UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

TRANSFER LEARNING EN JOCS MULTIAGENT

ÀLEX DOMÍNGUEZ RODRÍGUEZ

Director/a
JAVIER VAZQUEZ SALCEDA (Departament de Ciències de la Computació)

Codirector/a
SERGIO ÁLVAREZ NAPAGAO (Departament de Ciències de la Computació)

Titulació
Grau en Enginyeria Informàtica (Computació)

Memòria del treball de fi de grau

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

26/06/2024

Abstract

El *Transfer Learning* és una àrea de la Intel·ligència Artificial que busca transferir el coneixement adquirit per un sistema intel·ligent en la resolució d'un problema cap a la resolució d'un altre problema relacionat. Aquest projecte inicialment tenia com a objectiu aplicar *Transfer Learning* a un conjunt de quatre agents entrenats en un joc *Predator-Prey* amb aprenentatge per reforç multiagent per tal de jugar a un joc semblant, *Combat*, analitzant diferents estratègies de *Transfer Learning*.

Degut als recursos computacionals necessaris per dur a terme la proposta original, va caldre canviar l'objecte a jocs d'un sol agent. Això ens ha permès implementar i estudiar la tècnica d'*Introspective Action Advise* en un escenari amb una transferència d'aprenentatge reduïda, deixant per a treballs futurs l'aplicació d'aquesta tècnica a jocs multiagents.

Els principals reptes afrontats en aquest treball han estat l'adaptació de la tècnica a estudiar a un agent *DQN*, l'ús d'un sistema adaptatiu en l'aconsellament d'accions, i la implementació del sistema de classes per a l'experimentació. A través d'aquesta investigació, hem pogut explorar les capacitats de la tècnica d'*Introspective Action Advise* en un entorn controlat, proporcionant una base sòlida per a futures investigacions en entorns multiagents.

Agraïments

Moltes gràcies a tots aquells que m'han ajudat en aquest projecte.
Especialment, vull expressar la meva gratitud als meus directors, Javier i Sergio.
Reconeixement també a la meva família pel seu suport constant i incondicional.
Equip de professors de la UPC que m'han transmès els coneixements per posible aquest treball.
Departament de CS per proporcionar els recursos i coneixements necessaris.
Inequívocament a tots els i aquelles que han treballat en els camps estudiats abans que jo.
Tots els amics que van oferir ànims i consells, Miguel, Salvador, Aure, Tomasso i més.
Homenatge a tots els que han contribuït a aquest treball.

Gràcies de nou a tots per fer aquest projecte possible.
Realment, la vostra ajuda ha estat inestimable.
Espero poder retornar tot el que he rebut.
You have my deepest gratitude.

Índex

1 Introducció **15**
- 1.1 Context . . . 15
- 1.2 Conceptes Fonamentals . . . 16
- 1.3 Identificació del Problema . . . 20
- 1.4 Abast . . . 20
  - 1.4.1 Motivació . . . 20
  - 1.4.2 Objectius . . . 21
  - 1.4.3 Requeriments . . . 23
  - 1.4.4 Requeriments Principals . . . 23
  - 1.4.5 Requeriments Secundaris . . . 23
  - 1.4.6 Recursos . . . 24
- 1.5 Actualització de l'Abast al finalitzar el projecte . . . 25
- 1.6 Contexte legal: Lleis i Regulacions . . . 26
- 1.7 Estructura de la Memòria . . . 27
  - 1.7.1 Introducció . . . 27
  - 1.7.2 Fonaments Teòrics . . . 27
  - 1.7.3 Metodologia . . . 27
  - 1.7.4 Implementació del sistema d'Experimentació . . . 27
  - 1.7.5 Experimentació i Resultats . . . 27
  - 1.7.6 Conclusions . . . 27

2 Fonaments Teòrics **29**
- 2.1 Estat de l'art . . . 29
  - 2.1.1 Models d'agents per a l'aprenentatge per reforç . . . 29
  - 2.1.2 Algoritme per a la Transferencia d'Aprenentatge . . . 30
- 2.2 Recapitulació i Introducció de la Proposta . . . 32
- 2.3 La nostre proposta . . . 33
  - 2.3.1 *Budget Advise* . . . 33
  - 2.3.2 Mesura de Incertesa . . . 33
  - 2.3.3 Llindar d'Incertesa . . . 34
  - 2.3.4 Comparació dels Algoritmes . . . 34

3 Metodologia **37**
- 3.1 Mètode . . . 37
  - 3.1.1 Monitorització i rigor . . . 37
- 3.2 Metodologia d'implementació . . . 39
  - 3.2.1 Codi Base: Descripció General de les Classes i Funcions . . . 39
  - 3.2.2 Fase 1 de la Implementació: Encapsulament . . . 40
  - 3.2.3 Fase 2 de la Implementació: Refinament . . . 40
  - 3.2.4 Fase 3 de la Implementació: Experimentació . . . 41
- 3.3 Metodologia d'Experimentació . . . 41
  - 3.3.1 Justificació de l'entorn . . . 41
  - 3.3.2 Monitorització i Rigor en l'Experimentació . . . 42

4 Implementació del sistema d'experimentació **45**
4.1 Descripció de l'Entorn d'experimentació . . . 45
4.2 Exposició general del sistema de classes . . . 46
4.3 Descripció main i launchers . . . 48
4.3.1 Classe `A_main.py` . . . 48
4.3.2 Classe `TrainLauncher` . . . 49
4.3.3 Classe `TestLauncher` . . . 49
4.4 Descripció d'entrenaments . . . 50
4.4.1 Classe `BaseTrainingProcess` . . . 50
4.4.2 Descripció funcions `TrainDQN` i `TrainDQN_AA` . . . 53
4.4.3 Funcions comunes . . . 54
4.4.4 Funcions exclusives de `TrainDDQN_AA` . . . 55
4.4.5 Classe `TestProcess` . . . 56
4.5 Descripció DQNSolver i DQNAgent . . . 58
4.5.1 Classe `DQNSolver` . . . 58
4.5.2 Classe `DQNAgent` . . . 59
4.6 Classes auxiliars: `Tracker` i `Config Decoder` . . . 62
4.6.1 `Config Decoder` . . . 62
4.6.2 `Tracker` . . . 63

5 Experimentació i Resultats **65**
5.1 Definició de l'experimientació . . . 65
5.1.1 Nombre d'Episodis . . . 65
5.1.2 Experiments Planificats . . . 67
5.2 Tècniques Emprades . . . 69
5.2.1 Cota inferior: DQN . . . 69
5.2.2 Cota mitja: FT . . . 70
5.2.3 Cota superior: SS . . . 70
5.2.4 *Action Advise* . . . 71
5.3 Elecció de paràmetres . . . 71
5.3.1 Paràmetres de *Burn in* δ i *Decay* λ . . . 72
5.4 Resultats de l'Estudi de Paràmetres: Refinament DQN . . . 73
5.5 Resultats Estudi Paràmetres: *Drop Out* . . . 74
5.6 Resultats Estudi Paràmetres *Action Advise*: Learning Rate . . . 76
5.7 Interpretació de Mètriques d'*Action Advise* . . . 78
5.8 Interpretació de Mètriques Experimentals . . . 80
5.9 Interpretació de les gràfiques de recompensa i *Win Rate* . . . 82
5.10 Discussió de les Hipòtesis . . . 83
5.10.1 Discussió de l'Experiment 1 . . . 83
5.10.2 Discussió de l'Experiment 2 . . . 83
5.10.3 Discussió de l'Experiment 3 . . . 84
5.10.4 Discussió de l'Experiment 4 . . . 84

6 Conclusions **85**
6.1 Anàlisi dels resultats del projecte . . . 85
6.2 Treballs Futurs . . . 86
6.2.1 Línea d'investigació sobre algoritmes d'*Action Advise* . . . 86
6.2.2 Extenció del sistema d'experimentació . . . 86

A Planificació **89**
A.1 Planificació temporal . . . 89
A.1.1 Descripció de tasques . . . 89
A.2 Actualització de la Planificació . . . 96
A.2.1 Justificació de la planificació d'hores . . . 97

B Gestió Económica **101**
B.1 Gestió econòmica . . . 101

B.1.1 Cost de personal 101
B.1.2 Costos Tècnics 101
B.1.3 Contingencies 104
B.1.4 Mecanismes de Control de Gestió del Pressupost 104
B.2 Actualització Gestió Económica 105

C Informació adicional **107**
C.1 Funció de Recompensa 107
C.2 Gràfic de la Incertesa de l'agent `teacher` amb l'*outlier* 108

Índex de figures

2.1 Flux de decisió de SUA. Figura extreta de [14] . . . 31
2.2 Flux de decisió de SUA-AIR. Figura extreta de [14] . . . 31
2.3 Visió general de l'*Introspective Action Advising* . . . 32

4.1 Diagrama de Classes Generat per Doxygen . . . 51

5.1 Corba d'aprenentatge del blog amb 10.000 episodis (És erroni anomenar DDQN a la tècnica emprada; realment, la línia groga correspon a una execució de l'algorisme DQN i la blava a l'algorisme Q-Learning) . . . 65
5.2 Corba d'aprenentatge de Javier Montalvo amb 10.000 episodis . . . 66
5.3 Estudi de la transferència d'aprenentatge de Javier Montalvo amb 3.000 episodis . 66
5.4 Recompença per Episodi . . . 74
5.5 *Win Rate* per Episodi . . . 74
5.6 Recompença per Episodi . . . 76
5.7 Win Rate per Episodi . . . 76
5.8 Recompensa per Episodi . . . 77
5.9 Posició Final per Episodi . . . 77
5.10 *Win Rate* últims 100 episodis per Episodi . . . 77
5.11 *Loss Teacher Loss Student* per Episodi . . . 77
5.12 *Advise* per Espisodi . . . 78
5.13 Valor de λ_t per Espisodi . . . 78
5.14 Total d'*Advise* per Espisodi . . . 78
5.15 Evolució de la incertesa i el llindar d'incertesa del agent `student` per Episodi $\delta = 0.0$ 79
5.16 Evolució de la incertesa i el llindar d'incertesa del agent `teacher` per Episodi $\delta = 0.0$ 79
5.17 Evolució de la incertesa i el llindar d'incertesa del agent `student` per Episodi $\delta = 0.15$ 79
5.18 Evolució de la incertesa i el llindar d'incertesa del agent `teacher` per Episodi $\delta = 0.15$ 79
5.19 Recompensa per Episodi $\delta = 0.0$ . . . 82
5.20 Recompensa per Episodi $\delta = 0.15$ . . . 82
5.21 *Win Rate* per Episodi $\delta = 0.0$ . . . 83
5.22 *Win Rate* per Episodi $\delta = 0.15$ . . . 83

A.1 Diagrama de Gantt GEP . . . 95
A.2 Diagrama de Gantt TFG . . . 100

C.1 Gràfica de la incertesa del professor amb l'*outlier*. . . . 108

Índex de taules

1.1 Objectius del projecte 21

5.1 Descripció de les tècniques d'assessorament 71
5.2 Paràmetres utilitzats 72
5.3 Valors dels hiperparàmetres utilitzats per [3] 72
5.4 Valors de la funció λ per diferents iteracions 72
5.5 Valors de la funció λ per diferents episodis 73
5.6 Resultats de l'Experiment, Mitges de les execución en el nivells 1-1 i 4-1 75
5.7 Metricas Principals Estudi Learning Rate *Action Advise* 76
5.8 Total d'accions recomanades per cada tècnica amb $\delta = 0.0$ i $\delta = 0.15$, i el percentatge d'augment. 78
5.9 Resultats de l'Experiment amb Diferents Models i Paràmetres 80
5.10 Comparacions per *RGB AA Decay* 81
5.11 Comparacions per *RGB SUA* 81
5.12 Comparacions per *RGB TUA* 82
5.13 Comparacions per *RGB TSUA* 82

A.1 Descripció de Tasques 94
A.2 Resum de la Taula descripció de Tasques TFG 96
A.3 Resum de la Taula descripció de Tasques TFG 96

B.1 Salari anual dels diferents rols del projecte, considerant 1750 hores treballades l'any 101
B.2 Resum de costos tècnics 102
B.3 Descripció de Tasques amb els seus costos 103
B.4 Temps estimat i cost per a la gestió de riscos 104
B.5 Temps estimat i cost per a la gestió de riscos 105

Capítol 1

Introducció

En aquest capítol es presenta una visió general del projecte, establint el context, definint conceptes fonamentals i identificant el problema a resoldre. A més, es delimita l'abast del projecte, es descriuen les motivacions, objectius i requeriments, i es detallen els recursos disponibles. Finalment, es fa una actualització de l'abast del projecte al seu final i es considera el context legal aplicable.

1.1 Context

Als últims anys, hem assistit a la gran eclosió de la Intel·ligència Artificial, portant al públic no especialitzat eines que s'han desenvolupat durant les últimes tres dècades. Des dels primers *Bots* a *Twitter* o *Instagram*, fins a *ChatGPT* o *DALL-E*, ens trobem davant d'un nou paradigma per afrontar com es desenvoluparà la humanitat en els pròxims anys. Aaron Bastani, en el seu llibre "*Fully Automated Luxury Communism*"[2], exposa que els avanços tecnològics poden portar a una abundància que alliberi la humanitat del treball. Perquè això pugui arribar a ser possible, o apropar-nos-hi, el desenvolupament de tècniques d'Intel·ligència Artificial cada cop més generals, adaptatives i socialment i ecològicament sostenibles és vital per assegurar que aquest desenvolupament no trunqui el benestar, no només dels éssers humans, sinó de la biodiversitat del planeta.

En relació amb aquestes qüestions, hem plantejat un estudi sobre la *Transfer Learning* entre models d'agent guiats per Intel·ligència Artificial. La millora d'aquestes tècniques deriva beneficis no només en termes d'eficàcia i eficiència, sinó també en termes d'explicabilitat, tres qualitats que, posades al servei del benestar de la societat i del planeta, poden esdevenir claus per a la seva millora.

A la història de la humanitat, transferir l'aprenentatge ha estat clau per arribar a ser el que som avui. La comunicació, des de la verbal fins a la pictòrica, passant per l'escriptura, ha estat una eina essencial per transmetre coneixements. Les primeres escoles van aparèixer fa uns 4.500 anys a Mesopotàmia, i les universitats, com la de Bolonya, fundada el 1088, van ser institucions clau per a la disseminació del coneixement durant l'edat mitjana.

Al segle XV, la invenció de la premsa de Gutenberg va suposar una revolució en la difusió del coneixement, facilitant la producció en massa de llibres. Recentment, la ràdio, la televisió i la programació han estat mitjans poderosos per a la distribució d'informació i l'educació de les masses, emprant formats com l'àudio, el vídeo i els algoritmes per a l'automatització de tasques. La intel·ligència artificial, que va començar a desenvolupar-se com a camp acadèmic a la dècada de 1950, és l'últim esglaó en aquesta cadena de progressos tecnològics.

En aquest context, el *Transfer Learning*, esdevé un mecanisme clau. Aquesta tècnica permet a les màquines aprofitar el coneixement adquirit en una tasca, ja sigui per un humà o per una intel·ligència no humana, per aplicar-lo a una altra tasca relacionada, accelerant així el procés d'aprenentatge i millorant l'eficiència. Des dels seus inicis formals en la comunitat de recerca en intel·ligència artificial a finals de la dècada de 1990, el *Transfer Learning* ha demostrat ser una eina

potent en diversos àmbits, com la visió per computador, el processament del llenguatge natural i l'aprenentatge per reforç.

Per tant, el *Transfer Learning* és un reflex modern de la manera com els humans hem transmès el coneixements al llarg de la història per construir el món d'avui en dia. Des de les primeres formes de comunicació fins a les tècniques més avançades de Intel·ligència Artificial, la capacitat de transferir aprenentatges ha estat i continuarà sent el motor del progrés humà.

1.2 Conceptes Fonamentals

En aquesta secció es presenten i discuteixen alguns dels conceptes fonamentals per entendre aquest treball.

Machine Learning

El *Machine Learning* (ML) és una branca de la intel·ligència artificial (IA) que se centra en el desenvolupament d'algoritmes que permeten a les màquines aprendre a partir de dades i millorar el seu rendiment en tasques específiques sense ser explícitament programades per a cada cas concret. Els algorismes de ML identifiquen patrons i relacions en les dades, els quals són utilitzats per fer prediccions o prendre decisions.

Deep Learning

El *Deep Learning* (DL) és una subcategoria del *Machine Learning* que utilitza xarxes neuronals artificials profundes amb múltiples capes per modelar i resoldre problemes complexos. Aquestes xarxes neuronals profundes poden aprendre representacions de dades de diferents nivells d'abstracció, el que les fa especialment adequades per a tasques com el reconeixement d'imatges, el processament del llenguatge natural i la identificació de veu.

Diferències i Relacions entre ML i DL

- *Machine Learning*: Inclou un ampli ventall d'algoritmes com la regressió lineal, arbres de decisió, màquines de vector de suport (SVM), entre d'altres. Aquests algorismes poden ser supervisats, no supervisats o semi-supervisats.
- *Deep Learning*: Es centra principalment en les xarxes neuronals profundes. La seva estructura composta per múltiples capes li permet processar grans volums de dades i extreure característiques complexes de manera automàtica.

Exemple de Regressió Lineal en ML

Un exemple senzill de *Machine Learning* és la **regressió lineal**, on l'objectiu és ajustar una línia recta que millor s'ajusti als punts de dades. En aquest context:

- **Dades**: Un conjunt de punts (X, Y).
- **Funció de cost**: Mesura l'error entre les prediccions del model i les dades reals. Sovint es fa servir la log-versemblança negativa.
- **Model**: Una equació lineal del tipus $Y = Xw + b$.
- **Optimització**: Ajustament dels paràmetres w i b per minimitzar la funció de cost, que sovint es resol mitjançant mètodes com el descens de gradient.

Deep Learning i les Xarxes Neuronals Artificials

Les xarxes neuronals artificials són tècniques de *Machine Learning* que simulen el mecanisme d'aprenentatge dels organismes biològics. El sistema nerviós humà conté neurones connectades. Aquestes connexions es modifiquen en resposta a estímuls externs, permetent l'aprenentatge.

A les xarxes neuronals artificials, les unitats de càlcul s'anomenen neurones i estan connectades per pesos que actuen com les sinapsis biològiques. Cada entrada a una neurona es multiplica per un pes, afectant la funció calculada. L'aprenentatge es produeix ajustant els pesos en resposta a errors de predicció, amb l'objectiu de millorar les prediccions futures.

Per exemple, les xarxes s'entrenen amb dades d'entrada (com imatges) i les seves etiquetes (com "pastanaga"o "plàtan"). Els errors en les prediccions proporcionen retroalimentació per ajustar els pesos. Aquest procés d'ajust continu permet a la xarxa generalitzar, és a dir, fer prediccions precises sobre dades noves no vistes durant l'entrenament.

La combinació de múltiples unitats de càlcul bàsiques, augmenta la seva capacitat per aprendre funcions complexes. Quan es combinen moltes unitats i s'entrenen conjuntament, les xarxes neuronals poden aprendre funcions més complicades que els models clàssics de machine learning. L'efectivitat de les xarxes neuronals depèn de la combinació adequada d'aquestes unitats i de la disponibilitat de dades suficients per entrenar-les.

Reinforcement Learning

El *Reinforcement Learning* és una tècnica d'intel·ligència artificial on una xarxa neuronal aprèn a prendre accions en situacions dinàmiques i canviants, sense coneixement previ de les accions adequades. Aquest aprenentatge es fa mitjançant recompenses basades en les accions preses.

Un exemple és entrenar un sistema per jugar a un videojoc des de zero, sense conèixer les regles. Els videojocs són bons per al *Reinforcement Learning* perquè simulen situacions de la vida real, amb nombrosos estats possibles i moviments òptims depenent del context. El sistema d'aprenentatge recull dades a través de les seves accions, creant un entorn d'aprenentatge desafiant.

El procés d'aprenentatge es basa en la interacció de l'agent amb l'entorn: l'agent realitza accions, l'entorn respon a aquestes accions amb canvis d'estat i recompenses, i l'agent ajusta les seves polítiques d'acció en funció de l'experiència adquirida. Els components principals d'un sistema d'aprenentatge per reforç són els següents:

- **Agent**: És l'entitat que pren accions en l'entorn per tal de maximitzar la recompensa. L'agent aprèn a partir de les seves experiències a través d'una política que determina les accions a prendre en cada estat.
- **Entorn (Environment)**: És tot allò amb què l'agent interactua. L'entorn proveeix els estats i les recompenses a l'agent en resposta a les accions que aquest pren.
- **Estat (State)**: Representa una configuració del sistema o l'entorn en un moment donat. Els estats proporcionen informació a l'agent sobre l'entorn.
- **Accions (Actions)**: Són les decisions que l'agent pot prendre. Les accions afecten l'estat de l'entorn i, en conseqüència, les recompenses que l'agent rep.
- **Recompensa (Reward)**: És el senyal de retroalimentació que l'agent rep després de prendre una acció en un estat particular. La recompensa informa l'agent sobre l'eficàcia de les seves accions.
- **Política (Policy)**: És la funció que l'agent utilitza per determinar quina acció prendre en cada estat. La política pot ser estocàstica (probabilística) o determinista.
- **Funció de valor (Value Function)**: Estima la recompensa a llarg termini de ser en un estat (o de prendre una acció en un estat) sota una política determinada.
- **Model (opcional)**: En alguns mètodes d'aprenentatge per reforç, l'agent utilitza un model de l'entorn per fer prediccions sobre els resultats de les accions.

L'objectiu principal de l'agent en l'aprenentatge per reforç és aprendre una política òptima que maximitzi la recompensa total esperada al llarg del temps. Això implica explorar diverses accions i estats, així com explotar el coneixement adquirit per prendre decisions òptimes.

Donada la definició anterior, a continuació es presenten alguns exemples de com es poden modelar diversos problemes per solucionar-los entrenant un agent mitjançant l'aprenentatge per reforç.

Control d'un Robot en una Fàbrica

- **Agent**: El robot que ha de realitzar tasques específiques dins de la fàbrica, com moure objectes d'un lloc a un altre.
- **Entorn (Environment)**: La fàbrica amb les seves màquines, passadissos, objectes i altres obstacles.
- **Estat (State)**: La posició actual del robot, l'estat dels objectes (si estan recollits o no), i altres paràmetres com la quantitat de bateria restant.
- **Accions (Actions)**: Moviments del robot (endavant, enrere, girar, recollir objecte, deixar objecte).
- **Recompensa (Reward)**: Punts positius per completar tasques com recollir i moure objectes correctament, i punts negatius per col·lisions o moviments ineficients.
- **Política (Policy)**: Una funció que determina la millor acció a prendre segons l'estat actual per maximitzar la recompensa acumulada.
- **Funció de valor (Value Function)**: Estima la recompensa futura esperada en ser en un estat determinat o en prendre una acció específica.
- **Model del món (opcional)**: Una simulació de la fàbrica que permet predir els resultats de les accions sense realitzar-les en el món real.

Escacs

- **Agent**: El programa d'ordinador que juga a ajedrez.
- **Entorn (Environment)**: El tauler d'ajedrez amb totes les seves peces.
- **Estat (State)**: La disposició actual de les peces al tauler.
- **Accions (Actions)**: Moviments possibles de les peces segons les regles del joc.
- **Recompensa (Reward)**: Punts positius per captures de peces, escacs i escacs mat, i punts negatius per pèrdua de peces o situacions desfavorables.
- **Política (Policy)**: Estratègia de joc que determina el millor moviment a realitzar en cada situació del joc.
- **Funció de valor (Value Function)**: Estima la probabilitat de guanyar des d'una disposició específica del tauler.
- **Model del món (opcional)**: Un motor de joc que simula moviments futurs per avaluar possibles resultats.

Vehicles Autònoms

- **Agent**: El vehicle autònom que ha de conduir de manera segura i eficient.
- **Entorn (Environment)**: Les carreteres, amb altres vehicles, senyals de trànsit, vianants i altres elements del trànsit.
- **Estat (State)**: La velocitat actual, la posició del vehicle, la proximitat a altres vehicles, i informació de sensors sobre l'entorn.
- **Accions (Actions)**: Accelerar, frenar, girar, canviar de carril.
- **Recompensa (Reward)**: Punts positius per mantenir una conducció segura, arribar a la destinació a temps, i punts negatius per infraccions de trànsit o situacions de risc.
- **Política (Policy)**: Algoritme que selecciona les millors accions per conduir de manera segura i eficient.

- **Funció de valor (Value Function)**: Estima la seguretat i eficiència de ser en un estat determinat o prendre una acció específica.
- **Model del món (opcional)**: Simulador de conducció que permet predir les conseqüències de les accions sense posar en risc el vehicle real.

Transfer Learning

El *Transfer Learning* és una tècnica en la qual el coneixement adquirit durant l'aprenentatge d'una tasca específica és reutilitzat com a punt de partida per a l'aprenentatge d'una nova tasca relacionada. Aquesta estratègia es basa en la premissa que diverses tasques poden compartir característiques subjacents o estructura que permeten que les solucions d'una tasca puguin ser parcialment aplicables a una altra. El transfer learning és especialment valuós quan les dades per a la nova tasca són limitades o quan l'aprenentatge des de zero és computacionalment costós o ineficient. Exemples d'Aplicació del Transfer Learning

A continuació es presenten exemples de com s'aplicaria el transfer learning en els exemples d'ús de l'aprenentatge per reforç esmentats anteriorment:

Control d'un Robot en una Fàbrica Si un agent ha estat entrenat per controlar un robot en una línia de producció, aquest coneixement es pot transferir per entrenar un nou robot encarregat de tasques de manteniment. Les habilitats adquirides pel primer robot en navegació i manipulació d'objectes poden ser utilitzades com a base per reduir el temps d'entrenament del segon robot en les seves noves funcions.

Escacs Un agent que ja ha après a jugar a escacs pot utilitzar-se com a base per aprendre a jugar a un altre joc de taula com el shogi. Les estratègies generals i la comprensió tàctica desenvolupades durant l'entrenament en escacs poden ser aplicades per accelerar l'aprenentatge en el nou joc.

Vehicles Autònoms Si un vehicle autònom ha estat entrenat per conduir en entorns urbans, aquest coneixement es pot transferir per entrenar el mateix vehicle a conduir en entorns rurals. Les capacitats desenvolupades en detecció de senyals de trànsit, evasió d'obstacles i navegació urbana poden ajudar a reduir el temps i els recursos necessaris per adaptar el vehicle a les condicions específiques de les carreteres rurals.

Learning from Demonstration i l'*Action Advising*

L'*Learning from Demonstration* (*LfD*) i l'*Action Advising* són dues solucions àmpliament acceptades per al problema d'ineficiència de mostres en els algorismes d'aprenentatge per reforç (*RL*). Aquest problema es refereix a la necessitat d'una gran quantitat de mostres per aprendre una política que resolgui una tasca. Aquesta necessitat és més evident en el *deep RL*, on les xarxes neuronals s'utilitzen com a aproximadors de funcions i requereixen moltes dades per convergir. *LfD* és una tècnica que utilitza demostracions d'experts per iniciar l'aprenentatge dels agents, i normalment es fa de manera *offline*[1]. L'*Action Advising*, en canvi, és una tècnica *online*[2] on un agent *RL* rep consells d'accions d'un altre agent professor o d'un expert humà.

El paradigma de l'*Action Advising* s'utilitza extensament en el marc de treball *student-teacher*. L'agent estudiant és considerat un novell que pot millorar en presència d'un agent professor preentrenat. El professor està disponible per a un nombre limitat d'interaccions, conegut com el pressupost de consells. Torrey et al. [18] van introduir l'*Action Advising* juntament amb diverses heurístiques per decidir com proporcionar els consells, com el *Early Advising* i el *Importance Advising*. També es van explorar les heurístiques basades en la incertesa epistèmica de l'agent estudiant i la novetat del consell per decidir quan demanar consell. Amir et al. [1] van investigar l'*Jointly-Initiated Action Advising*, on tant l'estudiant com el professor poden iniciar la sol·licitud de consell. Da Silva et al. [16] van estendre aquest marc per acomodar múltiples agents, permetentlos entrenar simultàniament utilitzant consells dels altres.

[1] Una tècnica *offline* implica que l'aprenentatge es fa abans de l'entrenament real, utilitzant dades precompilades.
[2] Una tècnica *online* implica que l'aprenentatge es fa durant l'entrenament real, utilitzant dades generades en temps real.

Pel que fa a la reutilització de consells, hi ha pocs treballs que busquen col·lectar consells per a un ús posterior. Això és útil perquè permet a l'estudiant utilitzar el pressupost del professor de manera més eficient. Zhu et al. [20] van introduir la idea de reutilitzar consells col·lectats en un marc *student-teacher* basat en diferents heurístiques com el *Q-Change*, la reutilització de pressupost i la probabilitat de reutilització decreixent en algorismes *tabular RL*. Ilhan et al. [7] [8] van adaptar aquest treball per ser compatible amb el *deep RL*, utilitzant l'aprenentatge supervisat per entrenar un model del professor a partir dels consells col·lectats anteriorment. Aquest model prediu una acció que es pot reutilitzar si es compleix un llindar de probabilitat predeterminat. No obstant això, la limitació del seu treball és la dependència de l'estudiant en el model del professor per a la recollida de consells. Proposem un marc metòdic per abordar aquests problemes.

1.3 Identificació del Problema

En el paper *Introspective Action Advising for Interpretable Transfer Learning* presentat a la CoLLAs 2023, Campbell et al. [3] proposen una aproximació alternativa al *Transfer Learning* basada en *Action Advising*, on un professor, entrenat en una tasca font, guia activament l'exploració d'un estudiant en una tasca objectiu. Aquesta metodologia permet la transferència de coneixement entre polítiques sense importar les representacions subjacents, i s'ha demostrat que millora les taxes de convergència en entorns de Gridworld i Atari, alhora que proporciona una visió sobre quin coneixement ha estat transferit.

En les conclusions del paper, es mostra que la tècnica *Introspective Action Advising* (IAA) és aplicable a altres algorismes, però no hem trobat investigacions que aprofundeixin en aquest aspecte. A més, l'estudi de com trobar un paràmetre de llindar adaptatiu és interessant i presenta una àrea de millora potencial. En l'algorisme original, s'utilitza un paràmetre d'exploració, però aquest pot no ser útil en determinats entorns on l'exploració d'accions aleatòries convé que sigui mínima i per tant decau ràpidament. A més, les conclusions del treball indiquen que l'algorisme pot no ser tan eficient en situacions on la tasca font i la tasca objectiu disten més que les estudiades per ells. Això obre la porta a investigar si l'algorisme funciona millor en tasques més similars, suggerint així una possible relació directa entre l'eficiència d'aquesta tècnica innovadora i la semblança entre les tasques. Per aquest motiu, hem decidit continuar la investigació de l'algorisme IAA per explorar aquestes àrees i optimitzar el seu rendiment en diverses aplicacions pràctiques.

Per aquest estudi, el problema a resoldre és aplicar l'algorisme *Introspective Action Advising* (IAA) per transferir el coneixement d'un agent entrenat en un nivell del joc NES Super Mario a un agent que s'està entrenant en un nivell més complex. Aquesta transferència de coneixement busca millorar l'eficiència de l'agent en el nou nivell, avaluant com la semblança entre la tasca font i la tasca objectiu influeix en l'eficàcia de l'algorisme IAA.

1.4 Abast

1.4.1 Motivació

La realització d'aquest projecte m'ha aportat un primer contacte real amb un projecte d'investigació d'envergadura moderada, cosa que també m'ha ajudat a veure si aquesta és la sortida professional que desitjo. Aquest projecte m'ha permès adquirir experiència pràctica en la gestió i execució de projectes d'investigació, així com desenvolupar habilitats en la resolució de problemes tècnics i en la presa de decisions basades en dades. Aquest aprenentatge marca un abans i un després en el meu creixement professional i personal.

Actualment, el problema que volia abordar es resol amb models més sofisticades i computacionalment més costosos. De tota manera, aquesta investigació obre la porta a valorar l'ús d'aquesta tècnica en models més complexos. Ja que, totes aquestes tècniques compten amb sistemes de control del *Budget Advise*, que es el que nosaltres estem estudiant. A més, estudiar com d'eficient és aquesta tècnica davant d'altres amb una eficàcia provada en l'entorn d'estudi ens permetrà determinar si aquesta tècnica no és adequada per a l'entorn proposat, un resultat que també seria significatiu per a futures investigacions. En l'àmbit social, aquest projecte no només aporta un gra

de sorra a l'estudi d'estratègies innovadores, sinó que també nodreix la comunitat d'investigadors i investigadores amb un entorn flexible i extensible, aprofitable per a qualsevol persona que vulgui seguir aquesta investigació o en vulgui fer-ne de semblants.

El desenvolupament de noves tècniques, així com la millora i perfeccionament de les mateixes, necessita investigadors que les estudiïn i les provin en entorns on els seus creadors no ho han fet. Així es demostren les seves virtuts i carències, obrint noves vies d'investigació i possibilitant que se'n pugui fer un ús productiu. En el nostre cas, la tècnica que estudiem no s'havia investigat prèviament, per tant, hem expandit l'estat de l'art en aquesta matèria. Aquest projecte ha tingut com a objectiu avaluar la viabilitat i l'eficàcia d'aquesta nova tècnica, contribuint així al coneixement i al desenvolupament tecnològic en el camp. A més, aquest estudi proporciona noves perspectives que poden ser aplicades en futurs projectes més sofisticats i extensos, beneficiant tant la comunitat científica com la indústria.

1.4.2 Objectius

Durant el desenvolupament del projecte, es van realitzar ajustaments significatius en els objectius inicials. Inicialment, l'objectiu era explorar la implementació de diverses tècniques de *Transfer Learning* en agents *Deep Q-Network* (DQN). Tanmateix, amb el progrés del projecte, es va descobrir que algunes de les tècniques seleccionades no eren adequades per als problemes que estàvem intentant resoldre.

Això ens va portar a revisar els objectius i a reorientar-los cap a tècniques que demostraven ser més efectives en el context dels agents DQN dins de l'entorn experimental definit. Aquesta actualització va ser essencial per alinear els objectius amb les necessitats específiques i les condicions de l'experimentació.

Els objectius proposats a l'inici del projecte eren els següents:

Codi	Objectiu	Descripció
OP1	Estudiar la tècnica proposada amb l'algorisme DQN	Avaluar l'eficàcia de l'*Introspective Action Advising* quan s'aplica a un algorisme de RL diferent del que s'ha utilitzat en estudis previs.
OP2	Entorn: SuperMario de la llibreria Gym	Utilitzar un entorn conegut i ben definit per estudiar la transferència de coneixement entre nivells.
OP3	Millorar el Llindar d'Introspecció	Ajustar el llindar d'introspecció per optimitzar la transferència de coneixement i minimitzar els errors de predicció.

Taula 1.1: Objectius del projecte

Després de veure que el codi que empravem per realitzar els experiments no era suficientment flexible per permetre'ns fer els experiments amb comoditat, de la mateixa manera que no hagués suportat una extensió del codi per implementar altres tècniques d'aprenentatge per reforç com A2C o PPO, vam decidir estendre els objectius. Justament per aquest motiu, canviàrem la nomenclatura dels objectius de OB_ a OP_ per tractar els objectius principals, els objectius de l'experimentació, i OS_ pels objectius secundaris, els objectius d'implementació. A continuació detallem quins objectius s'han modificat i quins s'han afegit.

- Objectiu Principal:
  - Descripció: Estudiar la tècnica proposada amb l'algoritme DQN.
  - Subobjectius:

    - OP1: Avaluar el rendiment i l'adaptació de la tècnica proposada per Sahir [14] comparant-la amb la perspectiva proposada per Campbell [3] utilitzant l'algoritme DQN.
    - OP2: Analitzar el rendiment d'aquestes tècniques en l'entorn seleccionat.
    - OP3: Proposar una millora en el llindar de recomanació.
  - Justificacions i canvis:
    - OP1: S'han identificat estudis molt relacionats amb els realitzats per Sahir et al. [14] per a un model DQN, i Campbell et al. [3] per a un model PPO. Les tècniques guiades per la incertesa (*Uncertainty Advice*) són molt semblants a la proposta de Campbell et al.[3], amb la diferència que una es basa en la incertesa de l'estudiant (*Student Uncertainty Advice*), mentre que Campbell et al. [3] utilitza el llindar d'introspecció per guiar la incertesa, basant aquest càlcul en el professor.
    - OP2: Determinar el llindar que identifica quines accions es recomanen és un dels aspectes a investigar mencionats en ambdós treballs.

    Com es pot apreciar, tots tres objectius amplien l'estat de l'art en diferents àmbits de la investigació i desenvolupament de tècniques d'*Action Advice*.

- Objectiu Secundari:
  - Descripció: Implementar un sistema de classes que permeti implementar diferents tècniques d'aprenentatge per reforç.
  - Subobjectius:
    - OS1: Possibilitat d'estendre el sistema amb diferents algoritmes d'aprenentatge per reforç, com PPO o A2C.
    - OS2: Possibilitat d'estendre el sistema amb diferents tècniques de *Transfer Learning*.
    - OS3: Possibilitat d'estendre el sistema amb diferents entorns on estudiar les possibles combinacions de les tècniques.
  - Justificació: La implementació d'aquest sistema de classes facilita la integració i l'avaluació de les diverses tècniques d'aprenentatge per reforç en un entorn unificat. Les possibilitats d'estendre el sistema amb algoritmes i tècniques variades, així com amb diferents entorns d'estudi, garanteixen que la plataforma serà flexible i adaptable a diferents necessitats i escenaris de recerca.

Els objectius plantejats, tant principals com secundaris, han estat definits per abordar noves vies d'investigació i millora. L'avaluació comparativa dels diferents enfocaments proposats per als equips d'investigació de Sahir et al. [14] i Campbell et al. [3], proporcionant una visió més clara sobre l'eficàcia d'aquestes tècniques en la transferencia d'aprenentatge entre tàsques molt semblants.

Aquest treball no només busca validar les tècniques existents, sinó que també aspira a establir una base per a futures investigacions en l'àrea de l'aprenentatge per reforç. La implementació d'un sistema flexible i extensible permetrà als futurs investigadors adaptar i expandir les seves eines amb facilitat, promovent així el desenvolupament continu i la innovació en aquest camp. Amb aquests objectius ben definits i una metodologia robusta, s'espera que aquesta recerca faci una contribució valuosa a la comunitat científica i acadèmica.

1.4.3 Requeriments

Per assolir els objectius definits en aquest projecte, s'han identificat diversos requeriments essencials. Aquests requeriments s'organitzen en funció dels objectius principals (OP) i secundaris (OS), assegurant que cada aspecte del projecte estigui alineat amb les necessitats específiques de l'experimentació i la implementació.

1.4.4 Requeriments Principals

RP1: Requisits per a l'Avaluació del Rendiment de la Tècnica Proposada amb l'Algoritme DQN

- **RP1.1**: Integrar la tècnica d'*Introspective Action Advising* en un algorisme DQN.
- **RP1.2**: Desenvolupar un sistema de comparació per avaluar el rendiment de la tècnica proposada per Sahir [14] i Campbell [3].
- **RP1.3**: Implementar mètriques de rendiment per analitzar l'eficàcia de la tècnica en diferents entorns.

RP2: Requisits per a l'Avaluació en l'Entorn Seleccionat

- **RP2.1**: Configurar l'entorn de SuperMario de la biblioteca Gym [6] per permetre la transferència de coneixement entre nivells.
- **RP2.2**: Recopilar i analitzar dades experimentals per validar la transferència de coneixement.

RP3: Requisits per a la Millora del Llindar d'Introspecció

- **RP3.1**: Desenvolupar un algoritme adaptatiu per ajustar el llindar d'introspecció basat en l'anàlisi de rendiment.
- **RP3.2**: Implementar un sistema per identificar accions recomanades en funció del llindar d'introspecció.
- **RP3.3**: Validar la millora del llindar d'introspecció mitjançant experiments comparatius.

1.4.5 Requeriments Secundaris

RS1: Requisits per a la Implementació d'un Sistema de Classes per a Diverses Tècniques d'Aprenentatge per Reforç

- **RS1.1**: Desenvolupar un sistema de classes flexible que permeti la implementació de diferents algoritmes d'aprenentatge per reforç, com PPO o A2C.
- **RS1.2**: Assegurar la compatibilitat del sistema amb diverses tècniques de *Transfer Learning*.
- **RS1.3**: Facilitar l'extensió del sistema per permetre l'estudi en diferents entorns.

RS2: Requisits per a l'Extensibilitat i la Flexibilitat del Sistema

- **RS2.1**: Dissenyar una arquitectura modular que permeti fàcilment afegir nous algoritmes i tècniques.
- **RS2.2**: Implementar interfícies clares per a la integració de noves funcionalitats.
- **RS2.3**: Proporcionar documentació exhaustiva per facilitar la utilització i l'extensió del sistema per part de futurs investigadors.

1.4.6 Recursos

Per al correcte desenvolupament del projecte, es van identificar els següents recursos, classificats en recursos humans, informàtics i d'informació.

Recursos Humans

Els recursos humans inclouen:

- **Director del Projecte**: Responsable de guiar i supervisar el desenvolupament del projecte, oferint suport tècnic i acadèmic, així com revisant els avenços i proporcionant feedback constructiu.
- **Investigador Principal (Jo mateix)**: Responsable de dur a terme la investigació, implementació i anàlisi dels resultats. També s'encarrega de la documentació i presentació final del projecte.
- **Col·laboradors i Mentors**: Altres experts en el camp que puguin proporcionar consells i orientació addicionals durant les reunions setmanals.

Recursos Informàtics

Per dur a terme l'experiment proposat, ha sigut essencial comptar amb els recursos adequats tant de *hardware*, com de *software*. A continuació es detallen els requeriments necessaris, en relació amb els requisits identificats:

- ***Hardware***
  - **Ordinador de Desenvolupament** (per a RP1.1 i RS1.1): Un ordinador personal amb capacitats suficients per a la programació i proves inicials. Es recomana un processador amb almenys quatre nuclis, 16 GB de memòria RAM i una GPU compatible amb CUDA per accelerar les operacions de Deep Learning.
  - **Servidor d'Entrenament** (per a RP1.1, RP2.1, RP3.1 i RS1.2): Un servidor dedicat per a l'entrenament de models de Reinforcement Learning. Les especificacions mínimes inclouen:
    - **CPU**: Processador d'alt rendiment.
    - **GPU**: Almenys una GPU NVIDIA amb suport CUDA, per tal de manejar les càrregues de treball intensives de Deep Learning.
    - **RAM**: Mínim 64 GB de memòria RAM per a assegurar una execució fluida dels entrenaments.
    - **Emmagatzematge**: SSD d'alta velocitat amb almenys 1 TB d'espai per a emmagatzemar models, dades i resultats.
- ***Software***
  - **Llibreries i Entorns de Programació**:
    - **Python 3.8+**: Llenguatge de programació principal per al desenvolupament de l'experiment (necessari per a RP1.1, RP2.1, RP3.1 i RS1.1).
    - **Pytorch**: Llibreria de Deep Learning que proporciona eines per a la creació i entrenament de models de xarxes neuronals. Instal·lació de Pytorch amb suport CUDA per a l'acceleració GPU (per a RP1.1, RP2.1, RP3.1 i RS1.1).
    - **Gym**: Biblioteca per a desenvolupar i comparar algorismes de Reinforcement Learning. Proporciona l'entorn SuperMario (per a RP2.1).
    - **Numpy i Pandas**: Necessàries per a gestionar i analitzar dades de manera ràpida i eficient (per a RP2.1 i RS1.1).

    - **Weights and Biases**: Eina per al seguiment d'experiments de Machine Learning. Facilita el registre, la visualització i l'anàlisi de metadades d'entrenament. És necessari tenir un compte en aquesta plataforma (per a RP3.1 i RS1.1).
    - **Hydra**: Biblioteca de configuració que permet la gestió flexible d'experiments mitjançant fitxers de configuració YAML i línies de comandes. Facilita l'organització i reproducció d'experiments (per a RP1.1, RP2.1, RP3.1 i RS1.1).
  - **Control de Versions** (per a RS1.3): Git per al control de versions del codi i col·laboració en el desenvolupament.
  - **Entorn de Desenvolupament Integrat (IDE)**: Visual Studio Code o PyCharm per facilitar la programació i el depurament del codi (per a RS1.3).

En conjunt, aquests requeriments asseguraran que l'experiment es pugui dur a terme de manera eficient, permetent una implementació i entrenament òptims de l'algorisme *Introspective Action Advising (IAA)* amb DQN en l'entorn de SuperMario.

Recursos d'Informació

Els recursos d'informació inclouen totes les fonts de coneixement necessàries per dur a terme la investigació i desenvolupar el projecte. Aquests inclouen:

- **Google Scholar**: Plataforma de recerca acadèmica per accedir a articles, papers i altres documents acadèmics rellevants (per a RP1.1, RP2.1, RP3.1 i RS1.1).
- **Documentació GEP**: Documents i guies que proporcionen informació sobre els estàndards i les millors pràctiques per a la gestió de projectes (per a RS1.3).
- **Apunts SID**: Material acadèmic i apunts sobre Sistemes d'Informació i Decisió, utilitzats com a referència per al desenvolupament del projecte (per a RP1.1, RP2.1, RP3.1 i RS1.1).

Aquests recursos són essencials per assegurar un desenvolupament fluït i eficient del projecte, proporcionant el suport necessari en cada etapa del procés.

1.5 Actualització de l'Abast al finalitzar el projecte

En aquesta secció es presenten els objectius actualitzats del nostre projecte, dividits en objectius principals i secundaris. A continuació, es detallen els objectius amb el seu abast, els treballs futurs que es derivaran d'ells, i la justificació corresponent per a cadascun. Aquesta revisió ens permetrà clarificar el propòsit de cada objectiu i establir les bases per a futurs desenvolupaments i extencions, tant del projecte, com de la investigació.

Abast, Futurs Treballs i Justificació dels Objectius

- **OP1: Avaluar el rendiment i l'adaptació de la tècnica proposada**
  - **Abast**: S'estudiarà com funcionen les adaptacions de les diferents tècniques en l'entorn seleccionat.
  - **Futurs Treballs**: Serà necessari estudiar aquestes tècniques amb els experiments on totes dues han sigut presentades.
  - **Justificació**: Degut al termini limitat, no és possible dur a terme aquests estudis addicionals en la present investigació.
- **OP2: Analitzar el rendiment d'aquestes tècniques en l'entorn seleccionat**
  - **Abast**: Es podrà completar l'objectiu tal com està plantejat.
  - **Futurs Treballs**: No s'ha identificat la necessitat de treballs futurs addicionals per a aquest objectiu.

  - **Justificació**: L'objectiu es pot complir completament dins del termini de la investigació.

- **OP3: Proposar una millora en el llindar de recomanació**
  - **Abast**: S'implementarà i avaluarà la nostra proposta de millora en el llindar de recomanació.
  - **Futurs Treballs**: No es compararà amb les propostes ja existents.
  - **Justificació**: Es busca establir la nostra proposta perquè quedi constància en la literatura. La secció de metodologia desenvoluparà millor l'abast d'aquest objectiu.

- **OS1, OS2, OS3: Implementar un sistema de classes que permeti implementar diferents tècniques d'aprenentatge per reforç**
  - **Abast**: El codi està dissenyat per suportar tots els objectius plantejats.
  - **Futurs Treballs**: L'extensió del sistema no és trivial i requerirà desenvolupar diferents funcions en diverses parts de l'entorn.
  - **Justificació**: El sistema presentat es capaç de dur a terme les tècniques i algoritmes utilitzats en aquest estudi, deixant la implementació de tècniques addicionals per a futurs treballs.

Els objectius presentats en aquesta taula reflecteixen l'evolució del nostre projecte i les adaptacions que hem hagut de realitzar en resposta a les limitacions temporals i tècniques. Tot i les restriccions, hem assegurat que els aspectes més crítics del nostre estudi siguin abordats, deixant constància clara de les nostres propostes i preparant el camí per a investigacions futures.

1.6 Contexte legal: Lleis i Regulacions

Actualment, els aspectes tècnics d'aquest projecte no estan regulats per cap llei o normativa específica. No obstant això, es interessant en tot projecte informàtic en general i en els projectes que involucren la Intel·ligència Artificial fer un anàlisi dels possibles usos de la tecnología desenvolupada.

Un aspecte important a analitzar es el possible impacte en la privacitat de les persones. En el cas que usos futurs del projecte involucressin dades personals o informació sensible, s'haurien de garantir el compliment de les regulacions pertinents, com el Reglament General de Protecció de Dades (GDPR) de la Unió Europea. Encara que aquest projecte no gestiona dades personals, es manté un compromís amb les millors pràctiques en termes de privacitat i seguretat de la informació.

El GDPR és una llei europea de dades que ha estat implementada dins la legislació espanyola en el marc de la Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD). Aquesta legislació assegura que qualsevol processament de dades personals sigui realitzat de manera lícita, lleial i transparent. La implementació d'aquesta llei a Espanya reforça la protecció dels drets dels individus respecte a les seves dades personals, assegurant que empreses i projectes, com el nostre, compleixin amb estrictes normes de protecció de dades.

A més, és important analitzar els possibles usos de la tecnologia que podrien veure's afectats per la propera llei europea de la Intel·ligència Artificial (AI Act), que s'espera entri en vigor durant 2025. Aquesta legislació establirà un marc regulador per a l'ús de la intel·ligència artificial a la Unió Europea, amb l'objectiu de garantir que les aplicacions d'IA siguin segures, transparents i respectuoses amb els drets fonamentals. Segons aquesta llei, els sistemes d'IA es classificaran segons el seu risc potencial, i es preveuen normes estrictes per als sistemes d'alt risc.

En conclusió, aquest projecte es basa en una metodologia sòlida i ben justificada, que inclou un enfocament modular per facilitar l'extensibilitat i la integració de coneixements interdisciplinaris per oferir solucions innovadores. A més, es manté una atenció conscient a les possibles implicacions legals, assegurant que el treball compleixi amb les regulacions aplicables en qualsevol moment.

1.7 Estructura de la Memòria

A continuació, es presenta una descripció detallada de l'estructura de la memòria, que està organitzada en diversos capítols. Cada capítol aborda aspectes específics de la investigació, des de la introducció del context fins a les conclusions i treballs futurs.

1.7.1 Introducció

Aquest capítol introdueix el projecte, establint les bases per a la seva comprensió. Inclou la justificació del context, els conceptes fonamentals necessaris, la identificació del problema que es vol resoldre, l'abast del projecte, i els objectius plantejats. També es detallen els requeriments i recursos necessaris per dur a terme el projecte, i es discuteix el context legal aplicable.

1.7.2 Fonaments Teòrics

En aquest capítol es presenten els fonaments teòrics del projecte, revisant l'estat de l'art en l'àmbit de l'aprenentatge per reforç i la transferència d'aprenentatge. S'hi descriuen els models d'agents i els algoritmes més rellevants, es recapitulen els conceptes clau i es presenta la proposta específica del treball.

1.7.3 Metodologia

Aquest capítol descriu la metodologia seguida per a la implementació i experimentació del projecte. Inclou el mètode utilitzat, les tècniques de monitorització i rigor emprades, i la metodologia d'implementació, que es divideix en fases d'encapsulament, refinament i experimentació. També es detalla la metodologia d'experimentació, justificant l'entorn i les pràctiques de monitorització.

1.7.4 Implementació del sistema d'Experimentació

En aquest capítol es descriu el sistema *software* d'experimentació desenvolupat en `python` utilitzat per a dur a terme el projecte. Es proporciona una visió general del sistema de classes i es detallen les funcions principals, els processos d'entrenament i les funcions comunes i exclusives per a cada tipus d'entrenament, així com les classes auxiliars.

1.7.5 Experimentació i Resultats

Aquest capítol defineix i descriu els experiments realitzats per avaluar les tècniques proposades. S'explica el nombre d'episodis utilitzats, els experiments planificats, i les tècniques emprades com a línies base (DQN, FT, SS). Es presenta l'elecció dels paràmetres i els resultats obtinguts en cada estudi, incloent el refinament DQN, el drop out, i les tècniques d'*Action Advise*. Finalment, es proporcionen interpretacions detallades de les mètriques experimentals i es discuteixen les hipòtesis plantejades.

1.7.6 Conclusions

Aquest capítol conclou el projecte resumint els resultats obtinguts i avaluant si s'han complert els objectius establerts. Es discuteix l'eficàcia de les tècniques proposades i s'ofereix una reflexió sobre el procés d'implementació. A més, es presenten possibles direccions per a treballs futurs, suggerint millores i ampliacions que podrien portar a noves investigacions i desenvolupaments en aquest camp.

Capítol 2

Fonaments Teòrics

En aquest capítol es proporcionen els conceptes teòrics i el context necessari per a comprendre les bases sobre les quals es desenvolupa aquest projecte. Es presenten les investigacions recents en els camps rellevants, així com les contribucions teòriques sobre les que hem construït la proposta del projecte.

2.1 Estat de l'art

2.1.1 Models d'agents per a l'aprenentatge per reforç

En l'aprenentatge per reforç el procés d'aprenentatge es basa en la idea d'optimitzar una política, que és un conjunt de regles o estratègies que l'agent segueix per decidir quina acció prendre en cada situació. Entre els models d'agents més utilitzats, trobem DQN, A2C i PPO, cadascun presenta diferens enginys per abordar l'aprenentatge. En aquesta secció farem un repàs general de les intucions en les que es basen aquests models.

DQN (*Deep Q-Network*)

El DQN (Deep Q-Network) es va introduir en el paper titulat *"Playing Atari with Deep Reinforcement Learning"* per Mnih et al. (2013) [12]. És un model que utilitza xarxes neuronals profundes per estimar la funció de estat-acció-valor Q. Aquesta funció Q és una predicció de la recompensa futura que un agent pot esperar obtenir en prendre una acció específica en un estat determinat. La clau de l'èxit del DQN rau en dues tècniques innovadores: la memòria d'experiència i la *target network*.

La memòria d'experiència és un mecanisme que emmagatzema les experiències passades de l'agent, que consisteixen en tuples d'estat, acció, recompensa i nou estat. Aquestes experiències permeten reutilitzar dades antigues i reduir la correlació entre les mostres d'entrenament. D'altra banda, la *target network* és una còpia de la xarxa Q que s'actualitza periòdicament i proporciona objectius més estables durant l'entrenament. Aquesta combinació de tècniques fa que el DQN sigui especialment adequat per a problemes amb espais d'acció discrets i entorns estables.

A2C (*Advantage Actor-Critic*)

El model A2C (*Advantage Actor-Critic*) es descriu en el paper *"Asynchronous Methods for Deep Reinforcement Learning"* per Mnih et al. (2016) [11]. Combina dos components principals: un actor que pren decisions i un crític que avalua aquestes decisions. L'actor genera una política, que és una distribució de probabilitats sobre les accions que pot prendre en cada estat. Aquesta política és el criteri que segueix l'agent per decidir les seves accions.

El crític, en canvi, avalua la qualitat de les accions preses per l'actor, utilitzant la funció de valor. Aquesta funció de valor mesura quant de bé o malament es va comportar l'actor en relació amb una

acció específica. Aquesta avaluació contínua permet a l'actor ajustar la seva política de manera que maximitzi les recompenses futures. La capacitat de separar la presa de decisions i l'avaluació fa que l'A2C tingui un bon rendiment, especialment, en entorns amb espais d'acció continus i on es necessita una avaluació contínua de les accions.

PPO (*Proximal Policy Optimization*)

El model PPO (*Proximal Policy Optimization*) es va presentar en el paper *"Proximal Policy Optimization Algorithms"* per Schulman et al. (2017) [15]. Representa una millora significativa en l'àmbit de l'aprenentatge per reforç, ja que optimitza la política amb restriccions per assegurar que els canvis siguin petits i segurs. Aquesta característica és crucial per mantenir l'estabilitat durant l'entrenament i per evitar que l'agent faci canvis massa dràstics que podrien resultar en un comportament subòptim.

El PPO utilitza dues tècniques principals per aconseguir-ho: el *clipping* i la proximitat de polítiques. El *clipping* restringeix les actualitzacions de la política, limitant la magnitud del canvi que es pot fer en cada pas d'entrenament. Això assegura que les noves polítiques no s'allunyin massa de les antigues, mantenint l'estabilitat. La proximitat de polítiques manté la nova política propera a l'antiga, millorant encara més l'estabilitat i la robustesa de l'entrenament. Aquest enfocament fa que el PPO sigui ideal per a una àmplia varietat de problemes, ja que combina els avantatges de models com el DQN i l'A2C, però amb una major estabilitat durant l'entrenament.

2.1.2 Algoritme per a la Transferencia d'Aprenentatge

Action Advise

L'assessorament d'acció en el marc estudiant-professor implica que un agent estudiant, novell en la tasca, millori gràcies a l'ajuda d'un agent professor ja entrenat. Aquest assessorament es limita a un nombre d'interaccions, conegut com el pressupost d'assessorament.

Els mètodes d'assessorament varien: poden ser iniciats pel professor o per l'estudiant, que demana ajuda quan té incertesa. Per reduir la càrrega del professor, es pot utilitzar un sistema conjunt on ambdós agents poden començar la interacció. També existeixen marcs on diversos agents es poden assessorar mútuament sense rols fixos. Finalment, algunes metodologies utilitzen funcions objectiu per decidir els moments i el contingut de l'assessorament, en lloc de basar-se en heurístiques fixes.

Uncertainty Action Advice

En l'àmbit de l'aprenentatge per reforç, el maneig de la incertesa és un aspecte crucial per millorar l'eficiència de l'aprenentatge. Shair et al. [14] repassen diferents tècniques en l'estudi de la incertesa en el paradigma *teacher-student*, on l'agent professor proporciona consells a l'agent estudiant basant-se en diversos criteris d'incertesa. Aquestes tècniques tenen com a objectiu principal augmentar la taxa d'aprenentatge de l'agent estudiant mentre es minimitza la dependència del professor, optimitzant així l'ús dels recursos disponibles.

(SUA) *Student Uncertainty-driven Advising*

L'algoritme *SUA* (*Student's Uncertainty-driven Advising*) utilitza les estimacions d'incertesa de l'agent estudiant per impulsar el procés de recollida de consells. Aquest mètode és una resposta a la necessitat de fer que els agents estudiants siguin més eficients en la sol·licitud d'ajuda, demanant consell només quan realment ho necessiten. En aquest enfocament, l'agent estudiant pot demanar consell a l'agent professor quan la seva incertesa, u_s, és superior a un llindar adaptatiu d'incertesa, *c1*. Això permet que l'agent estudiant sol·liciti consell només quan no està segur, optimitzant així l'ús del pressupost de consells disponible, b. Si l'agent professor encara té consells disponibles (és a dir, b ¿ 0), proporcionarà una acció consellada, a_t. Si no s'obté cap consell (per exemple, quan l'agent estudiant està segur o el pressupost de consells s'ha esgotat), l'agent estudiant continuarà seguint la seva pròpia política. Aquesta metodologia permet una col·lecció de consells més efectiva

i evita la dependència innecessària de l'agent estudiant en l'agent professor, millorant així la seva capacitat per generalitzar en l'espai d'estats.

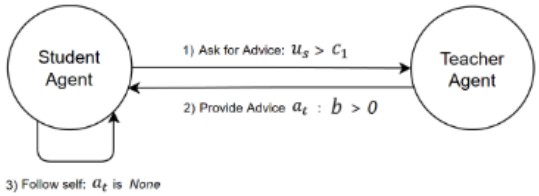


Figura 2.1: Flux de decisió de SUA. Figura extreta de [14]

(SUA-AIR) *Student Uncertainty-driven Advising with Advice Imitation & Reuse*

L'algoritme *SUA-AIR* (*Student's Uncertainty-driven Advising with Advice Imitation & Reuse*) amplia el concepte de *SUA* incorporant la reutilització del consell del model de l'agent professor. Aquesta tècnica no només permet que l'agent estudiant sol·liciti consell quan la seva incertesa és alta, sinó que també permet la reutilització dels consells anteriors, millorant així l'eficiència general de l'aprenentatge. En aquest enfocament, l'agent estudiant pot demanar consell a l'agent professor de manera similar a *SUA*, però si no s'obté consell (per exemple, quan el pressupost de consells s'ha esgotat), l'agent estudiant pot sol·licitar la reutilització del consell del model de l'agent professor. La reutilització del consell es permet només si la incertesa del model de l'agent professor, *u_m*, és inferior al seu propi llindar d'incertesa, *c2*, i si la probabilitat de reutilització, *p*, és major que una probabilitat aleatòria. Aquest mecanisme permet a l'agent estudiant obtenir consells de manera més independent i eficient, fent ús del model del professor quan sigui apropiat.

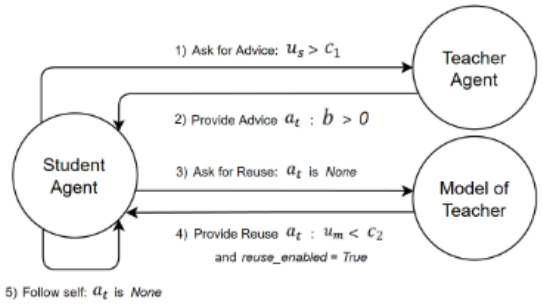


Figura 2.2: Flux de decisió de SUA-AIR. Figura extreta de [14]

Els algorismes *SUA* i *SUA-AIR* presenten un avanç significatiu en l'ús de la incertesa per a la col·lecció i reutilització de consells en entorns d'aprenentatge per reforç. *SUA* serveix com a mètode de referència per als nostres experiments, mentre que *SUA-AIR* aprofita la reutilització de consells per permetre que l'agent estudiant sigui més autònom en la sol·licitud de consells, en lloc de recollir consells aleatòriament o dependre exclusivament del model del professor.

Introspective Action Advising (IAA)

Campbell et al. [3] proposen un nou enfocament per a l'aprenentatge per transferència en el context de l'aprenentatge profund per reforç que es realitza dinàmicament durant el procés d'entrenament del model objectiu. Inspirat en el paradigma *teacher-student* en el consell d'accions (*Action Advising*), el seu mètode introdueix un professor que té accés al model font i transfereix selectivament consells en forma de parelles estat-acció a un estudiant mentre aquest s'està entrenant en la tasca objectiu.

A diferència de les tècniques tradicionals que es basen en heurístiques temporals simples, el mètode de Campbell et al. [3] introdueix un professor intel·ligent que utilitza la introspecció per deter-

minar quan el seu consell seria útil. Aquesta introspecció es realitza contrastant la recompensa esperada per una acció en la tasca font amb la recompensa observada en la tasca objectiu. Si la diferència entre aquestes recompenses és petita, el consell es considera transferible i es proporciona a l'estudiant.

Avantatges del Mètode

En l'estudi conclouen que, aquest efocament te diversos avantatges respecte al *fine-tuning*:

- El coneixement es transfereix selectivament, permetent distingir entre el coneixement específic de la tasca i el coneixement generalitzable.
- El coneixement transferit és interpretable, ja que les parelles estat-acció proporcionen una visió clara del consell donat.
- El mètode és agnòstic respecte a les arquitectures dels models subjacents, ja que el coneixement es transfereix en termes de parelles estat-acció.

Resultats Empírics

Campbell et al. [3] que aquest enfocament condueix a millores en les taxes de convergència respecte al *fine-tuning* entre tasques en entorns *Gridworld* i *Atari*, permetent alhora obtenir insights qualitatius sobre on la transferència és útil.

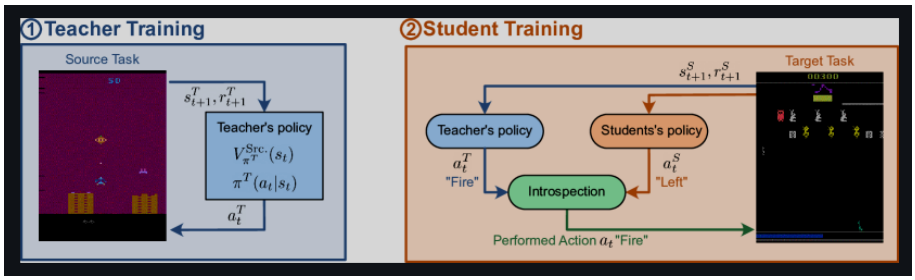


Figura 2.3: Visió general de l'*Introspective Action Advising*

2.2 Recapitulació i Introducció de la Proposta

En la secció anterior, hem revisat els principals models d'agents per a l'aprenentatge per reforç i diverses tècniques avançades que s'han proposat per millorar l'eficiència de l'aprenentatge, especialment en entorns amb limitacions de mostres. Hem abordat els models DQN, A2C i PPO, destacant les seves característiques clau i avantatges en diferents situacions. A més, hem discutit diverses tècniques d'aprenentatge per transferència, com el *Action Advising*, i hem presentat els mètodes *SUA* i *SUA-AIR*, que utilitzen la incertesa de l'estudiant per optimitzar la col·lecció i reutilització de consells.

La proposta més recent, *Introspective Action Advising (IAA)*, desenvolupada per Campbell et al. [3], destaca per la seva capacitat de transferir coneixement de manera dinàmica durant l'entrenament del model objectiu. Aquest mètode introdueix un professor intel·ligent que utilitza la introspecció per determinar quan el consell seria útil, basant-se en la comparació entre la recompensa esperada en la tasca font i la recompensa observada en la tasca objectiu.

Els mètodes existents presenten diversos avantatges:

- **DQN**: Utilitza la memòria d'experiència i la *target network* per a una estimació més estable de la funció Q, adequat per a problemes amb espais d'acció discrets i entorns estables.
- **A2C**: Separa la presa de decisions i l'avaluació, permetent ajustar contínuament la política per maximitzar les recompenses futures, especialment en entorns amb espais d'acció continus.
- **PPO**: Combina els avantatges de DQN i A2C amb una major estabilitat durant l'entrenament, gràcies al *clipping* i la proximitat de polítiques.

- **Action Advising**: Permet a un agent estudiant millorar gràcies a l'ajuda d'un agent professor, utilitzant tècniques com *SUA* i *SUA-AIR* per optimitzar la col·lecció i reutilització de consells.
- **IAA**: Selecciona i transfereix coneixement de manera intel·ligent, diferenciant entre coneixement generalitzable i específic de la tasca, amb resultats empírics que mostren una millora significativa en les taxes de convergència.

Basat en les tècniques revisades, proposem una nova metodologia que integra els avantatges de *SUA*, *SUA-AIR* i *IAA*. La nostra proposta es centra en utilitzar la incertesa tant de l'estudiant com del professor per optimitzar no només la col·lecció i reutilització de consells, sinó també la transferència dinàmica de coneixement durant l'entrenament. Aquesta integració busca maximitzar l'eficiència de l'aprenentatge i la generalització en una àmplia varietat de tasques i entorns.

En la següents seccions, detallarem la nostra proposta, secció 2.3, incloent els algorismes desenvolupats, secció 5.2, les estratègies per estudiar els entrenaments, secció 3.3, i els resultats esperats, secció 5. Ens centrarem en com aquesta combinació de tècniques pot superar les limitacions dels mètodes existents i oferir una noves solucions per a l'aprenentatge per reforç en entorns complexes.

2.3 La nostre proposta

L'Introspective Action Advising permet a l'agent `teacher` ajustar la seva estimació de la funció de valor de l'estat en la tasca objectiu utilitzant les exploracions de l'agent `student`. El procés implica utilitzar una còpia de la funció de valor de l'estat de l'agent `teacher` de la tasca font, que es refina amb les recompenses observades en la tasca objectiu. Per al càlcul del pressupost de consells, hem utilitzat la mateixa estratègia proposada originalment, mantenint els paràmetres de decaïment (λ) i cremada (δ).

D'altra banda, la mesura d'incertesa s'ha adaptat al model d'agent que estem utilitzant, ajustant la funció estat-acció-valor Q en comptes de la funció de estat-valor V. Finalment, el llindar d'incertesa és una proposta pròpia que resol la limitació d'utilitzar un llindar fix. Utilitzar un llindar fix implica que ha de ser estudiat per a cada entorn específic, mentre que el nostre llindar adaptatiu proporciona una solució més flexible i adaptativa.

2.3.1 *Budget Advise*

- **Paràmetre de Decaïment**: Introducció d'un hiperparàmetre de decaïment (λ) per ajustar la probabilitat d'emetre consells al llarg del temps.
- **Equilibri d'Observacions i Consells**: Hiperparàmetre de cremada (δ) per equilibrar entre les observacions que el agent `teacher` ha d'incorporar abans d'emetre consells i la necessitat de donar consells aviat en l'entrenament per obtenir el màxim efecte.

La formulació del càlcul del paràmetre de decaïment és la següent:

$$\lambda_t = \lambda^{t-\delta}$$

2.3.2 Mesura de Incertesa

- **Estimació de la Funció de Valor**: El agent `teacher` refina la seva estimació de la funció de valor de l'estat en la tasca objectiu utilitzant les exploracions de l'agent `student`.
- **Ajust de la Funció de Valor**: S'utilitza una còpia de la funció de valor de l'estat del agent `teacher` de la tasca font, afinada amb les recompenses observades en la tasca objectiu.

La formulació del càlcul de la funció de valor és la següent:

$$|V^{\text{new}}_{\pi_T}(s_t) - V^{\text{src}}_{\pi_T}(s_t)| \leq \epsilon$$

Nosaltres, al estar utilitzant DQN, hem utilitzat la funció estat-acció Q en comptes de la V i reformulem la formulació de la següent manera:

$$|Q^{\text{new}}_{\pi_T}(s_t, a_t) - Q^{\text{src}}_{\pi_T}(s_t, a_t)| \leq \tau_t$$

2.3.3 Llindar d'Incertesa

A la proposta presentada en el paper, s'utilitza un llindar fix (ϵ). Si la incertesa és menor a ϵ, es recomana l'acció. Un dels punts a millorar esmentats a les conclusions del paper és no utilitzar un llindar adaptatiu.

Per aquest motiu, nosaltres hem utilitzat un llindar adaptatiu basat en la mitjana dels últims 100 valors d'incertesa més la desviació estàndard multiplicada per λ. La formulació és la següent:

$$\mu_t = \frac{1}{100} \sum_{i=t-100}^{t} U_i$$

$$\sigma_t = \sqrt{\frac{1}{100} \sum_{i=t-100}^{t} (U_i - \mu_t)^2}$$

$$\tau_t = \mu_t + \lambda \sigma_t$$

On:

- μ_t és la mitjana dels últims 100 valors d'incertesa.
- σ_t és la desviació estàndard dels últims 100 valors d'incertesa.
- τ_t és el llindar adaptatiu utilitzat per decidir si es recomana una acció.

D'aquesta manera, el llindar d'incertesa s'ajusta dinàmicament basant-se en el comportament recent de l'agent, proporcionant una manera més adaptativa i potencialment més eficaç de donar consells.

2.3.4 Comparació dels Algoritmes

Com es pot apreciar a l'Algorisme 1, l'algorisme *Introspect* original es basa en la comparació de la funció de valor V entre l'estat actual i una estimació prèvia del agent `teacher`. La probabilitat de donar consells (λ) decreix amb el temps, ajustada amb un paràmetre de cremada (δ). Si la diferència entre les estimacions de la funció de valor és menor que un llindar fix (ϵ), es dona el consell.

Com es pot apreciar a l'Algorisme 1, l'algorisme *Introspect* original es basa en la comparació de la funció de valor V entre l'estat actual i una estimació prèvia del agent `teacher`. La probabilitat de donar consells (λ) decreix amb el temps, ajustada amb un paràmetre de cremada (δ). Si la diferència entre les estimacions de la funció de valor és menor que un llindar fix (ϵ), es dona el consell.

Algorithm 1 Introspect

Input: s_t, $V_{\pi_T}^{\text{src}}$, $V_{\pi_T}^{\text{new}}$, t, ϵ, λ
$h_t \leftarrow 0$
$p \sim \text{Bern}(\lambda^{\max(0,t-\delta)})$
if $t > \delta$ ***and*** $p = 1$ **then**
  if $|V_{\pi_T}^{new}(s_t, a_t) - V_{\pi_T}^{src}(s_t, a_t)| \leq \epsilon$ **then**
    $h_t \leftarrow 1$
  end
end
return h_t

Com es pot apreciar a l'algorisme 2, la diferència principal entre els dos algorismes es troba la condició del `if` més inten, on hem adaptat l'algoritme *Introspect* per utilitzar la funció estat-acció Q en lloc de la funció de valor V, per ajustar-se millor al model d'agent que estem utilitzant.

Algorithm 2 Introspect-Q

Input: s_t, $Q_{t\pi_T}^{\text{src}}$, $Q_{t\pi_T}^{\text{new}}$, t, μ, σ, λ
$h_t \leftarrow 0$
$p \sim \text{Bern}(\lambda^{\max(0,t-\delta)})$
if $t > \delta$ ***and*** $p = 1$ **then**
  $\tau_t \leftarrow \mu_t + \lambda\sigma_t$
  if $|Q_{t\pi_T}^{new}(s_t, a_t) - Q_{t\pi_T}^{src}(s_t, a_t)| \leq \tau_t$ **then**
    $h_t \leftarrow 1$
  end
end
return h_t

En la nostra proposta, introduïm un llindar d'incertesa adaptatiu basat en la mitjana i la desviació estàndard dels últims 100 valors d'incertesa, millorant així la capacitat d'adaptació i la precisió de l'algoritme en diversos entorns. La definició dels valors dels paràmetres està explicada a la secció de metodologia .

Capítol 3

Metodologia

En aquest capítol, es descriu detalladament la metodologia seguida per dur a terme el projecte. Es presenta el mètode utilitzat, les eines i tècniques de monitorització i rigor emprades, i es detalla la metodologia d'implementació, dividida en fases d'encapsulament, refinament i experimentació. També s'explica la metodologia d'experimentació, incloent la justificació de l'entorn i les pràctiques de monitorització i rigor aplicades durant els experiments.

3.1 Mètode

Aquesta secció detalla la metodologia seguida per dur a terme la nostra investigació. Es va partir del *software* d'entrenament desenvolupat per Javier Montalvo [13]. Aquest codi estava dissenyat per a l'entorn de SuperMario utilitzant la llibreria GYM de *python*, que permetia realitzar entrenaments amb l'algorisme *Fine-Tuning* i, si es desitjava, modificar la visualització de l'estat de l'entorn de l'agent mitjançant l'algorisme de *Semantic Segmentation* utilitzant una *ResNet50*.

Les tasques dutes a terme per al desenvolupament de la investigació es van realitzar en el següent ordre:

- **Preparació del Codi Base**: S'ha adaptat el codi desenvolupat per Javier Montalvo per a l'experiment, assegurant que el codi podia executar tant l'algorisme *Fine-Tuning* com l'ús de la *ResNet50* per l'algorisme *Semantic Segmentation*.
- **Integració de les Tècniques AA**: S'ha integrat la tècnica *Introspective Action Advising (IAA)* en l'agent DQN, modificant l'algorisme per permetre l'ús de consells d'introspecció durant l'entrenament.
- **Recollida de Dades amb WandB**: S'ha configurat la plataforma *Weights and Biases* (WandB) per recollir les dades d'entrenament i els resultats de manera detallada, permetent una monitorització en temps real de les mètriques d'entrenament.
- **Refinament de Paràmetres**: S'han definit i refinat els paràmetres dels agents DQN i alguns de l'algorisme IAA.
- **Procés d'Entrenament**: S'han descrit les condicions d'entrenament, descrit les hipòtesis i s'han realitzat els entranaments amb les diferents tècniques i configuracións proposades.
- **Avaluació i Anàlisi de Resultats**: S'han utilitzat mètodes d'avaluació dels resultats obtinguts.

3.1.1 Monitorització i rigor

Per a la correcta gestió i monitorització del nostre projecte, es van utilitzar diverses eines i tècniques que van permetre mantenir una estructura ordenada i eficient. Aquestes eines van incloure GitHub, Weights and Biases (WandB), Hydra i reunions setmanals amb els directors del projecte.

GitHub El codi del projecte es va gestionar mitjançant un repositori a GitHub. Es van utilitzar dues branques principals: **develop** i **main**.

- **Branch develop**: Aquesta branca es va utilitzar per al desenvolupament actiu i la implementació de noves característiques i millores. Tots els canvis i actualitzacions es feien primer a aquesta branca.

- **Branch main**: Aquesta va ser la branca de producció, que contenia el codi estable i provat. Només els canvis completament verificats i funcionals es fusionaven a la branca **main**.

Aquesta estratègia va permetre una clara separació entre el codi en desenvolupament i el codi preparat per a producció, minimitzant el risc d'introduir errors a la branca principal del projecte.

Weights and Biases (WandB) Per a la monitorització i la recollida de dades, es va utilitzar la plataforma *Weights and Biases* (WandB). Es van crear dos projectes dins de WandB per mantenir un seguiment clar de les diferents etapes del nostre treball:

- **Projecte RLDPTL SuperMario**: Aquest projecte es va destinar a la monitorització dels experiments definitius, que es van realitzar amb el codi i els paràmetres ja optimitzats.

- **Projecte RLDPTL SuperMario Develop**: Aquest projecte es va fer servir durant la fase de desenvolupament i prova, permetent-nos experimentar amb diferents configuracions i metodologies sense afectar les dades del projecte principal.

La separació dels projectes en WandB va permetre tenir un registre clar i ordenat del procés de desenvolupament, així com dels resultats finals dels experiments.

Hydra Es va utilitzar Hydra per gestionar el pas de paràmetres i l'organització dels models preentrenats. Hydra va permetre:

- **Configuració Dinàmica**: Gestionar diferents configuracions de manera eficient i flexible, permetent canviar paràmetres fàcilment sense modificar el codi base.

- **Organització dels Models Preentrenats**: Mantenir una estructura ordenada dels models preentrenats, facilitant-ne l'ús i la comparació durant els experiments.

L'ús de Hydra va proporcionar una gestió més eficaç i adaptativa dels nostres experiments, millorant la nostra capacitat d'optimització i ajustament de paràmetres.

Reunions Setmanals A més de les eines tecnològiques, es van realitzar reunions setmanals amb els directors del projecte. Aquestes reunions van ser crucials per a:

- **Monitorització del Procés**: Permetre una revisió constant de l'avanç del projecte i assegurar que es complissin els objectius establerts.

- **Solució de Dubtes Puntuals**: Proporcionar un espai per a la resolució de dubtes específics que poguessin sorgir durant el desenvolupament.

- **Guia de la Investigació**: Rebre orientació i consells dels directors per mantenir l'alineament del projecte amb els objectius de recerca i les millors pràctiques del camp.

L'us aquestes eines i pràctiques han estat fonamentals per assegurar una gestió eficient, un seguiment acurat i un bon desenvolupament del projecte, garantint l'alineament constant amb els objectius de establerts.

3.2 Metodologia d'implementació

3.2.1 Codi Base: Descripció General de les Classes i Funcions

Classes

- **MaxAndSkipEnv**: Aquesta classe és un `gym.Wrapper` que modifica l'entorn per retornar només cada `skip`-èssim fotograma. Això ajuda a reduir el nombre de fotogrames que s'han de processar, millorant l'eficiència del sistema i permetent un entrenament més ràpid.
- **ProcessFrame84**: Aquesta classe és un `gym.ObservationWrapper` que redimensiona les imatges a 84x84 píxels. També aplica segmentació semàntica si està configurat, la qual cosa pot millorar el rendiment de l'agent en detectar elements clau en l'entorn. Si no s'utilitza la segmentació semàntica, les imatges es converteixen a escala de grisos.
- **ImageToPyTorch**: Aquesta classe és un `gym.ObservationWrapper` que transforma les imatges a format `float32` i mou els canals de color al primer eix, adequant-les per ser utilitzades amb PyTorch. Això simplifica el processament de les dades d'entrada a la xarxa neuronal.
- **ScaledFloatFrame**: Aquesta classe és un `gym.ObservationWrapper` que normalitza els valors dels píxels de les imatges per estar en l'interval de 0 a 1. Aquesta normalització pot ajudar a millorar l'estabilitat i l'eficàcia de l'entrenament del model.
- **BufferWrapper**: Aquesta classe és un `gym.ObservationWrapper` que acumula les últimes observacions al llarg de l'eix del canal. Això permet que l'agent tingui en compte l'historial recent de fotogrames, proporcionant un context temporal que pot millorar la presa de decisions.
- **DQNSolver**: Aquesta classe defineix el model de xarxa neuronal utilitzat per l'agent DQN. Implementa una xarxa convolucional seguida de capes totalment connectades per processar les observacions de l'entorn i produir les accions òptimes. També inclou mètodes per obtenir les activacions i els gradients de les activacions, que poden ser útils per a la depuració i l'anàlisi del rendiment del model.
- **DQNAgent**: Aquesta classe implementa l'agent que utilitza el model DQN per prendre decisions i interactuar amb l'entorn. Gestiona la memòria d'experiències, l'estratègia d'exploració-explotació, l'actualització dels pesos del model i l'entrenament mitjançant l'experiència repetida. També inclou funcionalitats per carregar models preentrenats i guardar l'estat de l'agent.
- **Segmentator**: Aquesta classe s'encarrega de la segmentació semàntica de les imatges. Utilitza un model preentrenat de segmentació profunda basat en `DeepLabV3` amb una arquitectura `ResNet50`. La classe carrega el model, processa les imatges d'entrada i retorna les imatges segmentades o les etiquetes de segmentació. Això permet a l'agent treballar amb representacions més abstractes de l'entorn, millorant potencialment la seva capacitat per identificar objectes i prendre decisions informades.

Funcions Globals

- `make_env(env)`: Crea i retorna l'entorn, embolicat amb totes les capes de preprocessament definides.
- `vectorize_action(action, action_space)`: Donada una acció escalar, retorna una acció codificada en one-hot.
- `show_state(env, ep=0, info="")`: Mostra l'estat actual utilitzant OpenCV.
- `run(training_mode, pretrained)`: Funció principal que executa l'entrenament o la prova de l'agent DQN en l'entorn de Super Mario Bros.

3.2.2 Fase 1 de la Implementació: Encapsulament

En aquesta fase, es van encapsular diverses classes i funcions en diferents fitxers per millorar l'organització del codi i facilitar el manteniment. Les modificacions realitzades van ser mínimes per assegurar que cada classe i funció s'adaptés correctament a la nova estructura dels fitxers.

Encapsulació de la Classe `Segmentator`

La classe `Segmentator` es va encapsular al fitxer `X_segmentator.py`. Aquesta classe s'encarrega de la segmentació semàntica de les imatges utilitzant un model preentrenat basat en `DeepLabV3` amb una arquitectura `ResNet50`.

Encapsulació de les Classes de l'Entorn

Les classes `MaxAndSkipEnv`, `ProcessFrame84`, `ImageToPyTorch`, `ScaledFloatFrame` i `BufferWrapper`, juntament amb les funcions `make_env(env)` i `show_state`, es van encapsular al fitxer `Y_environment_wrappers.py`. Aquestes classes i funcions són responsables de preprocessar les imatges i gestionar l'entorn de `Super Mario Bros`.

Encapsulació de les Classes `DQNSolver` i `DQNAgent`

Les classes `DQNSolver` i `DQNAgent` es van encapsular al fitxer `D_agent.py`. La classe `DQNSolver` defineix el model de xarxa neuronal utilitzat per l'agent DQN, mentre que la classe `DQNAgent` implementa l'agent que utilitza el model DQN per prendre decisions i interactuar amb l'entorn.

Creació de la Classe `Config`

Es va crear la classe `Config` al fitxer `Z_config_decoder.py`. Aquesta classe s'encarrega de gestionar el pas de paràmetres a través de diccionaris, validant i processant la configuració necessària per al funcionament del sistema.

Funció `run`

La funció `run` es va mantenir al fitxer `A_main.py`, que és el punt d'entrada principal del programa. Aquesta funció executa l'entrenament o la prova de l'agent DQN en l'entorn de `Super Mario Bros`.

Eliminació de la Funció `vectorize_action`

Es va descartar la funció `vectorize_action`, ja que no era utilitzada en el codi. Això va ajudar a reduir la complexitat i mantenir el codi més net.

3.2.3 Fase 2 de la Implementació: Refinament

En aquesta fase, es va construir un sistema de superclasses per desglossar la funció `run`. Aquest procés va incloure la creació de diverses subclasses específiques per a diferents tasques d'entrenament i comprovacions generals.

Creació del Sistema de Superclasses

Es va establir una classe base anomenada `BaseProcess`, que serveix com a plantilla abstracta. Aquesta classe proporciona estructures i comprovacions generals que són implementades de manera específica per les subclasses. Les subclasses principals creades inclouen `BaseTrainingProcess`, `TrainDQN` i `TrainDQN_AA`. Aquestes subclasses es detallen en el fitxer `C_simulation_manager.py`.

Implementació de `TrainDQN_AA`

Encara que es va deixar implementada una versió preliminar de `TrainDQN_AA`, aquesta no es va utilitzar àmpliament. Només es va verificar el seu funcionament, centrant-se principalment en el mode Decay.

Implementació de la Classe `Tracker`

Es va implementar la classe `Tracker` en el fitxer `T_tracker.py`. Aquesta classe s'encarrega de recollir les mètriques dels experiments, facilitant l'anàlisi dels resultats.

Modificació de Classes

Les classes `DQNAgent` i `DQNSolver` es van modificar per permetre l'estudi de l'ús de diferents nombres de capes convolucionals i diferents valors de `dropout`. Aquestes modificacions van ampliar la flexibilitat dels models, permetent provar configuracions més variades. També es van fer modificacions a la classe `Config` per permetre configurar el procés de refinat.

Refinament dels Paràmetres d'Entrenament

Un cop fetes aquestes modificacions, es va procedir a refinar els paràmetres per a l'entrenament d'un agent `DQN Non Advising`. Això va incloure ajustar els hiperparàmetres per optimitzar el rendiment de l'agent.

Sobrecàrrega de la Funció `main`

Durant la implementació d'aquests canvis, la funció `main` es va sobrecarregar amb noves responsabilitats. Es va decidir deixar la funció `main` en aquest estat, posposant la creació de les classes `TrainLauncher` i `TestLauncher` per a una fase posterior. Aquestes classes s'encarregaran de gestionar les tasques d'entrenament i prova de manera més organitzada.

3.2.4 Fase 3 de la Implementació: Experimentació

Per a aquesta fase, es van implementar les classes `TrainLauncher` i `TestLauncher` per realitzar l'experimentació i estudiar el funcionament de les tècniques d'Action Advise proposades.

Implementació de les Classes `TrainLauncher` i `TestLauncher`

Aquestes classes es van crear per gestionar de manera organitzada les tasques d'entrenament i prova, facilitant l'experimentació amb les tècniques d'Action Advise.

Modificació de Classes per a l'Experimentació

Es van fer modificacions a diverses classes per cobrir les necessitats pròpies del procés d'experimentació:

- `Train_AA`: Es va modificar per implementar les tècniques `SUA`, `TUA` i `TSUA`.
- `DQNAgent`: Es va modificar per permetre el càlcul de la incertesa.
- `main`, `Tracker` i `Config`: Es van fer les modificacions pertinents per adaptar-se a les necessitats del procés d'experimentació.

3.3 Metodologia d'Experimentació

3.3.1 Justificació de l'entorn

En el treball de final de màster (TFM) de Javier Montalvo, titulat *Exploiting Semantic Segmentation to Boost Reinforcement Learning in Video Game Environments* [13], s'explora l'ús de la segmentació semàntica per millorar el rendiment dels algorismes d'aprenentatge per reforç en entorns de videojocs. El codi desenvolupat per Montalvo implementa l'entorn de SuperMario de la biblioteca Gym [6] i inclou tant la tècnica de *fine-tuning* com un segmentador semàntic basat en una ResNet50, una xarxa neuronal convolucional profunda composta per 50 capes que utilitza connexions residuals per millorar el flux de gradients durant l'entrenament. Aquesta arquitectura

està dissenyada per a la classificació d'imatges, però en aquest cas s'utilitza per a la segmentació semàntica de les imatges de l'entorn. A la secció **??** explicarem amb més profunditat en què consisteixen aquestes tècniques.

Per al nostre estudi, aprofitarem aquest codi base per diversos motius:

- L'experimentació ja feta sobre aquest entorn ens dóna una base sòlida sobre la qual definir els nostres experiments.
- La implementació existent del *fine-tuning* ens permetrà utilitzar aquesta tècnica com una de les línies base en les nostres comparacions, facilitant l'avaluació del rendiment de la tècnica d'estudi.
- El segmentador semàntic basat en ResNet50 ens permetrà comparar el rendiment del sistema amb i sense segmentació visual, proporcionant una cota superior per a les nostres comparacions.

Aquest enfocament ens permetrà centrar-nos en la integració i avaluació de la tècnica *Introspective Action Advising (IAA)* en lloc de desenvolupar tot el codi des de zero, accelerant així el procés de recerca.

Una restricció que ens imposa l'aprofitament d'aquest entorn és l'ús d'un model DQN, ja que tota l'experimentació feta per Javier Montalvo està feta amb aquest model. D'aquesta manera, podem basar-nos en alguns dels experiments realitzats en la seva investigació per a dissenyar els nostres, assegurant així que són experiments fiables i vàlids.

3.3.2 Monitorització i Rigor en l'Experimentació

Monitorització amb *Weights and Biases*

En la secció 3.1.1, s'ha esmentat l'us de l'eina *Weights and Biases* per a la monitorització dels experiments. En aquest apartat, esmentarem les estratègies principals que hem utilitzat a l'hora d'emprar aquesta eina.

- **Compte *Develop***: Durant les implementacions i el manteniment del codi, es van fer proves per verificar el funcionament de la recollida de paràmetres. Per a aquestes proves, vam considerar adequat l'ús d'un projecte *Develop* per assegurar-nos que tot funcionava correctament abans de llançar els processos en el compte d'experimentació principal.
- **Configuració organitzada**: La plataforma *Weights and Biases* permet discriminar les execucions a través dels paràmetres de configuració. L'ús d'un sistema organitzat ens permet assegurar-nos que els experiments es duen a terme amb els paràmetres correctes. Això facilita l'anàlisi i l'extracció de conclusions de manera més eficient.
- **Ús d'etiquetes**: Un dels paràmetres de configuració que *Weights and Biases* permet utilitzar són les etiquetes. El seu funcionament és semblant al d'un *hashtag* de Twitter, permetent-nos organitzar els experiments de manera més ràpida i eficient. Aquest sistema no només millora la claredat en la visualització de diferents configuracions dins d'un mateix experiment, sinó que també facilita la separació de les execucions entre diferents experiments, permetent-nos visualitzar només les execucions rellevants de manera senzilla.

Rigor en la Reproduibilitat

Per assegurar la rigorositat i reproduibilitat dels experiments, s'ha posat un èmfasi especial en l'ús de *seeds* i en l'organització en *seed_groups*. Aquesta pràctica permet replicar els resultats, garantir la consistència i rigor en l'entrenament i avaluació dels models.

Què és una *seed* i per què és important?

Una *seed* és un valor inicial utilitzat per a la generació de nombres aleatoris en els processos d'entrenament dels models. La seva importància radica en el fet que garanteix la reproductibilitat dels experiments. En utilitzar la mateixa *seed*, es pot assegurar que els processos aleatoris (com la

inicialització dels pesos i la inicialització del entorn) siguin consistents entre diferents execucions del mateix experiment. Això permet assegurar la validesa dels resultats i les conclusions obtingudes.

Organització en *seed_groups*

Per assegurar la consistència i la diversitat en els experiments, s'ha organitzat l'ús de *seeds* en *seed_groups*, seguint aquestes pautes:

- **Refinament:**
  - Cada *seed_group* consistia en tres nombres generats aleatòriament.
  - Es va evitar repetir *seed_groups* per garantir que cada conjunt de refinament fos únic i aportés variabilitat als resultats.
- **Experimentació:**
  - Cada *seed_group* consistia en dos nombres generats aleatòriament.
  - A cada *seed* se li van associar pesos diferents, preentrenats tots sota els mateixos paràmetres en el nivell 1-1, per assegurar una base consistent per a l'avaluació dels models en experiments posteriors.

Capítol 4

Implementació del sistema d'experimentació

Aquest capítol descriu l'entorn d'experimentació desenvolupat per al projecte. Es proporciona una visió general del sistema de classes, es detalla la implementació de les funcions principals i de les classes implementades com `TrainDQN_AA`, `TrainLauncher`, `TestLauncher`, i es descriuen els processos d'entrenament, incloent les funcions comunes i exclusives per a cada tipus d'entrenament. També s'expliquen les classes auxiliars com `Tracker` i `Config`.

4.1 Descripció de l'Entorn d'experimentació

L'entorn del joc Super Mario que utilitzem en aquesta pràctica es basa en una implementació mitjançant la llibreria `gym-super-mario-bros` [10], que permet la interacció amb el joc a través d'OpenAI Gym. A continuació es descriuen els elements clau de l'entorn:

- **Agent**: L'agent és el personatge de Mario que pren accions dins del joc per tal de maximitzar la seva recompensa. L'agent aprèn a partir de les seves experiències, desenvolupant una política que determina les accions a prendre en cada estat per tal de completar els nivells de manera eficient.
- **Entorn (*Environment*)**: L'entorn és el món del joc Super Mario, on hi trobem enemics, obstacles i objectes. L'entorn proporciona els estats i les recompenses a l'agent en resposta a les seves accions, afectant el seu progrés en el joc.
- **Estat (*State*)**: L'estat, en el context de Super Mario, es representa amb un *frame* amb codificació RGB de 96x96 píxels.
- **Accions (*Actions*)**: Les accions que l'agent pot prendre inclouen moure's cap a l'esquerra o la dreta, saltar, córrer i altres accions específiques al aconseguir un objecte determinat. En el nostre cas, per a simplificar l'espai de decisions, hem utilitzat el `flag RIGHTONLY`, per a permetre només saltar, moure's a la dreta i ajupir-se.
- **Recompensa (*Reward*)**: La recompensa, la senyal que l'agent rep després de prendre una acció, es calcula amb una fórmula complexa que té en compte la velocitat de desplaçament cap a la dreta, la penalització per temps aturat i la penalització per mort, i es troba detallada a l'annex C.1.
- **Política (*Policy*)**: La política és la funció que l'agent utilitza per determinar quina acció prendre en cada estat. En el nostre cas, generem aquesta política amb un agent DQN.
- **Funció de valor (*Value Function*)**: La funció de valor estima la recompensa a llarg termini de ser en un estat (o de prendre una acció en un estat) sota una política determinada. En el

nostre cas, aquesta funció és la funció Q, que ajuda l'agent a prendre decisions informades per maximitzar les recompenses futures.

- **Model**: En alguns mètodes d'aprenentatge per reforç, l'agent utilitza un model de l'entorn per fer prediccions sobre els resultats de les accions. En el nostre cas, utilitzem l'*experience replay* de l'algorisme DQN per emmagatzemar les experiències passades i utilitzar-les per entrenar la xarxa neuronal.

Un cop explicat com es defineix l'entorn d'experimentació, de del punt de vista d'un problema de *Reinforcement Learning*, exposarem el sistema de classes dissenyat per poder dur a terme l'entrenament i l'experimentació sobre les diferents tècniques d'aprenentatge per reforç.

4.2 Exposició general del sistema de classes

En aquest capítol, es descriuen les classes i fitxers que constitueixen el sistema desenvolupat per a l'experimentació amb tècniques de *Transfer Learning* i *Action Advise* en xarxes neuronals aplicades a la segmentació semàntica i la classificació d'imatges RGB. Limplementació està estructurada en quatre blocs:

- Les classes dels tres primers fitxers, `B_launchers`, `C_simulation_manager` i `D_agent`, són les encarregades de la producció de l'entrenament.
- Les classes dels fitxers `X_segmentator` i `Y_environment_wrappers` són encapsulaments del sistema de gestió de l'input. Es important remarcar que el codi d'aquest bloc ha estat extret del TFM de Javier Montalvo i que ha patit modificacions mínimes, únicament per permetre que pogués funcionar amb les màquines de Google.
- Les classes del fitxer `Z_config_decoder` s'encarreguen de gestionar el pas de paràmetres, tractant el diccionari de configuració que recullen.
- Finalment, les classes del fitxer `T_tracker` són les encarregades de recollir les mètriques i són les úniques que es comuniquen amb la plataforma *Weights and Biases* (`wandb`).

Fitxer `A_main.py`

Aquest fitxer conté la funció principal del sistema, configurada per utilitzar *Hydra* per a la gestió dinàmica de la configuració. La funció principal s'encarrega d'inicialitzar i executar els processos d'entrenament i prova basant-se en els paràmetres de configuració especificats.

Fitxer `B_launchers`

Aquest fitxer conté les classes responsables d'iniciar els processos d'entrenament i de prova. El propòsit d'aquestes classes és configurar i engegar els processos corresponents, assegurant-se que els paràmetres estiguin correctament establerts.

- **TrainLauncher**: Aquesta classe és responsable d'iniciar el procés d'entrenament. Configura els paràmetres necessaris i engega l'entrenament del model.
- **TestLauncher**: Aquesta classe és responsable d'iniciar el procés de prova. Configura els paràmetres necessaris i engega el procés de prova per avaluar el rendiment del model.

Fitxer `C_simulation_manager`

Aquest fitxer conté les classes que gestionen el procés de simulació. Aquestes classes proporcionen el marc estructural per als processos d'entrenament i prova, assegurant una gestió adequada del flux de treball.

- **BaseProcess**: Classe base per als processos que es gestionen en la simulació. Defineix les operacions bàsiques que han de tenir totes les subclasses.

- **BaseTrainingProcess**: Hereta de **BaseProcess** i proporciona funcionalitats específiques per als processos d'entrenament.
- **TrainDQN**: Especialització de **BaseTrainingProcess** per entrenar models DQN (Deep Q-Network).
- **TrainDQN_AA**: Especialització de **TrainDQN** que inclou tècniques d'Action Advise durant l'entrenament.
- **TestProcess**: Hereta de **BaseProcess** i proporciona funcionalitats específiques per als processos de prova.

Fitxer `D_agent`

Aquest fitxer conté les classes relacionades amb els agents d'aprenentatge. Aquestes classes implementen la lògica dels agents, incloent la definició del model i les estratègies d'aprenentatge.

- **DQNSolver**: Defineix el model DQN utilitzat pels agents. Inclou la xarxa neuronal i les funcions d'actualització dels pesos.
- **DQNAgent**: Implementa l'agent que utilitza el model DQN per prendre decisions. Defineix les interaccions de l'agent amb l'entorn i les estratègies d'aprenentatge.

Fitxer `T_tracker`

Aquest fitxer conté les classes relacionades amb el seguiment del rendiment dels models. Aquestes classes són responsables de recopilar, emmagatzemar i analitzar les mètriques de rendiment durant l'entrenament i la prova.

- **Tracker**: Classe que gestiona el seguiment del rendiment dels models durant l'entrenament i la prova. Emmagatzema i processa les mètriques de rendiment.

Fitxer `X_segmentator`

Aquest fitxer conté les classes relacionades amb la segmentació d'imatges. Aquestes classes implementen els algorismes de segmentació que divideixen les imatges en regions semàntiques.

- **Segmentator**: Implementa les funcionalitats per segmentar imatges. Aquesta classe s'encarrega de dividir les imatges en regions semàntiques utilitzant models de segmentació.

Fitxer `Y_environment_wrappers`

Aquest fitxer conté les classes que configuren i gestionen l'entorn de l'agent. Aquestes classes són responsables de processar les imatges de l'entorn i adaptar-les per a l'ús del model d'aprenentatge.

- **MaxAndSkipEnv**: Configura l'entorn de Mario per al procés d'entrenament i prova. Inclou funcionalitats específiques per a la configuració del joc.
- **ProcessFrame84**: Processa les imatges de l'entorn per ajustar-les a les dimensions requerides pel model.
- **ImageToPyTorch**: Converteix les imatges processades en tensors que es poden utilitzar en PyTorch.
- **ScaledFloatFrame**: Escala les imatges a valors de punt flotant normalitzats.
- **BufferWrapper**: Implementa una estructura de dades per emmagatzemar les experiències de l'agent durant l'entrenament.

Fitxer `Z_config_decoder`

Aquest fitxer conté les classes relacionades amb la configuració del sistema. Aquestes classes s'encarreguen de gestionar els paràmetres de configuració, tractant el diccionari de configuració que recullen durant l'entrenament i la prova.

- **Config**: Implementa la decodificació i gestió dels paràmetres de configuració del sistema. Aquesta classe s'encarrega de carregar, validar i proporcionar accés als paràmetres configurats per a l'entrenament i la prova dels models.

Aquest capítol proporciona una visió general de l'arquitectura del sistema, detallant les funcions i responsabilitats de cada classe, així com els fitxers on es defineixen. Això facilita la comprensió del codi i la seva extensibilitat per a futurs estudis de *Transfer Learning* i *Action Advise*.

4.3 Descripció main i launchers

4.3.1 Classe `A_main.py`

Descripció del propòsit

Aquest fitxer conté la funció principal del sistema, configurada per utilitzar *Hydra* per a la gestió dinàmica de la configuració. La funció principal s'encarrega d'inicialitzar i executar els processos d'entrenament i test basant-se en els paràmetres de configuració especificats.

Atributs d'inicialització

No aplica, ja que aquest fitxer no defineix cap classe directament sinó que conté la funció principal.

Classes relacionades

- `Config` (del fitxer `Z_config_decoder`)
- `TestLauncher` (del fitxer `B_launchers`)

Configuracions possibles

Permet tres modes d'execució:

- **Mode de test**: Si `config['process']['is_training']` és `False`, es realitza un test de nivell únic utilitzant la classe `TestLauncher` del fitxer `B_launchers`.
- **Mode d'entrenament**: Si `config['process']['is_training']` és `True`, es realitza l'entrenament del model amb llavors aleatòries o predefinides. Les funcions `load_seeds` (del fitxer `useful_functions`), `print_configs` (del fitxer `useful_functions`) i `training_by_seeds` (del fitxer `R_run_functions`) s'utilitzen per gestionar les llavors, imprimir les configuracions i executar l'entrenament respectivament.
- **Mode de test zero-shot**: Si `config['process']['zero_shot_test']` és `True`, es realitza un test zero-shot utilitzant les funcions `print_configs` (del fitxer `useful_functions`) i `zero_shot_test` (del fitxer `R_run_functions`).

Lògiques destacables

Utilitza *Hydra* per gestionar la configuració, *random* per generar llavors aleatòries i executa funcions d'entrenament i test segons els paràmetres configurats.

4.3.2 Classe `TrainLauncher`

Descripció del propòsit

Aquesta classe és responsable d'iniciar el procés d'entrenament. Configura els paràmetres necessaris, crea l'entorn de Super Mario i engega l'entrenament del model utilitzant les tècniques de DQN (Deep Q-Network) o DQN amb Action Advise.

Atributs d'inicialització

- `config`: Configuració específica per l'entrenament, incloent paràmetres generals i específics del procés.
- `agent_config`: Configuració específica per l'agent d'entrenament, incloent hiperparàmetres del model.

Classes i Llibreries relacionades

- `BaseTrainingProcess` (del fitxer `C_simulation_manager`)
- `TrainDQN` (del fitxer `C_simulation_manager`)
- `TrainDQN_AA` (del fitxer `C_simulation_manager`)
- `gym_super_mario_bros` (llibreria utilitzada per crear l'entorn de Super Mario)

Configuracions possibles

Les configuracions d'entrenament poden incloure:

- **Seed Mode**: El mode de llavor pot ser individual o de grup, establint la llavor per a cada entrenament.
- **Tipus d'entrenament**: El mode d'entrenament pot ser `train_dqn`, `train_ft` (fine-tuning) o `train_aa` (Action Advise).
- **Seguiment amb Weights and Biases (wandb)**: Es pot habilitar el seguiment dels experiments utilitzant wandb.

Lògiques destacables

- **setup_training**: Configura l'entrenament establint la llavor, el mode de llavor i prepara la configuració de wandb per passar-la al trainer si està habilitat.
- **setup_trainer**: Crea l'entorn de Super Mario Bros a través de la llibreria *gym_super_mario_bros* i inicialitza la classe d'entrenament adequada (`TrainDQN` o `TrainDQN_AA`) segons el mode especificat en la configuració.
- **execute**: Executa el procés d'entrenament utilitzant la configuració especificada. Si wandb està habilitat, utilitza la configuració de `wandb` per al seguiment.

4.3.3 Classe `TestLauncher`

Descripció del propòsit

Aquesta classe és responsable d'iniciar el procés de test. Configura els paràmetres necessaris i engega el procés de test per avaluar el rendiment del model en diferents nivells del joc Super Mario Bros.

Atributs d'inicialització

- `test_config`: Configuració específica per al test.
- `agent_config`: Configuració específica per a l'agent de test.

- `test_levels`: Nivells de test a executar.
- `path`: Ruta per carregar els models entrenats.
- `use_wandb`: Indicador d'ús de Weights and Biases (wandb) per al seguiment dels tests.

Classes i Llibreries relacionades

- `TestProcess` (del fitxer `C_simulation_manager`)
- `gym_super_mario_bros` (llibreria utilitzada per crear l'entorn de Super Mario)

Configuracions possibles

Les configuracions de test poden incloure:

- **Nivells de test**: Els diferents nivells del joc Super Mario Bros que es volen provar.
- **Ruta de càrrega**: La ruta on es troben els models entrenats que es volen avaluar.
- **Ús de Weights and Biases (wandb)**: Si s'utilitza wandb per al seguiment dels experiments.

Lògiques destacables

- **setup_testing**: Configura el test establint els nivells a provar, la configuració de l'agent de test i la ruta de càrrega dels models entrenats. Prepara la configuració de wandb si està habilitat.
- **single_level_test**: Realitza un test en un nivell únic del joc Super Mario Bros i recopila els resultats. Si wandb està habilitat, utilitza la configuració de wandb per al seguiment.
- **multi_level_test**: Realitza tests en múltiples nivells del joc Super Mario Bros i recopila els resultats. Si wandb està habilitat, utilitza la configuració de wandb per al seguiment.
- **get_results**: Retorna els resultats dels tests. Pot retornar els resultats en mode 'pure' o 'splited', on es separen les recompenses i els punts finals.

4.4 Descripció d'entrenaments

La superclasse `BaseProcess` defineix una interfície genèrica per a processos d'aprenentatge per reforç. Aquesta classe abstracta assegura que tots els tipus de processos (com entrenament i test) compleixin amb un patró d'inicialització estàndard i una interfície comuna. Inclou la configuració de l'entorn, la configuració de l'agent i les comprovacions necessàries basades en els paràmetres proporcionats.

Tot i que és essencial per establir la base de les subclasses com `BaseTrainingProcess`, `TrainDQN` i `TrainDQN_AA`, la `BaseProcess` no es detalla amb la mateixa profunditat perquè serveix principalment com a plantilla abstracta. La seva funció principal és proporcionar estructures i comprovacions generals que les subclasses implementen de manera específica.

4.4.1 Classe `BaseTrainingProcess`

Descripció del propòsit

Hereta de `BaseProcess` i proporciona funcionalitats específiques per als processos d'entrenament. Defineix les operacions bàsiques necessàries per entrenar un model.

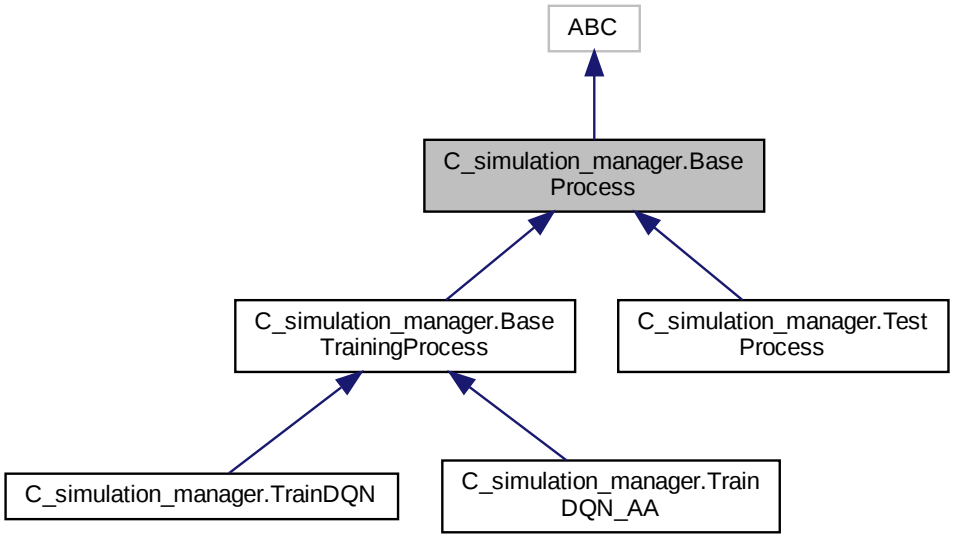


Figura 4.1: Diagrama de Classes Generat per Doxygen

Atributs d'inicialització

- `config`: Configuració general per l'entrenament.
- `agent_config`: Configuració específica per l'agent, incloent el nom del model.
- `input_env`: Identificador de l'entorn utilitzat per configurar la simulació.
- `epochs`: Nombre d'èpoques per l'entrenament.
- `mode`: Mode de procés especificat a la configuració.
- `save_best_model`: Indicador per guardar el millor model basat en el rendiment.
- `best_reward`: Millor recompensa obtinguda fins ara.
- `save_name`: Nom de fitxer per guardar el model.
- `env`: Entorn configurat per l'entrenament.

Classes i Llibreries relacionades

- `BaseProcess` (del fitxer `C_simulation_manager`)
- `DDQNAgent` (del fitxer `D_agent`)
- `make_env, show_state` (del fitxer `Y_environment_wrappers`)
- `Tracker` (del fitxer `T_tracker`)
- `gym_super_mario_bros` (llibreria utilitzada per crear l'entorn de Super Mario)

Configuracions possibles

Inclou paràmetres com el nombre d'episodis, el tipus de xarxa neuronal i els hiperparàmetres d'entrenament. També permet configurar si es guarda el millor model basat en el rendiment (`save_best_model`).

Lògiques destacables

- **check_save_conditions**: Avalua si el rendiment del model actual compleix els criteris per ser guardat. Aquesta funció no s'utilitza en aquesta pràctica ja que el flag `save_best_model` sempre es manté a fals.
- **train**: Executa el procés d'entrenament, gestionant l'entorn de simulació, les accions de l'agent i registrant els resultats a cada episodi. Utilitza un bucle per executar els episodis, reiniciant l'entorn cada vegada i cridant el mètode `act` de l'agent per triar accions basades en l'estat actual.

Algorithm 3 Funció train

Data: Configuració de la llavor, configuració de l'entorn, agent, epochs, mode, save_best_model
Result: Dades dels models guardats
Configurar la llavor aleatòria per a la reproductibilitat
Reiniciar l'entorn
Inicialitzar les dades dels models guardats
Inicialitzar el tracker per a seguir el rendiment

if *s'utilitza wandb* **then**
  if *wandb_config és None* **then**
    Mostrar missatge d'error i sortir
  end
  else
    Iniciar el seguiment de wandb amb la configuració proporcionada
  end
end
for *episodi* **do**
  Reiniciar l'entorn i obtenir l'estat inicial
  Inicialitzar `total_reward`, `steps`, `flag_get`, etc.
  while *no es completi l'episodi* **do**
    if *visualització habilitada* **then**
      Visualitzar l'estat de l'entorn
    end
    Obtenir l'acció de l'agent
    Executar l'acció a l'entorn i obtenir el nou estat, recompensa, etc.
    Actualitzar `total_reward` i `steps`
    if *es compleix la condició de guardat del model* **then**
      Guardar el model actual
    end
    Recordar la transició d'estat a l'agent i executar el *replay* d'experiències
    Actualitzar l'estat a l'estat següent
    if *l'episodi ha acabat* **then**
      Sortir del bucle
    end
  end
  Registrar els resultats de l'episodi amb el tracker
end
Acabar el seguiment amb el tracker
Guardar el model final després de l'entrenament
Tancar l'entorn
Retornar les dades dels models guardats

4.4.2 Descripció funcions `TrainDQN` i `TrainDQN_AA`

Aquest capítol presenta les classes utilitzades per entrenar models DQN (*Deep Q-Network*). Aquestes classes estan dissenyades per gestionar els agents de manera independent al tipus d'entrenament seguit, ja que la lògica d'entrenament està definida a la superclasse `BaseTrainingProcess`.

Les tècniques implementades a TrainDQN:

- **NA**: *Non Advising*
  Aquesta tècnica utilitza imatges sense cap tipus de consell.
- **FT**: *Fine-tuning*
  Aquesta tècnica aplica una afinació fina a les imatges, servint com a cota mitjana per a la comparació.

Les tècniques implementades a TrainDQN_AA:

- **AA Decay**: *Action Advise only Decay guided*
  Guia el consells únicament amb la funció de *Decay* λ_t.
- **SUA**: *Student Uncertainty guided Advise*
  Guia els consells basant-se en la incertesa de l'estudiant.
- **TUA**: *Teacher Uncertainty guided Advise*
  Guia els consells basant-se en la incertesa del professor.
- **TSUA**: *Teacher Student Uncertainty guided Advise*
  Combina la incertesa del professor i de l'estudiant per guiar els consells.

Primer es presentarà la Descripció del propòsit, Atributs d'inicialització, Classes i llibreries relacionades i Configuracions possibles de les classes. Després es descriuran les funcions comunes, les funcions comunes amb diferències d'implementació i les funcions exclusives de *Action Advise*. Finalment, es proporcionarà una descripció en pseudocodi de les funcions principals d' *Action Advise*.

Classe `TrainDQN`

Descripció del propòsit

Encapsula les funcions per a l'entrenament models DQN (*Deep Q-Network*). La classe hereta de `BaseTrainingProcess` i implementa lògiques específiques per l'entrenament de models DQN, gestionant el cicle de vida d'un episodi d'entrenament i les interaccions amb l'agent.

Atributs d'inicialització

- `config`: Configuració específica per l'entrenament del model DQN.
- `agent`: L'agent DQN utilitzat per prendre decisions i actualitzar les seves polítiques d'aprenentatge.
- `save_name`: Nom del fitxer utilitzat per emmagatzemar el model entrenat.
- `mode`: Mode d'entrenament (`train_dqn` o `train_ft`).
- `best_reward`: Millor recompensa obtinguda durant l'entrenament.

Classes i llibreries relacionades

- `BaseTrainingProcess` (del fitxer `C_simulation_manager`)
- `DQNAgent` (del fitxer `D_agent`)
- `Tracker` (del fitxer `T_tracker`)

Configuracions possibles

Inclou paràmetres com la taxa d'aprenentatge, la xarxa neuronal utilitzada, i altres hiperparàmetres específics per DQN. Les configuracions poden ser ajustades per optimitzar el rendiment de l'agent DQN durant l'entrenament.

Classe `TrainDQN_AA`

Descripció del propòsit

Especialització de `TrainDQN` que inclou tècniques d'*Action Advise* durant l'entrenament per millorar el rendiment del model. Aquesta classe gestiona tant l'agent professor com l'agent estudiant, així com la lògica per avaluar i donar consells d'acció basats en la incertesa.

Atributs d'inicialització

- `aa_config`: Configuració específica per l'*Action Advise*.
- `teacher_agent`: Agent professor utilitzat per donar consells d'acció basats en la seva experiència.
- `student_agent`: Agent estudiant que aprèn de la seva pròpia experiència i dels consells del professor.
- `save_name`: Nom del fitxer utilitzat per emmagatzemar el model de l'agent estudiant.
- `aa_mode`: Mode d'*Action Advise* utilitzat per avaluar les accions (SUA, TUA, TSUA).
- `balance`, `lambda_value`, `min_balance`, `delta_value`, `max_ep_recommendation`: Paràmetres específics per ajustar la lògica d'*Action Advise*.
- `student_Du`, `teacher_Du`: Vectors d'incertesa per als agents estudiant i professor respectivament.
- `student_threshold`, `teacher_threshold`: Llindars d'incertesa per decidir si es pren el consell.

Classes i llibreries relacionades

- `BaseTrainingProcess` (del fitxer `C_simulation_manager`)
- `DQNAgent` (del fitxer `D_agent`)
- `Tracker` (del fitxer `T_tracker`)

Configuracions possibles

Inclou paràmetres per a la configuració de l'*Action Advise*, com els criteris per a l'assessorament, el mode d'*Action Advise* (AA Decay, SUA, TUA, TSUA), i altres paràmetres per ajustar la interacció entre els agents professor i estudiant.

4.4.3 Funcions comunes

Funcions comunes idèntiques

- `get_ending_position`: Recupera la posició final de l'agent en l'entorn.
- `get_agent`: Recupera l'objecte agent utilitzat en el procés d'entrenament.
- `get_agent_save_name`: Recupera el nom de desat de l'agent.
- `agent_remember`: Passa l'experiència a la memòria de l'agent. Aquesta funció pren els següents paràmetres:
  - `state` (Tensor): L'estat actual de l'entorn.

  - `action` (Tensor): L'acció presa en l'estat actual.
  - `reward` (Tensor): La recompensa rebuda després de prendre l'acció.
  - `state_next` (Tensor): El següent estat de l'entorn després de l'acció.
  - `terminal` (Tensor): Indica si el següent estat és un estat terminal.

- `execute`: Executa el procés d'entrenament, diferenciant el comportament segons el mode d'entrenament. Aquesta funció inicialitza el procés d'entrenament, configura el tracker per a seguir el rendiment, i executa els episodis d'entrenament segons la configuració.

Funcions comunes amb diferències d'implementació

- `__init__`: La inicialització varia perquè les configuracions i els agents poden ser diferents segons la classe.
  - En `TrainDQN`, la inicialització es fa configurant l'agent segons el mode `train_dqn` o `train_ft`, utilitzant models preentrenats si s'està en mode de *fine-tuning*.
  - En `TrainDQN_AA`, es configura tant l'agent professor com l'agent estudiant, amb configuracions addicionals per als paràmetres d'*Action Advise*.
- `get_action`: La manera de seleccionar l'acció pot variar, especialment si s'utilitza algun tipus de consell d'acció (*Action Advise*).
  - En `TrainDQN`, l'acció és seleccionada per l'agent DQN directament.
  - En `TrainDDQN_AA`, l'acció pot ser seleccionada per l'agent professor o l'agent estudiant, depenent dels paràmetres d'incertesa i consell.
- `agent_experience_replay`: Activa el procés d'aprenentatge de l'agent des del buffer de memòria.
  - En `TrainDQN`, només l'agent DQN realitza el procés d'experiència *replay*.
  - En `TrainDQN_AA`, tant l'agent professor com l'agent estudiant poden realitzar el procés d'experiència *replay*, i es registren les pèrdues de cadascun. Si el profesor ja no pot fer aconcellaments, ja no fa aquest procés.
- `backup`: Els criteris per desar el model poden ser diferents.
  - En `TrainDQN`, es desa el model basat en les condicions de recompenses i l'episodi.
  - En `TrainDQN_AA`, es desa el model basat en criteris addicionals d'*Action Advise*.

4.4.4 Funcions exclusives de `TrainDDQN_AA`

- `set_mode`: Configura el mode d'*Action Advise* segons la configuració proporcionada. Estableix els noms dels models per a l'agent professor i l'agent estudiant basats en el mode d'*Action Advise*.
- `setup_aa_params`: Inicialitza els paràmetres d'*Action Advise* des de la configuració. Aquests inclouen el balanç mínim, el valor de δ, el paràmetre λ, la pèrdua màxima per a l'agent professor, i la recomanació màxima per episodi. També s'inicialitzen els vectors per a les incerteses dels estudiants i professors, així com els llindars per a cada un.
- `update_balance`: Actualitza el balanç basat en el número d'episodis actuals. El balanç es calcula com una funció del valor de λ elevat al màxim entre 0 i el número d'episodis menys el valor de δ.
- `teacher_working`: Avalua si l'agent professor ha d'estar actiu basant-se en el balanç actual i el número d'episodis. Si el balanç és major que el mínim balanç o el número d'episodis és menor que la recomanació màxima d'episodis, l'agent professor està actiu.

- `compute_thresholds`: Calcula els llindars d'incertesa per als agents estudiant i professor. La mitjana i la desviació estàndard es calculen a partir dels valors d'incertesa recollits i es multipliquen pel balanç actual.
- `update_uncertainty`: Actualitza els valors d'incertesa per als agents estudiant i professor. Aquests valors es limiten a una mida màxima per assegurar que no creixin indefinidament.
- `take_advice`: Avalua si s'ha de prendre el consell d'acció basant-se en el mode d'*Action Advise* i els valors d'incertesa actuals.
- `select_action_with_lambda_decay`: Selecciona si l'acció es pren basant-se en una distribució de probabilitat que es redueix amb el temps.

Algorithm 4 Funció update_balance

Data: Valor de λ, número d'episodi (t), valor de δ
Result: Balanç actualitzat
begin
  balance = $\lambda^{\max(0,t-\delta)}$
end

Algorithm 5 Funció teacher_working

Data: Mínim balanç, número d'episodis, recomanació màxima per episodi
Result: Indicador de si el professor està actiu
begin
  Actualitzar el balanç cridant `update_balance`
  if *balanç > mínim balanç* **then**
    return True
  end
  return False
end

Algorithm 6 Funció compute_thresholds

Data: Vector d'incertesa de l'estudiant (D_{uS}), vector d'incertesa del professor (D_{uT}), balanç
Result: Llindars d'incertesa per a l'estudiant i el professor
begin
  if D_{uS} *no està buit* **then**
    Convertir D_{uS} a array de numpy
    Calcula μ_t i σ_t de D_{uS}
    $\tau_{tS} = \mu_t + \sigma_t \cdot$ balance
  end
  if D_{uT} *no està buit* **then**
    Convertir D_{uT} a array de numpy
    Calcula μ_t i σ_t de D_{uT}
    $\tau_{tT} = \mu_t + \sigma_t \cdot$ balance
  end
end

4.4.5 Classe `TestProcess`

Descripció del propòsit: La classe `TestProcess` està dissenyada per conduir tests amb un agent preentrenat en diversos nivells del joc Super Mario Bros. Aquesta classe facilita l'avaluació del rendiment de l'agent en diferents escenaris de joc.

Algorithm 7 Funció update_uncertainty

Data: Incertesa de l'estudiant (u_S), incertesa del professor (u_T), mida màxima (100)
Result: Vectors d'incertesa actualitzats
begin
  if *u_S no és None* **then**
    Afegir u_S a D_{uS}
    if *mida de D_{uS} > max_size* **then**
      Eliminar el primer element de D_{uS}
    end
  end
  if *u_T no és None* **then**
    Afegir u_T a D_{uT}
    if *mida de D_{uT} > max_size* **then**
      Eliminar el primer element de D_{uT}
    end
  end
  Cridar `compute_thresholds`
end

Algorithm 8 Funció take_advice

Data: Estat actual, mode d'*Action Advise*, llindars d'incertesa
Result: Indicador de si es pren el consell
begin
  if *aa_mode és 'AA Decay'* **then**
    take_advice = True
  end
  else if *aa_mode és 'SUA'* **then**
    Calcula u_S a partir de l'estat
    take_advice = $u_S \geq \tau_{tS}$
    Actualitzar incertesa de l'estudiant cridant `update_uncertainty`
  end
  else if *aa_mode és 'TUA'* **then**
    Calcula u_T a partir de l'estat
    take_advice = $u_T < \tau_{tT}$
    Actualitzar incertesa del professor cridant `update_uncertainty`
  end
  else if *aa_mode és 'TSUA'* **then**
    Calcula u_S a partir de l'estat
    Calcula u_T a partir de l'estat
    take_advice = $u_S \geq \tau_{tS}$ **and** $u_T < \tau_{tT}$
    Actualitzar incerteses cridant `update_uncertainty`
  end
  return take_advice
end

Atributs d'inicialització:

- `config`: Configuració general del procés de test, que inclou paràmetres com el tipus d'entrada i si s'utilitza o no `Weights and Biases (wandb)`.
- `agent_config`: Configuració específica de l'agent, que inclou paràmetres per al model DQN.
- `input_type`: Tipus d'entrada utilitzada per a l'agent (definida a la configuració general).
- `use_wandb`: Indicador de si es fa servir `wandb` per al seguiment de resultats.
- `epochs`: Nombre d'èpoques o episodis durant els quals es realitzarà el test.
- `agent`: Instància de l'agent DQN preentrenat, carregat amb pesos preentrenats especificats en la configuració.

Classes i llibreries relacionades:

- `gym-super-mario-bros`: Llibreria utilitzada per crear l'entorn del joc.
- `DQNAgent`: Classe de l'agent DQN que s'utilitza per prendre decisions basades en l'aprenentatge preentrenat.
- `Tracker`: Utilitzat per fer el seguiment del temps i els resultats durant els tests, amb suport per `wandb`.

Configuracions possibles:

- `input_type`: Configuració del tipus d'entrada que l'agent rebrà de l'entorn.
- `use_wandb`: Configuració per habilitar o deshabilitar l'ús de `wandb` per al seguiment de resultats.
- `pretrained_path`: Ruta al fitxer amb els pesos preentrenats per a l'agent.
- `epochs`: Nombre d'episodis durant els quals es durà a terme el test.

Lògiques destacables:

- `execute_multi_level_test(test_levels, wandb_config)`: Funció per realitzar tests en múltiples nivells. Crea l'entorn per a cada nivell i executa el test, registrant els resultats.
- `execute_single_level_test(levels, wandb_config)`: Funció per realitzar un test en un sol nivell. Configura l'entorn i executa el test per al nivell especificat.
- `execute(wandb_config)`: Funció principal per executar un episodi de test. Reinicia l'entorn, segueix l'agent mentre pren accions, registra les recompenses i el progrés, i finalitza l'episodi quan l'agent arriba a un estat terminal.

Aquesta classe proporciona una eina integral per avaluar el rendiment d'un agent DQN preentrenat en l'entorn de Super Mario Bros, amb suport per a múltiples configuracions i seguiment de resultats.

4.5 Descripció DQNSolver i DQNAgent

4.5.1 Classe `DQNSolver`

Descripció del propòsit

Defineix el model DQN utilitzat pels agents. Inclou la xarxa neuronal i les funcions d'actualització dels pesos.

Atributs d'inicialització

- `solver_config`: Configuració específica per a la solució del model DQN.
- `input_shape`: La forma de les observacions d'entrada.

- `n_actions`: El nombre d'accions que l'agent pot prendre.
- `conv_layers`: Nombre de capes convolucionals, per defecte 3.
- `dropout_rate`: Taxa de dropout, per defecte 0.

Classes i llibreries relacionades

- `torch`: Llibreria principal per a operacions de tensor i creació de xarxes neuronals.
- `torch.nn as nn`: Mòdul de `torch` que conté definicions per a totes les capes de xarxes neuronals.
- `nn.Module`: Classe base per a tots els models de xarxes neuronals en PyTorch, proporciona la infraestructura per definir i entrenar xarxes neuronals.

Configuracions possibles

Inclou la configuració de la xarxa neuronal, la taxa d'aprenentatge, el descompte de recompensa, entre altres hiperparàmetres específics.

- `input_shape`: La forma de les observacions d'entrada.
- `n_actions`: El nombre d'accions que l'agent pot prendre.
- `conv_layers`: Nombre de capes convolucionals, 2, 3 o 4, per defecte 3.
- `dropout_rate`: Taxa de *dropout*, per defecte 0.

Lògiques destacables

Implementa l'algorisme DQN per entrenar la xarxa neuronal, incloent funcions per a la propagació cap enrere i l'actualització dels pesos.

- `__init__`: Inicialitza la xarxa neuronal amb capes convolucionals i totalment connectades.
- `_get_conv_out`: Calcula la sortida convolucional donada una entrada.
- `forward`: Propaga l'entrada a través de la xarxa.
- `activations_hook`: Guarda els gradients de les activacions.
- `get_activations_gradient`: Retorna els gradients de les activacions.
- `get_activations`: Retorna les activacions de les capes convolucionals.

4.5.2 Classe `DQNAgent`

Descripció del propòsit

Implementa l'agent que utilitza el model DQN per prendre decisions. Defineix les interaccions de l'agent amb l'entorn i les estratègies d'aprenentatge. Aquest agent implementa Q-learning amb experiència repetida i xarxes de destinació.

Atributs d'inicialització

- `config`: Diccionari de configuració que conté tots els paràmetres necessaris.
- `pretrained`: Indicador booleà per utilitzar models preentrenats.
- `pretrained_path`: Camí als models preentrenats.

Classes i llibreries relacionades

- `DQNSolver`: Definició de la xarxa neuronal utilitzada per l'agent DQN (del fitxer `D_agent`).
- `torch`: Llibreria principal per a operacions de tensor i creació de xarxes neuronals.

Configuracions possibles

Inclou la configuració de les estratègies d'aprenentatge, paràmetres d'exploració/explotació, i altres configuracions específiques de l'agent.

- `state_space`: La forma de l'espai d'estats.
- `action_space`: El nombre d'accions que l'agent pot prendre.
- `conv_layers`: Nombre de capes convolucionals de la xarxa neuronal.
- `dropout`: Taxa de dropout per a la regularització de la xarxa.
- `lr`: Taxa d'aprenentatge per a l'optimitzador.
- `target_update`: Freqüència d'actualització de la xarxa objectiu.
- `max_memory_size`: Mida màxima de la memòria d'experiències.
- `batch_size`: Mida del lot per a l'experiència repetida.
- `gamma`: Factor de descompte per a l'aprenentatge Q.
- `max_exploration_rate`: Taxa màxima d'exploració inicial.
- `min_exploration_rate`: Taxa mínima d'exploració.
- `exploration_decay`: Taxa de decaïment de l'exploració.
- `run_name`: Nom de l'execució per a guardar els models.
- `save_path`: Camí per guardar els models.

Lògiques destacables

Defineix les estratègies de presa de decisions de l'agent, incloent l'exploració de l'entorn i l'explotació del coneixement adquirit per maximitzar les recompenses.

- `__init__`: Inicialitza l'agent DQN, configura les xarxes neuronals i els components d'entrenament.
- `load_pretrained_models`: Carrega els pesos dels models preentrenats des del camí especificat.
- `setup_training_components`: Configura els components d'entrenament com l'optimitzador, la memòria i les taxes d'aprenentatge.
- `remember`: Emmagatzema les experiències a la memòria d'experiències.
- `recall`: Mostreja aleatòriament experiències de la memòria d'experiències.
- `act`: Selecciona una acció basada en la política epsilon-greedy.
- `update_model`: Actualitza els pesos de la xarxa local a la xarxa objectiu.
- `experience_replay`: Executa un pas d'entrenament utilitzant un minibatch de la memòria d'experiències.
- `calculate_uncertainty`: Calcula la incertesa dels valors Q per a un estat donat.
- `save_model`: Guarda el model entrenat en el camí especificat.
- `print_all_params`: Imprimeix tots els paràmetres de configuració.

Modificacións de la Classe DQNAgent

- **Paràmetres d'inicialització**: La versió original utilitza paràmetres explícits mentre que la versió modificada utilitza un diccionari de configuració, per això s'han implmementant funcions per facilitar el tractament dels diccionaris.

- **Memòries d'experiència**: La versió original inclou l'opció de carregar memòries prèviament guardades, mentre que la versió modificada s'ha descartat, ja que no s'emprava.
- **Funcions de còpia del model**: La funció de còpia del model té un nom diferent. En la versió original és `copy_model` i en la versió modificada és `update_model`, ja que ens ha semblat que el nom es més a curat a la funcionalitat.
- **Mètodes addicionals**: La versió modificada inclou mètodes addicionals que no es troben en la versió original.

Funcions Addicionals en la Versió Modificada

- `load_pretrained_models(pretrained_path, pretrained_name, abs_path)`
  - **Justificació**: Aquesta funció permet carregar els pesos dels models preentrenats des d'una ubicació especificada. Això aïlla la lògica de càrrega en una funció separada, fent el codi més modular i mantenible.
- `setup_training_components(config)`
  - **Justificació**: Aquesta funció configura els components d'entrenament, com ara l'optimitzador, la memòria i els paràmetres d'aprenentatge. Això permet separar la lògica d'inicialització dels components d'entrenament de la resta del codi, millorant la claredat i la modularitat.
- `calculate_uncertainty(state)`
  - **Justificació**: Aquesta funció calcula la incertesa dels valors Q per a un estat donat. Això pot ser útil per a analitzar la confiança del model en les seves prediccions i pot ajudar a guiar l'exploració de l'entorn.
- `save_model(ep_num, total_reward, end_training)`
  - **Justificació**: Aquesta funció guarda els pesos del model entrenat en un fitxer específicat. Això permet preservar l'estat del model després de l'entrenament, facilitant la continuació de l'entrenament o l'avaluació posterior.
- `print_all_params(config)`
  - **Justificació**: Aquesta funció imprimeix tots els paràmetres de configuració del model. Això és útil per a la verificació i depuració, permetent assegurar-se que tots els paràmetres s'han inicialitzat correctament.

Càlcul de la Incertesa

Algorithm 9 Càlcul de la Incertesa dels Valors Q

procedure CALCULATE_UNCERTAINTY(s_t)
  Input: s_t - L'estat per al qual es vol calcular la incertesa

  Output: La incertesa dels valors Q per a l'estat donat

  $a_t \leftarrow \text{act}(s_t)$ ▷ Selecciona l'acció utilitzant la política epsilon-greedy
  $Q_{t\pi}^{\text{local}}(s_t, a_t) \leftarrow Q_{local}(s_t, a_t)$ ▷ Calcula el valor Q utilitzant la xarxa local
  $Q_{t\pi}^{\text{target}}(s_t, a_t) \leftarrow Q_{target}(s_t, a_t)$ ▷ Calcula el valor Q utilitzant la xarxa objectiu
  Return $|Q_{t\pi}^{\text{target}}(s_t, a_t) - Q_{t\pi}^{\text{local}}(s_t, a_t)|$ ▷ Retorna la incertesa com l'absolut de la diferència entre Q_{local} i Q_{target}
end procedure

4.6 Classes auxiliars: `Tracker` i `Config Decoder`

4.6.1 `Config Decoder`

Descripció del propòsit

La classe `Config` s'utilitza per gestionar la configuració, validar-la i proporcionar dades de configuració. Analitza, valida i processa les dades de configuració necessàries per al funcionament del sistema.

Atributs d'inicialització

- `cfg`: Objecte de configuració d'`OmegaConf`.

Classes i llibreries relacionades

- `OmegaConf`: Llibreria per treballar amb fitxers de configuració YAML.
- `pickle`: Llibreria estàndard de Python per a la serialització i deserialització d'objectes Python.

Configuracions possibles

Inclou la configuració de les seccions requerides i la configuració específica de l'entorn.

- `trackers`: Configuració per al seguiment i registre dels experiments.
- `process`: Configuració del procés, incloent el mode de funcionament (entrenament o test).
- `agent`: Configuració específica de l'agent, incloent paràmetres d'entrenament i de prova.

Lògiques destacables

Defineix les funcions per validar, processar i extreure les configuracions necessàries per al sistema.

- `__init__`: Inicialitza l'objecte `Config`, valida les seccions necessàries i processa la configuració de l'entorn.
- `validate_cfg()`: Valida la presència de les seccions requerides en la configuració. Genera un error si falta alguna secció.
- `general_config()`: Extreu i retorna les configuracions específiques de l'entorn.
- `process_config`: Propietat que processa i retorna la configuració específica per al mode operatiu (entrenament o prova).
- `agent_config`: Propietat que processa i retorna la configuració específica de l'agent.
- `extract_train_config()`: Extreu la configuració específica per al mode d'entrenament a partir de la configuració YAML proporcionada.
- `extract_test_config()`: Extreu la configuració específica per al mode de prova a partir de la configuració YAML proporcionada.
- `_build_basic_config(agent_info)`: Construeix i retorna la configuració bàsica de l'agent a partir de la informació de l'agent.
- `_build_sub_config(sub_params, basic_config)`: Construeix i retorna la configuració de subcomponents a partir dels paràmetres `sub_params` i la configuració bàsica `basic_config`. Això inclou la configuració dels paràmetres d'exploració, mida de la memòria, mida del lot, gamma, taxa d'aprenentatge, taxa de `dropout` i altres paràmetres específics. La funció combina aquestes configuracions amb la configuració bàsica per generar una configuració completa per als subcomponents de l'agent.

4.6.2 Tracker

Descripció del propòsit

La classe `Tracker` s'utilitza per gestionar i fer un seguiment de l'estat d'un procés. Manté un registre del mode, l'hora d'inici i diversos comptadors. També té opcions per al seguiment i l'ús de `wandb` per a la registre.

Atributs d'inicialització

- `start_time`: L'hora en què el procés va començar.
- `use_wandb`: Indicador booleà que indica si el registre amb `wandb` està habilitat.

Classes i llibreries relacionades

- `time`: Llibreria estàndard de Python per treballar amb temps.
- `wandb`: Llibreria per a la registre i monitorització d'experiments de machine learning.

Configuracions possibles

Inclou la configuració de l'ús de `wandb` per a la registre, i altres configuracions específiques del procés de seguiment.

- `start_time`: L'hora en què el procés va començar.
- `use_wandb`: Indicador booleà per habilitar o deshabilitar el registre amb `wandb`.

Lògiques destacables

Defineix les estratègies de seguiment de l'estat del procés, incloent la inicialització, registre, i finalització del seguiment, així com el registre de mètriques específiques.

- `__init__`: Inicialitza una nova instància de la classe `Tracker` amb l'hora d'inici i l'opció d'usar `wandb`.
- `set_start_time()`: Estableix l'hora d'inici per a l'episodi actual.
- `start_tracking(project_name, tags, config, agent_config)`: Inicia el seguiment amb `wandb`, configurant el projecte, les etiquetes, i la configuració.
- `end_tracking()`: Finalitza el procés de seguiment.
- `log_time(log_name)`: Registra el temps passat des de l'última registre amb `wandb`.
- `log_loss(loss, teacher_loss=None)`: Registra la pèrdua amb `wandb`.
- `log_uncertainty(aa_mode, student_uncertainty, student_threshold, teacher_uncertainty, teacher_threshold)`: Registra la incertesa amb `wandb`.
- `log_episode(total_reward, ending_position, steps, flag_get, count_advice_given=None, balance=None)`: Registra les mètriques de l'episodi amb `wandb`.
- `log_test_results(reward, pos, steps, flag_get)`: Registra els resultats del test amb `wandb`.
- `get_accumulated_metrics()`: Retorna les mètriques acumulades.

Capítol 5

Experimentació i Resultats

En aquest capítol es defineixen i descriuen els experiments realitzats per avaluar les tècniques proposades. S'explica el nombre d'episodis utilitzats, els experiments planificats, i les tècniques emprades com a *base-lines* (DQN, FT, SS). Es presenta l'elecció dels paràmetres i els resultats obtinguts en cada estudi. Finalment, es proporcionen les interpretacions de les mètriques experimentals i es discuteixen les hipòtesis plantejades.

5.1 Definició de l'experimientació

En aquesta secció definim i justifiquem les condicions sota les quals s'han fet els experiments, alhora que es defineixen les hipòtesis i el procediment seguit per l'experimentació.

5.1.1 Nombre d'Episodis

Per escollir la llargada que havien de tenir els entrenaments, ens hem basat en el blog *Building a Double Deep Q-Network to Play Super Mario Bros* [4], on es presenta una primera implementació del software d'experimentació utilitzant l de Super Mario, adaptada per Javier Montalvo en el seu TFM [13], s'exposa que un entrenament òptim consta de 10.000 episodis. Com podem veure a les Figures 5.1 i 5.2, en l'experimentació de Javier Montalvo "4.3 Training in multiple game-levels"del seu TFM), l'ús de la segmentació semàntica de l'entrada aconsegueix augmentar el pendent de la corba d'aprenentatge, arribant a una mitjana de *reward* normalitzada del 50% abans dels 2.000 episodis.

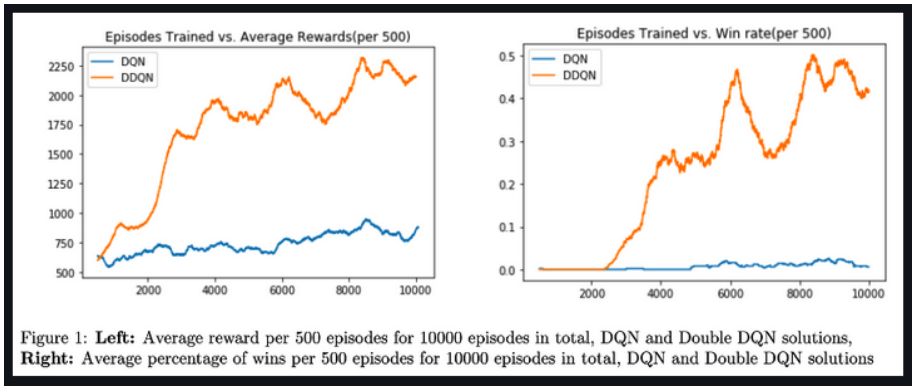


Figure 1: **Left:** Average reward per 500 episodes for 10000 episodes in total, DQN and Double DQN solutions, **Right:** Average percentage of wins per 500 episodes for 10000 episodes in total, DQN and Double DQN solutions

Figura 5.1: Corba d'aprenentatge del blog amb 10.000 episodis (És erroni anomenar DDQN a la tècnica emprada; realment, la línia groga correspon a una execució de l'algorisme DQN i la blava a l'algorisme Q-Learning)

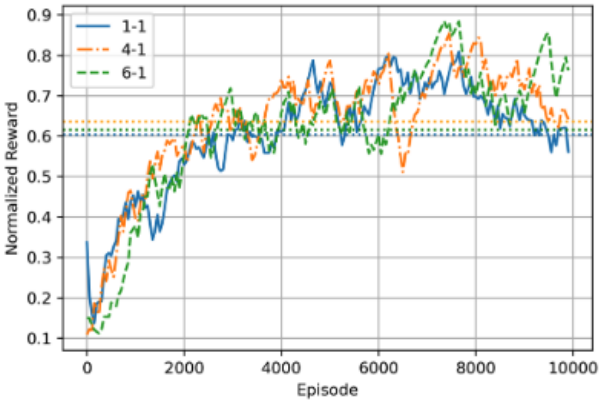


Fig. 9 Normalized per-game-level rewards while training on three levels simultaneously for the segmented configuration. Horizontal lines represent the average values for each level

Figura 5.2: Corba d'aprenentatge de Javier Montalvo amb 10.000 episodis

En la següent secció del seu treball (*4.4 Transfer learning*), Javier Montalvo exposa l'estudi de la transferència d'aprenentatge en experiments de 3.000 episodis, ja que és on hi ha el major aprenentatge i el que resulta determinant per al posterior entrenament.

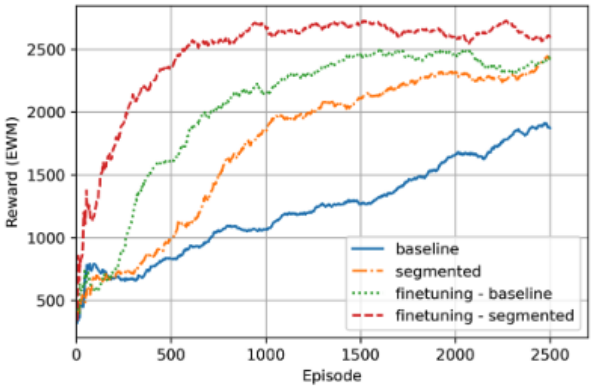


Fig. 10 Transfer learning between game-levels with similar appearance: rewards evolution training on 4-1 a model pre-trained on 1-1

Figura 5.3: Estudi de la transferència d'aprenentatge de Javier Montalvo amb 3.000 episodis

Per tant, atès que comptem amb un temps reduït, ens decantarem per fer estudis de 3.000 episodis, centrant-nos així en la secció més crítica de l'entrenament. Això deixaria per a treballs futurs l'experimentació completa. Decidir enfocar l'estudi d'aquesta manera ens impedeix estudiar com els models entrenats generalitzen la resolució del nivell, ja que l'entrenament no ha estat completat. Tot i així, en la majoria d'estudis de *Transfer Learning* revisats, no és comú l'ús d'aquestes proves per avaluar l'entrenament en aquests tipus d'entorns, ja que les recompenses obtingudes es consideren suficients per extreure conclusions sobre l'eficàcia de les tècniques estudiades.

Mètriques analitzades

- **Recompensa acumulada:** Aquesta mètrica mesura el rendiment global del model en termes de la suma total de recompenses obtingudes al llarg del temps. És un indicador directe de com de bé el model està assolint els seus objectius en l'entorn d'aprenentatge. Segons el treball de Taylor i Stone (2009) [17], en el qual s'analitzen diverses mètriques per avaluar

la transferencia d'aprenentatge, estableixen que la recompensa acumulada permet avaluar amb l'eficàcia i l'eficiència dels mètodes d'aprenentatge, quan la llargada de l'aprenentatge es acotada.

- **Temps total:** Mesura el temps total requerit per completar els episodis. Aquesta mètrica és crucial per avaluar l'eficiència temporal del model.
- **Recompensa per episodi:** Reflecteix la mitjana de recompenses obtingudes per episodi. Permet veure l'evolució del rendiment del model episodi per episodi.
- **Win rate (últims 100 episodis):** Indica la taxa d'èxit en els últims 100 episodis, proporcionant una visió clara de la consistència i convergència del model durant l'entrenament.

Mètriques construïdes:

- **Recompensa acumulada / `steps` acumulats:** Aquesta mètrica proporciona una mesura de l'eficiència del model en termes de la recompensa obtinguda per cada pas realitzat, ajudant a entendre la relació entre les accions del model i les recompenses.
- **Recompensa acumulada respecte temps transcorregut:** Proporciona una mesura de la recompensa obtinguda per unitat de temps, permetent avaluar l'eficiència temporal del model en aconseguir recompenses.

5.1.2 Experiments Planificats

Experiment 1: Efecte de l'Estudi de Paràmetres

Objectiu: Avaluar l'impacte de l'estudi de paràmetres en el rendiment de les tècniques utilitzades.

Hipòtesi: S'espera que tant RGB NA com SS NA millorin amb l'estudi previ de paràmetres. Tot i així, no s'espera que la versió millorada de RGB NA superi el rendiment de SS NA sense millorar. D'altra banda, s'espera que el *Fine-tune* (RGB FT), sent una tècnica que presenta un bon rendiment en aquest entorn, aconsegueixi un rendiment similar a SS NA sense haver estudiat els paràmetres.

Condicions de l'experiment:

- **Nivell**: 4-1.
- **Entrades**: RGB NA i SS NA (*Semantic Segmentation*) abans i després de l'estudi de paràmetres. La tècnica RGB FT només l'hem executat amb els paràmetres refinatas.
- `seed_group`:
  - `zeta_prima`: Utilitzat per les tècniques RGB NA i SS NA abans d'haver estudiat els paràmetres.
  - `Altres seed_groups`: Utilitzats per les tècniques RGB NA, RGB FT i SS NA en experiments posteriors.

Procediment:

1. Executar les tècniques RGB NA i SS NA amb paràmetres no estudiats sobre el `seed_group` `zeta_prima` durant 3000 episodis al nivell 4-1.
2. Utilitzar les execucions d'altres experiments on s'han estudiat els paràmetres i s'han aplicat les tècniques refinades.
3. Comparar el rendiment inicial amb els resultats posteriors utilitzant les mètriques escollides.

Experiment 2: Efecte del Dropout en el Rendiment del Model

Motivació: Avaluar la viabilitat de l'ús del *dropout*. El *dropout* és una tècnica de regularització utilitzada en xarxes neuronals per prevenir el sobreajustament, on unitats (neurones) d'una xarxa es desactiven aleatòriament durant l'entrenament. Això força la xarxa a aprendre representacions

més robustes. No obstant això, en entorns on es busca un sobreajustament de l'agent a una tasca concreta, com és superar un nivell específic d'un joc, l'ús del *dropout* pot ser contraproduent per a que l'agent resolgui el problema de manera òptima.

Objectiu: Analitzar l'impacte del *dropout* en el rendiment del model, comparant els resultats obtinguts amb i sense l'aplicació del *dropout*.

Hipòtesi: S'espera que l'ús del *dropout* no sigui adequat per aquest entorn específic i que, en conseqüència, degradi el rendiment del model.

Condicions de l'experiment:

- **Nivells:** 1-1 i 4-1.
- **Dropout:** 0.0 (sense *dropout*) i 0.1.
- **Entrades:** RGB NA i SS NA (*Semantic Segmentation*).
- `seed_group`:
  - **1-1**: `alpha_prima` i `beta_prima`.
  - **4-1**: `zeta_prima` i `xi_prima`.

Procediment:

1. Entrenar el model amb `dropout` = 0.0 durant 3000 episodis en els nivells 1-1 i 4-1, utilitzant tant entrades RGB NA com SS NA.
2. Entrenar el model amb `dropout` = 0.1 durant 3000 episodis en els mateixos nivells i tipus d'entrada.
3. Comparar el rendiment dels models en termes de les mètriques escollides.

Experiment 3: Rendiment amb *Action Advice* (AA) amb diferents *Learning Rates*

Motivació: Determinar com de determinant és el *learning rate* en l'aprenentatge de l'agent `teacher` en l'*Action Advice*, per al seu posterior ús en l'Experiment 4.

Objectiu: Estudiar l'efecte del *learning rate* en el rendiment del model utilitzant tècniques d'*Action Advice* (AA).

Hipòtesi: Es preveu que un *learning rate* menor (0.1) afavorirà el procés d'*Action Advice*, ja que és el valor habitualment utilitzat en aquests contextos. No obstant això, es considera que un *learning rate* més alt (0.4) podria tenir un impacte negatiu en l'eficiència de l'aprenentatge de l'agent.

Condicions de l'experiment:

- **Nivell:** 4-1, amb pre-entrenament al nivell 1-1.
- ***Learning Rates*:** 0.1 i 0.4.
- **Tècnica d'Action Advice:** *AA Decay* amb una delta de 0.15.
- `seed_group`:
  - **4-1**: `zeta_prima`, `xi_prima`.

Procediment:

1. Pre-entrenar el model en el nivell 1-1 durant 3000 episodis, utilitzant els models pre-entrenats corresponents a les dues execucions en el nivell 1-1 utilitzant de llavors el `seed_group` beta_prima.
2. Entrenar el model en el nivell 4-1 amb *learning rate* 0.1 durant 3000 episodis.
3. Entrenar el model en el nivell 4-1 amb *learning rate* 0.4 durant 3000 episodis.
4. Comparar el rendiment dels models en termes de les mètriques escollides.

Experiment 4: Avaluació de les Tècniques Proposades

Motivació: L'objectiu d'aquest experiment és estudiar com funcionen les tècniques d'*Action Advise* amb la implementació feta en un context on la distància entre la tasca font i la tasca objectiu és petita. A més, es vol analitzar l'efecte dels diferents heurístics en l'aconsellament d'accions, i del paràmetre *Burn in* δ, en el rendiment d'aquests algoritmes en l'entorn escollit.

Objectiu: Avaluar les tècniques proposades i comparar el rendiment, considerant l'impacte del paràmetre δ i els heurístics utilitzats per a l'aconsellament d'accions.

Hipòtesi 1: Ja que en el treball de [3] s'esmentava que no asseguraven que la tècnica pogués funcionar tan bé si les tasques font i objectiu eren massa diferents. En el nostre cas al ser més semblants, s'espera que el rendiment sigui encara millor.

Hipòtesi 2: La tècnica TSUA, sent la més restrictiva, mostrarà un millor rendiment al tenir en compte tant la incertesa del professor com de l'estudiant per decidir si es pren l'acció proposada pel professor.

Hipòtesi 3: En l'estudi del paràmetre δ, s'espera que l'algoritme SUA tingui un millor rendiment amb $\delta = 0.0$, mentre que l'algoritme TUA tindrà un rendiment òptim amb $\delta = 0.15$. Aquesta expectativa es basa en resultats previs d'investigacions que han seguit aquestes configuracions.

Condicions de l'experiment:

- **Nivell:** 4-1, amb pre-entrenament al nivell 1-1.
- **Pre-entrenament**:
  - Els models FT, AA Decay, SUA, TUA i TSUA s'han utilitzat els mateixos models pre-entrenats en el nivell 1-1 per a cadascuna de les *seeds*.
- `seed_group` **entrenament**:
  - **RGB NA i SS NA:** `zeta_prima`, `xi_prima`, `upsilon_prima`, `tau_prima`, `theta_prima`.
  - **FT:** `zeta_prima`, `xi_prima`, una *seed* de `theta_prima` i una altra de `upsilon_prima`.
  - **AA Decay:** `zeta_prima`, `xi_prima`.
  - **SUA, TUA i TSUA:** `zeta_prima`, `xi_prima`, `upsilon_prima`.

Procediment:

1. Pre-entrenar el model en el nivell 1-1 durant 3000 episodis.
2. Entrenar el model en el nivell 4-1 durant 3000 episodis, aplicant les tècniques proposades.
3. Comparar el rendiment dels models en termes de les mètriques escollides.

A continuació, expliquem el procés que hem seguit per escollir els paràmetres de configuració de l'agent, de l'entrenament i de les tècniques, per a la posterior discussió dels resultats.

5.2 Tècniques Emprades

En aquesta secció es presenten les tècniques utilitzades com a *base-lines* per a avaluar les tècniques estudiades. Les tècniques considerades són DQN com a cota inferior, *fine-tuning* com a cota mitjana, i segmentació semàntica com a cota superior. En el cas de DQN, ens referim a NA (*Non Advising*) a l'anàlisi de resultats, indicant que no s'utilitza cap tècnica d'*Action Advise*.

5.2.1 Cota inferior: DQN

La tècnica DQN està explicada a la secció 2.1.1. Per tant, en aquesta secció profunditzarem en com es realitza el procés d'entrenament del model, des de la recollida de l'estat de l'entorn, la generació d'una acció i l'actualització de la xarxa neuronal amb les mètriques que es recullen en resposta a aquesta acció.

Durant el procés d'entrenament, el model DQN segueix els següents passos:

- **Recollida de l'estat de l'entorn**: Es captura l'estat actual de l'entorn, que en el cas d'un joc podria ser una imatge de la pantalla o altres dades representatives de la situació actual del joc.
- **Generació d'una acció**: Basant-se en l'estat actual, la xarxa neuronal predir quina acció prendre. Això es fa mitjançant la funció de valor d'acció Q, que estima la recompensa esperada per cada acció possible en l'estat donat.
- **Execució de l'acció**: L'acció seleccionada es realitza en l'entorn, el que provoca una transició a un nou estat.
- **Recollida de recompenses i mètriques**: Després de l'execució de l'acció, es recull l'estat del entorn després del qual s'ha decidit prendre l'acció, l'acció presa, la recompensa obtinguda d'executar l'acció, l'estat del entorn després de l'execució de l'acció i si es troba en un estat terminal.
- **Actualització de la xarxa neuronal**: Amb les dades recollides (estat inicial, acció, recompensa, i nou estat), s'actualitza la xarxa neuronal. Això es fa ajustant els pesos de la xarxa per minimitzar la diferència entre la recompensa predita i la recompensa real observada.

Aquest procés es repeteix durant els episodis estipulats fins que el model aprèn una política òptima, suficientment bona per a la tasca específica o s'acaba el nombre d'episodis estipulats.

Aquest procediment permet al model DQN millorar iterativament les seves prediccions i decisions, aprenent a maximitzar les recompenses a llarg termini en l'entorn donat.

5.2.2 Cota mitja: FT

La tècnica del *Fine-Tuning* implica agafar una xarxa neuronal que ha estat prèviament entrenada en un conjunt de dades gran i general, com ImageNet, i ajustar-la per a una nova tasca específica. Això es fa utilitzant els pesos preentrenats com a punt de partida en lloc d'iniciar l'entrenament des de zero amb pesos inicialitzats aleatòriament. Aquesta tècnica és especialment útil quan el conjunt de dades per a la nova tasca és relativament petit, ja que aprofita el coneixement ja adquirit en la xarxa preentrenada.

La implementació del *Fine-Tuning* és relativament senzilla. Es tracta de carregar els pesos preentrenats des d'un fitxer i començar l'entrenament en el nou context, de la mateixa manera que es faria amb un model inicialitzat aleatòriament, com és comú en l'algorisme DQN. Això permet que el model mantingui les característiques generals aprengudes de les dades prèvies mentre s'ajusta als detalls específics de la nova tasca.

Un article en el qual es debat sobre la seva efectivitat és *How transferable are features in deep neural networks?*[19], on va discutir l'us de *Fine-Tuning* en el context de l'aprenentatge profund. En aquest article es demostra efectivitat per adaptar models preentrenats a noves tasques, maximitzant l'eficiència de l'entrenament i millorant el rendiment del model

5.2.3 Cota superior: SS

La segmentació semàntica és una tècnica que permet classificar cada píxel d'una imatge en una categoria específica, facilitant una interpretació més abstracta i simplificada de les imatges. Aquesta informació va ser extreta de la memòria del TFG de Javier Montalvo [13].

Per a la implementació de la segmentació semàntica, es va utilitzar la llibreria `Pytorch`. Per a desenvolupar la tècnica, es va utilitzar el model *DeepLabV3* amb l'arquitectura *ResNet50*.

Per entrenar el model, es va seguir un enfocament de *Fine-Tuning*. Primer, es va descarregar el model amb pesos preentrenats a *ImageNet*, i la capa classificador va ser substituïda per una nova capa de la classe `DeepLabHead` de la llibreria `torchvision`. Aquesta classe era una xarxa neuronal seqüencial composta per una convolució 3x3 amb 256 canals d'entrada i 256 canals de sortida,

seguida d'una normalització per lots 2D, una capa d'activació *ReLU*, i una altra convolució 1x1 amb 256 entrades i tantes sortides com classes a segmentar.

Els models es van entrenar utilitzant un conjunt de dades de 20.000 imatges, amb 16.000 imatges per a entrenament i 4.000 imatges per a validació. Es va utilitzar una augmentació de dades simple per retallar la imatge a 256x240 píxels des de la mida original de 272x240 píxels, minimitzant l'impacte de la graella utilitzada per generar els fotogrames.

Javier Montalvo va penjar la *ResNet50* amb els pesos ja entrenats a Google Drive, i són aquests els que hem utilitzat per a la nostre experimentació.

5.2.4 *Action Advise*

Per facilitar la comprensió de les diferents tècniques d'*Action Advise* implementades, es presenta a continuació una taula que resumeix cada tècnica, la seva abreviació, una breu explicació de la seva aplicació i la seva procedència.

Tècnica	**Explicació**	**Procedència**
Action Advise by Decay (`AA Decay`)	Aquesta tècnica acceptar consells sempre, sense importar les circumstàncies específiques, es veu només limitat pel llindar de recomanació.	Utilitzada com a *base-line* en el treball de Campbell et al. [3]
Student Uncertainty Advise (`SUA`)	Es calcula la incertesa de l'estudiant i es compara amb un llindar establert; els consells es prenen si la incertesa és menor que el llindar.	Presentada per Sahir et al. [14]
Teacher Uncertainty Advise (`TUA`)	Es calcula la incertesa del professor i es compara amb un llindar establert; els consells es prenen si la incertesa és menor que el llindar.	Homòloga a Introspective Action Advising en DQN
Teacher Student Uncertainty Advise (`TSUA`)	Es calculen les incerteses tant de l'estudiant com del professor i es comparen amb els llindars corresponents; els consells es prenen si ambdues incerteses són menors que els llindars.	Construïda a partir de les dues anteriors

Taula 5.1: Descripció de les tècniques d'assessorament

5.3 Elecció de paràmetres

El refinament dels paràmetres s'ha realitzat en la versió sense la implementació de Action Advise. Tot i que seria metodològicament més precís aplicar aquest refinament a tots els paràmetres, ens vam limitar a aquest conjunt inicial. Existeix una eina anomenada Sweeps, a la plataforma Weights and Biases (wandb), que permet una optimització automàtica, utilitzant tant la cerca bayesiana, cerca aleatòria, cerca en graella, lògica personalitzada i també suporta l'aturada anticipada, però la vam descobrir en una fase avançada del procés, per la qual cosa vam optar per un ajustament manual. En futures investigacions, es considerarà l'ús d'aquesta eina.

A la taula 5.2, es mostren els paràmetres utilitzats per als algorismes DQN i AA (Action Advise), incloent els valors provats durant el procés de refinament i la procedència d'aquests valors.

Alg.	Paràmetre	Valor	Procedència
DQN	Capes Convolucionals	2	Valors provats 2,3,4
DQN	Batch Size	32	Valors provats 8, 16, 32, 64
DQN	Learning Rate	0.0005	Valors provats 0.0001 0.00025 0.0005
DQN	Exploration Decay	0.999	Valors provats 0.99, 0.995, 0.999
DQN	Gamma	0.9	Valors provats 0.85, 0.9, 0.95
DQN	Target Update	500	Valors provats 500, 1000, 1500
DQN	Max Memory Size	4000	Valors provats 2000, 4000, 6000
AA	Batch Size	32	Idem DQN
AA	Learning Rate	0.0004	Valors provats 0.0001, 0.0004
AA	Exploration Decay	0.999	Idem DQN
AA	Gamma	0.9	Idem DQN
AA	Target Update	500	Idem DQN
AA	Max Memory Size	4000	Idem DQN
AA	Lambda	0.996	Càlcul propi
AA	Delta	0.0, 0.15	Càlcul propi

Taula 5.2: Paràmetres utilitzats

5.3.1 Paràmetres de *Burn in* δ i *Decay* λ

Els valors utilitzats en la investigació inicial de l'algorisme IAA són els següents:

Hiperparàmetre	Valor
Llindar d'Introspecció (ϵ)	{0.15, 0.30, 0.45}
Burn in (δ)	500000
Decay (λ)	0.999999

Taula 5.3: Valors dels hiperparàmetres utilitzats per [3]

Com es va exposar a la secció 2.3, el llindar d'introspecció (ϵ) es calcula dinàmicament, eliminant la necessitat de definir un valor fix. Per determinar els valors òptims dels altres dos paràmetres, considerant la restricció temporal, hem analitzat detalladament les conclusions derivades d'aquests valors.

En la proposta feta per Campbell [3], es van estudiar diverses quantitats d'iteracions: 2.5, 5 i 10 milions d'iteracions. Al resoldre la funció per identificar el punt d'inici de l'aconsellament i el punt en què es sobrepassa el valor mínim establert, hem obtingut els valors mostrats a la Taula 5.5.

It. Totals	It. Inici ($t = \delta$)	% d'Entr.	It. $\lambda_t = 0.01$	% d'Entr.
2.5M	500000	20%	5.105M	204%
5M	500000	10%	5.105M	102%
10M	500000	5%	5.105M	51%

Taula 5.4: Valors de la funció λ per diferents iteracions

Com es pot observar a la taula, l'ús de valors fixes resulta en què, si l'entrenament no és prou llarg, no hi ha iteracions on l'agent `student` aprengui sense la interacció de l'agent `teacher`. Això implica que, en entrenaments més curts, l'agent `student` depèn constantment dels consells de l'agent `teacher`, la qual cosa pot no ser òptima per a la independència i l'eficàcia de l'aprenentatge a llarg termini.

Revisant altres estudis en aquest camp, com el realitzat pel mateix equip [5] o el fet per l'equip de [9], es mostra que els pressupostos d'aconsellament es consumeixen en diferents moments de l'entrenament, depenent de l'entorn on s'estigui entrenant. De totes formes, s'aprecia que en la

majoria dels casos comencen a convergir al voltant del 20% de l'entrenament. Per tant, nosaltres hem fet el següent càlcul:

Veient que un entrenament efectiu en l'entorn on estem fent l'estudi correspon a 10000 episodis, un 15% correspon a 1500 episodis. Per tant, utilitzant una $\delta = 450$ episodis, el valor de λ necessari perquè als 1500 episodis el seu valor sigui inferior a 0.01, per tant es permetrà rebre consells de l'agent `teacher` amb una probabilitat inferior al 1%, és 0.996.

Els càlculs dels valors de la funció λ per diferents quantitats d'episodis són els següents:

1. Percentatge de l'Entrenament:

$$\frac{450}{3000} = 15\%$$

$$\frac{450}{5000} = 9\%$$

$$\frac{450}{10000} = 4.5\%$$

2. Episodis $\lambda_t = 0.01$:

$$t - 450 = \frac{\ln(0.01)}{\ln(0.996)} \approx 1148.99$$

$$t \approx 450 + 1148.99 = 1598.99 \text{ per 3000 episodis}$$

$$t \approx 450 + 1148.99 = 1948.99 \text{ per 5000 episodis}$$

$$t \approx 450 + 1148.99 = 2948.99 \text{ per 10000 episodis}$$

3. Percentatge de l'Entrenament:

$$\frac{1598.99}{3000} \approx 53.3\%$$

$$\frac{1948.99}{5000} \approx 39.0\%$$

$$\frac{2948.99}{10000} \approx 29.5\%$$

Ep. Totals	**Ep. Inici ($t = \delta$)**	**% d'Entr.**	**Ep. $\lambda_t = 0.01$**	**% d'Entr.**
3000	450	15%	1598.99	53.3%
5000	450	9%	1948.99	39.0%
10000	450	4.5%	2948.99	29.5%

Taula 5.5: Valors de la funció λ per diferents episodis

Com es pot observar a la taula, amb els valors establerts, el sistema d'aconsellament s'atura abans que el procés d'entrenament estigui complet, assegurant que l'agent `student` tingui l'oportunitat d'aprendre de manera independent durant una part significativa de l'entrenament. Aquesta afirmació no és vàlida per al nostre cas, ja que, com en un episodi es fan més de 6000 accions, un 1% de probabilitat és suficient perquè se'n recomanin diverses. Aquest efecte es veuria de manera absoluta en un entrenament complet de 10000 episodis.

5.4 Resultats de l'Estudi de Paràmetres: Refinament DQN

En aquest experiment es va investigar l'impacte d'haver realitzat un estudi de paràmetres previ a l'experimentació amb tècniques d'*Action Advise* (AA). La hipòtesi formulada era que tant RGB NA com SS NA millorarien amb l'estudi previ de paràmetres. Tot i així, no s'esperava que la versió millorada de RGB NA superés el rendiment de SS NA sense millorar. A més, es preveia que el *Fine-tune* (RGB FT), una tècnica que presenta un bon rendiment en aquest entorn, aconseguiria un rendiment similar a SS NA sense haver estudiat els paràmetres.

Per a aquest experiment, es va utilitzar el nivell 4-1. Les tècniques estudiades van ser RGB NA, RGB FT i SS NA abans i després de l'estudi de paràmetres. Els grups de llavors (`seed_group`) utilitzats van ser `zeta_prima` per a les tècniques RGB NA i SS NA abans de l'estudi de paràmetres, i altres grups de llavors per a les tècniques RGB NA, RGB FT i SS NA en experiments posteriors.

Els models es van entrenar durant 3000 episodis amb i sense l'estudi previ de paràmetres. Els resultats obtinguts es van comparar per avaluar la millora en el rendiment de les tècniques estudiades. A continuació es presenten i analitzen els resultats detallats d'aquest experiment.

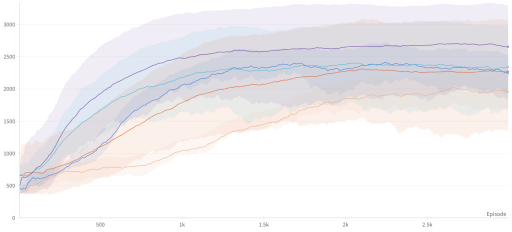


Figura 5.4: Recompença per Episodi

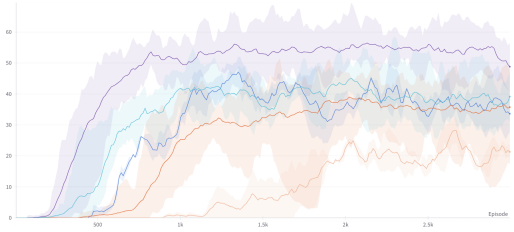
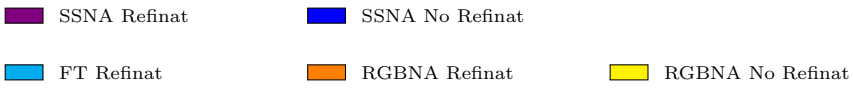


Figura 5.5: *Win Rate* per Episodi



Nota: La figura 5.4 utilitza al mitja mòbil.

Observem, com era d'esperar, que el model SS NA refinat presenta un rendiment significativament superior als altres models, assolint màxims de més de 3000 punts de recompensa. Amb 3500 punts de recompensa, es considera que el nivell 1-1 i el nivell 4-1 estan resolts. A més, aquest model aconsegueix màxims de win rate superiors al 60%.

Podem observar que el rendiment de l'algorisme *Fine-tune* (RGB FT), el SS NA sense refinar i el RGB NA refinat són molt semblants. Aquest fet subratlla la importància del refinament dels paràmetres per millorar el rendiment dels algorismes de *Reinforcement Learning*. Ja que amb una tècnica computacionalment menys costosa, estem conseguint uns resultats semblants a una de més costosa que no ha sigut tant ben configurada per al problema a resoldre.

En resum, els resultats obtinguts demostren que l'estudi previ de paràmetres és essencial per millorar el rendiment de les tècniques d'*Action Advise*. En particular, el refinament dels paràmetres del model SS NA ha donat lloc a un augment significatiu en la recompensa acumulada i el win rate, validant així la hipòtesi inicial. Els resultats també mostren que el *Fine-tune* i RGB NA pot aconseguir un rendiment comparable al de SS NA sense refinar, ressaltant la importància de l'ajust fi en els algorismes de *Reinforcement Learning*.

5.5 Resultats Estudi Paràmetres: *Drop Out*

En aquest experiment es va avaluar la viabilitat de l'ús del *dropout* en la resolució d'un nivell concret del joc SuperMario. Es va formular la hipòtesi que l'ús del *dropout* no seria adequat per aquest entorn específic i que, per tant, degradaria el rendiment del model. Això es basa en el fet que, en entorns on es busca un sobreajustament de l'agent a una tasca concreta, com és superar un nivell específic d'un joc, l'ús del *dropout* podria ser contraproduent per a que l'agent resolgui el problema de manera òptima.

Per a aquest experiment, es van utilitzar dos nivells (1-1 i 4-1) i es van aplicar dues configuracions de *dropout* (0.0 i 0.1). Les entrades utilitzades van ser RGB NA i SS NA (*Semantic Segmentation*). Els models es van entrenar durant 3000 episodis en ambdós nivells per a cada configuració de *dropout*, i es van utilitzar els grups de llavors (`seed_group`) següents: `alpha_prima` i `beta_prima` per al nivell 1-1, i `zeta_prima` i `xi_prima` per al nivell 4-1.

Els resultats obtinguts van ser comparats en termes de les mètriques escollides per tal de determinar l'impacte del *dropout* en el rendiment del model. A continuació es presenten i analitzen els resultats detallats d'aquest experiment.

Input Type	***Dropout***	**Recompensa**	**Temps(s)**	**Steps**	**Win Rate**
RGB	0.0	5368684.125	18322.532	618276.125	29.721
RGB	0.1	3411328.875	11,891.132	413994.625	2.123
SS	0.0	6639074.75	35475.137	669216.625	52.944
SS	0.1	5147571.625	27151.279	550791.5	17.053

Taula 5.6: Resultats de l'Experiment, Mitges de les execución en el nivells 1-1 i 4-1

Recompensa Acomulada

La recompensa acumulada presenta variacions significatives entre els diferents tipus d'entrada i valors de *dropout*. Per a l'entrada RGB, la recompensa és superior amb un *dropout* de 0.0 (5368684.125) en comparació amb un *dropout* de 0.1 (3411328.875). De manera similar, per a l'entrada SS, la recompensa és superior amb un *dropout* de 0.0 (6639074.75) en comparació amb un *dropout* de 0.1 (5147571.625). Això suggereix que un valor de *dropout* més baix tendeix a produir una recompensa acumulada més alta, independentment del tipus d'entrada.

Temps d'Entrenament

El temps total d'entrenament també varia entre els diferents paràmetres. Per a l'entrada RGB, el temps d'entrenament és menor amb un *dropout* de 0.1 (11891.132 segons) en comparació amb un *dropout* de 0.0 (18322.532 segons). Per a l'entrada SS, el temps d'entrenament és menor amb un *dropout* de 0.1 (27151.279 segons) en comparació amb un *dropout* de 0.0 (35475.137 segons). En aquest cas, sembla que els resultats no segueixen un patró consistent com en el cas de la recompensa acumulada.

Nombre de Passos (Steps)

El nombre de passos (steps) utilitzats durant l'entrenament també mostra diferències. Per a l'entrada RGB, el nombre de passos és superior amb un *dropout* de 0.0 (618276.125) en comparació amb un *dropout* de 0.1 (413994.625). Per a l'entrada SS, el nombre de passos és superior amb un *dropout* de 0.0 (669216.625) en comparació amb un *dropout* de 0.1 (550791.5). Això indica que un valor de *dropout* més baix tendeix a requerir més passos durant l'entrenament, independentment del tipus d'entrada.

Win Rate

El win rate és notablement més alt amb un *dropout* de 0.0 en comparació amb un *dropout* de 0.1 per ambdós tipus d'entrada. Per a l'entrada RGB, el win rate és 29.721% amb un *dropout* de 0.0 i només 2.123% amb un *dropout* de 0.1. Per a l'entrada SS, el win rate és 52.944% amb un *dropout* de 0.0 i 17.053% amb un *dropout* de 0.1. Aquestes dades suggereixen que un valor de *dropout* més baix afavoreix un millor rendiment en termes de win rate, independentment del tipus d'entrada.

Havent repassat els resultats de la Taula 5.7, es pot concloure que no utilitzar *dropout* en aquest entorn, produteix millors resultats en termes de recompensa acumulada i win rate. Aquestes observacions són vàlides tant per a les entrades RGB com per a les SS. A més, la Figura **??** corrobora perfectament aquests fenòmens, mostrant visualment les diferències en el rendiment del model en funció dels diferents valors de *dropout*.

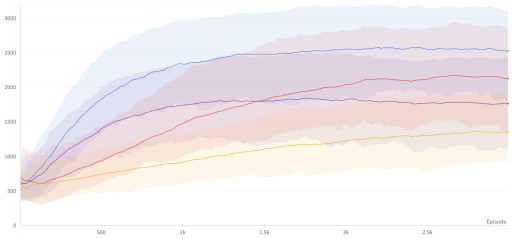


Figura 5.6: Recompença per Episodi

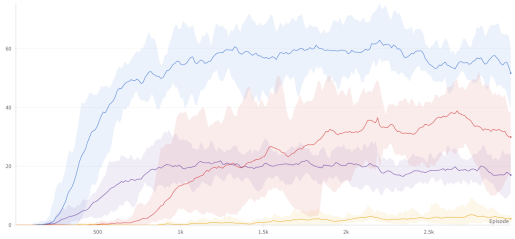


Figura 5.7: Win Rate per Episodi

5.6 Resultats Estudi Paràmetres *Action Advise*: Learning Rate

En aquest experiment es va examinar com diferents *learning rates* afecten el procés d'*Action Advice* i el rendiment global del model. La motivació d'aquest estudi era determinar com de determinant és el *learning rate* en l'aprenentatge de l'agent `teacher` en l'*Action Advice*, per al seu posterior ús en l'Experiment 4.

La hipòtesi formulada preveia que un *learning rate* menor (0.1) afavoriria el procés d'*Action Advice*, ja que és el valor habitualment utilitzat en aquests contextos. No obstant això, es considerava que un *learning rate* més alt (0.4) podria tenir un impacte negatiu en l'eficiència de l'aprenentatge de l'agent.

Per a aquest experiment, es va utilitzar el nivell 4-1 amb preentrenament al nivell 1-1. Les tècniques d'*Action Advice* utilitzades van ser *AA Decay* amb una delta de 0.15. Els grups de llavors (`seed_group`) utilitzats van ser `zeta_prima` i `xi_prima`.

Els models es van preentrenar al nivell 1-1 durant 3000 episodis utilitzant els models preentrenats corresponents a les dues execucions en el nivell 1-1 del `seed_group` `beta_prima`. Posteriorment, els models es van entrenar al nivell 4-1 amb *learning rates* de 0.1 i 0.4 durant 3000 episodis cadascun. Els resultats obtinguts es van comparar en termes de les mètriques escollides. A continuació es presenten i analitzen els resultats detallats d'aquest experiment.

lr	Recompensa	Temps(s)	Steps	Win Rate
0.0001	5892951.75	22598.607	596505.25	37.706
0.0004	5890816.75	22545.16	598789.75	39.105

Taula 5.7: Metricas Principals Estudi Learning Rate *Action Advise*

Recompensa Acomulada

Encara que la recompensa acumulada per a la *leaning Rate* de 0.0001 és lleugerament superior a la de 0.0004, aquesta diferència és molt petita. La recompensa acumulada amb 0.0001 és 0.036% més alta que amb 0.0004 (5892951.75 vs. 5890816.75).

Temps d'Entrenament

El temps total d'entrenament és lleugerament menor per a la *leaning Rate* de 0.0004. El temps d'entrenament amb 0.0004 és 0.24% més curt que amb 0.0001 (22545.16 segons vs. 22598.607 segons).

Nombre de Passos (Steps)

El nombre de passos utilitzats durant l'entrenament és lleugerament superior per a la *leaning Rate* de 0.0004. Els passos amb 0.0004 són 0.38% més alts que amb 0.0001 (598789.75 vs. 596505.25).

Win Rate

El *Win Rate* és notablement més alt amb la *leaning Rate* de 0.0004. El *Win Rate* amb 0.0004 és 3.71% més alt que amb 0.0001 (39.105% vs. 37.706%).

Havent repassat el resultats de la Taula 5.7, tot i que les diferències entre les taxes d'aprenentatge de 0.0001 i 0.0004 són mínimes en termes de recompensa acumulada, temps d'entrenament i nombre de passos, la *leaning Rate* de 0.0004 mostra un avantatge significatiu en termes de *Win Rate*.

Seguidament analitzarem el comportament d'altres mètriques per veure si através de veure com es comporten els valors d'aquestes al llarg dels episodis ens ajuda a discernir quina *leaning Rate* és més eficaç.

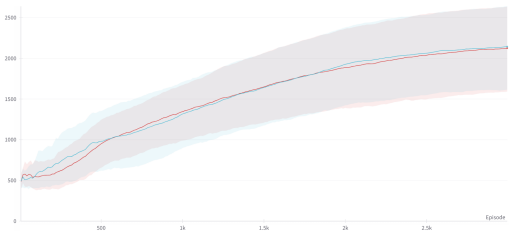


Figura 5.8: Recompensa per Episodi

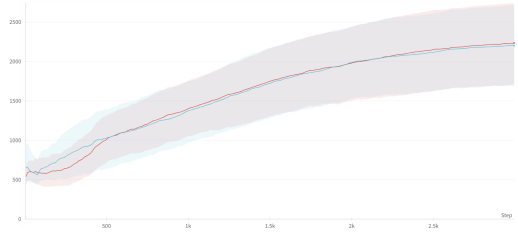


Figura 5.9: Posició Final per Episodi

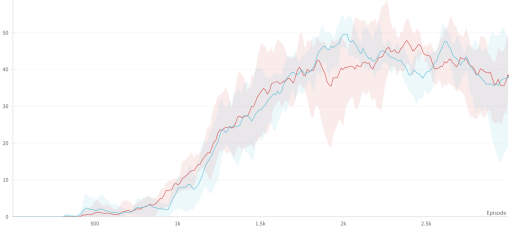


Figura 5.10: *Win Rate* últims 100 episodis per Episodi

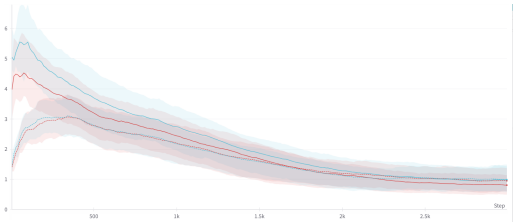


Figura 5.11: *Loss Teacher Loss Student* per Episodi

lr0001 lr0004 **Nota:** Totes les figures han estat suavitzades utilitzant una mitjana mòbil, menys la figura *Win Rate*

Recompenses, Posicions Finals i *Win Rate*

Les recompenses per episodi són una mica superiors amb la *leaning Rate* de 0.0001, que és el comportament esperat donat que la recompensa acumulada és superior, com podem veure a la taula 5.7.

En el gràfic de posicions finals es pot observar que la mitjana de les posicions finals és lleugerament superior per a la *leaning Rate* de 0.0004, de tota manera les diferències segueixen sent mínimes.

Per últim, encara que la taula 5.7 mostra que el *Win Rate* final és un 3.71% superior amb la *leaning Rate* de 0.0004, al observar el gràfic amb les mitjanes mòbils, es constata que la diferència també és mínima.

Pèrdua (*Loss*) del Model

Tant la pèrdua del `teacher` com la del `student` són menors amb una *leaning Rate* de 0.0004. A més, el punt de tall entre les pèrdues es produeix a les 1500 iteracions per la taxa de 0.0004, mentre

que per la taxa de 0.0001 sembla que es tallaran poc després dels 3000 episodis. Això indica una convergència més ràpida i eficient amb la *leaning Rate* de 0.0004.

Mètriques de Recompensa

Les mètriques de recompensa per episodi i de recompensa per temps són pràcticament idèntiques per ambdues taxes d'aprenentatge. Això suggereix que, malgrat les diferències en altres aspectes del rendiment, les taxes d'aprenentatge tenen un efecte mínim sobre aquestes mètriques específiques.

En resum, tot i que les diferències entre les taxes d'aprenentatge de 0.0001 i 0.0004 són mínimes en la majoria de mètriques, la *leaning Rate* de 0.0004 mostra alguns avantatges en termes de *Win Rate*, posicions finals mitjanes i velocitat de convergència de la pèrdua (*Loss*).

5.7 Interpretació de Mètriques d'*Action Advise*

Els resultats de la monitorització dels aconsellaments mostren l'efecte d'utilitzar diferents valors de δ. Aquests resultats confirmen el correcte funcionament del sistema d'*Action Advise* implementat.

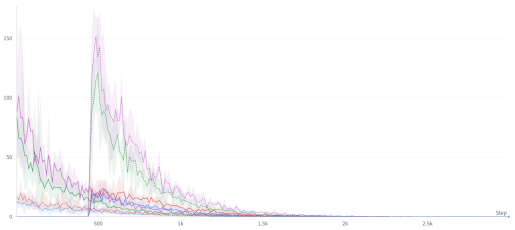


Figura 5.12: *Advise* per Espisodi

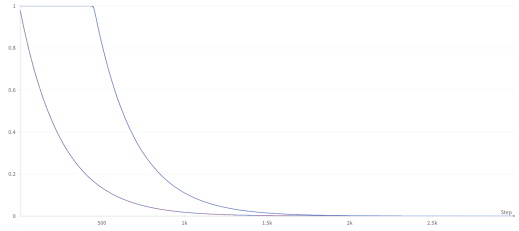


Figura 5.13: Valor de λ_t per Espisodi

AA Decay SUA TUA TSUA

Segons les expectatives, el control del *Budget Advise* a través de la funció *Decay* λ_t funciona correctament. A la Figura 5.14 i a la Taula 5.8 s'analitza el nombre total d'accions aconsellades per cada tècnica.

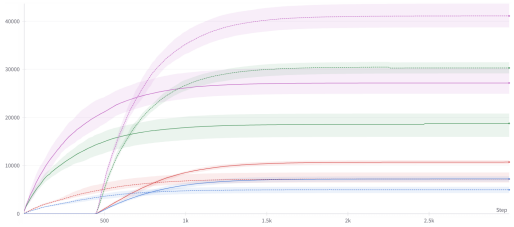


Figura 5.14: Total d'*Advise* per Espisodi

Tècnica	$\delta = 0.0$	$\delta = 0.15$	**Percentatge d'augment**
AA Decay	27170.5	41105	51.31%
SUA	7216.833	10741	48.83%
TUA	4989	7232.333	44.99%
TSUA	18800.5	30504.167	62.23%

Taula 5.8: Total d'accions recomanades per cada tècnica amb $\delta = 0.0$ i $\delta = 0.15$, i el percentatge d'augment.

Tant la Figura 5.14 com la Taula 5.8 confirmen aquests resultats. La següent secció 5.8 examina l'efecte que aquestes diferències substancials tenen en les recompenses obtingudes.

A continuació, es presenten els resultats obtinguts en la recollida dels valors de les incerteses, sobreposats als llindars d'incertesa. S'analitzen les observacions generals derivades de les gràfiques, així com les diferències específiques entre els agents `student` i `teacher`. Finalment, es compara entre els punts de tall u_S i τ_S, u_T i τ_T.

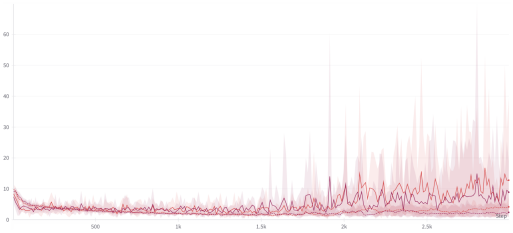


Figura 5.15: Evolució de la incertesa i el llindar d'incertesa del agent `student` per Episodi $\delta = 0.0$

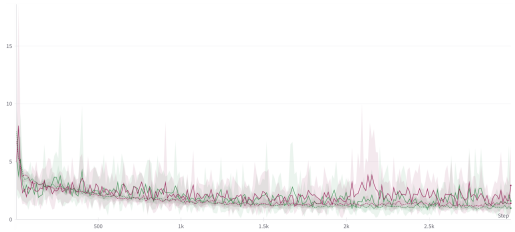


Figura 5.16: Evolució de la incertesa i el llindar d'incertesa del agent `teacher` per Episodi $\delta = 0.0$

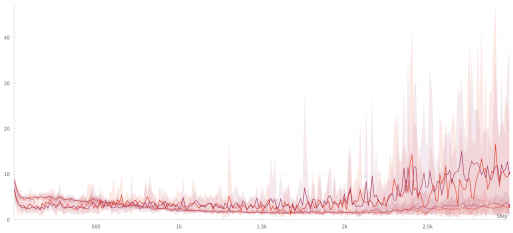


Figura 5.17: Evolució de la incertesa i el llindar d'incertesa del agent `student` per Episodi $\delta = 0.15$

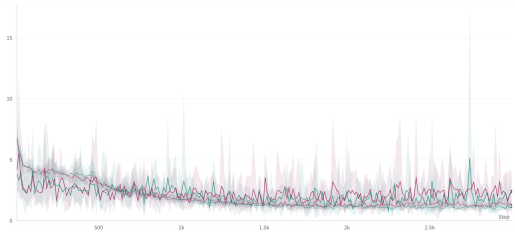


Figura 5.18: Evolució de la incertesa i el llindar d'incertesa del agent `teacher` per Episodi $\delta = 0.15$

SUA TUA TSUA

Note: La imatge 5.16 se li ha tret una execució que era un outlier que espallava la compreció de la grafica. Es pot trobar la imatge amb l'outlier a l'annex C.2

La imatge 5.16 se li ha tret una execució que era un outlier que espallava la compreció de la grafica. Es pot trobar la imatge amb l'outlier a l'annex C.2

Observem que el llindar d'incertesa s'estabilitza ràpidament al voltant de 2,5. Com que la fórmula utilitzada només multiplica la desviació, això fa que el valor mínim possible que pot prendre el llindar tendeixi a la mitjana dels últims 100 valors de la incertesa.

En les gràfiques del `student` (Figures 5.15 i 5.17), a partir de l'episodi 1500, el valor de la incertesa comença a créixer. Degut a que el multiplicador de la desviació és inferior a 0,01 a partir de la iteració 1150 en el cas $\delta = 0.0$ i 1600 en el cas de $\delta = 0.15$, l'efecte d'aquest increment és mínim sobre el llindar, encara que és més significatiu en el cas de l'algorisme SUA. Aquest efecte es discuteix en profunditat a les conclusions (**??**). És important tenir en compte que aquest fenomen és, a nivell efectiu, insignificant, ja que pel mateix motiu que l'afectació és mínima, el valor de λ_t, el nombre de recomanacions en aquest punt, és ínfim.

Encara que l'agent `student` ha arribat a màxims de 60 en alguna execució del TSUA amb $\delta = 0.0$, la majoria no superen els 40, igual que amb $\delta = 0.15$.

Pel que fa a les gràfiques del `teacher` (Figures 5.16 i 5.18), la incertesa és semblant a la del `student` fins a l'episodi on la del `student` es dispara. El gràfic del `teacher` es mostra molt estable. La incertesa del `teacher` és menor, com és raonable, ja que és l'agent preentrenat. L'algorisme TUA arriba a un màxim de 15 en alguna execució al final de l'entrenament.

Els punts de tall u_S i τ_S, u_T i τ_T varien considerablement i hi ha diverses iteracions. Encara que no es pot parlar d'un punt de tall definit, ens referirem al punt on visualment el llindar comença a ser inferior a la mitjana de la incertesa de l'agent en aquell episodi.

Es podria dir que, per a cada configuració, aquest punt és semblant, independentment de la tècnica utilitzada. No obstant això, en ambdós casos, l'episodi en què succeeix aquest fenomen és lleugerament posterior per a l'agent `teacher`. Així mateix, es pot apreciar que la distància entre aquests punts per a diferents configuracions de la mateixa tècnica és major en el cas del `teacher`.

5.8 Interpretació de Mètriques Experimentals

En aquest capítol es presenten les diferents tècniques utilitzades en l'experimentació. Cada tècnica té unes característiques específiques que es descriuen a continuació:

- **SS NA**:
  Cota Superior - *Semantic Segmentation Non Advising*
  Aquesta tècnica utilitza segmentació semàntica sense cap tipus de consell.

- **RGB FT**:
  Cota Mitja - *RGB Fine-tuning*
  Aquesta tècnica aplica una afinació fina a les imatges RGB. Serveix com a cota mitjana per a la comparació.

- **RGB NA**:
  Cota Inferior - *RGB Non Advising*
  Aquesta tècnica utilitza imatges RGB sense cap tipus de consell.

- **RGB AA Decay**:
  RGB Action Advise only Decay guided - Aquesta tècnica utilitza consells d'acció amb una guia de descomposició.

- **RGB SUA**:
  RGB Student Uncertainty guided Advise - Aquesta tècnica guia els consells basant-se en la incertesa de l'estudiant.

- **RGB TUA**:
  RGB Teacher Uncertainty guided Advise - Aquesta tècnica guia els consells basant-se en la incertesa del professor.

- **RGB TSUA**:
  RGB Teacher Student Uncertainty guided Advise - Aquesta tècnica combina la incertesa del professor i de l'estudiant per guiar els consells.

Model	Delta	Recompensa	Temps (s)	Steps	Win Rate (%)
SS NA	-	**7142352.25**	**33954.58**	**690859**	**690859**
RGB NA	-	5776623.583	15970.764	593035.667	33.32
RGB FT	-	6620605.167	17847.695	665508	39.209
RGB AA Decay	0	**6006498.25**	**22864.398**	**609857.75**	36.968
RGB AA Decay	0.15	5890816.75	22545.16	598789.75	**39.105**
RGB SUA	0	5224135.667	23562.621	552480.833	11.57
RGB SUA	0.15	**5593137.5**	**24870.624**	**575825**	**17.408**
RGB TUA	0	**5955739.5**	**25716.886**	**609222.333**	**36.83**
RGB TUA	0.15	5595085	24129.505	568792.667	36.101
RGB TSUA	0	**5613469.667**	**27027.325**	**576442.167**	**25.213**
RGB TSUA	0.15	5517709.333	27047.15	572157.333	12.501

Taula 5.9: Resultats de l'Experiment amb Diferents Models i Paràmetres

Interpretació

La taula 5.9 mostra els resultats obtinguts amb les diferents tècniques. A continuació, es fa una anàlisi comparativa centrada especialment en les tècniques que utilitzen diferents valors de δ, mentre que les tècniques sense δ es consideren *baselines*.

SS NA

Aquesta tècnica, *Semantic Segmentation Non Advising*, serveix com a punt de referència superior amb una recompensa acumulada de 7,142,352.25, un temps d'entrenament de 33,954.58 segons, i 690,859 *steps*. Això representa un *baseline* alt per a la segmentació semàntica, essent la cota superior contra la qual es comparen les altres tècniques.

RGB NA i RGB FT

Les tècniques *RGB Non Advising* i *RGB Fine-tuning* també serveixen com a *baselines*. *RGB NA* representa una cota mitjana amb una recompensa de 5,776,623.583, un temps de 15,970.764 segons, i 593,035.667 *steps*. *RGB FT*, amb una recompensa de 6,620,605.167, un temps de 17,847.695 segons, i 665,508 *steps*, i un *win rate* de 39.209%, representa una cota inferior per a la comparació.

RGB AA Decay

Quan es compara *RGB Action Advise only Decay guided* amb δ de 0 i 0.15, es veu una millora en el *win rate* del 5.77% (de 36.968% a 39.105%). La recompensa disminueix un 1.93%, de 6,006,498.25 a 5,890,816.75, mentre que el temps i els *steps* es redueixen lleugerament.

Comparació	$\delta = 0$	$\delta = 0.15$
Cota Superior (*SS NA*)	17.56% menor	17.52% menor
Cota Mitjana (*RGB FT*)	9.12% menor	11.02% menor
Cota Inferior (*RGB NA*)	3.86% superior	2.10% superior

Taula 5.10: Comparacions per *RGB AA Decay*

El millor valor de δ per a *RGB AA Decay* és 0.15, donada la millora en el *win rate*.

RGB SUA

En el cas de *RGB Student Uncertainty guided Advise*, un δ de 0.15 millora significativament la recompensa en un 7.06% (de 5,224,135.667 a 5,593,137.5), i el *win rate* en un 50.41% (de 11.57% a 17.408%). El temps augmenta un 5.54% (de 23,562.621 a 24,870.624 segons) i els *steps* augmenten un 4.23%.

Comparació	$\delta = 0$	$\delta = 0.15$
Cota Superior (*SS NA*)	26.83% menor	21.59% menor
Cota Mitjana (*RGB FT*)	21.09% menor	15.49% menor
Cota Inferior (*RGB NA*)	9.59% menor	3.17% menor

Taula 5.11: Comparacions per *RGB SUA*

El millor valor de δ per a *RGB SUA* és 0.15, donada la millora significativa en la recompensa i el *win rate*.

RGB TUA

Per a *RGB Teacher Uncertainty guided Advise*, el δ de 0.15 incrementa la recompensa en un 10.15% (de 5,955,739.5 a 5,595,085), i el *win rate* en un 0.73% (de 36.83% a 36.101%). El temps augmenta

Comparació	$\delta = 0$	$\delta = 0.15$
Cota Superior (*SS NA*)	16.50% menor	21.66% menor
Cota Mitjana (*RGB FT*)	10.12% menor	15.47% menor
Cota Inferior (*RGB NA*)	3.10% superior	3.18% menor

Taula 5.12: Comparacions per *RGB TUA*

un 5.55% (de 25,716.886 a 24,129.505 segons) i els *steps* disminueixen un 6.63% (de 609,222.333 a 568,792.667).

El millor valor de δ per a *RGB TUA* és 0.15, donada la millora en la recompensa i el *win rate*.

RGB TSUA

En la tècnica *RGB Teacher Student Uncertainty guided Advise*, un δ de 0.15 mostra una reducció en la recompensa del 1.70% (de 5,613,469.667 a 5,517,709.333) i el *win rate* disminueix en un 50.46% (de 25.213% a 12.501%). El temps és pràcticament el mateix, amb una lleugera variació del 0.07% (de 27,027.325 a 27,047.15 segons), i els *steps* també mostren una reducció mínima del 0.74%.

Comparació	$\delta = 0$	$\delta = 0.15$
Cota Superior (*SS NA*)	21.60% menor	22.74% menor
Cota Mitjana (*RGB FT*)	15.57% menor	16.61% menor
Cota Inferior (*RGB NA*)	2.12% menor	4.49% menor

Taula 5.13: Comparacions per *RGB TSUA*

El millor valor de δ per a *RGB TSUA* és 0, donada la menor reducció en la recompensa.

En resum, les tècniques que utilitzen diferents valors de *delta* mostren variacions significatives en el rendiment, amb millores notables en el *win rate* i la recompensa en alguns casos, mentre que altres mostren increments en el temps i els *steps* requerits.

5.9 Interpretació de les gràfiques de recompensa i *Win Rate*

Aquesta secció presenta una anàlisi comparativa de diverses tècniques utilitzades, basant-se en les gràfiques de recompensa i el *Win Rate*. Es proporcionen observacions detallades sobre el rendiment de les tècniques analitzades, destacant els aspectes clau que influeixen en els resultats obtinguts.

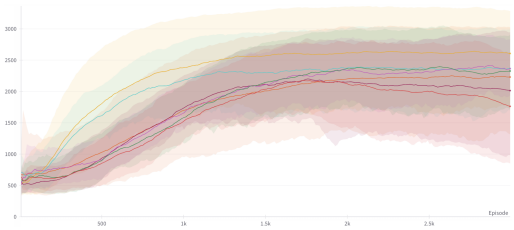


Figura 5.19: Recompensa per Episodi $\delta = 0.0$

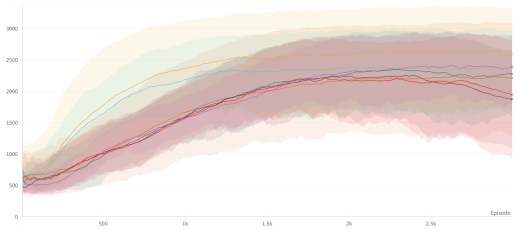


Figura 5.20: Recompensa per Episodi $\delta = 0.15$

Nota: Les figures han estat suavitzades utilitzant una mitjana mòbil.

En primera instància, observem que hi ha dues tècniques, SUA i TSUA, que es situen clarament per sota de la cota inferior. AA Decay i TUA mostren una tendència a aproximar-se a la cota mitjana

a partir de l'episodi 2000, especialment en el cas de $\delta = 0.0$. Quan $\delta = 0.15$, aquesta tendència és més evident en AA Decay que en TUA. Com era d'esperar, l'algorisme de Segmentació Semàntica, que representa la cota superior, obté els millors resultats.

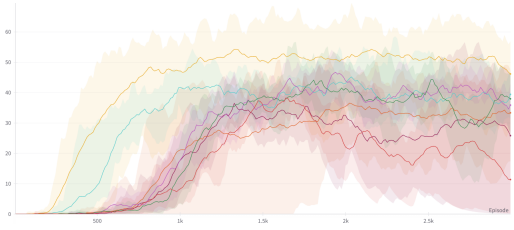


Figura 5.21: *Win Rate* per Episodi $\delta = 0.0$

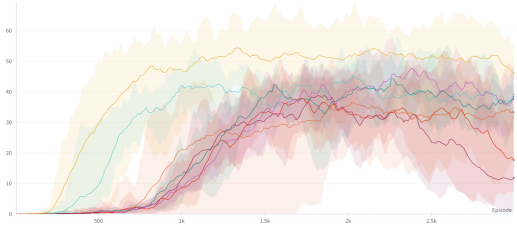


Figura 5.22: *Win Rate* per Episodi $\delta = 0.15$

En relació amb el *Win Rate*, es mantenen els patrons observats en la Figura **??**. Aquesta gràfica, que no utilitza la mitjana mòbil[1], permet observar l'evolució dels valors de *Win Rate* de manera més detallada. Els valors finals es poden consultar a la Taula 5.9 de la secció anterior, on es representa la proximitat entre els *Win Rate* observats.

5.10 Discussió de les Hipòtesis

En aquesta secció es discutiran les hipòtesis plantejades per a cadascun dels experiments, analitzant els resultats obtinguts i la seva coherència amb les hipòtesis formulades.

5.10.1 Discussió de l'Experiment 1

Hipòtesi: S'espera que tant RGB NA com SS NA millorin amb l'estudi previ de paràmetres. Tot i així, no s'espera que la versió millorada de RGB NA superi el rendiment de SS NA sense millorar. D'altra banda, s'espera que el *Fine-tune* (RGB FT), sent una tècnica que presenta un bon rendiment en aquest entorn, aconsegueixi un rendiment similar a SS NA sense haver estudiat els paràmetres.

Discussió: Els resultats de l'experiment 1 5.4 mostren que l'estudi previ de paràmetres ha millorat el rendiment tant de RGB NA com de SS NA. Com s'esperava, la versió millorada de RGB NA no ha superat el rendiment de SS NA sense millorar, però s'ha equiparat. Pel que fa al *Fine-tune* (RGB FT), els resultats han confirmat que aconsegueix un rendiment similar a SS NA sense haver estudiat els paràmetres, validant així la hipòtesi.

5.10.2 Discussió de l'Experiment 2

Hipòtesi: S'espera que l'ús del *dropout* no sigui adequat per aquest entorn específic i que, en conseqüència, degradi el rendiment del model.

Discussió: Els resultats de l'experiment 2 5.5 confirmen que l'ús del *dropout* no és adequat per aquest entorn específic. L'aplicació del *dropout* ha degradat el rendiment del model, tal com s'esperava, donant suport a la hipòtesi formulada. Això suggereix que, en entorns on es busca un sobre-ajustament per resoldre tasques concretes, el *dropout* pot ser contraproduent.

[1]La mitjana mòbil és una tècnica estadística que consisteix a suabitzar una sèrie temporal mitjançant el càlcul de la mitjana d'un conjunt de valors consecutius. Això permet reduir el soroll i mostrar tendències més clares en les dades.

5.10.3 Discussió de l'Experiment 3

Hipòtesi: Es preveu que un *learning rate* menor (0.1) afavorirà el procés d'*Action Advice*, ja que és el valor habitualment utilitzat en aquests contextos. No obstant això, es considera que un *learning rate* més alt (0.4) podria tenir un impacte negatiu en l'eficiència de l'aprenentatge de l'agent.

Discussió: Els resultats de l'experiment 3 **??** indiquen que les diferències entre les taxes d'aprenentatge de 0.1 i 0.4 són mínimes en la majoria de mètriques. Tot i això, la taxa d'aprenentatge de 0.4 mostra alguns avantatges en termes de *Win Rate*, posicions finals mitjanes i velocitat de convergència de la pèrdua (*Loss*). Això suggereix que, tot i que es preveia que un *learning rate* més alt tindria un impacte negatiu, en aquest cas hem observat un benefici mínim, contrariant parcialment la hipòtesi original.

5.10.4 Discussió de l'Experiment 4

Hipòtesi 1: Ja que en el treball de [3] s'esmentava que no asseguraven que la tècnica pogués funcionar tan bé si les tasques font i objectiu eren massa diferents, en el cas que fossin més semblants, s'espera que funcioni millor.

Discussió: Els resultats mostren que les tècniques d'*Action Advise* implementades en un context on la distància entre la tasca font i la tasca objectiu és petita, ha resultat no ser òptim per al problema a resoldre. Per tant, es rebutja la hipòtesis plantejada.

Hipòtesi 2: La tècnica TSUA, sent la més restrictiva, mostrarà un millor rendiment al tenir en compte tant la incertesa del professor com de l'estudiant per decidir si es pren l'acció proposada pel professor.

Discussió: Els resultats indiquen que la tècnica TSUA ha obtingut un rendiment inferior a la resta de tècniques, amb $\delta = 0.15$ i de la segona pitjor amb $\delta = 0.0$, rebutjant la hipotesis plantejada. Per tant, que les condicions d'aconsellament siguin més restrictives es tradueix en un millor rendiment.

Hipòtesi 3: En l'estudi del paràmetre δ, s'espera que l'algoritme SUA tingui un millor rendiment amb $\delta = 0.0$, mentre que l'algoritme TUA tindrà un rendiment òptim amb $\delta = 0.15$. Aquesta expectativa es basa en resultats previs d'investigacions que han seguit aquestes configuracions.

Discussió: Els resultats experimentals mostren que tant SUA com TUA tenen un rendiment similar amb tots dos valors de δ. No obstant això, mentre que el rendiment de TUA és gairebé idèntic amb $\delta = 0.0$ i $\delta = 0.15$, l'algoritme SUA sí que mostra una millora notable quan s'utilitza $\delta = 0.15$. Per tant, la primera part de la hipòtesi queda rebutjada, ja que una $\delta = 0.15$ dona millor rendiment en l'algorisme SUA, i la segona també, ja que donen tots dos valors de δ ens donen rendiments marginalment similars.

Capítol 6

Conclusions

En aquest projecte, s'han estudiat diverses tècniques d'*Action Advise* en l'àmbit del *Reinforcement Learning*, analitzant el seu rendiment en diferents condicions experimentals. En aquesta secció farem un resum dels resultats principals obtinguts i de les línies de recerca que s'obren a futur.

6.1 Anàlisi dels resultats del projecte

En primer lloc, es va investigar l'impacte de l'estudi previ de paràmetres, demostrant que aquest millora significativament el rendiment de les tècniques RGB NA i SS NA. La tècnica *Fine-tune* (RGB FT) ha aconseguit un rendiment comparable al de SS NA sense millorar, validant així la hipòtesi inicial.

En segon lloc, es va analitzar l'ús del *dropout*, confirmant que aquest no és adequat per a entorns que requereixen un sobreajustament per a resoldre tasques concretes, ja que degrada el rendiment del model. Pel que fa al *learning rate*, els resultats van mostrar que, tot i que es preveia un impacte negatiu d'una taxa d'aprenentatge més alta, la diferència entre les taxes de 0.1 i 0.4 va ser mínima en la majoria de mètriques, amb alguns avantatges observats per a la taxa de 0.4.

Finalment, l'avaluació de les tècniques d'*Action Advise* en un context on la distància entre la tasca font i la tasca objectiu és petita no va resultar òptima, refutant la hipòtesi que aquestes tècniques funcionarien millor en aquest context. A més, la tècnica TSUA, sent la més restrictiva, va mostrar un rendiment inferior, contrariant la hipòtesi que condicions més restrictives millorarien el rendiment. En l'estudi del paràmetre δ, es va observar que tant SUA com TUA tenen rendiments similars amb ambdós valors de δ, tot i que SUA mostra una millora notable amb $\delta = 0.15$, refutant parcialment la hipòtesi plantejada.

En relació al procés d'implementació del *Virtual Environment*, ha permès assolir els objectius. Hem dut a terme la investigació proposada, aconseguint desenvolupar un sistema flexible i adaptatiu. Aquest sistema no només compleix els requisits inicials, sinó que també obre la porta a futures extensions i millores.

El sistema implementat permet la integració senzilla de noves configuracions d'agents, com ara AgentPPO o AgentA2C. Afegir aquestes noves configuracions implica només la creació de noves classes d'agents i funcions d'entrenament específiques per a cada algorisme. Per exemple, per integrar l'algorisme PPO, només caldria afegir una classe `AgentPPO` i una funció `TrainPPO` que tingui en compte les particularitats de l'actualització de política d'aquest algorisme.

6.2 Treballs Futurs

6.2.1 Línea d'investigació sobre algoritmes d'*Action Advise*

Els resultats d'aquesta investigació han obert diverses línies de treball futur per a continuar millorant i comprenent les tècniques d'*Action Advise* i els seus paràmetres en entorns de *Reinforcement Learning*.

- **Estudiar el llindar d'incertesa:** Profunditazar en l'estudi del càlcul del llindar d'incertesa en experimentacións o sistemes amb *Budged Advise* ha donat bons resultats per estudiar si existeix una millora amb el sistema λ *Decay*.
- **Estudi de nous paràmetres:** Investigació de l'impacte d'altres paràmetres no considerats en aquest estudi per a millorar el rendiment de les tècniques d'*Action Advise*.
- **Refinament de tècniques existents:** Desenvolupar noves variants de les tècniques SUA, TUA i TSUA que puguin abordar millor les limitacions observades en aquest estudi. Com per exemple aplicar utilitzar una xarxa neuronal per controlar l'aconsellament d'accions o aplicar l'algorisme AIR-SUA, presentat en el paper Sahir et al. [14].
- **Combinació de tècniques:** Explorar la combinació de tècniques d'*Action Advise* amb altres estratègies d'aprenentatge per a optimitzar el rendiment global dels models.

Aquests treballs futurs permetran aprofundir en la comprensió i millora de les tècniques d'*Action Advise*, contribuint a l'avanç de la recerca en *Reinforcement Learning*.

6.2.2 Extenció del sistema d'experimentació

Tot i que el projecte ha complert amb èxit els objectius establerts, existeixen diverses vies per a futurs treballs que podrien ampliar i millorar encara més el sistema implementat. A continuació es presenten algunes direccions potencials per a la recerca i desenvolupament futurs:

- **Integració de nous algorismes d'aprenentatge per reforç**: Una de les extensions més prometedores és la integració d'algorismes addicionals com *Proximal Policy Optimization* (PPO) i *Advantage Actor-Critic* (A2C). Afegir aquests algorismes al sistema actual és senzill, ja que només caldria crear noves classes d'agents (`AgentPPO` i `AgentA2C`) i funcions d'entrenament (`TrainPPO` i `TrainA2C`) que tinguin en compte les particularitats d'actualització de política de cada algorisme.
- **Utilització de nous jocs**: Afegir la possibilitat d'entrenar agents en jocs diversos pot enriquir significativament la utilitat del sistema per a investigacions diverses. No obstant això, caldran modificacions en la manera com es rep l'estat de l'entorn per assegurar que els nous entorns es poden integrar correctament amb el sistema existent. Això permetria l'estudi de tècniques de *Transfer Learning* per a la transferència efectiva entre diferents entorns.
- **Aplicació de noves tècniques de *Transfer Learning***: La possibilitat d'incorporar diferents tècniques de *Transfer Learning* permetrà explorar com es poden transferir coneixements d'un entorn o tasca a un altre. Això requerirà petites modificacions en la manera com es gestiona la transferència de coneixement i l'adaptació de les polítiques entre diferents entorns.

Aquests treballs futurs no només augmentaran la capacitat i flexibilitat del sistema actual, sinó que també permetran la seva aplicació en una àmplia gamma d'entorns i tasques, fomentant així l'avanç continu en el camp de l'aprenentatge per reforç.

Bibliografia

[1] Ofra Amir, Ece Kamar, Andrey Kolobov, and Barbara Grosz. Interactive teaching strategies for agent training. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[2] Aaron Bastani. *Fully Automated Luxury Communism*. Verso, London, 2019.

[3] Joseph Campbell, Yue Guo, Fiona Xie, Simon Stepputtis, and Katia Sycara. Introspective action advising for interpretable transfer learning. In *Proceedings of The 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pages 1072–1090, 2023.

[4] Andrew Grebenisan. Building a double deep q-network to play super mario bros. `https://blog.paperspace.com/building-double-deep-q-network-super-mario-bros/`, 2018. Accessed: 19-Jun-2024.

[5] Yue Guo, Joseph Campbell, Simon Stepputtis, Ruiyu Li, Dana Hughes, Fei Fang, and Katia Sycara. Explainable action advising for multi-agent reinforcement learning. *arXiv preprint arXiv:2211.07882*, 2023.

[6] Gym Documentation. Atari environments. `https://www.gymlibrary.dev/environments/atari/index.html`, 2022.

[7] Ercument Ilhan, Jeremy Gow, and Diego Perez Liebana. Action advising with advice imitation in deep reinforcement learning. In *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems*, pages 629–637, 2021.

[8] Ercument Ilhan, Jeremy Gow, and Diego Perez Liebana. Learning on a budget via teacher imitation. *CoRR*, abs/2104.08440, 2021.

[9] Ercument Ilhan and Diego Perez Liebana. Student-initiated action advising via advice novelty. *CoRR*, abs/2010.00381, 2020.

[10] Christian Kauten. Super Mario Bros for OpenAI Gym. GitHub, 2018.

[11] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, 2016.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[13] Javier Montalvo, Álvaro García-Martín, and Jesús Bescós. Exploiting semantic segmentation to boost reinforcement learning in video game environments. *Multimedia Tools and Applications*, 2022. Received: 29 March 2022 / Revised: 29 May 2022 / Accepted: 15 August 2022.

[14] Ilhan E. Sahir, Subhojoy Das, and Matthew E. Taylor. Methodical advice collection and reuse in deep reinforcement learning. In *Proc. of the Adaptive and Learning Agents Workshop (ALA 2022)*, pages 51–65. ALA, 2022. Online: ALA. Cruz, Hayes, da Silva, Santos (Eds.). Alberta

Machine Intelligence Institute (Amii) & University of Alberta & Queen Mary University of London. [Enllaç](https://ala2022.github.io/).

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[16] Felipe Leno Da Silva, Ruben Glatt, and Anna Helena Reali Costa. Simultaneously learning and advising in multiagent reinforcement learning. In *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*, pages 1100–1108, 2017.

[17] Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, July 2009. Submitted 6/08; Revised 5/09; Published 7/09.

[18] Lisa Torrey and Matthew Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, pages 1053–1060, 2013.

[19] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 27. NIPS Foundation, Curran Associates, Inc., 2014.

[20] Changxi Zhu, Yi Cai, Ho fung Leung, and Shuyue Hu. Learning by reusing previous advice in teacher-student paradigm. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1674–1682, 2020.

Apèndix A

Planificació

A.1 Planificació temporal

Aquest projecte te un temps de realitació de 4 mesos aproximadament. En aquests 4 mesos s'han de realitzar tota una sèrie de tasques que seràn detallades en les següents subseccions.

A.1.1 Descripció de tasques

Recerca

En aquesta secció es detallaran les diferents seccions i etapes que han conformat el desenvolupament de l'estudi previ d'aquest projecte. Com el projecte, ja en els seus inicis, va patir un canvi de tema, l'extensió de l'estudi previ ha sigut més llarga del que es podria considerar desitjable en un projecte de 4 mesos.

Un cop produït aquest viratge, l'estudi previ va comptar amb les següents etapes:

- **EP1:** Recerca de la tècnica a estudiar: 30h (08/03/2024 - 21/03/2024)
- **EP2:** Estudi de l'entorn on aplicar la tècnica: 10h (08/03/2024 - 21/03/2024)

Administració del Projecte

Administrar correctament el projecte és clau per al seu èxit, doncs proporciona un marc organitzat que assegura el compliment dels objectius de manera eficaç i dins dels terminis i pressupostos establerts, tot gestionant de manera apropiada els riscos i mantenint els estàndards de qualitat. Es preveu que l'administració del projecte duri unes 135 hores i es desglossarà en diverses subtasques com s'especifica a continuació:

- **GP1 - Definició de l'Abast:** Abans de començar un projecte, sobretot un projecte d'investigació en un temps reduït com amb el que comptem, és vital definir l'abast del treball, els requisits que ha de complir una investigació, els objectius i subobjectius, i els obstacles derivats d'una gran càrrega de desenvolupament, que tindran un pes notori en el desenvolupament del treball i la seva realització. S'han assignat unes 5 hores a aquesta fase inicial (22/03/2024 - 29/03/2024).
- **GP2 - Planificació Temporal:** La planificació acurada del projecte implica establir les activitats, distribuir responsabilitats i preveure els terminis i recursos necessaris. Aquesta planificació pretén optimitzar el temps i recursos, reduint riscos i garantint el seguiment dels objectius del projecte. S'han dedicat unes 5 hores a aquesta etapa (22/03/2024 - 29/03/2024).
- **GP3 - Pressupost:** Es desenvoluparà un pressupost detallat que inclogui tots els costos del projecte, tant de personal com de materials, a més de considerar despeses generals i contingències. Elaborar un bon pressupost permet obtenir una visió realista del cost del

projecte per a una distribució eficient dels recursos. Aquesta tasca prendrà unes 5 hores (22/03/2024 - 29/03/2024).

- **GP4 - Informe de Sostenibilitat:** Aquesta fase inclou l'avaluació de l'impacte ambiental, social i econòmic del projecte. La durada estimada per a aquesta etapa és de 3 hores (29/03/2024 - 01/04/2024).
- **GP5 - Seguiment del Projecte:** El seguiment inclourà reunions setmanals amb els directors del projecte els dijous de 11 a 12 del matí. A vegades seran en línia i altres presencials, depenent de les necessitats i situacions setmanals. Aquestes reunions són importants per a la monitorització del procés, la solució de dubtes puntuals i la guia de la investigació. Es preveu dedicar unes 14 hores a aquesta tasca en reunions formals(08/03/2024 - 16/06/2024).
- **GP6 - Documentació:** La preparació d'una memòria final és crucial per a l'avaluació del projecte, i considerant la seva importància per a la qualificació final, es dedicarà bastant temps a aquesta tasca. La documentació es realitzarà de forma paral·lela al desenvolupament del codi, registrant els progressos. Es dedicaran unes 80 hores a aquesta tasca, donada l'extensió dels documents a elaborar. Els caps de setmana es dedicaran a la documentació fins arribar a l'experimentació, que ja es farà diariament (22/03/2024 - 10/06/2024).
- **GP7 - Defensa del Projecte:** Preparació per a la defensa de la tesi, prevista entre l'11/06/2024 i el 28/06/2024, amb una durada de 20 hores.

Les etapes d'abast, planificació temporal i pressupost s'estudiaran entre el 22 de març i el 29 de març. Tot i això, es pararà un dia després de la implementació i després del refinament, per revisar aquestes seccions i adaptar-les a les necessitats o dificultats que puguin sorgir.

Implementació

La tècnica a estudiar es una tècnica enfocada a un paradigma que no es amb el que nosatres treballarem, per tant trobar una adaptació d'aquesta tècnica implicarà un primer desenvolupament en aquesta via.

Per un canto, encara que l'entorna amb el que treballem no es un entorn nou per nosaltres, una primera etapa de familiarització amb l'entorn es imprescidible abans de començar a implementar la tècnica a estudiar. Entendre les funcionalitats que ofereix un entorn que no ha desenvolupat un mateix, entendre les possibilitats que ofereix i com es pot utilitzar per al correcte desenvolupament del nostre estudi.

Seguidament, s'haura d'adaptar l'entorn a les necessitats de la nostre tècnica, a les dades que es vulguin recollir i a qualsevol modificació del codi que requerim per poder millora la comoditat, l'eficiencia o el rigor del nostre estudi.

Per últim, com a tot projecte informàtic, el manteniment del codi s'espera que sigui constant ja que es possible que hi hagi funcionalitats implementades que alhora de posar-les en pràctica poden necessitar ajustos i correccions.

I0 - Familitarització (10h) I1 - Implementació d'un prototip de la tècnica (40h) I2 - Implementar estratègies de seguiment (5h) M - Manteniment del codi (20h)

RP - Refinament de paràmetres

En aquesta secció es descriuran els diferents paràmetres que es refinaran per optimitzar el rendiment de l'algorisme Double Deep Q-Network (DDQN) i la tècnica *Introspective Action Advising (IAA)*.

Les execucions del codi consistiran en 3000 episodis. Les primeres execucions del codi en mostren que triga unes 3 hores en realitzar aquest nombre d'episodis. Per a cada configuració, es realitzaran 6 execucions, cadescuna amb una llavor diferent, provant 3 valors diferents per paràmetre, resultant en un temps total de 54 hores per configuració. Suposant que serà possible tenir dues configuracions diferents treballant alhora, el temps total d'execució es reduirà a 27 hores per paràmetre a estudiar.

A continuació, es detallen els paràmetres que es refinaran per als algorismes DDQN i IAA:

Paràmetres del DDQN

- **Capes Convolucionals**: Ajustar el nombre i la configuració de les capes convolucionals per millorar la capacitat d'aprenentatge de l'algorisme.
- **Batch Size**: Determinar la mida òptima del lot per a l'entrenament, equilibrant la velocitat de convergència i l'ús eficient dels recursos computacionals.
- **Learning Rate**: Ajustar la taxa d'aprenentatge per assegurar una òptima adaptació del model als patrons de les dades.
- **Exploration Decay**: Configurar la taxa de decreixement de l'exploració per equilibrar l'exploració de nous estats i l'explotació del coneixement adquirit.
- **Gamma**: Ajustar el factor de descompte per controlar la importància de les recompenses futures en la presa de decisions.
- **Target Update**: Determinar la freqüència d'actualització de la xarxa objectiu per estabilitzar l'entrenament.
- **Max Memory Size**: Configurar la mida màxima de la memòria d'experiència per assegurar que l'algorisme tingui accés a una quantitat suficient de dades històriques.

Paràmetres de l'IAA

- **Batch Size**: Determinar la mida òptima del lot per a l'entrenament, assegurant una gestió eficient de les dades d'aprenentatge.
- **Learning Rate**: Ajustar la taxa d'aprenentatge per optimitzar la capacitat d'adaptació de l'algorisme.
- **Exploration Decay**: Configurar la taxa de decreixement de l'exploració per trobar un bon equilibri entre l'exploració i l'explotació.
- **Gamma**: Ajustar el factor de descompte per assegurar que les recompenses futures s'incorporin de manera adequada en la presa de decisions.
- **Target Update**: Determinar la freqüència d'actualització de la xarxa objectiu per estabilitzar l'aprenentatge i evitar grans variacions en els resultats.
- **Max Memory Size**: Configurar la mida màxima de la memòria d'experiència per assegurar que l'algorisme pugui aprendre de manera eficient a partir de les dades acumulades.

El temps total de refinament per a cada paràmetre es calcula com la suma del temps d'execució de totes les configuracions provades. Amb un total de 13 paràmetres a estudiar (7 per al DDQN i 6 per a l'IAA), cada un requerint 27 hores d'execució, el temps total de refinament serà:

$$\text{Temps total de refinament} = 13 \text{ paràmetres} \times 27 \text{ hores/paràmetre} = 351 \text{ hores}$$

Suposant que les execucions es poden realitzar de forma ininterrompuda, això equivaldria a aproximadament 14,625 dies d'execució contínua. Si es dediquen dos dies per cada paràmetre, el temps total estimat serà:

$$\text{Temps total estimat} = 13 \text{ paràmetres} \times 2 \text{ dies/paràmetre} = 26 \text{ dies}$$

EX - Experimentació

En aquesta secció es detallen els experiments que es realitzaran per avaluar el rendiment dels algorismes DDQN i IAA en diferents configuracions. S'espera poder estudiar, com a minim, les diferens tècniques en dos nivells diferens 1-1 i 4-1. Fent 3000 iteracions per execució i realitzant un total de 10 execucions amb llavors diferents.

Línies Base del DDQN

- **RGB DDQN Baseline**: Avaluar el rendiment de l'algorisme DDQN utilitzant dades d'entrada RGB sense segmentació semàntica.
- **SS DDQN Baseline**: Avaluar el rendiment de l'algorisme DDQN utilitzant segmentació semàntica per a les dades d'entrada.

Millors Configuracions del DDQN

- **RGB DDQN BestConfig**: Experimentar amb la millor configuració trobada per al DDQN utilitzant dades d'entrada RGB.
- **SS DDQN BestConfig**: Experimentar amb la millor configuració trobada per al DDQN utilitzant segmentació semàntica per a les dades d'entrada.

Experiments amb IAA i Fine-Tuning

- **RGB IAA**: Avaluar el rendiment de la tècnica *Introspective Action Advising (IAA)* utilitzant dades d'entrada RGB.
- **RGB FT**: Avaluar el rendiment del *fine-tuning* utilitzant dades d'entrada RGB com a línia base per a la comparació amb IAA.

Aquestes tasques ens permetran obtenir una comprensió detallada del rendiment dels nostres algorismes en diferents escenaris, assegurant que podem identificar les configuracions òptimes per a la seva aplicació pràctica.

El temps requerit per a l'experimentació serà un total de:

$$\begin{aligned}\text{Temps total d'experimentació} = 2 \text{ nivells} \times 10 \text{ execucions} \times \\ 3 \text{ hores/execució} \times 6 \text{ configuracions} = 360 \text{ hores}\end{aligned}$$

Suposant que les execucions es poden realitzar de forma ininterrompuda, això equivaldria a aproximadament 15 dies d'execució contínua. Pel mateix motiu que en el refinament, deixarem 5 dies de marge per si succeeix qualsevol complicació, de manera que el temps total estimat serà:

$$\text{Temps total estimat} = 15 \text{ dies} + 5 \text{ dies de marge} = 20 \text{ dies}$$

AR - Anàlisis de Resultats

En aquesta última secció es durar a terme un anàlisis detallat dels resultats obtinguts en la comparació dels mètodes més comuntment utilitzats per desenvolupar la tasca que estem estudiant, amb la tècnica estudiada.

Per evaluar l'eficàcia de les diferents tècniques, utilitzarem una mètrica fonamental:

- Recompensa acumulada, que mesura l'èxit acumulat durant tot el període d'entrenament, reflectint la capacitat de la tècnica per maximitzar els beneficis al llarg del temps. Segons el paper Taylor, Matthew E., i Peter Stone. "Transfer Learning for Reinforcement Learning Domains: A Survey." la recompensa acumulada es una de les mesures fonamentals sobre les que es poden compara tècniques de Transfer Learning.

Per evaluar l'eficiencia de les diferents tècniques, utilitzarem les mètriques de:

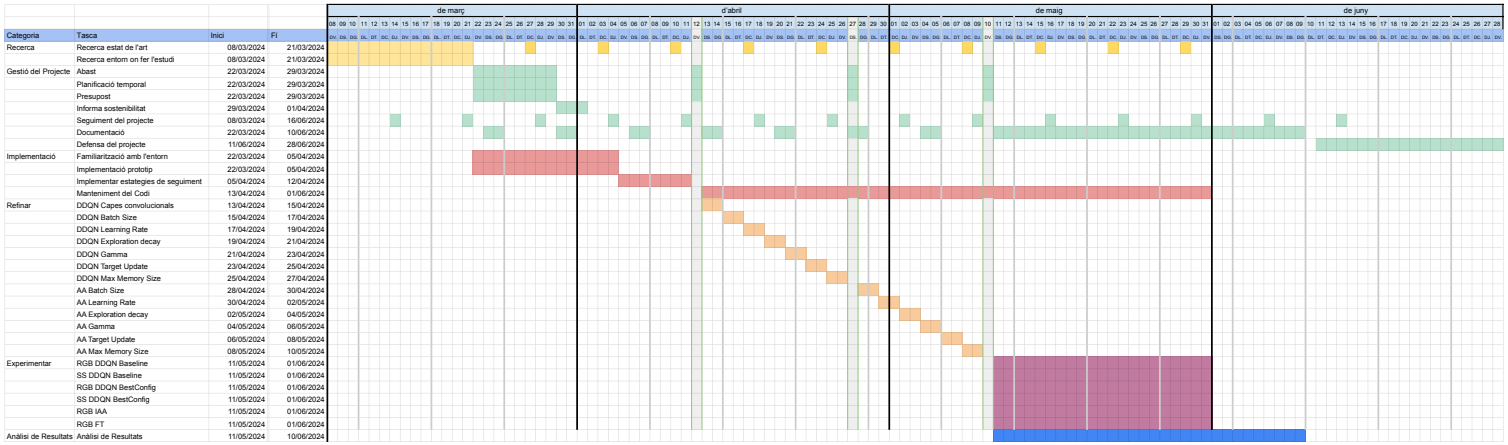
- Temps total d'entrenament, paràmetre que resulta crític per nosaltres, ja que tenim un temps d'experimentació limitat.

Finalment, per oferir una visió més completa de l'eficiència, considerarem la recompensa acumulada per segon, una mètrica que combina les dues mètriques anteriors per determinar quina tècnica

optimitza millor la relació entre benefici i temps invertit. A demès, es menester veure com evoluciona la gràfica de les recompenses de cada episodi al llarg del temps. Encara que no sigui molt explicativa, es d'on parteix la construcció de la mètrica amb la que analitzem la eficàcia.

Id Tasca	Tasca	Temps	Recursos	Dependencies
EP1	Estudi de la tècnica a estudiar	30 hores	PC, Overleaf, GitHub, Google Scholar	-
EP2	Estudi de l'entorn on aplicar la tècnica	10 hores	PC, Overleaf, GitHub, Google Scholar	-
GP1	Definició de l'Abast	5 hores	PC, Overleaf, Documentació GEP	-
GP2	Planificació Temporal	5 hores	PC, Overleaf, Documentació GEP	GP1
GP3	Pressupost	5 hores	PC, Overleaf, Documentació GEP	GP1
GP4	Informe de Sostenibilitat	3 hores	PC, Overleaf, Documentació GEP	GP1
GP5	Seguiment del Projecte	14 hores	PC, Overleaf, Documentació GEP	GP1,2,3,4
GP6	Documentació	60 hores	PC, Overleaf, Documentació GEP	GP1,2,3,4,5
GP6	Defensa del Projecte	20 hores	PC, Overleaf	GP6
I0	Familiarització	10 hores	PC, GitHub, Visual Studio Code	-
I1	Implementació d'un prototip de la tècnica	40 hores	PC, GitHub, Visual Studio Code, Apunts SID	I0
I2	Implementar estratègies de seguiment	5 hores	PC, GitHub, Visual Studio Code, WandB	I1
M	Manteniment del codi	20 hores	PC, GitHub, Visual Studio Code	I1, I2
RP1	DDQN Capes Convolucionals	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP2	DDQN Batch Size	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP3	DDQN Learning Rate	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP4	DDQN Exploration Decay	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP5	DDQN Gamma	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP6	DDQN Target Update	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP7	DDQN Max Memory Size	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP8	IAA Batch Size	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP9	IAA Learning Rate	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP10	IAA Exploration Decay	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP11	IAA Gamma	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP12	IAA Target Update	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
RP13	IAA Max Memory Size	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	I1, I2
EX1	RGB DDQN Baseline	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	RP1,2,3,4,5,6,7
EX2	SS DDQN Baseline	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	RP1,2,3,4,5,6,7
EX3	RGB DDQN BestConfig	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	RP1,2,3,4,5,6,7
EX4	SS DDQN BestConfig	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	RP1,2,3,4,5,6,7
EX5	RGB IAA	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	RP8,9,10,11,12,13
EX6	RGB FT	27 hores	PC, GitHub, Visual Studio Code, WandB, GPU NVIDIA	RP8,9,10,11,12,13
AR	Anàlisi de Resultats	60 hores	PC, Overleaf, Visual Studio Code, WandB	I0,1,2

Taula A.1: Descripció de Tasques



Categoria	Tasca	Inici	Fi
Recerca	Recerca estat de l'art	08/03/2024	21/03/2024
	Recerca entorn on fer l'estudi	08/03/2024	21/03/2024
Gestió del Projecte	Abast	22/03/2024	29/03/2024
	Planificació temporal	22/03/2024	29/03/2024
	Presupost	22/03/2024	29/03/2024
	Informa sostenibilitat	29/03/2024	01/04/2024
	Seguiment del projecte	08/03/2024	16/06/2024
	Documentació	22/03/2024	10/06/2024
	Defensa del projecte	11/06/2024	28/06/2024
Implementació	Familiarització amb l'entorn	22/03/2024	05/04/2024
	Implementació prototip	22/03/2024	05/04/2024
	Implementar estategies de seguiment	05/04/2024	12/04/2024
	Manteniment del Codi	13/04/2024	01/06/2024
Refinar	DDQN Capes convolucionals	13/04/2024	15/04/2024
	DDQN Batch Size	15/04/2024	17/04/2024
	DDQN Learning Rate	17/04/2024	19/04/2024
	DDQN Exploration decay	19/04/2024	21/04/2024
	DDQN Gamma	21/04/2024	23/04/2024
	DDQN Target Update	23/04/2024	25/04/2024
	DDQN Max Memory Size	25/04/2024	27/04/2024
	AA Batch Size	28/04/2024	30/04/2024
	AA Learning Rate	30/04/2024	02/05/2024
	AA Exploration decay	02/05/2024	04/05/2024
	AA Gamma	04/05/2024	06/05/2024
	AA Target Update	06/05/2024	08/05/2024
	AA Max Memory Size	08/05/2024	10/05/2024
Experimentar	RGB DDQN Baseline	11/05/2024	01/06/2024
	SS DDQN Baseline	11/05/2024	01/06/2024
	RGB DDQN BestConfig	11/05/2024	01/06/2024
	SS DDQN BestConfig	11/05/2024	01/06/2024
	RGB IAA	11/05/2024	01/06/2024
	RGB FT	11/05/2024	01/06/2024
Anàlisi de Resultats	Anàlisi de Resultats	11/05/2024	10/06/2024

Figura A.1: Diagrama de Gantt GEP

A.2 Actualització de la Planificació

La planificació original no preveia una dedicació suficient de temps per a la implementació de la part pràctica del projecte. Com a resultat, hem incrementat el nombre d'hores assignades a aquesta fase, redistribuint el temps de tasques menys crítiques.

Per evitar experiments decisius en les setmanes finals del projecte, hem optat per sacrificar part del temps destinat al refinament dels paràmetres de l'agent Teacher. La Figura A.2 de l'Annex presenta la planificació inicial, destacant la distribució del temps entre les diferents fases.

Fase	**Descripció**	**Hores Totals**
Implementació	Inclou I0, I1, I2 i M	75 hores
Refinament DQN	Inclou RP1 a RP7	189 hores
Refinament IAA	Inclou RP8 a RP13	162 hores

Taula A.2: Resum de la Taula descripció de Tasques TFG

Amb la nova planificació, hem fet els següents canvis:

- Hem afegit noves tasques d'implementació per assegurar una millor cobertura del desenvolupament del projecte. Aquestes tasques inclouen la Familiarització amb l'entorn, la Implementació dels Prototips v0, v1 i v2, el Disseny de classes, l'Implementació d'estratègies de seguiment i l'Ajust de recollida de mètriques.
- Hem revisat i actualitzat les tasques de refinament per al DQN i AA per reflectir millor les necessitats del projecte.
- Hem ajustat les hores assignades per a cadascuna d'aquestes tasques per assegurar que el projecte tingui el temps adequat per completar cada fase de manera efectiva.

En la nova planificació es poden veure els següents canvis:

Fase	**Descripció**	**Hores Totals**
Implementació	Inclou I0 a I7	140 hores
Refinament DQN	Inclou RP1 a RP7	60.5 hores
Refinament AA	Inclou RP8	40 hores
Baselines	Inclou RP9, RP10	61.67 hores
Experimentació	Inclou RP11 a RP16	45 hores
Anàlisi de Resultats	Inclou AR	60 hores

Taula A.3: Resum de la Taula descripció de Tasques TFG

A.2.1 Justificació de la planificació d'hores

A continuació es justifica el càlcul de la planificació d'hores per als diferents experiments i baselines realitzats. Per a cada paràmetre estudiat, es va provar una mitjana de 3 execucions amb cadascun dels valors. Les execucions tenien una durada mitjana de 5 hores.

Refinament DQN:

- **Capes Convolucionals:** Valors provats: 2, 3, 4
- **Batch Size:** Valors provats: 8, 16, 32, 64
- **Learning Rate:** Valors provats: 0.0001, 0.00025, 0.0005
- **Exploration Decay:** Valors provats: 0.99, 0.995, 0.999
- **Gamma:** Valors provats: 0.85, 0.9, 0.95
- **Target Update:** Valors provats: 500, 1000, 1500
- **Max Memory Size:** Valors provats: 2000, 4000, 6000

Total hores per Refinament DQN:

$$(3 \text{ valors} \times 3 \text{ execucions} \times 5 \text{ hores} \times 6 \text{ paràmetres})$$
$$+(4 \text{ valors} \times 3 \text{ execucions} \times 5 \text{ hores})$$
$$= 270 \text{ hores} + 60 \text{ hores} = \mathbf{330 \ \ hores}$$

Refinament AA:

- **Learning Rate 0.0001:** 7 hores * 4 execucions = 28 hores
- **Learning Rate 0.0004:** 7 hores * 4 execucions = 28 hores

Total hores per Refinament AA: 28 hores + 28 hores = **56 hores**

Baselines:

- **SS:** 10 hores * 10 execucions * 2 nivells = 200 hores
- **RGB:** 5 hores * 10 execucions * 2 nivells = 100 hores
- **FT:** 5 hores * 8 execucions * 1 nivell = 40 hores

Total hores per Baselines: 200 hores (SS) + 100 hores (RGB) + 40 hores (FT) = **340 hores**

Experimentació:

- **SUA:** 7 hores * 6 execucions * 2 deltas * 1 nivell = 84 hores
- **TUA:** 7 hores * 6 execucions * 2 deltas * 1 nivell = 84 hores
- **TSUA:** 7 hores * 6 execucions * 2 deltas * 1 nivell = 84 hores

Total hores per Experimentació: 84 hores (SUA) + 84 hores (TUA) + 84 hores (TSUA) = **252 hores**

Total hores planificades: 340 hores + 252 hores+ 56 hores + 330 hores = **978 hores**

Considerant que podem executar 6 configuracions alhora, el temps total requerit es redueix a:

Temps total per Refinament DQN:

$$\frac{330 \text{ hores}}{6} = 55 \text{ hores}$$

Temps total per Refinament AA:

$$\frac{56 \text{ hores}}{6} = 9.33 \text{ hores}$$

Temps total per Baselines:

$$\frac{340 \text{ hores}}{6} = 56.67 \text{ hores}$$

Temps total per Experimentació:

$$\frac{252 \text{ hores}}{6} = 42 \text{ hores}$$

Temps total: 56.67 hores (baselines) + 42 hores (experimentació) + 9.33 hores (Refinament AA) + 55 hores (Refinament DQN) = **163 hores**

Per obtenir una estimació més precisa del temps real de treball, utilitzarem un càlcul basat en una dedicació de 5 minuts per execució. Aquest increment té en compte el temps addicional necessari per gestionar els experiments, assegurar-se que s'estan executant amb els valors predeterminats, atendre possibles interrupcions i repetir execucions si cal. Aquestes hores addicionals es comptabilitzaran en el còmput total d'hores de l'investigador.

El càlcul de les hores dedicades de l'investigador basat en 5 minuts per execució és el següent:

Refinament DQN: Execucions per als 6 paràmetres = 6 paràmetres ×3 valors ×3 execucions = 54 execucions

Execucions per al paràmetre = 1 paràmetre ×4 valors × 3 execucions = 12 execucions

Número total d'execucions = 54 execucions + 12 execucions = 66 execucions

66 execucions $\times \frac{5 \text{ minuts}}{60} = 5.5$ hores

Refinament AA:

$$\text{Número total d'execucions} = 8 \text{ execucions}$$

$$8 \text{ execucions} \times \frac{5 \text{ minuts}}{60} = 0.67 \text{ hores}$$

Baselines:

$$\text{Número total d'execucions} = (10 \text{ execucions} \times 2 \text{ nivells} \times 3 \text{ baselines}) = 60 \text{ execucions}$$

$$60 \text{ execucions} \times \frac{5 \text{ minuts}}{60} = 5 \text{ hores}$$

Experimentació:

$$\text{Número total d'execucions} = (6 \text{ execucions} \times 2 \text{ deltas} \times 3 \text{ experiments}) = 36 \text{ execucions}$$

$$36 \text{ execucions} \times \frac{5 \text{ minuts}}{60} = 3 \text{ hores}$$

Per tant, el temps total dedicat de l'investigador per baselines, experimentació, refinament AA i refinament DQN serà:

5 hores (baselines) + 3 hores (experimentació) + 0.67 hores (refinament AA) + 5.5 hores (refinament DQN) = **14.17 hores**

Així, el temps total dedicat a cada tasca és:

Refinament DQN:

$$55 \text{ hores (comput)} + 5.5 \text{ hores (investigador)} = 60.5 \text{ hores}$$

Refinament AA:

$$9.33 \text{ hores (comput)} + 0.67 \text{ hores (investigador)} = 10 \text{ hores}$$

Baselines:

$$56.67 \text{ hores (comput)} + 5 \text{ hores (investigador)} = 61.67 \text{ hores}$$

Experimentació:

$$42 \text{ hores (comput)} + 3 \text{ hores (investigador)} = 45 \text{ hores}$$

El total combinat de temps dedicat a totes les tasques és:

$$163 \text{ hores (comput)} + 14.17 \text{ hores (investigador)} = \mathbf{177.17 \ \ hores}$$

Aquestes hores addicionals es comptabilitzaran en el còmput total d'hores de l'investigador. L'experimentació ha resultat ser més farragosa degut a l'ús de Google Colab per a llançar els experiments, ja que és una eina poc professional i orientada a usuaris no especialitzats. En no estar dissenyada per a execucions prolongades, s'ha hagut d'invertir temps addicional per assegurar el correcte funcionament dels experiments. De cara al futur, es buscaran serveis més adequats per a aquest tipus de tasques i es realitzarà un estudi de mercat per identificar les millors opcions disponibles.

Categoria	Tasca	Inici	Durac	Fi
Recerca	Recerca estat de l'art	08/03/2024	13	21/03/2024
	Recerca entorn on fer l'estudi	08/03/2024	13	21/03/2024
Gestió del Projecte	Abast	22/03/2024	7	29/03/2024
	Planificació temporal	22/03/2024	7	29/03/2024
	Presupost	22/03/2024	7	29/03/2024
	Informa sostenibilitat	29/03/2024	3	01/04/2024
	Seguiment del projecte	08/03/2024	100	16/06/2024
	Documentació	22/03/2024	80	10/06/2024
	Defensa del projecte	11/06/2024	17	28/06/2024
Implementació	Familiarització amb l'entorn	22/03/2024	14	05/04/2024
	Implementació Prototip v0	22/03/2024	14	05/04/2024
	Implementar estategies de seguiment	05/04/2024	7	12/04/2024
	Disenv de classes	12/04/2024	3	15/04/2024
	Implementació Prototip v1	15/04/2024	10	25/04/2024
	Manteniment del Codi	26/04/2024	14	10/05/2024
	Implementació Prototip v2	10/05/2024	5	15/05/2024
	Ajustar recollida de mètriques	15/05/2024	2	17/05/2024
Refinar	DQN Capes convolucionals	26/04/2024	2	28/04/2024
	DQN Batch Size	28/04/2024	2	30/04/2024
	DQN Learning Rate	30/04/2024	2	02/05/2024
	DQN Exploration decay	02/05/2024	2	04/05/2024
	DQN Gamma	04/05/2024	2	06/05/2024
	DQN Target Update	06/05/2024	2	08/05/2024
	DQN Max Memory Size	08/05/2024	2	10/05/2024
	AA Learning Rate	16/05/2024	2	18/05/2024
Experimentar	RGB DQN Baseline	18/05/2024	14	01/06/2024
	SS DQN Baseline	18/05/2024	14	01/06/2024
	RGB DQN BestConfig	18/05/2024	14	01/06/2024
	SS DQN BestConfig	18/05/2024	14	01/06/2024
	RGB EA	18/05/2024	14	01/06/2024
	RGB SUA	18/05/2024	14	01/06/2024
	RGB TUA	18/05/2024	14	01/06/2024
	RGB TSUA	18/05/2024	14	01/06/2024
Análisi de Resultats	Análisi de Resultats	18/05/2024	23	10/06/2024

Figura A.2: Diagrama de Gantt TFG

Apèndix B

Gestió Económica

B.1 Gestió econòmica

B.1.1 Cost de personal

Costos de Personal per Activitat

En aquesta secció calcularem el cost total de cada tasca definida en la secció anterior i detallat en la Taula B.1. El cost de les tasques es calcularà en funció del salari que estimem que es pagaria a la persona que les realitza.

En aquest treball suposarem que hi ha tres tipus de rols professionals diferents:

- **Desenvolupador**: El desenvolupador és responsable d'adaptar el codi desenvolupat per Javier Montalvo a les necessitats específiques de la investigació. Aquesta tasca inclou la implementació de noves funcionalitats, la resolució de problemes tècnics i l'optimització del codi per assegurar que funciona correctament en l'entorn d'experimentació.
- **Investigador**: L'investigador és l'encarregat de desenvolupar la investigació, dissenyar els experiments, recollir i analitzar les dades resultants. Aquesta tasca inclou la formulació d'hipòtesis, la realització d'experiments controlats i l'anàlisi dels resultats per extreure conclusions significatives.
- **Project Manager**: El Project Manager s'encarrega de supervisar el desenvolupament del projecte en les diferents etapes. Això inclou la planificació del projecte, la coordinació de les tasques, la gestió dels recursos i el seguiment del progrés per assegurar que es compleixen els objectius i els terminis establerts.

Rol	Salari Anual (€)	Preu per hora (€)	Dut a terme per
Project Manager	40.000	22,86	TGEP, D, A
Desenvolupador	25.000	14,29	A
Investigador	30.000	17,14	A

Taula B.1: Salari anual dels diferents rols del projecte, considerant 1750 hores treballades l'any

B.1.2 Costos Tècnics

Cost de Desplaçament

- **Lloguer d'una bicicleta elèctrica al servei municipal**: un viatge al mes
- **Nombre total de viatges**: 4 viatges durant el projecte, 8 desplaçaments

- **Cost del transport**: 0,35€ per desplaçament + 50€ de tarifa anual dividida pels 12 mesos de l'any i multiplicada pels 4 mesos del projecte

$$\text{Cost desp.} = 8 \text{ desp.} \times 0,35€/\text{desp.} + \left(\frac{50€ \text{ de tarifa}}{12 \text{ mesos}} \times 4 \text{ mesos} \right) = 46,66$$

Espai de Treball

- **Lloguer de l'espai de treball**: Escriptori a un coworking, 200€/mes. Dins de la tarifa sens ofereix accés a aigua, electricitat i internet il·limitat
- **Cost total de l'espai per 4 mesos**:

$$\text{Cost de l'espai} = 200€/\text{mes} \times 4 \text{ mesos} = 1.000$$

Costos de Computació

En aquesta subsecció es detallen els costos associats a la computació necessària per al refinament i l'experimentació del projecte, utilitzant una targeta gràfica NVIDIA T4 llogada a Google.

- **Cost per hora de lloguer d'una NVIDIA T4**: 1,75 crèdits.
- **Cost de 500 crèdits**: 51,95€.
- **Cost per crèdit**:

$$\text{Cost per crèdit} = \frac{51,95}{500 \text{ crèdits}} = 0,1039 \text{ per crèdit}$$

- **Hores de refinament estimades**: 351 hores.
- **Hores d'experimentació estimades**: 360 hores.
- **Hores totals estimades**:

$$\text{Hores totals} = 351 \text{ hores} + 360 \text{ hores} = 711 \text{ hores}$$

- **Crèdits totals estimats**:

$$\text{Crèdits totals} = 711 \text{ hores} \times 1,75 \text{ crèdits/hora} = 1244,25 \text{ crèdits}$$

- **Cost total de computació**:

$$\text{Cost total} = 1244,25 \text{ crèdits} \times 0,1039 \text{ €/crèdit} = 129,31$$

Per tant, el cost total associat a la computació necessària per al refinament i l'experimentació del projecte, utilitzant una targeta gràfica NVIDIA T4 llogada a Google, s'estima que rondaria els serà de 129,31€.

Categoria	**Detall**	**Cost (€)**
Desplaçament	Lloguer bicicleta elèctrica + tarifa anual	46,66
Espai de Treball	Lloguer escriptori a coworking	1.000,00
Costos de Computació	Lloguer NVIDIA T4 per 711 hores	129,31

Taula B.2: Resum de costos tècnics

No considerem el cost d'amortització, ja que la màquina que utilitzem ha superat els quatre anys de vida útil estimada per a la seva amortització. Per tant, el seu valor comptable ja s'ha amortitzat completament i no representa un cost addicional en el pressupost actual del projecte.

Id Tasca	Tasca	Cost (€)	Càlcul
EP1	Estudi de la tècnica a estudiar	514,20	30 h ×17,14 €/h (Investigador)
EP2	Estudi de l'entorn on aplicar la tècnica	171,40	10 h ×17,14 €/h (Investigador)
GP1	Definició de l'Abast	114,30	5 h ×22,86 €/h (Project Manager)
GP2	Planificació Temporal	114,30	5 h ×22,86 €/h (Project Manager)
GP3	Pressupost	114,30	5 h ×22,86 €/h (Project Manager)
GP4	Informe de Sostenibilitat	68,58	3 h ×22,86 €/h (Project Manager)
GP5	Seguiment del Projecte	366,70	14 h ×22,86 €/h (Project Manager) + 46,66 € (Transport)
GP6	Documentació	1371,60	60 h ×22,86 €/h (Project Manager)
GP6	Defensa del Projecte	457,20	20 h ×22,86 €/h (Project Manager)
I0	Familiarització	142,90	10 h ×14,29 €/h (Desenvolupador)
I1	Implementació d'un prototip de la tècnica	571,60	40 h ×14,29 €/h (Desenvolupador)
I2	Implementar estratègies de seguiment	71,45	5 h ×14,29 €/h (Desenvolupador)
M	Manteniment del codi	285,80	20 h ×14,29 €/h (Desenvolupador)
RP1	DDQN Capes Convolucionals	4,86	27 h ×0,18 €/h (GPU)
RP2	DDQN Batch Size	4,86	27 h ×0,18 €/h (GPU)
RP3	DDQN Learning Rate	4,86	27 h ×0,18 €/h (GPU)
RP4	DDQN Exploration Decay	4,86	27 h ×0,18 €/h (GPU)
RP5	DDQN Gamma	4,86	27 h ×0,18 €/h (GPU)
RP6	DDQN Target Update	4,86	27 h ×0,18 €/h (GPU)
RP7	DDQN Max Memory Size	4,86	27 h ×0,18 €/h (GPU)
RP8	IAA Batch Size	4,86	27 h ×0,18 €/h (GPU)
RP9	IAA Learning Rate	4,86	27 h ×0,18 €/h (GPU)
RP10	IAA Exploration Decay	4,86	27 h ×0,18 €/h (GPU)
RP11	IAA Gamma	4,86	27 h ×0,18 €/h (GPU)
RP12	IAA Target Update	4,86	27 h ×0,18 €/h (GPU)
RP13	IAA Max Memory Size	4,86	27 h ×0,18 €/h (GPU)
EX1	RGB DDQN Baseline	4,86	27 h ×0,18 €/h (GPU)
EX2	SS DDQN Baseline	4,86	27 h ×0,18 €/h (GPU)
EX3	RGB DDQN BestConfig	4,86	27 h ×0,18 €/h (GPU)
EX4	SS DDQN BestConfig	4,86	27 h ×0,18 €/h (GPU)
EX5	RGB IAA	4,86	27 h ×0,18 €/h (GPU)
EX6	RGB FT	4,86	27 h ×0,18 €/h (GPU)
AR	Anàlisi de Resultats	1028,40	60 h ×17,14 €/h (Investigador)
	Total	8078,52	

Taula B.3: Descripció de Tasques amb els seus costos

B.1.3 Contingencies

Les contingències inclouen qualsevol problema, imprevist o contratemps que pugui aparèixer mentre es duu a terme el projecte.

El cost mitjà per hora dels dos professionals és:

$$\text{Cost mitjà per hora} = \frac{17,14 + 14,29}{2} = 15,715\,€/\text{h}$$

Per calcular el cost de cada risc, hem utilitzat la següent fórmula:

$$\text{Cost} = \text{Temps estimat (h)} \times \text{Cost per hora} \times \text{Probabilitat}$$

En aquest càlcul per a la control i prevenció de riscos en els resultats, hem sumat el cost de l'investigador i el cost de la GPU:

$$\text{Cost (Control i Prevenció de Riscos en els Resultats)} = 70 \times 0,18 \times \frac{50}{100} = 6,30\,€$$

Risc	**Nivell**	**Prob.(%)**	**Temps est.(h)**	**Cost est.(€)**
Deadline	Alt	70	40	439,78
Bugs	Baix	5	25	19,64
Codi no propi	Alt	80	50	628,60
Dimensió del projecte	Mitjà	40	30	188,58
Experimentació	Baix	15	15	35,96
Integració noves tècniques	Mitjà	40	35	220,01
Resultats	Mitjà	50	70	6,30
Total				**1538,87**

Taula B.4: Temps estimat i cost per a la gestió de riscos

B.1.4 Mecanismes de Control de Gestió del Pressupost

Per assegurar una gestió adequada del pressupost del projecte, s'implementaran els següents mecanismes de control:

- **Revisions Periòdiques**: Es realitzaran revisions periòdiques del progrés del projecte per comparar els costos i temps reals amb els planificats. Aquestes revisions es faran mensualment per garantir un seguiment proper.
- **Observació de Desviacions**: En cada revisió, es calcularan les desviacions tant temporals com econòmiques mitjançant la comparació dels temps reals invertits i els costos reals amb els previstos. Això permetrà identificar qualsevol discrepància i ajustar els plans segons sigui necessari.
- **Càlcul de Diferències**: Les desviacions es calcularan com la diferència entre el valor real i el valor planificat per cada tasca. Per les desviacions temporals, es calcularà la diferència en dies o hores; per les desviacions econòmiques, es calcularà la diferència en euros.
- **Ajustaments del Pressupost i Calendari**: Si es detecten desviacions significatives, es realitzaran ajustaments en el pressupost i el calendari per reflectir les noves estimacions. Aquest procés permetrà mantenir el projecte en camí i dins del pressupost.

Aquest enfocament sistemàtic permetrà un control rigorós del pressupost i el temps, assegurant que el projecte es mantingui en els límits previstos i es pugui reaccionar ràpidament a qualsevol desviació que pugui sorgir.

B.2 Actualització Gestió Económica

Les hores de desenvolupament han sigut augmentades, passant de 75 hores a 140 hores. Això implica que, costant el treball per hora del desenvolupador 14,29 €, el cost total del desenvolupament ha augmentat a:

40 hores (I1) + 5 hores (I2) + 15 hores (I3) + 40 hores (I4) + 20 hores (I5) + 10 hores (I6) + 10 hores (I7) = 140 hores 140 hores $\times 14,29$ €/hora = 2.000,60 €

Les hores de comput respecte a la planificació inicial han augmentat de 711 hores a 978 hores, a més de les 14.17 hores que l'investigador ha dedicat. Per tant, sent el cost de comput 14,29€ €/hora, el cost total és de:

$$978 \text{ hores} \times 0,18 \text{ €/hora} = 176,04 \text{ €}$$

El cost total de l'investigador és:

$$14,17 \text{ hores} \times 0,18 \text{ €/hora} = 2,55 \text{ €}$$

Per tant, el cost total combinat de comput és:

$$176,04 \text{ €} + 2,55 \text{ €} = 178,59 \text{ €}$$

Sent el cost total previst en la etapa de implementació de 1071,75€ i en computació de 129,31€, el cost en aquestes secció ha augmentat en 928,85€ i 49,28€ respectivament. Les contingències previstes contaven amb un sobrecost de 1538,87€, com podem veure a la següent taula:

Risc	**Nivell**	**Prob.(%)**	**Temps est.(h)**	**Cost est.(€)**
Deadline	Alt	70	40	439,78
Bugs	Baix	5	25	19,64
Codi no propi	Alt	80	50	628,60
Dimensió del projecte	Mitjà	40	30	188,58
Experimentació	Baix	15	15	35,96
Integració noves tècniques	Mitjà	40	35	220,01
Resultats	Mitjà	50	70	6,30
Total				**1538,87**

Taula B.5: Temps estimat i cost per a la gestió de riscos

Amb l'augment de les hores de desenvolupament de 75 a 140 hores, el cost total del desenvolupament ha incrementat significativament fins a 2.000,60 €. A més, les hores de computació han augmentat de 711 a 978 hores, sumant un cost de 176,04 €, més les hores dedicades per l'investigador, resultant en un cost combinat de comput de 378,60 €. Comparant amb els costos previstos inicialment, hem observat un augment de 928,85 € i 249,29 € respectivament. Tot i així, les contingències previstes de 1538,87 € han estat suficients per cobrir aquests increments de cost.

Apèndix C

Informació adicional

C.1 Funció de Recompensa

La funció de recompensa assumeix que l'objectiu del joc és avançar tant com sigui possible cap a la dreta (augmentant el valor de x de l'agent), tan ràpid com sigui possible, sense morir. Per modelar aquest joc, la recompensa es compon de tres variables separades:

- **v**: la diferència en els valors de x de l'agent entre estats
  - en aquest cas, és la velocitat instantània per al pas donat
  - v = x1 - x0
    - x0 és la posició x abans del pas
    - x1 és la posició x després del pas
  - moure's a la dreta si només si $v > 0$
  - moure's a l'esquerra si només si $v < 0$
  - no moure's si només si $v = 0$
- **c**: la diferència en el rellotge del joc entre fotogrames
  - la penalització evita que l'agent es quedi quiet
  - c = c0 - c1
    - c0 és la lectura del rellotge abans del pas
    - c1 és la lectura del rellotge després del pas
  - no hi ha tic del rellotge si només si $c = 0$
  - tic del rellotge si només si $c < 0$
- **d**: una penalització per mort que penalitza l'agent per morir en un estat
  - aquesta penalització encoratja l'agent a evitar morir
  - viu si només si $d = 0$
  - mort si només si d = -15

La recompensa total es calcula com:

$$r = v + c + d$$

La recompensa es limita al rang (-15, 15). Aquesta limitació assegura que les penalitzacions severes no afectin desproporcionadament l'aprenentatge de l'agent, permetent un entrenament més estable i efectiu.

C.2 Gràfic de la Incertesa de l'agent `teacher` amb l'*outlier*

En la presentació dels resultats de la incertesa del professor, es va observar que una de les execucions era un *outlier* que distorsionava significativament la interpretació gràfica. Aquesta execució presentava una incertesa de 30, mentre que la resta de les execucions mostraven valors d'incertesa al voltant de 5. La presència d'aquest *outlier* dificultava l'anàlisi precisa dels resultats, ja que els valors extrems impactaven negativament en la visualització global de les dades.

Per il·lustrar aquesta situació, es presenta a continuació la gràfica original amb l'*outlier* inclòs. Com es pot observar, l'execució anòmala espatlla la representació gràfica, fent difícil la comparació i l'avaluació dels resultats de la resta de les execucions.

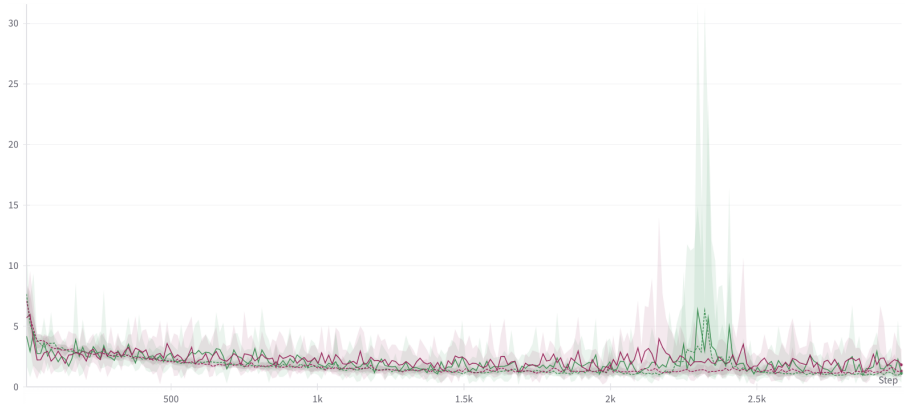


Figura C.1: Gràfica de la incertesa del professor amb l'*outlier*.

Degut a aquesta distorsió, es va decidir eliminar l'execució *outlier* de l'anàlisi final per obtenir una representació més clara i precisa de les dades. Aquesta acció va permetre visualitzar de manera més clara les tendències generals de la incertesa del professor, facilitant una millor comprensió dels resultats.