartes comunitàries** descobertes al centre de la taula (el *flop*). Aquestes cartes les pot utilitzar qualsevol jugador juntament amb les seves cartes privades per formar una mà de cinc cartes. Després de revelar el flop, té lloc una segona ronda d'apostes, començant pel primer jugador actiu a l'esquerra del *dealer*. Ara els jugadors també tenen l'opció de **passar** (*check*) si no hi ha cap aposta prèvia en la ronda, la qual cosa vol dir cedir el torn al següent jugador sense apostar però mantenint-se a la mà. Si algun jugador aposta, els altres han de almenys igualar aquesta aposta per continuar.

- **Turn:** Es revela una quarta carta comunitària (anomenada *turn* o *fourth street*). Segueix una altra ronda d'apostes similar a la del flop (amb els jugadors actius restant).

- **River:** Es revela la cinquena i última carta comunitària (el *river*). Es fa la ronda final d'apostes, de nou amb la mateixa mecànica. Després d'aquesta ronda, si encara hi ha més d'un jugador actiu, es passa al **showdown**.

Al *showdown*, els jugadors restants mostren les seves cartes privades. Cadascun forma la millor **mà de cinc cartes** possible combinant les seves dues privades amb les cinc comunitàries (o fins i tot pot usar només les comunitàries si això forma la millor mà). El jugador amb el millor **ranking de mà de pòquer** guanya el pot acumulat d'apostes. Les categories de mans de pòquer, de més forta a més dèbil, són: escala de color (royal flush és l'escala de color més alta possible), pòquer (quatre iguals), full, color (flush), escala, trio, doble parella, parella, i carta alta. Aquestes categories són estàndard i determinades per combinacions de cartes, i defineixen qui guanya en mostrar les cartes (Cornell University, Department of Mathematics, 2006). Si hi ha un empat exacte en valor de mà, el pot es reparteix.

En resum, una mà de Texas Hold'em té **quatre rondes d'aposta** (pre-flop, flop, turn, river) intercalades amb la revelació de cartes comunitàries, i pot acabar abans si tothom menys un es retira. Cada jugador sempre disposa de les accions bàsices fold, call (o check si no hi ha aposta) i bet/raise en el seu torn, tot i que la quantitat i possibilitat de raisear depèn de la situació (en

No-Limit Hold'em qualsevol pujada pot ser de qualsevol import per sobre del mínim fins a totes les seves fisches, mentre que en Limit Hold'em les apostes estan quantitzades i limitades en nombre). La descripció formal d'aquest joc és la d'un **joc d'extensió amb informació imperfecta**: hi ha nodes de decisió per a cada jugador i nodes d'atzar (repartiment de cartes), i els jugadors no poden distingir alguns estats (per exemple, abans del showdown no saben quines cartes té l'adversari, de manera que per a un jugador diversos estats del món possibles són **informacionalment equivalents**). El Texas Hold'em, per tant, introdueix incertesa tant per la baralla (aleatorietat en cartes repartides) com per la informació oculta dels oponents.

## 4.2 Terminologia i estratègies en partides competitives

En el pòquer de nivell alt s'utilitza un argot específic per descriure accions i situacions. A més, l'estratègia òptima depèn de variables com la posició i la fase del torneig. A continuació, definim alguns termes importants i conceptes d'estratègia:

- **Glossari essencial:**
  - **Blinds (SB/BB)** – Apostes obligatòries.
  - **Bet / Raise** – Aposta inicial o pujada.
  - **3-bet / 4-bet** – Re-pujades pre-flop.
  - **Call / Check** – Igualar / passar.
  - **Fold** – Abandonar la mà.
  - **All-in** – Apostar tot el *stack*.
  - **Pot** – Fitxes acumulades.
  - **Pot-odds** – Ràtio entre aposta i pot.
  - **Implied odds** – Valor afegit futur.
  - **Equity** – Probabilitat de guanyar.
  - **Outs** – Cartes que milloren la mà.
  - **Kicker** – Carta desempatadora.
  - **Nuts** – Millor mà possible.
  - **C-bet** – Aposta de continuació.
  - **Bluff / Semi-bluff** – Aposta sense mà / amb projecte.
  - **Fold equity** – Probabilitat de fer fold.
  - **Range** – Conjunt de mans possibles.
  - **GTO** – Estratègia no explotable.
  - **SPR** – Relació *stack/pot*.
  - **Showdown** – Es mostren cartes.
- **Apostes i raises:** Quan encara no hi ha apostes en una ronda i un jugador posa fitxes, això és un **bet** (apostar). Si ja hi ha una aposta i un altre jugador la augmenta, això és un **raise** (pujar). Es parla de **3-bet** per referir-se a una *re-raise* (repujada) abans del flop que seria la tercera ronda d'aposta: per exemple, un jugador obre amb un raise (2-bet comptant blinds com 1-bet), un adversari fa una 3-bet (segona pujada), etc. Una **4-bet** seria tornar a pujar sobre la 3-bet, i així successivament. En general, en partides no-limit, una 3-bet preflop denota molta força (sovint AA-TT, AK, bluffs ocasionals), una 4-bet encara més força (KK+ normalment, i algun bluff molt seleccionat).
- **Posicions a la Taula:** En una partida de pòquer, la posició des d'on actua cada jugador té un impacte clau en l'estratègia. Els primers que han de parlar (posicions inicials") estan en desavantatge perquè han de prendre decisions sense saber què faran els altres, mentre que els últims (com el *button* o dealer) tenen més informació i poden jugar un rang de mans molt més ampli. En una taula completa (9 o 10 jugadors), aquestes posicions es divideixen en UTG, MP, HJ, CO i BTN, entre d'altres, però en formats més reduïts com el **4-max**, això es simplifica.

## 4.3 Estratègies òptimes en pòquer: teoria de jocs i equilibris

El pòquer Texas Hold'em és un joc de **suma zero** (ignorant comissions): el que guanya un jugador ho perden els altres. En el context de pòquer, una estratègia en equilibri (també dita **estratègia òptima no explotable**) seria una manera de jugar tal que si l'adversari també juga en equilibri, tots dos tindrien expectació de guany zero contra l'altre (és a dir, no hi ha explotació possible). Si un dels jugadors desviés la seva estratègia, l'altre com a mínim no perdria *value* i probablement el guanyaria. Trobar l'equilibri exacte de Texas Hold'em és extraordinàriament complicat a causa de l'immens espai de decisions.

Un aspecte fonamental és que l'estratègia òptima en pòquer sovint és **aleatoritzada**. Per evitar ser explotat, un jugador òptim barreja bluffs i jocs de valor, juga algunes mans febles de vegades i de vegades les tira, etc., amb certes freqüències equilibrades. Això evita que l'oponent pugui endevinar fàcilment les cartes o tendències. Les **estratègies explotables** són aquelles que, desviant-se de l'equilibri, poden generar més guany contra determinades estratègies però obren la porta a ser aprofitades per un contra-jugador òptim.

La teoria de jocs subjacient ens diu que hi ha certs patrons ideals (per exemple, quina freqüència de *bluff* en relació a bets de valor fa que l'oponent sigui indiferent entre veure o foldejar, conegut com a **Game Theory Optimal (GTO) ratios**). Un entrenament adequat multiagent hauria de portar els agents cap a un estil de joc **competent** que, si no és exactament l'equilibri, almenys *s'apropa* en el sentit de no contenir fuites manifestes explotables fàcilment.

## 4.4 Taules de rang de mans i estadístiques de probabilitat

Un concepte central en l'estratègia de pòquer són els **rang de mans** que un jugador pot tenir sota cert context, i quines mans escull jugar. Un jugador no juga cada mà de la mateixa manera: depèn de la **posició** (jugadors en posició inicial, UTG, solen ser més selectius que els de posició tardana com el *button*), de les **accions prèvies** (si algú ha pujat fort pre-flop, un jugador típic només veurà la puja amb mans fortes; si tothom ha foldejat fins a ell en *late position*, pot pujar amb un rang més ampli per robar blinds), i d'altres factors com l'etapa del torneig o stack sizes.

**Taules de rang pre-flop:** En la teoria i pràctica del pòquer, és comú representar quines mans inicials (parelles, connectors, cartes altes, etc.) són jugables des de quina posició. Per exemple, en una taula de 9 jugadors, un jugador *under the gun* pot només obrir pujant amb aproximadament el top 10-15% de les mans (parelles grans com AA, KK, QQ; Broadway cards com AK, AQ, AJ, KQ; etc.), mentre que un jugador al *button* (últim a parlar preflop) pot obrir amb fins al 40-50% de mans si els de darrere són tight, ja que té avantatge de posició postflop. Aquestes percentatges s'han estimat tant per teòrics com per jugadors professionals al llarg dels anys, i sovint es recullen en **gràfics de rangs**. Per exemple, *pocket aces (AA)* és la millor mà inicial (aproximadament el 0.45% superior de mans) i contra un oponent aleatori té al voltant d'un **85% de probabilitat de guanyar preflop**. Mans fortes com KK, QQ, AKs segueixen darrere. Per contra, mans com 7-2 offsuit són de les pitjors i gairebé sempre es foldegen (7-2o és famosa per ser la mà inicial més feble). Si ordenem totes les 1.326 combinacions de dues cartes (169 classes per valor i pal), podem assignar-les a percentils: per exemple, top 5% inclouria AA, KK, QQ, JJ, AKs, AQs, AK, etc.; top 20% arriben a parelles mitjanes, connectors suitetes alts, etc. Aquestes llistes s'utilitzen per decidir un **rang d'obertura**. També hi ha **taules de 3-bet/4-bet**: per exemple, si estàs a les blinds i un jugador en cutoff puja, amb quines mans li faràs una *3-bet* (re-puja) possiblement mescles mans premiums (per valor) i alguns bluffs com A5s (un as petit suited) per equilibrar.

**Post-flop:** La força relativa d'una mà canvia amb les comunitàries. Es defineix sovint el concepte d'**equity** d'una mà contra el rang o mà d'un adversari: per exemple, un projecte de color (*flush draw*) al flop té ∼35% d'equitat contra una mà feta mediocre; dues overcards tenen cert ∼25% d'equitat contra una parella baixa, etc. Estadísticament es poden calcular les probabilitats que una mà concreta guanyi en showdown contra un rang donat. Aquestes estadístiques guien decisions òptimes: si la nostra **probabilitat de guanyar** (equity) és major que el cost relatiu de veure una aposta, matemàticament convé veure (principi d'**odd pot**). Així, molts jugadors coneixen percentatges clau: per exemple, un projecte de color amb 9 outs té ∼36% de completar-se amb dues cartes per venir (turn+river), un projecte d'escala obert ∼31%, etc. Aquests valors influencien quan fer *call* a apostes (si l'aposta ofereix pot odds favorables a la nostra equity).

**Distribucions de mans guanyadores:** En comunitats de pòquer s'han analitzat quines categories de mans guanyen més sovint en showdown. Per exemple, en taules grosses multiway, sol guanyar una **mà feta alta** (color, escalas, fulls) més sovint que no pas una simple parella, perquè amb molts jugadors és probable que algú lligui fort. En canvi, heads-up, sovint una parella alta pot guanyar al showdown. Aquest coneixement qualitatiu ajuda els jugadors a calibrar el risc: en un pot multi-jugador al river, intentar un gran farol és més arriscat si la probabilitat que algú tingui una mà molt forta és alta.

En síntesi, dominar pòquer requereix gestionar **rang de mans** (quines mans jugar i com), estimar **equities i outs**, i adaptar-se a la **taula i posició**.

# 5 Aprenentatge per Reforç (RL)

L'aprenentatge per reforç de Sutton i Barto (2020) és un paradigma de **machine learning** en què un agent aprèn a prendre decisions mitjançant interacció amb un entorn, rebre recompenses i penalitzacions, i adaptant el seu comportament per maximitzar la recompensa acumulada. Formalment, molts problemes de RL es modelen com un *Markov decision process* (MDP) definit per un conjunt d'estats $S$, accions $A$, funcions de transició $T(s, a, s')$ i recompenses $R(s, a)$, amb un factor de descompte $\gamma$. L'agent intenta aprendre una **política** $\pi : S \to A$ òptima que maximitzi la recompensa esperada acumulada a llarg termini.

Segons Sutton i Barto, *"reinforcement learning is learning what to do, how to map situations to actions, so as to maximize a numerical reward signal"*, destacant que l'agent ha d'explorar accions i rebre recompenses retardades en el temps. L'agent no només considera la recompensa immediata d'una acció, sinó també com aquesta acció afecta futurs estats i recompenses, cosa que requereix equilibrar **exploració** i **explotació**.

## 5.1 Agents

### 5.1.1 Funcions de valor i equació de Bellman

Un concepte clau en RL és la *funció de valor*, que mesura la qualitat d'un estat (o d'una acció) en relació amb la recompensa futura esperada. Si considerem una *política* $\pi$ (una estratègia que defineix com escollir les accions), la *funció de valor d'estat* $V^\pi(s)$ és la recompensa total esperada a llarg termini començant en l'estat $s$ i seguint la política $\pi$. De manera anàloga, la *funció de valor d'acció* $Q^\pi(s, a)$ quantifica la recompensa total esperada en prendre l'acció $a$ a l'estat $s$ i després continuar amb $\pi$.

Aquestes funcions de valor satisfan la *equació de Bellman*, que expressa el valor d'un estat (o acció) en termes de la recompensa immediata i del valor dels estats successius. En el cas òptim (és a dir, sota la política òptima), la versió de l'equació de Bellman per a $Q^*$ és:

$$Q^*(s, a) \;=\; \mathbb{E}\Big[r(s, a) + \gamma \max_{a'} Q^*(s', a') \;\Big|\; s, a\Big],$$

on $r(s, a)$ és la recompensa immediata al fer l'acció $a$ en l'estat $s$, $s'$ és el següent estat, i $\gamma \in [0, 1]$ el factor de descompte que valora la importància de recompenses futures.

### 5.1.2 Mètodes basats en valor i basats en política

En el camp de l'RL, trobem dues famílies principals d'algorismes:

- **Mètodes basats en valor**: busquen aprendre primer la funció de valor (per exemple, $Q(s, a)$) i derivar-ne la política òptima. Un exemple representatiu és *Q-learning*, un mètode off-policy que actualitza iterativament $Q(s, a)$ segons la discrepància entre l'estimació actual i la de l'equació de Bellman òptima.
- **Mètodes basats en política**: aprenen directament la política òptima sense haver d'estimar explícitament una funció de valor. Aquests mètodes (p. ex. *PPO*) ajusten els paràmetres d'una política $\pi_\theta(a \mid s)$ per maximitzar la recompensa esperada via gradient de política.

També existeixen aproximacions híbrides, anomenades *actor-critic*, que combinen un *actor* (que aprèn la política) amb un *critic* (que aproxima la funció de valor).

### 5.1.3 Q-learning

El **Q-learning** (Watkins, 1989), és un algoritme model-free de RL basat en valors que aprèn la **funció de valor d'acció** òptima $Q^*(s, a)$ mitjançant iteracions successives. En cada pas d'interacció, l'agent observa un estat $s$, pren una acció $a$, rep una recompensa $r$ i observa el nou estat $s'$. Els valors $Q$ es van actualitzant cap a l'òptim amb la regla de Bellman:

$$Q_{\text{nou}}(s, a) \leftarrow Q(s, a) + \alpha\Big[r + \gamma \max_{a'} Q(s', a') - Q(s, a)\Big],$$

on $\alpha$ és la taxa d'aprenentatge i $\gamma$ el factor de descompte. Aquesta fórmula ajusta $Q(s, a)$ en direcció a la suma de la recompensa immediata $r$ més la millor estimació del *reward* futur

des de $s'$ (terme $\max_{a'} Q(s', a')$). Intuïtivament, $Q(s, a)$ acaba representant la utilitat esperada d'executar acció $a$ en estat $s$ i seguir la política òptima a partir d'aleshores. En particular, Watkins i Dayan (1992) van demostrar la convergència amb probabilitat 1 sota MDP finits amb recompenses acotades. El Q-learning és un mètode off-policy, ja que aprèn la funció $Q^*$ òptima independentment de la política que segueixi durant l'exploració, gràcies al terme $\max_{a'} Q(s', a')$, que assumeix accions òptimes futures encara que l'agent temporalment explori accions subòptimes.

En paraules de Watkins (1989), el Q-learning permet a un agent **aprendre a actuar òptimament en dominis Markovians experimentant les conseqüències de les accions sense necessitat de modelar explícitament l'entorn**. L'agent avalua les accions en base a les recompenses immediates i els valors futurs estimats; amb prou exploració de tots els estats i accions, acaba aprenent quines accions són globalment millors en termes de recompensa acumulada a llarg termini.

**Convergència i limitacions:** Tot i que el Q-learning funciona molt bé en MDP simples, presenta dificultats en entorns *estocàstics complexos* o *parcialment observables*. Si l'agent no pot observar completament l'estat real (entorns *POMDP*, com el pòquer on part de l'estat, les cartes ocultes dels oponents, és desconegut), la dinàmica que afronta és **no-Markoviana**, i la convergència de Q-learning ja no està garantida teòricament. En contextos multiagent, el Q-learning independent s'enfronta a una *no-estacionarietat* severa: mentre un agent aprèn, els altres canvien de política, violant l'assumpció d'un entorn fix; això dificulta o impedeix la convergència a una solució estable.

### 5.1.4 Proximal Policy Optimization (PPO)

Els mètodes de **policy gradient** constitueixen una altra família important d'algoritmes d'RL, on l'objectiu és aprendre directament una **política paramètrica** $\pi_\theta(a|s)$ (per exemple, representada per una xarxa neuronal) que maximitzi la recompensa esperada. A diferència dels mètodes basats en valor (com Q-learning, DQN, etc.), aquí es calcula explícitament el gradient de la funció objectiu respecte als paràmetres $\theta$ i s'actualitzen en la direcció que millora la política. **Proximal Policy Optimization (PPO)** és un algoritme avançat de *policy gradient* introduït per Schulman et al. (2017) per tal d'aconseguir **actualitzacions més estables i fiables** de la política. PPO pertany als mètodes *actor-critic*, on hi ha un *actor* (la política $\pi_\theta$ que decideix accions) i un *critic* (una funció de valor $V_\phi(s)$ estimada que avalua els estats, o un avantatge $A(s, a)$). El *critic* s'utilitza per reduir la variància del gradient (estimant l'**avantatge** $A(s, a) \approx Q(s, a) - V(s)$) i guiar l'actor.

**Idea clau de PPO:** en comptes de fer grans canvis en la política (que podrien destruir el que s'ha après i desestabilitzar l'entrenament), PPO limita de forma explícita la magnitud del canvi de política en cada iteració. Concretament, defineix un **objectiu surrogate** que utilitza el *rati* entre la política nova i l'antiga per a cada acció: $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$. L'objectiu a maximitzar és:

$$L(\theta) = \mathbb{E}_t\Big[\min\big(r_t(\theta)\,A_t,\ \text{clip}(r_t(\theta),\,1-\varepsilon,\,1+\varepsilon)\,A_t\big)\Big],$$

on $A_t$ és l'avantatge estimat en el pas $t$ i $\varepsilon$ és un hiperparàmetre (p.ex. 0.1 o 0.2). Aquest terme $\min(\cdot, \cdot)$ amb el *clipping* força que si el rati $r_t(\theta)$ es desvia més d'un $\varepsilon$ (és a dir, la nova política s'allunya massa de l'antiga en probabilitat relativa), efectivament es tronca l'avantatge per evitar un gradient excessiu. Això impedix actualitzacions massa "llargues" que podrien degradar la política. En la pràctica, PPO implementa una sèrie de passes de gradient ascent sobre $L(\theta)$ utilitzant conjunts de dades de trajectòries obtingudes amb la política actual (per tant, és un mètode *on-policy*).

**Avantatges i inconvenients en pòquer multiagent:** En un entorn multiagent de pòquer, on diversos agents s'entrenen simultàniament, l'ús de PPO per a cada agent pot aportar **robustesa** en l'aprenentatge. El fet d'actualitzar la política de manera progressiva i limitada ajuda a evitar oscil·lacions dràstiques en resposta als canvis dels altres jugadors. No obstant, un desavantatge és el cost en mostres: en ser on-policy, cada canvi de política requereix noves partides simulades; en entorns com el pòquer, on l'espai d'estats és enorme, això pot requerir moltíssimes mans simulades per assolir una bona estratègia. Malgrat tot, la seva **simplicitat d'implementació, estabilitat en l'entrenament i rendiment contrastat** el fan una elecció atractiva com a *baseline* en un entorn multiagent de pòquer.

### 5.1.5 Altres algoritmes d'RL rellevants (DQN, SAC, APPO)

A més de Q-learning i PPO, existeixen altres algoritmes d'RL que han demostrat ser útils i que poden considerar-se en el context del pòquer Texas Hold'em, especialment combinats amb enfocaments multiagent o d'auto-joc (*self-play*). A continuació es presenten breument:

- **Deep Q-Network (DQN):** És la versió amb xarxes neuronals profundes del Q-learning, proposada per Mnih et al. (2015). La idea principal és usar una xarxa neuronal $Q(s, a; \theta)$ per aproximar la funció de valor d'acció, i entrenar-la amb variants de la regla de Q-learning, emprant tècniques com **replay buffer** (memòria d'experiències) per trencar la correlació entre mostres, i una **xarxa objectiu** fixa temporalmemnt per estabilitzar les etiquetes de Q. En pòquer, DQN pot aplicar-se si es defineixen estats numèrics (per exemple, característiques de la situació de joc) i accions discretes (p.ex., decisions d'aposta quantitzades). Tanmateix, el pòquer és un entorn *partially observable* i multiagent, cosa que dificulta l'aplicació directa de DQN: l'experiència d'entrenament provindria de jugar contra oponents que també evolucionen, violant l'estacionarietat requerida per usar un *replay buffer* clàssic. Malgrat això, DQN pot servir de base en escenaris simplificats (per exemple, pòquer heads-up amb espai d'estats reduït) o combinat amb tècniques per manejar la no-estacionarietat.

- **Soft Actor-Critic (SAC):** És un algoritme *off-policy actor-critic* , basat en la idea d'**aprenentatge de màxima entropia**. SAC (Haarnoja et al., 2018) entrena simultàniament una política estocàstica $\pi_\theta$ i una funció Q critic, però a diferència dels mètodes tradicionals, afegeix a la funció objectiu un terme d'entropia de la política que l'agent intenta maximitzar juntament amb la recompensa. En altres paraules, l'agent aprèn a **maximitzar la recompensa esperada** alhora que maximitza l'entropia de la política, és a dir, manté les seves accions tan imprevisibles com sigui possible mentre compleix la tasca. Aquest criteri fomenta l'**exploració** contínua i evita polítiques massa deterministes o fràgils. SAC ha mostrat un rendiment excel·lent en entorns de control continu complexos, sovint superant l'estat de l'art anterior en aprenentatge continu (Haarnoja, 2018).

- **Asynchronous Proximal Policy Optimization (APPO)**

  Una versió escalable i distribuïda de PPO. Aquesta variant millora significativament l'eficiència d'entrenament en entorns multiagent i paral·lels. APPO es basa en una optimització de política proximal, però incorpora característiques addicionals que li permeten assolir un comportament més robust i adaptable. En concret, utilitza *V-trace*, una tècnica per corregir el desajust entre la política de comportament (amb què es generen les dades) i la política objectiu (amb què s'entrena), mantenint estabilitat tot i recollir dades de forma asincrònica. També fa ús de **Generalized Advantage Estimation (GAE)** per reduir la variància de les actualitzacions, i pot aprofitar **RNNs** (xarxes neuronals recurrents) per integrar informació seqüencial, la qual és clau en jocs amb observabilitat parcial. A més, l'ús opcional d'un **critic compartit** entre agents permet aprofitar informació del valor estimat de l'estat de manera col·lectiva.

## 5.2 Aprenentatge per Reforç Multiagent (MARL)

Quan apliquem RL en entorns on múltiples agents aprenen i interaccionen simultàniament, entrem en l'àmbit de l'**aprenentatge per reforç multiagent (MARL)** (Buşoniu et al., 2010). En aquest cas, cada agent ja no només s'enfronta a estocasticitat de l'entorn, sinó que l'ambient efectiu ve determinat en part per les polítiques (en evolució) dels altres agents. Això introdueix reptes addicionals significatius respecte al RL d'un sol agent.

### 5.2.1 Complexitats addicionals en entorns multiagent

En RL estàndard (un sol agent en un entorn fix), es pot assumir que les dinàmiques de transició $T(s, a, s')$ i distribució de recompenses $R(s, a)$ són estacionàries (no canvien en el temps). No obstant, en un entorn multiagent on diversos jugadors estan aprenent alhora, des de la perspectiva de qualsevol agent individual l'entorn és altament **no-estacionari**. Això és perquè mentre un agent actualitza la seva política, els altres també ho fan; per tant, la "funció objectiu" de l'agent (la recompensa que pot esperar per a una estratègia determinada) està en constant moviment. En

terminologia de RL, l'agent té un "*moving target*": el $Q^*(s, a)$ depèn de les polítiques dels altres, que canvien contínuament durant l'entrenament.

Aquesta no-estacionarietat provoca diversos problemes pràctics:

- **Convergència i estabilitat:** No hi ha garanties generals que múltiples agents autònoms aprendran cap equilibri estable. De fet, són comuns els cicles d'estratègies: l'agent A aprèn a explotar agent B, llavors B canvia de política per contrarestar A, llavors A adapta de nou, i així successivament sense convergir. Per tant, cal dissenyar els algoritmes MARL perquè tendeixin a estats propers a equilibri (p. ex. mitjançant exploració i decaïment d'aprenentatge) o almenys mitigui oscil·lacions.
- **Entorn no-estacionari i "experience replay":** Molts algoritmes de *deep RL* es recolzen en memòries de replay (com DQN) per reentrenar la xarxa amb experiències passades. En MARL, reutilitzar transicions antigues és delicat perquè aquestes provenen d'una dinàmica de joc diferent (polítiques rivals antigues); són *off-policy* respecte a la situació actual. Això pot introduir *bias* important o fins i tot divergència en l'entrenament. Lowe et al. (2017) remarquen que l'ús directe de replay per a Q-learning en multiagent és problemàtic sense ajustos. Una solució pot ser fer replay només de les pròpies experiències però etiquetades amb la política actual dels altres (complex), o escurçar molt la memòria, o usar estratègies com importància ponderada. Els mètodes *on-policy* tipus PPO tenen l'avantatge que entrenen amb dades fresques de la política actual, però pateixen en eficiència de mostra.

## 5.3 Entorns

### 5.3.1 Fonament d'un entorn d'aprenentatge per reforç

En un escenari clàssic d'aprenentatge per reforç (RL) s'assumeix un únic agent que interactua amb un **entorn** modelitzat com un *procés de decisió de Markov* (MDP)

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle,$$

on $\mathcal{S}$ és el conjunt d'estats, $\mathcal{A}$ el conjunt d'accions, $P(s' \mid s, a)$ la dinàmica de transició, $R(s, a)$ la funció de recompensa i $\gamma \in (0, 1]$ el factor de descompte. A cada pas discret $t$ el cicle bàsic és:

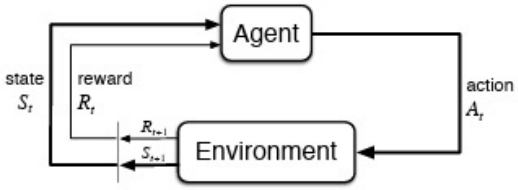



Figura 1: Markov decision process (MDP)

1. **Observació** $s_t \longrightarrow$ l'agent rep l'estat (o observació parcial) actual.
2. **Deguda decisió** $a_t \sim \pi(\cdot \mid s_t)$: l'agent selecciona una acció segons la seva política $\pi$.
3. **Transició** $(s_t, a_t) \longrightarrow s_{t+1}$ seguint $P$; l'entorn calcula la recompensa $r_t = R(s_t, a_t)$.
4. **Actualització**: l'agent ajusta $\pi$ (i, si escau, un valor $V$ o $Q$) amb la nova transició.

Un episodi acaba quan s'arriba a un estat terminal o es depassa un límit de passos. La tasca de l'agent és aprendre una política $\pi^*$ que maximitzi $\mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t r_t\big]$. El disseny d'un *entorn RL* consisteix, doncs, a codificar:

- `reset()`: retorna l'estat inicial i reinicia variables internes.
- `step(action)`: implementa la lògica de transició, calcula la recompensa i indica si l'episodi ha conclòs.
- `render()`: (opcional) visualitza l'estat actual.
- `observation_space` i `action_space`: objectes `gymnasium.spaces` que defineixen domini i rang.

### 5.3.2 Particularitats en entorns multiagent

Quan hi intervenen $N$ agents, el model esdevé un **joc estocàstic de Markov** (o *Markov game*) $\langle \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, P, \{R_i\}_{i=1}^N, \gamma \rangle$. Cada agent $i$ té la seva política $\pi_i$ i recompensa $R_i$. A la pràctica això introdueix diverses diferències clau:

1. **Espai d'accions conjunt**
   La dinàmica es regeix pel vector d'accions $\boldsymbol{a}_t = (a_t^{(1)}, \ldots, a_t^{(N)})$, de manera que $P$ i $R_i$ depenen de totes les accions simultànies. Implementacionalment, un `step()` rep o bé el diccionari d'accions de tots els agents (en entorns paral·lels) o bé la següent acció d'un agent segons un planificador de torns (en l'estil AEC).

2. **No-estacionarietat percebuda**
   Des del punt de vista d'un agent, l'"entorn" inclou els altres $N-1$ agents, la política dels quals canvia al mateix temps que ell aprèn. Això viola l'assumpció d'estacionarietat que utilitzen molts algoritmes de RL d'un sol agent (p. ex. Q-learning), i exigeix estratègies com critics centralitzats o actualitzacions simultànies acotades (p. ex. PPO).

3. **Recompenses múltiples**
   Les recompenses poden ser cooperatives ($R_i = R_j$), competitives (suma zero) o mixtes. L'entorn ha de retornar un vector `rewards[agent_id]`. Sovint la distribució de crèdit (credit assignment) és més difícil: un agent pot rebre recompensa tardana que depèn de les decisions de tots.

4. **Gestió del torn**
   En jocs seqüencials els agents actuen en ordre. PettingZoo proposa dos modes: *parallel* (accions de tots a la vegada) i *AEC* (Agent Environment Cycle), on un `agent_selector` decideix quin agent té el torn i l'entorn avança immediatament després de rebre la seva acció.

5. **Observacions locals**
   Sovint cada agent té només una observació parcial $o_i(s)$, generant un *joc d'informació imperfecta*. L'entorn retorna observacions diferents per ID: `observations[agent_id]`. Això obliga els agents a raonar sobre incertesa (creences) i sol empitjorar la complexitat de l'aprenentatge.

**Resum conceptual** Un entorn multiagent requereix:

- *Espais* i *recompenses* indexats per agent,
- una política clara de torns o d'accions simultànies,
- coherència entre la lògica de joc interna i la informació que es revela a cada jugador,
- i mecanismes per mantenir l'equitat (p. ex. rotació de posicions).

Aquestes diferències fan que les estratègies i els algoritmes que funcionen en MDP convencionals hagin de ser adaptats o repensats en contextos multiagent, tal com es posa de manifest en el disseny del nostre `PokerTournamentEnv`.

### 5.3.3 Entorns existents

En els darrers anys s'han desenvolupat diversos entorns i biblioteques per a entrenar agents mitjançant aprenentatge per reforç en el pòquer *Texas Hold'em*. A continuació es revisen breument alguns dels més rellevants, destacant les seves característiques i limitacions, especialment en context multiagent:

- **OpenSpiel (DeepMind):** OpenSpiel (Lanctot et al., 2019) és un marc general per a jocs amb Aprenentatge per Reforç que inclou variacions de pòquer (fins a 10 jugadors) mitjançant la integració del motor de l'ACPC (Annual Computer Poker Competition). Proporciona una base robusta per a investigació en teoria de jocs i algoritmes de resolució (CFR, Fictitious Play, ...), però no està dissenyat específicament per facilitar experiments pràctics en entorns multiagent personalitzats. L'API és de baix nivell (C++ amb *bindings* a Python) i gestionar

partides complexes requereix programació addicional. Tampoc contempla modes de torneig ni configuracions de taula flexibles (estructura de blinds, eliminació de jugadors, ...). En resum, és molt potent però menys *usable* per a experiments ràpids.

- **RLCard (Data Lab, Univ. Texas A&M):** Toolkit centrat exclusivament en jocs de cartes enfocat a fer de pont entre RL i jocs d'informació imperfecta. Inclou dues variants de Texas Hold'em: *Limit* i *No-limit*. En la variant *Limit*, l'entorn per defecte és *heads-up* (2 jugadors) i cada episodi correspon a una sola mà amb aposta fixa (màxim 4 pujades per ronda). L'estat és un vector de 72 booleans (52 cartes + 20 indicadors d'historial d'apostes) i l'acció és discreta amb quatre moviments (**Fold**, **Call**, **Raise**, **Check**). Tot i que es pot modificar el codi, no ofereix de sèrie entorns multi-jugador, mecàniques de torneig ni recompenses pensades per a competició prolongada. (Zha et al., 2020).

- **PokerRL (Eric Steinberger):** Marc especialitzat d'aprenentatge profund multiagent per a pòquer, amb suport per a NFSP, Deep CFR i computació distribuïda amb *Ray*. El motor és general per a $N$ jugadors, però moltes funcionalitats es centren en escenaris *heads-up*. És una llibreria complexa orientada a investigadors, amb corba d'aprenentatge elevada i sense modalitat de torneig multi-jugador "llesta per usar". (Steinberger, 2019).

- **Altres entorns comunitaris:** Hi ha wrappers Gym com *OpenAI Gym Hold'em* (Wenkel) o *NeuronPoker* (dickreuter) que proporcionen entorns *heads-up* amb No-limit i eines de visualització. La llibreria `clubs` (`clubs_gym`) permet simular variants arbitràries (Leduc, Omaha, Hold'em 6–9 jugadors). No obstant, cap d'aquests entorns aborda de manera integral un torneig amb escalada de blinds i eliminació de jugadors. (Terry et al., 2021).

## 5.4 Modelatge matemàtic del pòquer com a problema d'RL

Des del punt de vista d'aprenentatge per reforç, podem modelar una partida de Texas Hold'em com un entorn on els **estats** inclouen tota la informació disponible en un moment donat (cartes comunitàries revelades, historial d'apostes en la ronda, stack de fitxes de cada jugador, posició del dealer, etc.). Per a un sol agent, però, l'entorn és **parcialment observable**: l'agent no coneix les cartes privades dels altres ni les cartes que faltin per repartir. Podem definir l'**observació** de l'agent com la informació que veu: les seves pròpies cartes, les comunitàries visibles, la seqüència d'apostes realitzades en la ronda (p. ex., si algú ha pujat, quant, etc.), el nombre de jugadors actius, els tamanys dels stacks, el pot acumulat, i possiblement informació derivada com la seva posició (UTG, middle, cutoff, button, blinds). L'**acció** de l'agent en cada estat és la decisió de *fold/call/bet* i, si aposta, l'import. En context d'implementació, sovint es discretitza l'import de l'aposta (p. ex., pujar a un múltiple determinat de la big blind, o all-in) per tenir un espai d'accions manejable, encara que en No-Limit teòricament l'acció és contínua. La **transició d'estats** es regeix per la lògica del joc: després que tots els agents hagin actuat en una ronda, es destapa la següent carta comunitària (a no ser que la mà acabi abans), i així successivament fins al showdown o abandó de tots menys un. L'**reward** final per a cada agent pot definir-se com el guany net de fitxes al final de la mà (p. ex., +X si guanya el pot, -Y si perd Y fitxes apostades). Durant la mà, es podrien donar recompenses intermedies per facilitar l'aprenentatge, però habitualment en pòquer l'única recompensa significativa és la resultant de la conclusió de la mà. També es poden penalitzar algunes accions per motius d'exploració (per exemple, penalitzar lleugerament fer *fold* constant per incentivar l'agent a jugar alguna mà).

Formalment, un sol agent que aprèn a jugar pòquer contra altres pot considerar-se en un **Entorn de Markov Particionalment Observable (POMDP)**, o bé podem modelar tots els jugadors junts com un **joc d'aprenentatge multiagent** (on definim un *Markov Game* o *Stochastic Game* entre agents). L'objectiu de l'agent RL és desenvolupar una **política òptima** que maximitzi la seva utilitat (guany esperat de fitxes) enfront dels altres jugadors.

El modelatge RL del pòquer ha de bregar amb dues complicacions principals: la **combinatòria enorme de l'estat** (52 cartes, distribucions combinacionals de mans i taulers possibles, històrics d'apostes molt diversos) i la **incompletitud de la informació**. Això fa que l'espai d'estats efectiu (considerant la història observable com a estat) sigui gegantí i es requereixen tècniques d'aproximació de funcions (xarxes neuronals) i fins i tot abstraccions del joc (agrupar estats similars, reduir opcions d'aposta) per fer viable l'aprenentatge.

# 6 Implementació de l'entorn de pòquer

L'entorn base utilitzat per entrenar els agents és una extensió personalitzada del marc `PettingZoo`, adaptat per simular tornejos de *Texas Hold'em Limit 4-max* amb múltiples agents. A continuació es detallen les decisions de disseny més rellevants, tant a nivell funcional com d'informació disponible per als agents i càlcul de recompenses.

## 6.1 Accions disponibles

Cada agent disposa de quatre accions discretes:

- `Fold` (0): abandonar la mà.
- `Call` (1): igualar la quantitat requerida per continuar.
- `MinRaise` (2): pujar el mínim permès (10 fitxes).
- `All-in` (3): apostar tot el *stack* restant.

Aquest conjunt es manté constant, però una màscara d'accions `action_mask` informa en cada moment de quines són vàlides.

## 6.2 Observacions per agent

Cada agent rep una observació estructurada, en forma de diccionari `Dict`, amb les següents components:

- Cartes pròpies (`hole_cards`), comunitàries, fase del joc (`phase`), *stack* propi i posició a la taula.
- Informació contextual com `pot_size`, quantitat a igualar (`to_call`), aposta actual i pròpia.
- Indicadors de situació: nombre de jugadors actius, `pot odds`, `stack ratio`, i posició relativa (dealer, small blind, big blind).
- Valors numèrics de força (`strength`) i potencial (`potential`) de la mà actual.
- `action_mask` amb accions vàlides (vector binari).

## 6.3 Càlcul de recompenses

Per tal de guiar l'entrenament i incentivar un estil realista, es defineixen recompenses per mà amb components diversos:

- **Reward base per guanyar:** `WIN_HAND_BONUS = +1.0` per quedar-se el pot per fold adversari.
- **Bonus addicional per showdown:** `WIN_SHOWDOWN_BONUS = +2.0` si es guanya en mostrar cartes.
- **Pèrdua per perdre la mà:** `LOSE_PENALTY = -0.25`, si no es guanya però s'arriba al final.
- **Penalització per all-in prematur:** `LOSE_EARLY_PENALTY = -2.0` si es fa all-in abans del Turn.
- **Penalització per quedar eliminat:** `BUSTED_PENALTY = -5.0` per quedar sense fitxes.

**Survival Bonus.** El *bonus de supervivència* és una recompensa dinàmica que busca incentivar que els agents sobrevisquin a les primeres fases del torneig. Aquest valor comença en `+0.20` a la mà 1 i decreix linealment fins arribar a 0 a la mà 20, moment en què es torna negatiu. El valor mínim que pot assolir és `-1.0`, per tal d'evitar penalitzacions excessives en mans molt avançades:

$$\text{survival\_bonus}(h) = \begin{cases} 0.20 \cdot \left(1 - \frac{h}{20}\right) & \text{si } h < 20 \\ \max\left(-0.05 \cdot (h - 19),\ -1.0\right) & \text{si } h \geq 20 \end{cases}$$

Aquest bonus es torna negatiu a partir de la mà 20 per penalitzar estratègies massa conservadores a llarg termini. Aquest component es calcula al final de cada mà i s'afegeix a la recompensa acumulada de cada agent encara viu.

**Pot reward.** El guanyador d'una mà, ja sigui per fold dels oponents o en un showdown, rep a més una recompensa proporcional al pot acumulat:

$$\text{pot\_reward} = \frac{\text{pot}}{\text{STACK\_SCALE}}$$

Amb `STACK_SCALE = 1000`, això normalitza la quantitat de fitxes i evita desequilibris numèrics entre màximes i mínimes recompenses.

**Reward màxim per mà.** El cas més favorable és guanyar un pot gran en un showdown, sense penalitzacions:

$$= \frac{\text{pot}}{\text{STACK\_SCALE}} + \text{WIN\_HAND\_BONUS} + \text{WIN\_SHOWDOWN\_BONUS} + \text{survival bonus} \tag{1}$$

$$\approx 3.0 + 1.0 + 2.0 + 0.20 = \mathbf{6.20} \tag{2}$$

**Reward mínim per mà.** El pitjor cas és quedar eliminat després d'un all-in prematur i perdre la mà:

$$= \text{BUSTED\_PENALTY} + \text{LOSE\_EARLY\_PENALTY} + \text{LOSE\_PENALTY} + \text{survival (negatiu)} \tag{3}$$

$$\approx -5.0 + (-2.0) + (-0.25) + (-1.0) = \mathbf{-8.25} \tag{4}$$

## 6.4 Aportacions i resolució de limitacions

- **Compatibilitat MARL:** En adoptar PettingZoo, l'entorn opera com un *Gym-like* multiagent compatible amb RLlib, Stable-Baselines3, CleanRL, ...
- **Simetria entre agents:** Tots els jugadors comparteixen espai d'estat i accions; les posicions es roten automàticament, evitant biaixos.
- **Mecànica de torneig integrada:** *Stacks* finits, escalat de blinds i eliminació de jugadors permeten estudiar estratègies d'adaptació i supervivència.
- **Blinds configurables:** L'usuari pot definir calendari d'increments (p. ex. doblar cada 10 mans).
- **Gestió eficient de torns:** *agent_selector* salta jugadors inactius i reinicia correctament l'ordre després de cada ronda.
- **Extensibilitat:** El codi és modular; es pot estendre a No-Limit Hold'em, variar el nombre de jugadors o ajustar límits de *raises*.

En conclusió, `PokerTournamentEnv` omple un buit important en la recerca d'entorns d'aprenentatge per reforç per a pòquer, facilitant experiments realistes en format de torneig multijugador i mantenint una interfície senzilla i estàndard.

# 7 Resultats

En aquesta secció presentem una revisió completa dels experiments amb `PokerTournamentEnv`. S'han comparat dos agents d'*Aprenentatge per Reforç Multiagent*: un **APPO** (Asynchronous Proximal Policy Optimization) i un **DQN** independent, tots dos enfrontats a tres rivals aleatoris.

## 7.1 Anàlisi

### 7.1.1 Paràmetres d'entrenament i decisions de configuració

Durant el desenvolupament dels experiments, s'han seleccionat acuradament alguns paràmetres amb l'objectiu de garantir estabilitat i rendiment en un entorn complex com el pòquer multiagent. Tot seguit es detallen els més significatius per a cada algoritme.

**Configuració DQN**

- **lr = 5e-5**: Una taxa d'aprenentatge especialment baixa. Aquesta decisió s'ha pres perquè, en entorns amb informació parcial i variabilitat estocàstica entre episodis, els valors Q poden oscilar molt. Amb un `lr` petit, les actualitzacions són més suaus i el valor de les accions es torna més estable.
- **gamma = 0.997**: Discount alt, ja que volem que els agents valorin molt les recompenses futures, cosa rellevant en partides llargues amb estructures d'aposta múltiples com al Hold'em.
- **double_q = True, dueling = True, n_step = 3**: S'han activat tècniques avançades per millorar l'estabilitat del Q-learning. El doble Q evita sobreestimacions, el dueling separa valor d'estat i avantatge d'acció, i el n-step ajuda a propagar millor la informació de recompensa.
- **Prioritized Replay Buffer (alpha = 0.6, beta = 0.4)**: Amb aquest buffer, les transicions més informatives tenen més probabilitat de ser reentrenades. Els valors d'`alpha` i `beta` balancegen entre exploració de casos rars i estabilitat d'entrenament.
- **Exploració amb EpsilonGreedy (decai lent)**: Comencem amb `epsilon = 1.0` i el fem decaure molt lentament (200 000 `timesteps`) per permetre exploració extensa en una etapa on encara no hi ha política útil.
- **Arquitectura FeedForward: [512, 512]**: Xarxa bastant gran per capturar la complexitat estratègica del pòquer, però sense arribar a ser costosa computacionalment.

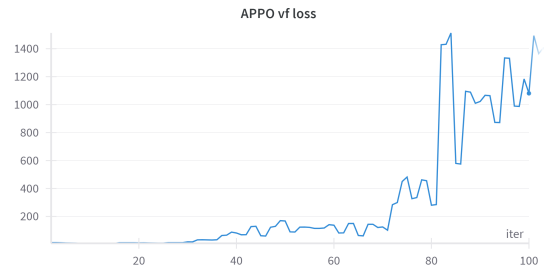
**Configuració APPO**

- **lr = 1e-4, clip_param = 0.3**: Una taxa d'aprenentatge lleugerament més alta que a DQN, ja que APPO és més estable gràcies a la política actualitzada per confiança. El paràmetre de `clipping` protegeix contra actualitzacions massa agressives.
- **use_gae = True, lambda = 0.95**: L'ús de *Generalized Advantage Estimation* millora la reducció de variància i la convergència en entorns amb recompensa esparsa com el pòquer. El valor de $\lambda = 0.95$ busca un bon equilibri entre biais i variància.
- **vtrace = True**: Aquesta opció activa el càlcul de V-Trace, un mètode de correcció off-policy que permet que les mostres recollides amb una política antiga segueixin sent útils per actualitzar la política actual.
- **entropy_coeff = 0.01**: S'ha afegit una petita penalització d'entropia per incentivar l'exploració durant les primeres etapes de l'entrenament.
- **model amb LSTM (256)**: Es fa servir una xarxa recurrent perquè el pòquer és un joc amb informació parcial i seqüencial. Aquesta memòria ajuda a capturar patrons d'aposta o accions passades del rival.
- **kl_coeff = 0.5**: Penalització de divergència KL per mantenir la política propera a l'anterior i evitar exploració descontrolada.

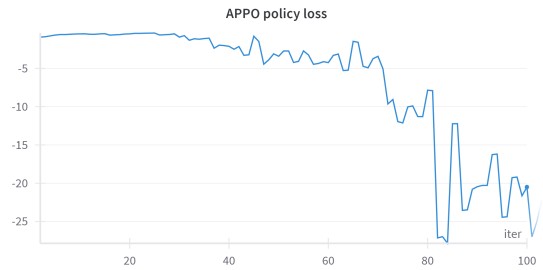
Aquestes configuracions s'han escollit amb la intenció de facilitar un aprenentatge inicial estable, encara que no òptim. No s'ha intentat maximitzar el rendiment final sinó explorar com responen els agents a condicions raonables. En futures iteracions es podria fer una cerca més fina d'hiperparàmetres, entrenar durant més timesteps o introduir tècniques avançades com replay buffer compartit, entrenament centralitzat amb observadors globals o estratègies *meta-learning*.

### 7.1.2 Corbes d'entrenament

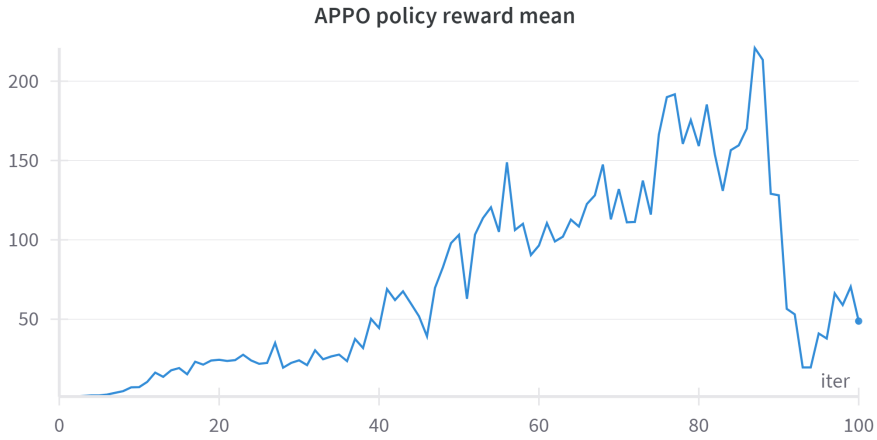
Les Figures 2 i 3 mostren les principals mètriques enregistrades via Weights & Biases (W&B).




(a) APPO *value-function* loss



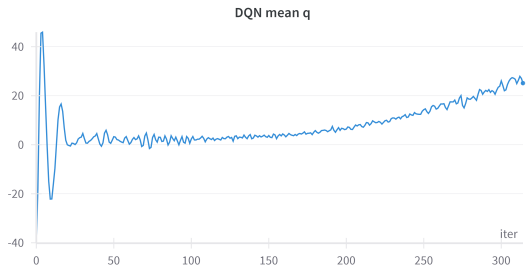

(b) APPO policy loss



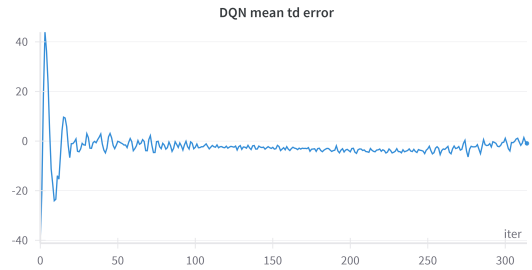

(c) APPO recompensa mitjana per iteració

Figura 2: Evolució de mètriques APPO (100 iteracions)

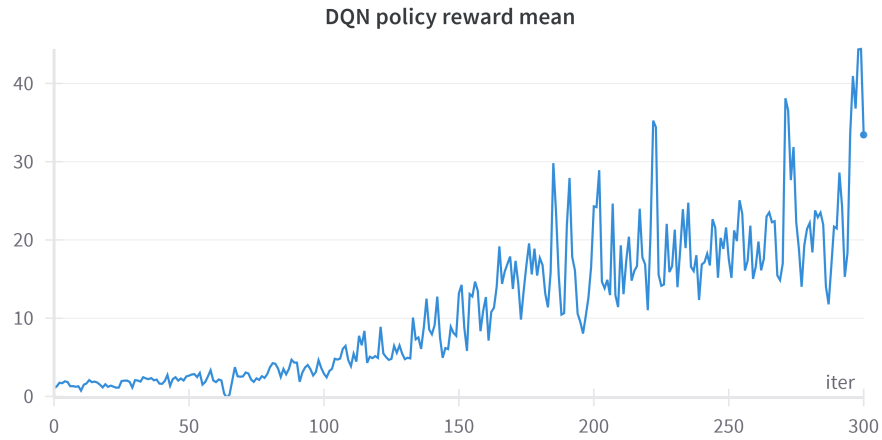
Les corbes d'entrenament mostren diferències importants entre els dos enfocaments. En el cas d'APPO, l'evolució inicial és força prometedora, amb un augment sostingut de la *policy reward mean* fins aproximadament la iteració 80. Tanmateix, a partir d'aquest punt s'observa una caiguda abrupta del rendiment, que coincideix amb una explosió sobtada del *value function loss*, indicant una desestabilització del *critic*. Aquest fenomen pot estar relacionat amb un mal càlcul dels avantatges, acumulació de desajustos entre política i valor estimat, o una combinació d'ambdós efectes, especialment tenint en compte la naturalesa estocàstica del pòquer i la presència d'observacions parcials.




(a) DQN recompensa mitjana




(b) DQN error TD mitjà




(c) DQN valor Q mitjà

Figura 3: Evolució de mètriques DQN (300 iteracions)

D'altra banda, la política entrenada amb DQN mostra una evolució més suau però constant, amb increments progressius de la *policy reward mean* fins a la iteració 300. El *TD error* s'estabilitza ràpidament, la qual cosa suggereix que l'actualització del Q-valor és coherent i convergent. Això es veu reforçat pel creixement gradual però sostingut del valor mitjà estimat dels Q-valors (*mean q*), que indica un reconeixement creixent de situacions avantatjoses.

En resum, mentre APPO presenta un comportament més volàtil i depèn fortament de l'estabilitat del *critic*, DQN evoluciona de manera més robusta, tot i que a un ritme inicialment més lent. Aquestes diferències evidencien la sensibilitat dels algoritmes d'actor-critic en entorns complexos com el pòquer, on la informació parcial i les múltiples rondes d'apostes poden amplificar errors en la propagació de valors. Això reforça la importància d'una exploració acurada i d'una parametrització estable en algoritmes com APPO.

### 7.1.3 Victòries per torneig

Taula 1: APPO vs. Random (100 tornejos)

| Jugador | Algorisme | Victòries (%) |
|---|---|---|
| player_0 | APPO | 32.0 |
| player_3 | Random | 25.0 |
| player_1 | Random | 24.0 |
| player_2 | Random | 19.0 |

Taula 2: DQN vs. Random (100 tornejos)

| Jugador | Algorisme | Victòries (%) |
|---|---|---|
| player_0 | DQN | 43.3 |
| player_1 | Random | 20.3 |
| player_2 | Random | 18.9 |
| player_3 | Random | 17.5 |

Taula 3: Distribució d'estil de joc i agressivitat

| | Fold % | Call % | Raise % | All-in % | Agressivitat |
|---|---|---|---|---|---|
| APPO | 72.0 | 5.8 | 22.0 | 0.1 | 79.2 % |
| DQN | 79.3 | 0.0 | 17.5 | 3.2 | 100.0 % |

#### 7.1.4 Indicadors d'estil de joc

**Càlcul de l'agressivitat**

Per tal de caracteritzar l'estil de joc dels agents, una mètrica rellevant és la **agressivitat**, que mesura la proporció d'accions actives (*raise*, *all-in*) en relació amb les accions no-passives (*call*, *raise*, *all-in*). Aquesta mesura ens permet entendre si l'agent tendeix a pressionar els oponents o a jugar de manera més passiva.

Formalment, l'agressivitat es defineix com:

$$\text{Agressivitat} = \frac{\#Raise + \#All\text{-}in}{\#Call + \#Raise + \#All\text{-}in},$$

on el numerador representa les accions ofensives i el denominador el conjunt d'accions on l'agent decideix continuar jugant (s'exclouen els *folds*).

Aquesta mètrica es basa en el principi que fer *raise* o *all-in* comporta prendre la iniciativa, mentre que fer *call* implica una actitud més reactiva. El valor resultant oscil·la entre 0 (agent totalment passiu) i 1 (agent purament agressiu).

### 7.2 Discussió

**Rendiment global.** A primera vista, DQN obté més victòries (43,3%) que APPO (32,0%), però abans de concloure que és superior, cal tenir en compte la incertesa estadística associada a aquests valors. Tots dos agents han estat avaluats sobre 100 tornejos, de manera que podem comparar directament les proporcions utilitzant el model binomial per calcular un interval de confiança del 95 %.

L'error estàndard (*standard error*) d'una proporció $p$ es calcula amb la fórmula:

$$\text{EE} = \sqrt{\frac{p(1-p)}{n}},$$

i l'interval de confiança del 95 % és:

$$\text{IC}_{95\%} = p \pm 1{,}96 \cdot \text{EE}.$$

Apliquem-ho als dos agents:

- **DQN**: $p = 0{,}433$, $n = 100 \Rightarrow \text{EE} \approx 0{,}0493$
  $\text{IC} = [0{,}433 \pm 1{,}96 \cdot 0{,}0493] = [0{,}336,\ 0{,}530]$
- **APPO**: $p = 0{,}320$, $n = 100 \Rightarrow \text{EE} \approx 0{,}0466$
  $\text{IC} = [0{,}320 \pm 1{,}96 \cdot 0{,}0466] = [0{,}229,\ 0{,}411]$

Tot i que els intervals es solapen parcialment, el de DQN es desplaça cap a valors més alts. Això suggereix que DQN té, amb una confiança raonable, millor capacitat de convertir decisions en victòries en aquest entorn específic. No obstant això, donat el solapament parcial, no podem afirmar-ho amb certesa absoluta. Una mostra més gran ajudaria a reduir l'ambigüitat i confirmar o refutar estadísticament aquesta hipòtesi.

En resum, **DQN presenta una mitjana més alta i un interval de confiança més orientat cap a l'èxit**, fet que apunta a un rendiment potencialment superior.

**Dinàmica d'entrenament.** Les figures confirmen la percepció qualitativa:

- **APPO**. Entre les iteracions 40 i 80 la recompensa mitjana passa de 0$ a 200$, però a partir de la iteració 85 la corba s'enfonsa sobtadament (Fig. 2c). El pic de *vf_ loss* (1.500) indica que el critic va deixar de predir correctament i la *policy loss* es va disparar a valors molt negatius. En terminologia PPO, l'actualització va creuar el llindar de *trust region* i va degradar la política. En tornejos posteriors això es tradueix en ales aturades de victòria sobre la meitat d'episodis.
- **DQN**. L'agent basat en valor mostra una pujada més lenta però sostinguda (Fig. 3a). El *TD error* (b) cau ràpidament a zero i es manté estable; el valor Q mitjà (c) creix fins a 25 fitxes, coherent amb la banda superior de recompensa observada. Aquesta estabilitat del critic explica el menor risc de col·lapse de política.

**Estil de joc.** Les mètriques d'estil (Taula 3) retraten dues personalitats oposades:

**APPO** Selecciona mans: 72% de *fold*, només 0.1% d'all-ins. El model sembla empènyer la *policy entropy* cap a mínims, privilegiant la supervivència; això correlaciona amb la taxa de victòria moderada però sòlida.

**DQN** Agressiu al màxim: 79% de *fold*, però cap `call` (0%) i un 17% de `raise`. La manca de `call`s suggereix que la xarxa Q penalitza moviments passius i prefereix apostar o llançar-se. Contra rivals humans podria ser fàcilment explotable, però contra oponents aleatoris capitalitza.

**Limitacions.** Cal matisar que els rivals són *estàtics*; no hi ha co-adaptació. Un entorn realista exigiria que tots els jugadors aprenessin simultàniament, de manera que la corba de DQN hauria de conviure amb un entorn no-estacionari.

En conjunt, aquests resultats mostren que el marc de torneig 4-max és prou ric per distingir patrons d'aprenentatge i que encara hi ha marge per a millorar tant la robustesa d'APPO com la sofisticació estratègica de DQN.

# Conclusió

Aquest Treball de Fi de Grau ha estat una oportunitat per aprofundir en un àmbit de la intel·ligència artificial que em motiva especialment: l'aprenentatge per reforç multiagent en entorns amb informació incompleta. Tot i que gran part dels èxits de l'Aprenentatge per Reforç Profund (DRL) s'han assolit en contextos d'informació perfecta, com ara videojocs clàssics o escacs, el pòquer i en particular la seva modalitat Limit Hold'em 4-max ofereix un repte molt més proper al món real, on la incertesa i la interacció estratègica entre agents són constants.

L'objectiu principal era construir un entorn funcional, obert i estandarditzat de Limit Hold'em 4-max basat en PettingZoo. Aquest objectiu s'ha assolit amb èxit: l'entorn desenvolupat és compatible amb biblioteques DRL habituals (Stable-Baselines3, RLlib, CleanRL...), permet visualització i enregistrament d'històries de mans, i suporta fàcilment extensions com panells de monitoratge i tornejos. Aquesta eina és, per si sola, una contribució útil per a futurs treballs acadèmics o experimentals dins del camp MARL.

Pel que fa als objectius científics, he aconseguit implementar i entrenar dos enfocaments bàsics però representatius: Independent DQN i APPO. Tot i les limitacions d'aquests mètodes, els resultats han permès observar patrons clars: DQN presenta una corba d'aprenentatge més estable, mentre que APPO mostra pics inicials més alts però també més volatilitat. En cap cas s'ha arribat a estratègies properes a l'equilibri, cosa que era esperable donada la complexitat del domini i la simplicitat dels algoritmes provats. Per tant, tot i que no s'ha assolit un rendiment òptim dels agents, sí que s'ha complert l'objectiu d'explorar i avaluar les dificultats de l'aprenentatge en escenaris multiagent amb informació incompleta.

Les limitacions del treball també són evidents i obren la porta a múltiples millores futures. A nivell de l'entorn, es podria ampliar amb suport per a *no-limit*, variacions amb sis jugadors, o funcionalitats per a entrenament centralitzat. També seria útil afegir mètriques més avançades per avaluar l'exploitabilitat i eines per analitzar el joc post-flop amb més detall. A nivell d'entrenament, els algoritmes utilitzats han estat intencionadament senzills per mantenir el projecte manejable, però seria molt interessant explorar enfocaments més potents com SAC, Q-Mix, o mètodes amb crítica centralitzada, la implementació de *self-play iteratiu* i mecanismes de *policy distillation* podrien ajudar a estabilitzar l'aprenentatge.

En resum, aquest TFG ha estat un primer pas sòlid cap a la investigació en aprenentatge per reforç en jocs d'informació incompleta. He après a combinar disseny d'entorns, enginyeria de simuladors i entrenament d'agents en un marc complex però apassionant. La feina feta ha estat exigent, però ha valgut la pena, i obre diverses línies futures tant per continuar aprofundint com per contribuir amb eines útils a la comunitat.

Per a qui vulgui replicar o ampliar aquest treball, tot el codi i els recursos utilitzats estan disponibles a GitHub: https://github.com/JOAKOF/PokerEntorn.

# Referències


Buşoniu, L., Babuška, R., & De Schutter, B. (2010). Multi-agent Reinforcement Learning: An Overview. A G. Weiss et al. (Ed.), *Innovations in Multi-Agent Systems and Applications - 1* (p. 183 - 221). Springer. https://doi.org/10.1007/978-3-642-14435-6_9

Cornell University, Department of Mathematics. (2006). Texas Hold'em Poker [Accedit el 9 de juny de 2025].

Haarnoja, T. (2018, desembre). *Acquiring Diverse Robot Skills via Maximum Entropy Deep Reinforcement Learning* (Technical Report No. UCB/EECS-2018-176) (Accedit el 9 de juny de 2025). EECS Department, University of California, Berkeley. https://www2.eecs.berkeley.edu/Pubs/TechRpts/2018/EECS-2018-176.pdf

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor [Accedit el 9 de juny de 2025]. *Proceedings of the 35th International Conference on Machine Learning (ICML)*. https://arxiv.org/pdf/1801.01290

Lanctot, M., Lockhart, E., Lespiau, J.-B., Zambaldi, V., Upadhyay, S., Pérolat, J., Gruslys, A., Müller, T., Tuyls, K., & Graepel, T. (2019). OpenSpiel: A Framework for Reinforcement Learning in Games [Accedit el 9 de juny de 2025]. *arXiv preprint arXiv:1908.09453*. https://arxiv.org/abs/1908.09453

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments [Accedit el 9 de juny de 2025]. *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/1706.02275

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529 - 533. https://doi.org/10.1038/nature14236

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms [Accedit el 9 de juny de 2025]. *arXiv preprint arXiv:1707.06347*. https://arxiv.org/pdf/1707.06347

Steinberger, E. (2019). PokerRL: Open Source Framework for Deep Reinforcement Learning in Poker [Accedit el 9 de juny de 2025]. *arXiv preprint arXiv:1905.04341*. https://arxiv.org/abs/1905.04341

Sutton, R. S., & Barto, A. G. (2020). *Reinforcement Learning: An Introduction* (2nd). MIT Press. http://incompleteideas.net/book/RLbook2020.pdf

Terry, J., Black, A., Jayakumar, M., Hari, A., Santos, R., Ravi, S., Perez, R., Dhariwal, P., Anandkumar, A., & Leibo, J. Z. (2021). PettingZoo: Gym for Multi-Agent Reinforcement Learning [Accedit el 9 de juny de 2025]. *arXiv preprint arXiv:2009.14471*. https://arxiv.org/abs/2009.14471

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards* [PhD thesis]. University of Cambridge. https://www.cs.rhul.ac.uk/home/chrisw/thesis.html

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3–4), 279 - 292. https://doi.org/10.1007/BF00992698

Zha, D., Zhang, H., Huang, Y., Wu, J., Tan, M., & Wang, X. (2020). RLCard: A Toolkit for Reinforcement Learning in Card Games [Accedit el 9 de juny de 2025]. *arXiv preprint arXiv:1910.04376*. https://arxiv.org/abs/1910.04376