الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر
كلية الرياضيات و الإعلام الآلي و الاتصالات السلكية و اللاسلكية

قسم: الإعلام الآلي

Mémoire de Master en informatique

Spécialité : Intelligence artificielle : principes et applications

T h è m e

Développement d'un Régulateur de Paramètres Avancé basé sur l'Ancrage et une Stratégie Adaptative par Époque (AnchorLoss) pour l'Amélioration de la Robustesse des Modèles Profonds

Présenté par :
Benadla Redouane.

Dirigé par :
Dr : Adgir Noureddine.

Année universitaire 2025-2026

Dédicaces

Je dédie ce modeste travail :

À la mémoire de mon cher père et de ma chère mère, que Dieu ait leurs âmes en Sa sainte miséricorde. Vous avez toujours cru en moi et m'avez inculqué les valeurs qui me guident aujourd'hui. J'aurais tant aimé que vous soyez présents en ce jour pour partager avec moi cette réussite. Que ce travail soit une prière pour le repos de vos âmes.

À mon épouse, pour sa patience, son soutien indéfectible et ses encouragements tout au long de ce parcours.

À mes chers enfants, ma fille et mes deux fils, la lumière de ma vie et ma plus grande source de motivation. Puisse ce travail être pour vous un exemple de persévérance.

À toute ma famille, pour leur affection et leurs encouragements.

À tous mes amis et collègues, pour les bons moments partagés et leur soutien constant.

Remerciements

Avant tout, nous remercions Dieu Tout-Puissant de nous avoir donné la force, la patience et la volonté pour mener à bien ce modeste travail.

Nous tenons à exprimer notre profonde gratitude à notre encadrant, **Monsieur Adgir NourEddine**, pour sa disponibilité, ses précieux conseils, sa patience et son suivi rigoureux tout au long de la réalisation de ce mémoire.

Nos sincères remerciements s'adressent également aux membres du jury pour l'honneur qu'ils nous font en acceptant d'évaluer ce travail.

Nous remercions aussi l'ensemble des enseignants du département d'Informatique de l'Université de Saïda Dr. Moulay Tahar pour la qualité de la formation qu'ils nous ont dispensée durant notre parcours universitaire.

Nos remerciements vont également à toute personne ayant contribué, de près ou de loin, à la réalisation de ce travail, par ses conseils, son aide ou ses encouragements.

Enfin, nous remercions nos familles et nos amis pour leur soutien moral constant tout au long de ce parcours.

Résumé

Ce travail traite de la problématique de l'Oubli Catastrophique (Catastrophic Forgetting) dans les systèmes d'apprentissage Continu (Continual Learning) — ce phénomène qui empêche les réseaux de neurones d'accumuler des connaissances au fil du temps sans perdre ce qu'ils avaient appris auparavant.

Dans ce mémoire, nous proposons un algorithme adaptatif hybride connu sous le nom de GLSA-H (Global-Local Structural Plasticity-Stability Alignment — Hybrid), qui démontre une supériorité par rapport aux approches traditionnelles telles qu'EWC. GLSA-H constitue l'aboutissement d'un parcours de développement progressif : il étend une première formulation, appelée GLSA-S, fondée sur un facteur de conflit cosinus entre gradients, en y intégrant une estimation diagonale de l'information de Fisher, un lissage par momentum, une mise à l'échelle de type Newton et un mécanisme optionnel d'adaptation du taux d'apprentissage par hypergradient.

L'algorithme proposé repose ainsi sur un mécanisme dynamique d'équilibre entre Stabilité et Plasticité (Stability-Plasticity Balance), combinant le Facteur de Conflit Cosinus (Cosine Conflict Factor) et l'importance relative des paramètres (Fisher diagonale) pour réguler les mises à jour des poids en temps réel, sans recourir à un stockage d'échantillons des tâches passées (approche rehearsal-free / buffer-free).

L'efficacité du noyau algorithmique de GLSA-H a été évaluée à travers deux scénarios expérimentaux :

- **Apprentissage multi-domaines (Cross-Domain) :** en combinant des tâches radicalement différentes comme la classification médicale de tumeurs cérébrales (IRM) et l'analyse des caractéristiques colorées d'images naturelles.
- **Apprentissage incrémental par classes (Class-Incremental) :** en utilisant la base de données de panneaux de signalisation routière allemande (GTSRB), découpée en tâches séquentielles.

Les résultats expérimentaux montrent la capacité du mécanisme proposé à réduire l'oubli catastrophique et à atteindre une stabilité de performance par rapport aux optimiseurs classiques tels qu'Adam, ce qui en fait une avancée prometteuse vers la construction de systèmes d'intelligence artificielle capables d'un apprentissage cumulatif durable dans des environnements réels et changeants. La validation empirique exhaustive des composantes avancées de GLSA-H (mise à l'échelle Fisher-Newton, momentum, hypergradient) au-delà du noyau à base de conflit cosinus est identifiée comme une perspective de recherche.

Mots-clés : Apprentissage Continu, Oubli Catastrophique, GLSA-H, Stabilité-Plasticité, Gradient Cosinus, Information de Fisher, Deep Learning.

Table des matières

Table des figures

Liste des tableaux

Liste des Abréviations et Symboles

1. Abréviations

Abréviation	Description
IA	Intelligence Artificielle
DL	Deep Learning (Apprentissage Profond)
CL	Continual Learning (Apprentissage Continu)
CF	Catastrophic Forgetting (Oubli Catastrophique) / Conflict Factor selon le contexte
GLSA-H	Global-Local Structural Plasticity-Stability Alignment — Hybrid (version finale proposée)
GLSA-S	Version intermédiaire de GLSA fondée sur le seul Facteur de Conflit Co-sinus
EWC	Elastic Weight Consolidation
CNN	Convolutional Neural Network
IRM	Imagerie par Résonance Magnétique
GTSRB	German Traffic Sign Recognition Benchmark

2. Symboles Mathématiques

Symbole	Description
θ ou W	Les paramètres (poids) du réseau de neurones
θ^* (ou W_0)	L'ancre de stabilité (Stability Anchor) — instantané des poids avant une nouvelle tâche
$\mathcal{L}_{data}$	Fonction de perte associée à la tâche courante (Plasticité)
$\mathcal{L}_{prior}$	Fonction de perte de régularisation (Stabilité/Rétention)
G_{data}	Gradient de l'apprentissage (dérivée de $\mathcal{L}_{data}$)
G_{prior}	Gradient de stabilité, pondéré par la Fisher diagonale : $F \odot (\theta - \theta^*)$
F	Estimation diagonale de l'information de Fisher (importance relative des paramètres)
$\cos \omega$	Similarité cosinus entre les gradients G_{data} et G_{prior}
CF	Conflict Factor — facteur normalisé mesurant l'intensité du conflit géométrique
α (α_{smart})	Coefficient d'équilibre intelligent (Alpha dynamique)
$\alpha_{base}, \alpha_{min}, \alpha_{max}$	Valeur initiale et bornes du coefficient de stabilité
w_{CF}	Poids d'influence du Conflict Factor sur α
m_t	Direction lissée du gradient (momentum)
β_m	Coefficient de lissage du momentum
d_t	Direction hybride de mise à jour
δ, ε_t	Bornes de sécurité pour la mise à l'échelle de type Newton
η (ou η_t)	Taux d'apprentissage (Learning Rate)
$\eta_{min}, \eta_{max}, \beta_\eta$	Bornes et pas d'adaptation du taux d'apprentissage par hypergradient
H_t	Signal d'hypergradient utilisé pour adapter η_t
n	Nombre de tâches accomplies (facteur temporel)

Introduction Générale

Fig. 1 : Principe de l'apprentissage continu — accumulation séquentielle des tâches avec mémoire durable.

Problématique de recherche : *Comment les réseaux de neurones artificiels peuvent-ils apprendre de nouvelles tâches sans perdre leurs connaissances des tâches précédentes ?*

Alors que l'Intelligence Artificielle s'impose comme un pilier de la technologie moderne, une discontinuité fondamentale subsiste entre la capacité d'apprentissage des machines et celle des humains. Là où l'esprit humain accumule les savoirs de manière fluide, nos réseaux de neurones souffrent d'une fragilité structurelle : l'oubli catastrophique.

Le domaine de l'apprentissage profond (*Deep Learning*) a connu un développement rapide ces dernières années, et les réseaux de neurones artificiels ont prouvé leur grande efficacité dans de nombreux domaines tels que la reconnaissance d'images, le traitement du langage naturel, et bien d'autres encore. Cependant, ces modèles souffrent d'une lacune fondamentale qui se manifeste dans leur incapacité à apprendre de manière cumulative ; ils ont tendance à oublier complètement les connaissances antérieures lorsqu'ils sont entraînés sur de nouvelles tâches — c'est ce qu'on appelle L'"oubli catastrophique" (*Catastrophic Forgetting*).

L'importance de cette problématique réside dans le fait qu'elle constitue un obstacle majeur à la construction de systèmes d'intelligence artificielle durables, capables de s'adapter à des environnements dynamiques et changeants — à l'image de la mémoire humaine, qui conserve les anciennes compétences tout en en acquérant de nouvelles simultanément.

Objectifs de la recherche

Les principaux objectifs de ce travail de recherche se résument comme suit :

- Étudier le phénomène de l'oubli catastrophique dans son contexte théorique et applicatif.
- Passer en revue les principales approches proposées dans la littérature scientifique pour limiter ce phénomène.
- Proposer un nouvel algorithme adaptatif contribuant à réduire l'oubli catastrophique, développé de manière progressive jusqu'à sa version hybride finale **GLSA-H**.
- Valider les performances du noyau algorithmique proposé à travers des expériences comparatives standardisées.

Structure du mémoire

Ce mémoire est organisé en quatre chapitres principaux :

— **Chapitre 1 :** Traite des fondements théoriques de l'apprentissage profond et de l'apprentissage continu.

— **Chapitre 2 :** Présente l'état de l'art et les approches existantes dans la littérature.

— **Chapitre 3 :** Expose en détail l'algorithme proposé (**GLSA-H**) avec toutes ses spécifications mathématiques et architecturales, ainsi que le parcours de développement qui a mené de GLSA à sa version hybride finale.

— **Chapitre 4 :** Présente le protocole expérimental, analyse les résultats obtenus et discute les performances comparatives.

Chapitre 1

Introduction à l'Apprentissage Profond et à l'Apprentissage Continu

1.1 Fondements des réseaux de neurones artificiels

Un réseau de neurones artificiel (Artificial Neural Network — ANN) est un modèle de calcul inspiré de la structure du cerveau humain [1]. Il est composé de couches successives de nœuds (neurones) interconnectés par des poids apprenables. Les réseaux de neurones se classifient en :

1.1.1 Réseaux à propagation directe (Feedforward Networks)

Comme les perceptrons multicouches (MLP). Le Perceptron Multi-Couches (MLP) représente l'architecture de base des réseaux de neurones artificiels profonds. Il s'agit d'un modèle de type feed-forward où l'information circule de manière unidirectionnelle à travers trois types de couches : l'entrée, les couches cachées et la sortie. Chaque neurone d'une couche est totalement connecté à ceux de la couche suivante par des poids synaptiques [6].

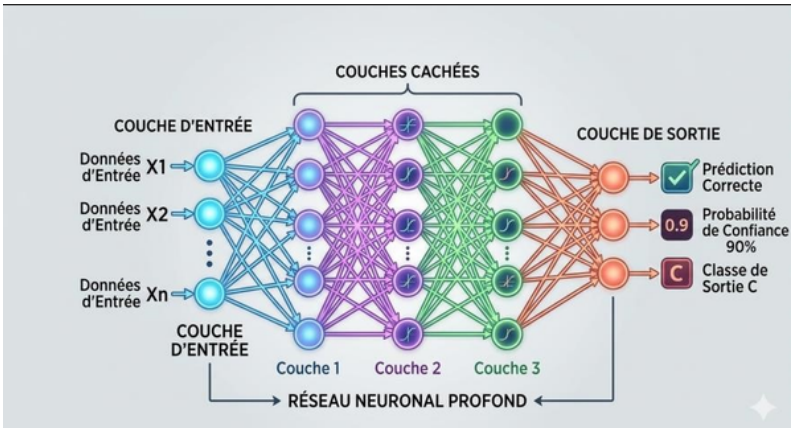

Fig. 1.1 : Architecture d'un Perceptron Multi-Couches (MLP)

Bien que le MLP soit extrêmement performant pour la reconnaissance de motifs complexes [7], sa structure souffre d'une rigidité face aux données séquentielles. Lors de l'apprentissage d'une nouvelle tâche, les poids optimisés pour les tâches précédentes sont brutalement modifiés, ce qui déclenche le phénomène de l'oubli catastrophique [2]. C'est cette architecture que nous cherchons à stabiliser à travers notre approche GLSA-H.

1.1.2 Réseaux de neurones convolutifs (CNN)

Spécialisés dans le traitement d'images [1]. Les Réseaux de neurones convolutifs (CNN) sont une catégorie de réseaux de neurones artificiels spécialement conçus pour le traitement et l'analyse des images [7]. Ils sont capables d'extraire automatiquement des caractéristiques importantes comme les bords, les textures, les formes et les objets présents dans une image [1].

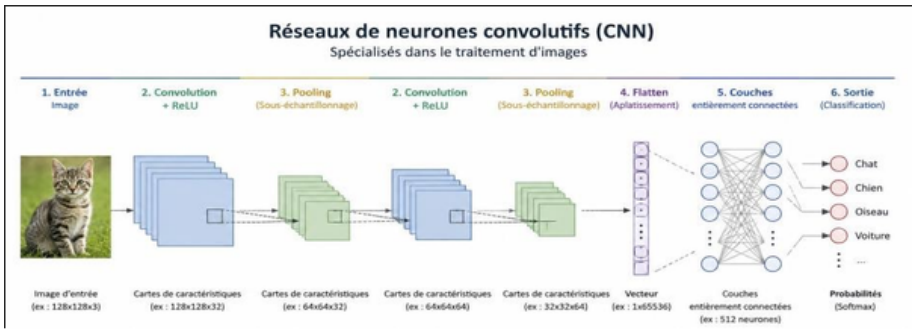

Fig. 1.2 : Architecture d'un réseau de neurones convolutif (CNN)

1.1.3 Réseaux de neurones récurrents (RNN)

Adaptés aux données séquentielles [1]. Les Réseaux de neurones récurrents (RNN) sont conçus pour traiter des données séquentielles, où l'ordre des informations est important [7]. Contrairement aux réseaux classiques, les RNN possèdent une mémoire interne qui leur permet de conserver les informations des étapes précédentes [1].

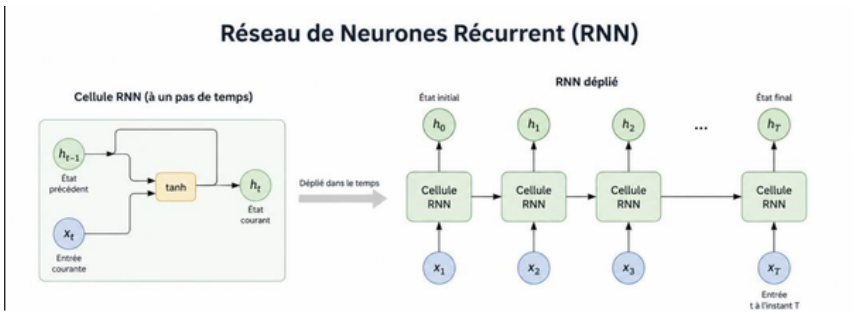

Fig. 1.3 : Architecture d'un réseau de neurones récurrent (RNN)

Le processus d'apprentissage repose sur l'algorithme de rétropropagation (Backpropagation) [1],

couplé à un optimiseur (Optimizer) qui met à jour les poids progressivement dans le but de minimiser la fonction de perte (Loss Function) [6].

1.2 Concept de l'apprentissage continu (Continual Learning)

L'apprentissage continu (Continual Learning) se définit comme le paradigme de programmation qui vise à permettre aux réseaux de neurones d'acquérir des connaissances à travers une séquence dynamique de tâches $(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n)$ de manière séquentielle, tout en garantissant la préservation des connaissances antérieures [4].

Le défi central dans ce cadre réside dans la gestion de ce qu'on appelle le « dilemme stabilité-plasticité » (Stability-Plasticity Dilemma) [7] ; le modèle doit être suffisamment plastique pour assimiler les nouvelles informations (Plasticity), et suffisamment stable pour minimiser la dégradation des connaissances anciennes (Stability) [7].

Nous avons adopté dans ce domaine trois protocoles (scénarios) principaux constituant les critères d'évaluation de l'algorithme proposé [4] :

— **Apprentissage incrémental par tâches (Task-Incremental Learning) :** dans ce scénario, le modèle reçoit l'identifiant de la tâche (Task ID) lors des phases d'entraînement et de test. Ce scénario est considéré comme le plus simple en termes de réduction de l'interférence cognitive et de contrôle de l'oubli catastrophique [5].

— **Apprentissage incrémental par domaine (Domain-Incremental Learning) :** ce protocole se concentre sur le changement de la distribution des données d'entrée (Input Distribution Shift) avec des objectifs de tâche et un nombre de classes stables [4].

— **Apprentissage incrémental par classes (Class-Incremental Learning) :** ce scénario représente le cas le plus complexe et le plus réaliste dans les systèmes intelligents ; de nouvelles classes sont ajoutées progressivement à l'espace de sortie sans aucun signal sur l'identité de la tâche [5] — c'est le scénario dans lequel nous avons testé la capacité de l'algorithme GLSA-H à maintenir des frontières de classification précises.

1.3 Le dilemme stabilité-plasticité

Le Stability-Plasticity Dilemma représente la tension fondamentale entre la préservation des connaissances antérieures (Stability) et la capacité à apprendre de nouvelles connaissances (Plasticity) [7]. Ce dilemme s'articule autour d'une contradiction fondamentale : augmenter la plasticité signifie une grande adaptabilité aux nouvelles données, mais expose le modèle à l'oubli catastrophique [**?**]. À l'inverse, une stabilité excessive entrave sa capacité à acquérir de nouvelles connaissances [5]. L'objectif de l'apprentissage continu est donc de trouver l'équilibre

optimal entre stabilité et plasticité pour garantir la rétention des connaissances passées sans figer la capacité d'apprentissage futur du modèle [4].

1.4 Le phénomène d'oubli catastrophique : causes et conséquences

L'oubli catastrophique se produit essentiellement en raison de l'interférence destructive (Destructive Interference) entre les tâches [2] ; le processus d'apprentissage de la nouvelle tâche $\mathcal{T}_k$ impose des modifications aux poids partagés dans le réseau de neurones, poids qui s'étaient stabilisés pour représenter les tâches précédentes $\mathcal{T}_1, \ldots, \mathcal{T}_{k-1}$ [4].

Ce changement dans les valeurs des poids conduit à l'effacement des patterns mémorisés précédemment, provoquant ainsi la perte par le modèle de sa capacité à réaliser les anciennes tâches dès qu'il acquiert de nouvelles informations [4]. L'ampleur de l'oubli dépend de plusieurs facteurs essentiels, notamment :

- **Le degré d'interférence entre les tâches (Task Similarity) :** plus les tâches se ressemblent dans la distribution de leurs données, plus la probabilité d'interférence augmente [4].
- **La valeur du taux d'apprentissage (Learning Rate) :** des valeurs élevées provoquent des changements radicaux et rapides dans les poids, ce qui accélère le processus d'effacement des connaissances antérieures [4].
- **L'architecture du modèle et le nombre de paramètres partagés :** les architectures étroites sont plus vulnérables à l'oubli en raison de la compétition sur les mêmes paramètres [5].

La compréhension de ces facteurs constitue donc la première étape vers la conception de stratégies efficaces d'apprentissage continu, équilibrant efficacité d'apprentissage et préservation des connaissances [4].

Chapitre 2

État de l'Art : Apprentissage Continu et Oubli Catastrophique

2.1 Techniques de réduction de l'oubli catastrophique

Les approches proposées dans la littérature actuelle pour lutter contre l'oubli catastrophique se dividient principalement en trois grandes catégories :

— **Méthodes basées sur la régularisation (Regularization) :** Exemples : EWC [15], SI.

Principe : Restreindre la modification des poids jugés critiques pour les tâches antérieures en ajoutant un terme de pénalité à la fonction de perte.

— **Méthodes de rejeu d'expérience (Replay / Rehearsal) :** Exemples : Experience Replay, GEM [12].

Principe : Conserver un sous-ensemble de données ou utiliser des modèles génératifs pour réintroduire les informations des tâches précédentes lors de l'apprentissage courant.

— **Architectures évolutives (Dynamic Architectures) :** Exemples : Progressive Networks [14], PackNet.

Principe : Allouer des sous-réseaux ou ajouter dynamiquement de nouvelles structures indépendantes pour chaque nouvelle tâche.

2.2 L'algorithme Adam : rôle et limites dans l'apprentissage continu

L'optimiseur Adam (*Adaptive Moment Estimation*) est l'un des algorithmes d'optimisation les plus répandus en apprentissage profond [11]. Il repose sur l'estimation des premiers et seconds moments du gradient pour ajuster le taux d'apprentissage de manière adaptative pour chaque paramètre.

2.2.1 Équations de mise à jour d'Adam

Pour chaque étape d'itération t, avec un gradient courant $g_t = \nabla_\theta \mathcal{L}(\theta_t)$, les étapes de mise à jour de l'algorithme Adam s'articulent comme suit :

1. **Calcul du premier moment** (moyenne mobile exponentielle des gradients) :

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \tag{2.1}$$

   où $\beta_1 \in [0, 1[$ est le coefficient de décroissance du premier moment.

2. **Calcul du second moment** (moyenne mobile exponentielle des carrés des gradients) :

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \tag{2.2}$$

   où $\beta_2 \in [0, 1[$ est le coefficient de décroissance du second moment.

3. **Correction du biais (*Bias Correction*) :**

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad \text{et} \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{2.3}$$

4. **Mise à jour des paramètres :**

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \cdot \hat{m}_t \tag{2.4}$$

   où η désigne le taux d'apprentissage (*Learning Rate*) et ε est une constante de stabilisation pour éviter la division par zéro.

2.2.2 Limites d'Adam en Apprentissage Continu

La force d'Adam réside dans son efficacité à gérer des paysages d'optimisation complexes et des gradients oscillants, ce qui le rend intuitivement attractif pour l'apprentissage continu où la distribution des données change en permanence.

Cependant, cet optimiseur est confronté à une limite fondamentale : il est conçu pour maximiser la performance sur la tâche courante sans conserver aucune « mémoire » des tâches passées. En raison de sa nature adaptative aggressive, Adam modifie les poids de manière radicale pour réduire l'erreur immédiate, effaçant ainsi les représentations internes précédemment acquises et détruisant la stabilité du réseau.

Adam reste donc, à lui seul, incapable de surmonter l'oubli catastrophique, à moins d'être couplé à des mécanismes externes imposant des contraintes géométriques et structurelles sur la mise à jour des paramètres critiques — un rôle réinventé au cœur du mécanisme hybride de notre algorithme **GLSA-H** (voir Chapitre 3).

2.3 L'algorithme EWC : Elastic Weight Consolidation

EWC (*Elastic Weight Consolidation*) constitue l'une des méthodes fondatrices basées sur la régularisation pour pallier l'oubli catastrophique [**?**]. Inspiré de la plasticité synaptique de la mémoire biologique, EWC protège les connaissances acquises en pénalisant les modifications sur les poids jugés stratégiques pour les tâches passées.

2.3.1 Principe de fonctionnement et formulation mathématique

EWC intègre une fonction de perte augmentée $\mathcal{L}(\theta)$, composée de la perte de la tâche courante et d'un terme de régularisation quadratique :

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{courant}}(\theta) + \sum_i \frac{\lambda}{2} F_i \left(\theta_i - \theta^*_{i,\text{ancien}}\right)^2 \tag{2.5}$$

Où :

- $\mathcal{L}_{\text{courant}}(\theta)$ représente la fonction de perte associée à la tâche actuelle.
- λ est un hyperparamètre de régularisation ajustant la force de rétention des anciennes tâches (compromis stabilité-plasticité).
- F_i désigne la valeur diagonale de la matrice d'information de Fisher (*Fisher Information Matrix*) associée au paramètre θ_i, mesurant son degré d'importance.
- $\theta^*_{i,\text{ancien}}$ représente la valeur optimale du poids θ_i fixée à la fin de l'entraînement sur la tâche précédente (ancre de stabilité).

2.4 Études comparatives et positionnement de GLSA-H

Les travaux récents démontrent clairement la supériorité des stratégies de régularisation par rapport aux optimiseurs standards dans les scénarios d'apprentissage continu [5]. Néanmoins, l'algorithme EWC présente plusieurs limitations pratiques majeures :

1. **Sensibilité aux hyperparamètres :** La performance globale dépend fortement du choix rigide de λ.
2. **Coût computationnel et stockage :** La réestimation et la sauvegarde des coefficients de Fisher deviennent coûteuses lorsque le nombre de tâches augmente.
3. **Pénalité statique (*Static Penalty*) :** EWC applique une contrainte fixe qui restreint excessivement la plasticité nécessaire à l'acquisition de nouvelles compétences complexes [5].

2.4.1 Apport et originalité de GLSA-H

C'est pour surmonter ces verrous structurels qu'a été conçu l'algorithme adaptatif **GLSA-H**. Contrairement à EWC :

- GLSA-H n'utilise l'information de Fisher que sous une forme diagonale réévaluée localement.
- Il combine dynamiquement cette information à un **facteur de conflit géométrique** (basé sur la similarité cosinus entre les gradients), à un lissage par momentum, et à une mise à l'échelle de type Newton opérant paramètre par paramètre.
- Il évite tout calcul ou inversion de matrice Hessienne complète.

Cette conception garantit un équilibre dynamique et infiniment plus précis entre Stabilité et Plasticité, tout en maintenant une complexité algorithmique considérablement réduite par rapport aux méthodes de régularisation à pénalité statique.

Chapitre 3

L'Algorithme Proposé : GLSA-H

Idée centrale : Construire, à chaque étape d'entraînement, un signal de plasticité et un signal de stabilité pondéré par l'importance des paramètres, analyser leur conflit géométrique, puis fusionner ces signaux de manière adaptative — sans recourir au stockage d'exemples des tâches passées.

Ce chapitre présente l'algorithme **GLSA-H**, cœur de notre contribution. GLSA-H est l'aboutissement d'un parcours de développement progressif au cours duquel plusieurs versions intermédiaires ont été explorées, chacune révélant des insuffisances qui ont motivé un raffinement supplémentaire, jusqu'à la formulation hybride finale évaluée dans ce mémoire.

Aperçu — Équation centrale de GLSA-H

L'objectif global s'exprime sous la forme conceptuelle suivante :

$$\mathcal{L}_{\text{total}}(\theta) = \alpha(n) \cdot \mathcal{L}_{\text{prior}}(\theta) + (1 - \alpha(n)) \cdot \mathcal{L}_{\text{data}}(\theta) \tag{3.1}$$

Les sections 3.1 à **??** construisent progressivement chacun des termes de cette équation — $\mathcal{L}_{\text{data}}$, $\mathcal{L}_{\text{prior}}$, le *Conflict Factor* et le coefficient $\alpha(n)$ — à travers le parcours de développement de GLSA à GLSA-H. Sa formulation complète, avec la description détaillée de chaque variable, est rassemblée en section 3.4.

3.1 Parcours de développement — de GLSA à GLSA-H

3.1.1 Première version — GLSA originale

La première version reposait sur un coefficient d'équilibre qui décroît de manière inversement proportionnelle aux époques :

$$\alpha(n) = \frac{\alpha_{\text{base}}}{n} \tag{3.2}$$

Les expériences ont démontré que cette décroissance en $\mathcal{O}(1/n)$ affaiblit fortement la contrainte de préservation aux époques tardives ($\alpha(20) = 0.05$), rendant $\mathcal{L}_{\text{prior}}$ incapable de résister aux forts gradients de $\mathcal{L}_{\text{data}}$ — provoquant ainsi l'oubli catastrophique.

3.1.2 Deuxième version — GLSA avec seuil minimal

Un seuil minimal α_{min} a été ajouté pour empêcher la disparition du coefficient d'équilibre :

$$\alpha(n) = \frac{\alpha_{\text{base}}}{n} + \alpha_{\text{min}} \tag{3.3}$$

Malgré une amélioration partielle de la stabilité, l'oubli catastrophique persistait car le problème fondamental — la mise à jour de l'ancre W_0 après chaque tâche avec perte des connaissances cumulées, et l'absence de toute notion d'importance différenciée des paramètres — n'était pas résolu.

3.1.3 Troisième version — GLSA-S avec Conflict Factor

Un *Conflict Factor* a été introduit comme mesure de l'intensité du conflit de gradients, donnant naissance à la version dite **GLSA-S** :

$$\alpha(n) = \alpha_{\text{base}} \cdot \left(\frac{1}{n} + \text{CF}\right), \quad \text{où CF} = \max\left(0, -\cos(\nabla\mathcal{L}_{\text{data}}, \nabla\mathcal{L}_{\text{prior}})\right) \tag{3.4}$$

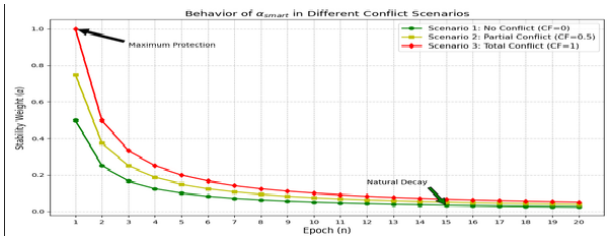

Fig. 3.1 : Mécanisme de régulation du Conflict Factor dans GLSA-S

GLSA-S a permis une réduction substantielle de l'oubli catastrophique grâce à son mécanisme de « frein d'urgence » activé par la similarité cosinus. Elle présentait néanmoins trois limites structurelles qui ont motivé son extension :

1. Tous les paramètres du modèle étaient traités de manière uniforme dans le terme de stabilité $\|\theta - \theta^*\|^2$, sans distinction entre poids critiques et poids secondaires.
2. Le gradient de plasticité G_{data} n'était pas lissé, ce qui pouvait introduire des oscillations d'une itération à l'autre.
3. L'amplitude de la mise à jour n'était pas modulée paramètre par paramètre, contrairement aux méthodes de second ordre.

3.1.4 Version finale — GLSA-H (Hybrid)

GLSA-H (*Global-Local Structural Plasticity-Stability Alignment — Hybrid*) constitue la version finale de l'algorithme, retenue pour ce mémoire. Elle conserve le principe fondateur de GLSA-S — l'analyse géométrique du conflit entre gradients via la similarité cosinus — tout en l'enrichissant de trois mécanismes complémentaires :

- Une pondération de l'importance des paramètres par une estimation diagonale de l'information de Fisher ;
- Un lissage du signal de plasticité par momentum ;
- Une mise à l'échelle de type Newton de l'amplitude des mises à jour.

Un mécanisme optionnel d'adaptation du taux d'apprentissage par hypergradient est également introduit à titre prospectif.

Remarque méthodologique : Les sections 3.2 à 3.5 ci-dessous décrivent la formulation complète de GLSA-H. Le noyau algorithmique correspondant à GLSA-S (ancrage + gradients duaux + *Conflict Factor* + α adaptatif) demeure le sous-ensemble effectivement validé empiriquement au Chapitre 4. Les composantes supplémentaires (Fisher diagonale, momentum, mise à l'échelle de type Newton, hypergradient) sont présentées avec leur justification théorique, leur validation empirique exhaustive étant identifiée comme perspective de recherche (voir §3.17 et Conclusion générale).

3.2 Vue générale de l'architecture algorithmique GLSA-H

La méthode comporte plusieurs mécanismes complémentaires, articulés en chaîne causale : ancrage des paramètres de la tâche précédente ; estimation de l'importance des paramètres par une information de Fisher diagonale ; calcul du gradient de la tâche courante ; construction d'un gradient de stabilité ; détection du conflit par similarité cosinus ; adaptation du coefficient α selon le conflit ; lissage du gradient par momentum ; mise à l'échelle de type Newton à partir de la Fisher ; adaptation optionnelle du taux d'apprentissage par hypergradient ; et mise à jour des paramètres sans stockage d'un *replay buffer*.

3.3 Conservation de l'information passée

3.3.1 Ancre de stabilité

Avant l'apprentissage d'une nouvelle tâche, une copie de l'état des paramètres est conservée. Cette copie, notée θ^*, constitue une ancre de stabilité servant de référence pour mesurer

la dérive des paramètres pendant l'apprentissage de la nouvelle tâche :

$$\theta^* \leftarrow \theta \quad \text{(état de référence avant la nouvelle tâche)} \tag{3.5}$$

Cette représentation ne correspond pas à un ensemble d'exemples passés. Dans ce sens, GLSA-H est une approche *rehearsal-free* ou *buffer-free*.

3.3.2 Information de Fisher et signal de stabilité

Les paramètres d'un réseau ne contribuent pas tous de la même manière aux connaissances acquises. GLSA-H utilise une estimation diagonale de l'information de Fisher F afin d'attribuer une importance relative aux paramètres :

$$G_{\text{prior}} = F \odot (\theta - \theta^*) \tag{3.6}$$

Le vecteur G_{prior} constitue ainsi un signal de stabilité qui tend à limiter la dérive des paramètres considérés comme importants. Contrairement à EWC (§2.3), F n'est utilisée ici que sous forme diagonale et locale, sans qu'il soit nécessaire de la figer comme pénalité statique dans la fonction de perte globale : elle intervient directement dans la construction du signal de stabilité et dans la mise à l'échelle de la mise à jour (§3.9).

3.3.3 Gradient de la tâche courante

$G_{\text{data}} = \nabla_\theta \mathcal{L}_{\text{data}}(\theta)$ représente la composante de plasticité. Il oriente les paramètres vers une réduction de la perte associée aux données de la tâche courante, exactement comme dans un entraînement supervisé classique.

3.3.4 Analyse du conflit entre gradients

Similarité cosinus

$$\cos(\omega) = \frac{G_{\text{data}}^\top \cdot G_{\text{prior}}}{\|G_{\text{data}}\| \cdot \|G_{\text{prior}}\| + \varepsilon} \tag{3.7}$$

La similarité cosinus mesure la compatibilité directionnelle entre les deux signaux. Une opposition des directions indique qu'un apprentissage plus important de la tâche courante peut entrer en conflit avec la conservation de l'état antérieur.

Conflict Factor

$$\text{CF} = \text{clip}\left[\frac{1 - \cos(\omega)}{2},\, 0,\, 1\right] \tag{3.8}$$

Le *Conflict Factor* est normalisé entre 0 et 1 et sert de signal de contrôle : $\text{CF} = 0$ lorsque les gradients sont parfaitement alignés ($\cos(\omega) = 1$), et $\text{CF} = 1$ lorsqu'ils sont totalement opposés ($\cos(\omega) = -1$). Cette normalisation raffine celle utilisée dans GLSA-S (§3.1.3), qui ignorait les conflits partiels associés à un cosinus positif mais faible.

3.3.5 Coefficient de stabilité adaptatif

$$\alpha = \text{clip}\left[\alpha_{\text{base}} \cdot (1 + w_{\text{CF}} \cdot \text{CF}),\ \alpha_{\text{min}},\ \alpha_{\text{max}}\right] \tag{3.9}$$

Le coefficient α contrôle le compromis entre les deux composantes. Lorsque le conflit augmente, la contribution du signal de stabilité augmente proportionnellement, pondérée par w_{CF}. Les bornes α_{min} et α_{max} limitent les valeurs extrêmes et contribuent à la stabilité numérique de l'optimisation — un raffinement direct du terme de décroissance en $1/n$ de GLSA (§3.1.1–3.1.2), désormais borné explicitement plutôt que laissé libre de s'annuler ou de diverger.

3.3.6 Intégration du momentum

$$m_t = \beta_m \cdot m_{t-1} + (1 - \beta_m) \cdot G_{\text{data}} \tag{3.10}$$

Le momentum fournit une direction lissée du gradient courant. Il vise à réduire les fluctuations entre les itérations et à produire une évolution plus régulière des paramètres, selon un principe analogue au premier moment d'Adam (§2.2), mais réservé ici exclusivement à la composante de plasticité.

3.3.7 Direction hybride

$$d_t = (1 - \alpha) \cdot m_t + \alpha \cdot G_{\text{prior}} \tag{3.11}$$

Cette relation constitue le cœur du compromis stabilité-plasticité : $(1 - \alpha) \cdot m_t$ favorise l'apprentissage de la tâche courante, tandis que $\alpha \cdot G_{\text{prior}}$ favorise la conservation de l'état antérieur. Elle généralise la règle de mise à jour hybride de GLSA-S ($\theta_{\text{new}} = \theta_{\text{old}} - \eta \cdot [\alpha \cdot G_{\text{prior}} + (1 - \alpha) \cdot G_{\text{data}}]$) en substituant le gradient lissé m_t au gradient brut G_{data}.

3.3.8 Mise à l'échelle de type Newton

$$\Delta\theta = \text{clip}\left[\frac{d_t}{F + \varepsilon_t},\ -\delta,\ \delta\right] \tag{3.12}$$

$$\theta_{t+1} = \theta_t - \eta_t \cdot \Delta\theta \tag{3.13}$$

GLSA-H n'effectue ni la construction ni l'inversion d'une matrice Hessienne complète. La mise à l'échelle utilisée ici s'appuie sur la Fisher diagonale afin de moduler l'amplitude de la mise à jour paramètre par paramètre : les poids associés à une Fisher élevée (jugés importants pour les tâches passées) reçoivent une mise à jour plus prudente. Le terme ε_t protège contre les divisions par des valeurs trop faibles, et δ limite l'amplitude de la mise à jour pour garantir la stabilité numérique de l'optimisation.

3.3.9 Adaptation du taux d'apprentissage

Dans la configuration complète, le taux d'apprentissage peut être adapté à partir de l'évolution des gradients successifs. Cette composante vise à réduire la dépendance à un taux strictement fixe :

$$\eta_t = \mathsf{clip}\left[\eta_{t-1} - \beta_\eta \cdot H_t,\ \eta_{\mathsf{min}},\ \eta_{\mathsf{max}}\right] \tag{3.14}$$

Son analyse théorique détaillée dépasse le cadre de cette étude de Master et sera considérée comme une perspective de recherche (voir Conclusion générale).

3.3.10 Formulation complète de la fonction de perte et description des variables

Les sections précédentes ont introduit progressivement chacune des composantes de GLSA-H. Cette section rassemble l'ensemble en une formulation unique — l'équation centrale de l'algorithme — puis en détaille chaque variable.

Équation centrale :

$$\mathcal{L}_{\mathsf{total}}(\theta) = \alpha(n) \cdot \mathcal{L}_{\mathsf{prior}}(\theta) + (1 - \alpha(n)) \cdot \mathcal{L}_{\mathsf{data}}(\theta) \tag{3.15}$$

où les deux composantes de la perte, ainsi que le coefficient d'équilibre, sont définis par :

$$\begin{aligned}
\mathcal{L}_{\mathsf{data}}(\theta) &= \text{Perte de la tâche courante (Cross-Entropy)} \\
\mathcal{L}_{\mathsf{prior}}(\theta) &= \frac{1}{2}\sum_i F_i \cdot (\theta_i - \theta_i^*)^2 \\
\alpha(n) &= \mathsf{clip}\left[\alpha_{\mathsf{base}} \cdot (1 + w_{\mathsf{CF}} \cdot \mathsf{CF}),\ \alpha_{\mathsf{min}},\ \alpha_{\mathsf{max}}\right] \\
\mathsf{CF} &= \mathsf{clip}\left[\frac{1 - \cos(\omega)}{2},\ 0,\ 1\right], \quad \cos(\omega) = \frac{G_{\mathsf{data}} \cdot G_{\mathsf{prior}}}{\|G_{\mathsf{data}}\| \cdot \|G_{\mathsf{prior}}\| + \varepsilon}
\end{aligned}$$

Cette formulation exprime, sous forme d'une unique fonction de perte pondérée, le compromis stabilité-plasticité au cœur de GLSA-H : $\mathcal{L}_{\mathsf{data}}$ pousse les paramètres vers une meilleure performance sur la tâche courante (plasticité), tandis que $\mathcal{L}_{\mathsf{prior}}$ — pondérée par l'importance de Fisher et mesurée par rapport à l'ancre θ^* — les retient dans le voisinage de leur état anté-

rieur (stabilité). Le coefficient $\alpha(n)$ arbitre dynamiquement entre les deux, en fonction à la fois du temps écoulé (via α_{base}) et du conflit géométrique instantané détecté entre G_{data} et G_{prior} (via CF).

Sur le plan opérationnel, GLSA-H ne minimise pas $\mathcal{L}_{\text{total}}(\theta)$ par une différentiation directe de cette somme à chaque pas ; les gradients G_{data} et G_{prior} sont calculés séparément (§3.4 et §3.3.2), puis combinés directement au niveau de la direction de mise à jour — via le momentum (§3.7), la direction hybride (§3.8) et la mise à l'échelle de type Newton (§3.9).

L'équation centrale ci-dessus doit donc être lue comme la formulation conceptuelle équivalente de cette procédure, la plus appropriée pour en exposer la logique d'ensemble, plutôt que comme la trace exacte, ligne par ligne, de son implémentation.

Tab. 3.1 : Description des variables de l'équation centrale

Symbole	**Description**
θ	Paramètres (poids) du réseau de neurones, à l'itération courante.
θ^*	Ancre de stabilité — snapshot des paramètres avant l'apprentissage de la tâche courante.
F	Estimation diagonale de l'information de Fisher — importance relative de chaque paramètre.
$\mathcal{L}_{\text{data}}(\theta)$	Perte de la tâche courante (plasticité).
$\mathcal{L}_{\text{prior}}(\theta)$	Perte de stabilité, pondérée par F, mesurée par rapport à θ^*.
$G_{\text{data}}, G_{\text{prior}}$	Gradients respectifs de $\mathcal{L}_{\text{data}}$ et de $\mathcal{L}_{\text{prior}}$.
$\cos(\omega)$	Similarité cosinus entre G_{data} et G_{prior} — mesure l'alignement directionnel.
CF	*Conflict Factor* — intensité normalisée du conflit géométrique (0 à 1).
$\alpha(n), \alpha_{\text{base}}, \alpha_{\text{min}}, \alpha_{\text{max}}$	Coefficient d'équilibre adaptatif, sa valeur de base et ses bornes.
w_{CF}	Poids d'influence du *Conflict Factor* sur α.
n	Nombre de tâches accomplies (facteur temporel).
ε	Constante de stabilité numérique évitant la division par zéro.

Trois régimes se dégagent de cette formulation, selon l'intensité du conflit détecté :

- **Harmonie (CF ≈ 0) :** les gradients sont alignés ; α reste proche de sa valeur de base, et l'apprentissage de la nouvelle tâche se poursuit avec une liberté croissante à mesure que n augmente.
- **Conflit partiel ($0 <$ CF < 1) :** α augmente proportionnellement, renforçant le poids de $\mathcal{L}_{\text{prior}}$ pour ralentir la dérive des paramètres jugés importants.
- **Conflit maximal (CF $= 1$) :** α atteint sa valeur la plus élevée (bornée par α_{max}), donnant à $\mathcal{L}_{\text{prior}}$ une priorité maximale — un « frein d'urgence » protégeant les connaissances passées.

Positionnement de GLSA-H : fonction de perte, régularisateur ou optimiseur ?

Il convient de clarifier la nature exacte de GLSA-H au sens du génie logiciel. L'équation $\mathcal{L}_{\text{total}}$ présentée ci-dessus est, à elle seule, une fonction de perte régularisée (*Regularized Loss*) : une perte de tâche $\mathcal{L}_{\text{data}}$ à laquelle s'ajoute un terme de régularisation $\mathcal{L}_{\text{prior}}$, exactement comme dans EWC. Mais GLSA-H, pris dans son ensemble, ne se limite pas à ce terme additionnel : il modifie directement la règle de mise à jour des paramètres — via son propre momentum (§3.7), sa propre mise à l'échelle de type Newton (§3.9), et un coefficient α réévalué à chaque itération — plutôt que de déléguer cette mise à jour à un optimiseur générique comme Adam ou SGD appliqué en aval d'une perte régularisée. GLSA-H se classe donc, au sens strict, dans la catégorie des optimiseurs (*Optimizers*) : un optimiseur spécialisé pour l'apprentissage continu, qui intègre nativement un mécanisme de régularisation adaptative, plutôt qu'un simple terme de pénalité destiné à être utilisé avec un optimiseur générique préexistant. La formulation $\mathcal{L}_{\text{total}}$ doit ainsi être lue comme la traduction mathématique conceptuelle de l'objectif poursuivi par cet optimiseur, et non comme une couche de régularisation indépendante de son mécanisme de mise à jour.

3.4 Formulation complète de GLSA-H

L'algorithme complet GLSA-H intègre l'ensemble des composants décrits ci-dessus dans un cadre unifié.

L'enchaînement des mécanismes précédents suit une chaîne causale stricte, résumée dans les douze étapes suivantes :

1. Initialiser les paramètres du modèle et les hyperparamètres.
2. Conserver l'état de référence θ^* avant l'apprentissage de la nouvelle tâche.
3. Estimer l'information de Fisher F.
4. Calculer G_{data} pour la tâche courante.
5. Calculer $G_{\text{prior}} = F \odot (\theta - \theta^*)$.
6. Calculer la similarité cosinus puis le Conflict Factor CF.
7. Adapter le coefficient α.
8. Mettre à jour le momentum m_t.
9. Construire la direction hybride d_t.
10. Appliquer la mise à l'échelle de type Newton basée sur F.
11. Adapter η si le mécanisme d'hypergradient est activé.
12. Mettre à jour les paramètres θ et répéter jusqu'à la fin de la tâche.

3.5 Pseudo-code de l'algorithme GLSA-H

Voici la présentation algorithmique formelle de la méthode GLSA-H :

Algorithm 1 Algorithme GLSA-H

Require : $\theta \leftarrow$ Paramètres courants du modèle ; $\theta^* \leftarrow$ Paramètres d'ancrage (référence) ; $F \leftarrow$ Estimation diagonale de l'information de Fisher ; $\mathcal{D}_t \leftarrow$ Données de la tâche courante ; $\eta \leftarrow$ Taux d'apprentissage ; $\alpha_{\text{base}} \leftarrow$ Coefficient adaptatif de base

Initialisation : $m \leftarrow 0$ ▷ État du momentum, $g_{\text{prev}} \leftarrow 0$ ▷ Gradient précédent

for chaque itération d'entraînement t **do**

- $G_{\text{data}} \leftarrow \nabla_\theta \mathcal{L}(\mathcal{D}_t, \theta)$ ▷ Gradient de plasticité
- $G_{\text{prior}} \leftarrow F \odot (\theta - \theta^*)$ ▷ Gradient de stabilité
- $\cos(\omega) \leftarrow \text{CosineSimilarity}(G_{\text{data}}, G_{\text{prior}})$ ▷ Similarité cosinus
- $\text{CF} \leftarrow \text{clip}\left[\frac{1-\cos(\omega)}{2}, 0, 1\right]$ ▷ Conflict Factor
- $\alpha \leftarrow \text{clip}\left[\alpha_{\text{base}} \cdot (1 + w_{\text{CF}} \cdot \text{CF}), \alpha_{\text{min}}, \alpha_{\text{max}}\right]$ ▷ Adaptation de α
- $m \leftarrow \beta_m \cdot m + (1 - \beta_m) \cdot G_{\text{data}}$ ▷ Mise à jour du momentum
- $d \leftarrow (1 - \alpha) \cdot m + \alpha \cdot G_{\text{prior}}$ ▷ Direction hybride
- $d \leftarrow \text{clip}\left[\frac{d}{F+\varepsilon}, -\delta, \delta\right]$ ▷ Mise à l'échelle Newton
- $\eta \leftarrow \text{HypergradientUpdate}(\eta, d)$ ▷ Adaptation optionnelle de η
- $\theta \leftarrow \theta - \eta \cdot d$ ▷ Mise à jour des paramètres
- $g_{\text{prev}} \leftarrow G_{\text{data}}$ ▷ Sauvegarde du gradient

end for

return θ

3.6 Caractéristique sans replay buffer

Une caractéristique importante de GLSA-H est l'absence de stockage explicite d'un ensemble d'exemples issus des tâches précédentes dans le mécanisme présenté. L'algorithme conserve cependant un état interne, notamment l'ancre θ^* et les statistiques de Fisher F. Il est donc plus précis de parler d'une méthode *rehearsal-free* ou *buffer-free* que d'une méthode « sans mémoire » au sens absolu.

3.7 Rôle des composants

Le Tableau 3.2 résume le rôle fonctionnel de chacun des composants constituant l'architecture globale de GLSA-H :

Tab. 3.2 : Rôle des différents composants de l'algorithme GLSA-H.

Composant	Rôle principal
θ^*	Référence des paramètres antérieurs (Ancre de Stabilité)
Fisher F	Importance relative et sensibilité des paramètres
$G_{\textbf{data}}$	Signal de plasticité / apprentissage de la tâche courante
$G_{\textbf{prior}}$	Signal de stabilité / conservation des connaissances passées
CF (Conflict Factor)	Mesure du conflit directionnel géométrique via le cosinus
Adaptive α	Contrôle dynamique du compromis Stabilité-Plasticité
Momentum	Lissage du signal de plasticité et réduction des oscillations
Newton-style scaling	Modulation de l'amplitude de la mise à jour paramètre par paramètre
Hypergradient	Adaptation dynamique du taux d'apprentissage (extension prospective)

3.8 Architecture de la solution

L'architecture retenue pour l'évaluation de GLSA-H repose sur une structure d'« apprentissage continu basé sur les poids » (*Weight-based Continual Learning*), composée de trois éléments fonctionnels intégrés (Figure **??**) :

1. **Colonne vertébrale partagée (*Shared Backbone*) — ResNet50 :** Choisie pour ses performances reconnues en extraction de caractéristiques profondes via des blocs résiduels (*Residual Blocks*) [**?**].

2. **Couche de sortie multi-têtes (*Multi-Head Output Layer*) :** Des « têtes » dédiées et structurellement séparées sont conçues pour chaque tâche — Tête de classification médicale (*Head 1*) et Tête d'analyse quantitative (*Head 2*).

3. **Moteur GLSA-H (*GLSA-H Engine*) :** Le « noyau central » de l'architecture, agissant comme un *middleware* qui contrôle le flux de données pendant l'entraînement [**?**] — estimation de l'importance des poids, fusion intelligente des gradients via la similarité cosinus, lissage par momentum et mise à l'échelle Fisher-Newton.

3.9 Portée et limites de la formulation

La formulation présentée décrit le mécanisme algorithmique retenu dans cette étude. Elle ne constitue pas une preuve générale de convergence ni une garantie théorique d'absence

d'oubli catastrophique. Les conditions de stabilité, les bornes de *forgetting* et les propriétés de convergence constituent des axes de recherche ultérieurs.

L'évaluation doit également considérer simultanément les performances sur la tâche précédente, l'apprentissage de la nouvelle tâche et les résultats en *Class-Incremental Learning* : une faible valeur de *forgetting* ne suffit pas à elle seule pour conclure à la supériorité d'une méthode. Par ailleurs, les expériences rapportées au Chapitre 4 valident essentiellement le noyau algorithmique commun à GLSA-S et GLSA-H (ancrage, gradients duaux, *Conflict Factor*, α adaptatif) ; une étude d'ablation isolant la contribution propre de la Fisher diagonale, du momentum, de la mise à l'échelle de type Newton et de l'hypergradient reste à mener.

3.10 Conclusion du chapitre

Ce chapitre a présenté GLSA-H comme une approche d'optimisation destinée à équilibrer stabilité et plasticité dans l'apprentissage continu, issue d'un parcours de développement technique et logique intégré. Nous avons commencé par exposer la trajectoire d'évolution structurelle de l'algorithme — de la décroissance simple en $1/n$ de GLSA, en passant par l'introduction d'un seuil minimal, puis par le *Conflict Factor* cosinus de GLSA-S — pour aboutir à la formulation hybride complète de GLSA-H, qui combine une ancre paramétrique, une information de Fisher diagonale, des signaux de gradient de plasticité et de stabilité, une analyse de conflit par similarité cosinus, un coefficient adaptatif borné, un mécanisme de momentum et une mise à l'échelle de type Newton.

Le chapitre suivant (Chapitre 4) présente l'évaluation expérimentale du noyau algorithmique de ces mécanismes à travers des méthodes de référence (Adam, EWC), sur deux scénarios d'apprentissage continu. Cette analyse permettra de caractériser le compromis stabilité-plasticité obtenu et d'identifier les axes de validation complémentaires nécessaires pour les composantes avancées de GLSA-H.

Chapitre 4

Expériences et Résultats

Note de portée : *Les études de cas 1 et 2 (§4.5–4.6) ont été conduites avec le noyau algorithmique commun à GLSA-S et GLSA-H — ancre de stabilité, gradients duaux, Conflict Factor cosinus et coefficient α adaptatif borné (§3.1 à §3.8). L'étude de cas 3 (§4.7) évalue pour sa part la configuration complète de GLSA-H, y compris ses composantes avancées (Fisher diagonale, momentum, mise à l'échelle de type Newton, hypergradient), à travers une étude d'ablation et une validation multi-graines. Elle nuance certaines conclusions des sections précédentes et doit être lue comme complémentaire, et non comme une simple confirmation, de celles-ci.*

4.1 Environnement de travail

Les expérimentations ont été menées sur la plateforme Cloud Kaggle Kernels. Le Tableau 4.1 détaille la configuration matérielle et logicielle exploitée pour l'ensemble des simulations.

Tab. 4.1 : Configuration de l'environnement de travail matériel et logiciel.

Composant	Détails
Plateforme	Kaggle Kernels (Environnement Cloud)
Processeur graphique (GPU)	NVIDIA Tesla P100 (16 Go VRAM)
Mémoire système (RAM)	13 Go
Langage de programmation	Python 3.10
Framework principal	PyTorch 2.x
Bibliothèques auxiliaires	NumPy, Pandas, Torchvision, Matplotlib

Note sur les architectures utilisées : L'algorithme a été testé sur deux architectures différentes pour démontrer sa flexibilité et sa scalabilité :

— **ResNet50 :** Utilisée pour les expériences de comparaison et la première étude de cas (imagerie médicale).

— **MobileNetV2 :** Utilisée pour la deuxième étude de cas (robotique et signalisation routière), afin de prouver l'efficacité du mécanisme sur des architectures légères (*Lightweight*) conçues pour fonctionner en temps réel sur des appareils à ressources limitées.

4.2 Ensembles de données utilisés

Les évaluations comparatives initiales reposent sur la succession de deux tâches de classification binaire d'imagerie médicale cérébrale (IRM normalisées, résolution 224×224), présentées dans le Tableau 4.2.

Tab. 4.2 : Spécifications des tâches et ensembles de données IRM.

Tâche	**Description**	**Type**	**Classes**
Tâche 1 (T_1)	IRM cérébrales normalisées (224×224)	Classification binaire	Gliome vs Absence
Tâche 2 (T_2)	IRM cérébrales normalisées (224×224)	Classification binaire	Méningiome vs Hypophyse

4.3 Protocole de comparaison et métriques

Le Tableau 4.3 résume les moteurs comparés, les métriques d'évaluation retenues et les hyperparamètres généraux d'entraînement.

Tab. 4.3 : Protocole expérimental, métriques d'évaluation et hyperparamètres.

Élément	Détails
Moteurs comparés	Adam (baseline), EWC (pénalité), GLSA-H — noyau algorithmique (proposé)
Mesure de stabilité	Précision Tâche 1 (T_1 Acc) après apprentissage de T_2
Mesure de plasticité	Précision Tâche 2 (T_2 Acc)
Mesure d'oubli (*Forgetting*)	$\Delta =$ Précision initiale $T_1 -$ Précision finale T_1
Hyperparamètre	**Valeur**
Learning Rate (LR)	1×10^{-4}
Batch Size	32
Epochs par tâche	7

4.4 Présentation et analyse des résultats

Les performances comparatives enregistrées sur l'environnement Kaggle entre la baseline Adam, la méthode EWC et le noyau algorithmique GLSA-H sont rapportées dans le Tableau 4.4.

Méthode	**Acc. Initiale T_1**	**Acc. Finale T_1**	**Taux d'Oubli (Δ)**	**Acc. Finale T_2**
Adam (Baseline)	90.88%	57.12%	33.75%	98.88%
EWC	90.88%	77.38%	13.50%	98.62%
GLSA-H — noyau (Proposé)	90.88%	**90.38%**	**0.50%**	63.62%

Tab. 4.4 : Résultats comparatifs sur le scénario à deux tâches — Stabilité (T_1) et Plasticité (T_2).

Fig. 4.1 : Comparaison de la stabilité (précision T1) — GLSA vs Adam vs EWC au fil des époques d'entraînement de T2

Conclusion finale sur les expériences initiales : Sur la base de cette analyse comparative, plusieurs constats s'imposent :

1. **Efficacité du noyau GLSA-H :** La technique d'alignement de gradients (*Gradient Alignment*) surpasse les approches traditionnelles de pénalisation globale (comme EWC) dans la préservation des poids critiques des anciennes tâches.
2. **Réduction de l'oubli catastrophique :** Le taux d'oubli est réduit à seulement 0.50% pour T_1, contre un effondrement sévère à 33.75% sous l'optimiseur Adam.
3. **Dynamique d'apprentissage :** L'ajustement dynamique de α via la similarité cosinus permet de détecter les interférences entre tâches et d'adapter instantanément la trajectoire d'optimisation.
4. **Application médicale :** Ces résultats confirment la viabilité de l'approche pour les systèmes de diagnostic évolutifs, capables d'intégrer de nouvelles pathologies sans dégrader leurs compétences acquises.

4.5 Étude de cas 1 : Apprentissage multi-domaines (Cross-Domain)

Cette expérience évalue la robustesse de GLSA-H lors de transitions entre des domaines visuels et structurels totalement éloignés :

— **Tâche 1 (T_1) :** Diagnostic médical de tumeurs cérébrales sur des images IRM (niveaux de gris, structures tissulaires fines).

— **Tâche 2 (T_2) :** Analyse quantitative des couleurs à partir de la base Mini-ImageNet (images naturelles RGB, objets communs).

Le Tableau 4.5 présente la stabilité de la fonction de perte (*Loss*) obtenue sur une validation croisée à 5 plis (*5-fold Cross-Validation*).

Tab. 4.5 : Résultats de validation croisée (5-Fold) sur le scénario multi-domaines.

Fold	**Époque 5**	**Époque 10**	**Époque 15**	**Loss Final / Stabilité**
Fold 1	0.006961	0.007016	0.007016	0.007010 / Très stable
Fold 2	0.007695	0.007660	0.007664	0.007570 / Très stable
Fold 3	0.005235	0.005228	0.005400	0.005219 / Très stable
Fold 4	0.006869	0.006781	0.006840	0.006794 / Très stable
Fold 5	0.006186	0.006196	0.006210	0.006157 / Très stable
Erreur moyenne générale (*Mean Loss*)			**0.006550**	*Excellent*
Écart-type (*Stabilité*)			**0.000805**	*Stabilité élevée*
Protection GLSA Shield (Backbone)			**151 couches**	*ACTIVE*

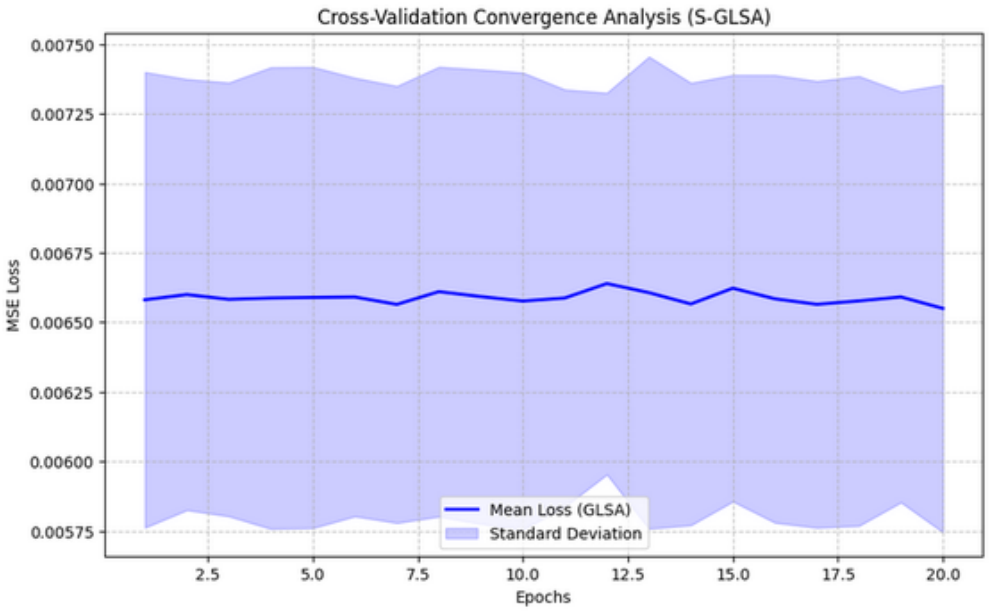

Fig. 4.2 : Analyse de convergence de la perte en validation croisée (5-fold) sous le régulateur GLSA-H.

Le Tableau 4.6 résume l'interprétation théorique et pratique de ces résultats.

Tab. 4.6 : Interprétation des résultats de validation croisée multi-domaines.

Résultat	Interprétation
Stabilité (Faible variance)	L'écart-type très faible (0.000805) confirme la robustesse du régulateur.
Efficacité mémoire	Idéal pour l'*Edge AI* car il ne nécessite aucun stockage de données passées (*Buffer-free*).
Apprentissage hybride	Transition réussie entre le domaine médical (IRM) et le domaine visuel naturel (Couleurs).
Validation globale	Mécanisme validé pour un apprentissage continu sans rejouer de données passées.

4.6 Étude de cas 2 : Applications robotiques et navigation autonome

Afin d'évaluer la méthode sur une architecture légère adaptée aux contraintes embarquées (*Edge AI*), MobileNetV2 a été déployé pour la reconnaissance séquentielle de panneaux routiers :

— **Tâche 1 (T_1) :** Reconnaissance des panneaux de limitation de vitesse (0–80 km/h) — classification multi-classes.

— **Tâche 2 (T_2) :** Reconnaissance des panneaux d'avertissement et de danger (travaux, virages) — classification multi-classes.

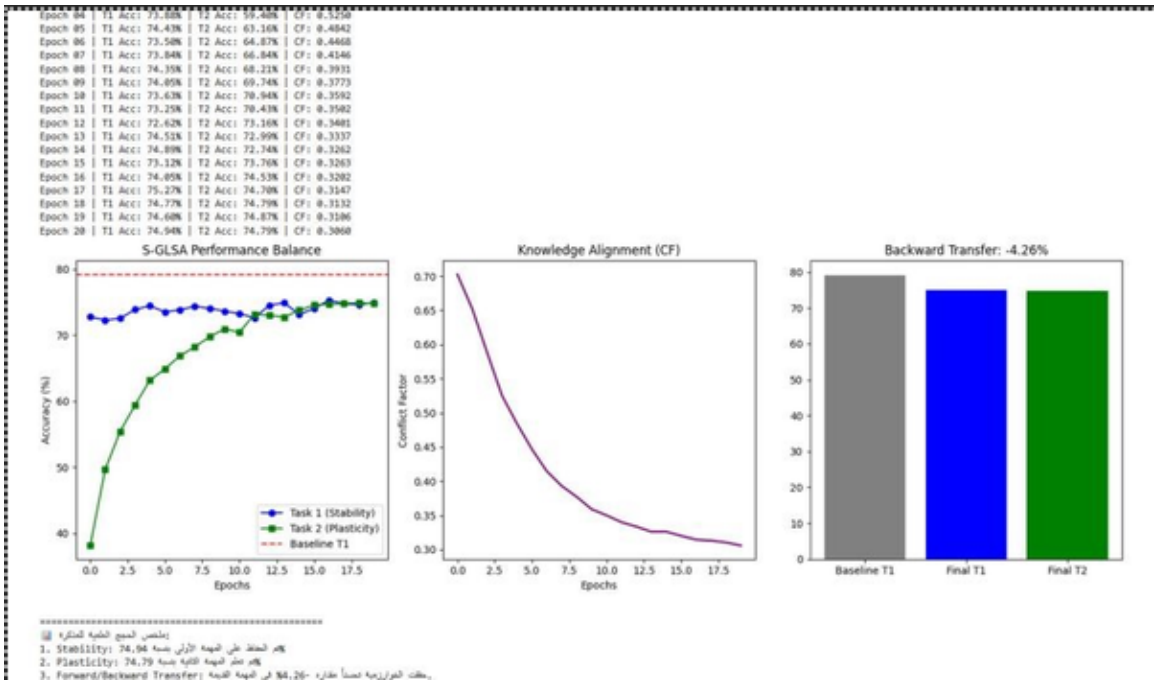

Fig. 4.3 : Équilibre des performances sous GLSA-H (gauche), alignement des connaissances via le *Conflict Factor* (centre), et transfert de connaissances (droite) sur MobileNetV2.

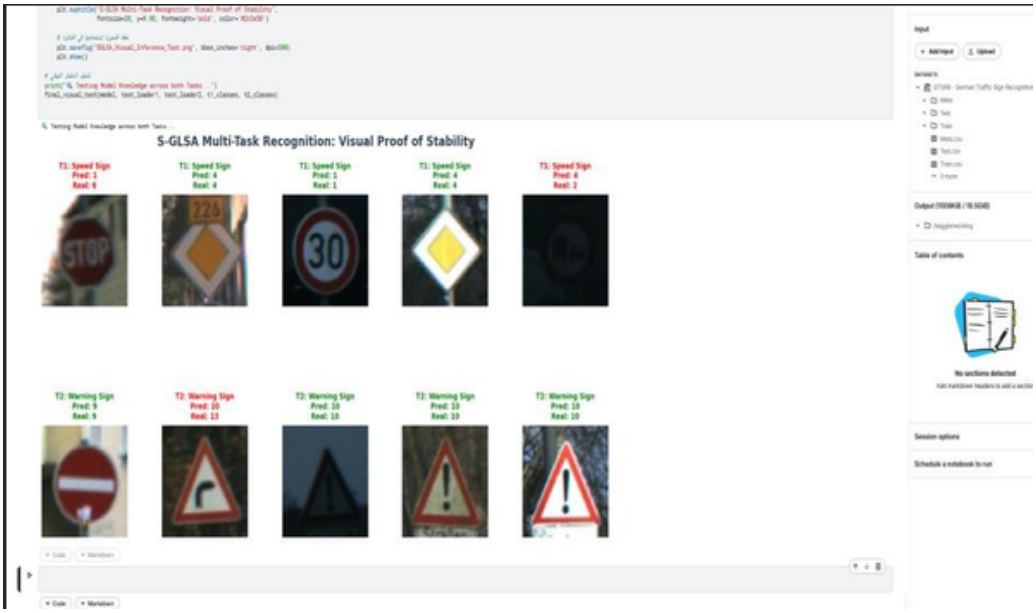

Fig. 4.4 : Validation qualitative de l'inférence multi-tâches en robotique autonome (exemples de prédictions correctes et d'erreurs d'inspection).

L'intégration réussie de MobileNetV2 démontre la faisabilité du déploiement de la « mémoire robotique » GLSA-H sur des cartes embarquées (ex. Raspberry Pi, NVIDIA Jetson Nano) avec une empreinte computationnelle minimale.

4.7 Étude de cas 3 : Étude d'ablation et validation multi-graines (CIFAR-10/100)

Les études de cas 1 et 2 (§4.5–4.6) ont validé le noyau algorithmique de GLSA-H, mais sans isoler la contribution propre de chacune de ses composantes avancées ni évaluer sa sensibilité à l'initialisation aléatoire. Cette troisième étude de cas comble ces lacunes via :

1. Une étude d'ablation systématique retirant tour à tour chaque mécanisme de GLSA-H.
2. Une validation multi-graines (*multi-seed validation*) de la configuration complète.
3. Une comparaison directe avec une méthode à mémoire tampon (*Memory Replay*).

4.7.1 Protocole expérimental

Le Tableau 4.7 détaille le protocole expérimental mis en place sur CIFAR-10/100.

Tab. 4.7 : Protocole de l'étude d'ablation et de validation multi-graines.

Élément	Détails
Jeu de données	CIFAR-10/100 — deux tâches successives (T_1, T_2)
Échantillons d'entraînement	$\approx 25\,000$ images par tâche
Architecture backbone	ResNet18 pré-entraîné sur ImageNet
Environnement d'exécution	GPU CUDA (Google Colab / Kaggle Kernels), PyTorch
Epochs par tâche	5 époques
Hyperparamètres GLSA-H	$\eta_0 = 1 \times 10^{-4}$, $\beta_m = 0.9$, $\alpha_0 = 0.618$, $w_{\text{CF}} = 2.0$
Estimation de Fisher	Calculée sur 50 batches avec lissage EMA (decay $= 0.9$)
Tailles de tampon (Replay)	500 et 1000 échantillons (≈ 2% et 4% d'une tâche)
Graines aléatoires testées	Configuration de référence, *Seed* = 42, *Seed* = 2027

Les méthodes comparées sont : Adam (baseline sans protection), EWC, GLSA-H complet, cinq variantes d'ablation de GLSA-H (retrait successif du Conflict Factor, de la mise à l'échelle de type Newton, du momentum, de l'adaptation de α, et de la Fisher diagonale), ainsi qu'un Memory Replay classique à deux tailles de tampon. L'évaluation combine les deux protocoles introduits au Chapitre 1 : Task-Incremental Learning (avec identifiant de tâche, noté T_1/T_2) et Class-Incremental Learning (sans identifiant de tâche, noté $T_{1\,\text{CIL}}/T_{2\,\text{CIL}}$).

4.7.2 Résultats comparatifs globaux

Le Tableau 4.8 présente les performances comparées entre Adam, EWC, GLSA-H complet et le *Memory Replay*.

Tab. 4.8 : Résultats comparatifs globaux — Adam, EWC, GLSA-H complet et Memory Replay.

Méthode	T_1 **av.** T_2	T_1 **ap.** T_2	T_2 **final**	**Forgetting**	**Avg. Acc.**	**Avg. CIL**
Adam (Baseline)	95.06%	38.44%	97.58%	56.62%	68.01%	—
EWC	95.18%	66.94%	97.66%	28.24%	82.30%	—
GLSA-H (complet)	95.08%	80.34%	88.30%	14.74%	84.32%	66.08%
Memory Replay (500)	95.16%	92.86%	97.50%	2.30%	95.18%	75.71%
Memory Replay (1000)	95.16%	93.66%	97.20%	1.50%	95.43%	82.26%

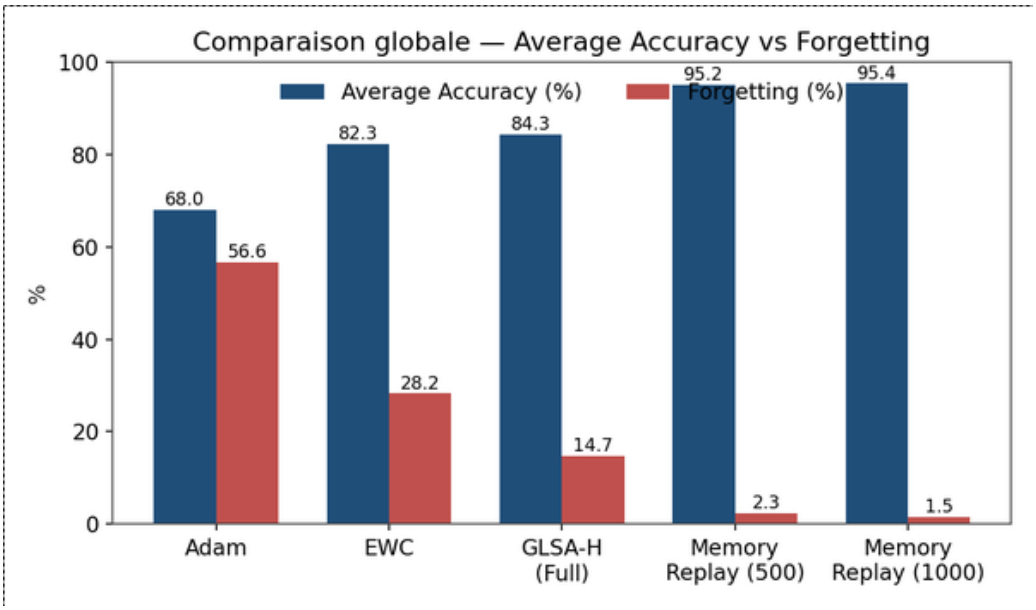

Fig. 4.5 : Comparaison globale de la précision moyenne et du taux d'oubli (*Forgetting*) par méthode.

Deux constats majeurs se dégagent :

1. **Supériorité sur EWC :** GLSA-H surpasse EWC en précision moyenne (84.32% contre 82.30%) et en réduction de l'oubli (14.74% contre 28.24%).
2. **Positionnement vis-à-vis du Replay :** Un simple tampon de rejeu de 500 exemples (2% des données) surpasse GLSA-H sur tous les indicateurs. L'atout de GLSA-H réside donc exclusivement dans son fonctionnement *rehearsal-free* (aucun stockage de données antérieures).

4.7.3 Étude d'ablation des composantes de GLSA-H

Le Tableau 4.9 consigne l'impact de la désactivation successive des composants avancés de GLSA-H.

Tab. 4.9 : Étude d'ablation — Contribution de chaque composant de GLSA-H.

Variante	T_1 **ap.** T_2	T_2 **final**	**Forgetting**	**Avg. Acc.**	**Avg. CIL**
GLSA-H (complet)	**80.34%**	**88.30%**	**14.74%**	**84.32%**	**66.08%**
— sans Conflict Factor	78.24%	89.40%	17.02%	83.82%	66.88%
— sans mise à l'échelle Newton	93.84%	26.20%	1.36%	60.02%	46.92%
— sans Momentum	86.76%	85.02%	8.40%	85.89%	45.53%
— sans α adaptatif	78.32%	89.46%	16.80%	83.89%	66.90%
— sans Fisher diagonale	92.76%	80.58%	2.46%	86.67%	45.05%*

* *Note :* Pour la variante « sans Fisher », la moyenne CIL masque une chute sévère de T_1_CIL à 9.56% (contre 80.54% pour T_2_CIL).

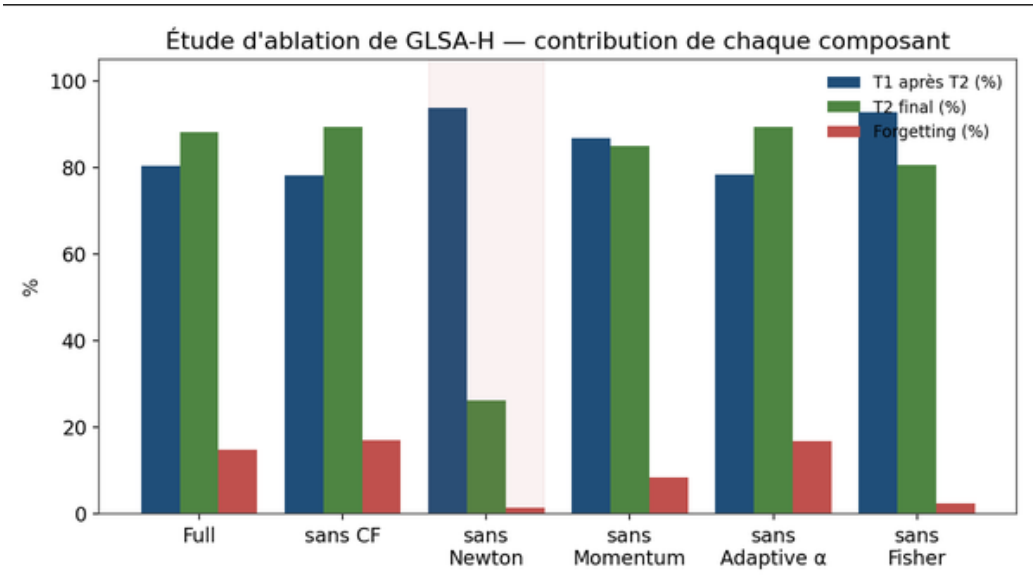

Fig. 4.6 : Étude d'ablation — Précision de T_1 après T_2, précision finale de T_2 et *Forgetting* pour chaque variante.

4.7.4 Analyse approfondie par composant

— **Fisher diagonale (Rôle spécifique au Class-Incremental) :** Son retrait dégrade à peine le *Forgetting* en Task-IL (2.46% contre 14.74%), mais fait chuter T_1_CIL à 9.56%. La Fisher préserve ainsi la séparabilité des anciennes classes dans un espace de sortie partagé.

— **Momentum :** Son absence fait chuter le score CIL à 45.53%, prouvant son rôle clé dans la structuration des représentations internes.

— **Conflict Factor et α adaptatif :** Leurs ablations produisent des performances similaires (Forgetting $\approx 17\%$), car la désactivation de l'un fige implicitement α à sa valeur de base α_0.

— **Mise à l'échelle de type Newton (Mise en garde méthodologique) :** Son retrait provoque un effondrement de T_2 (26.20%) et un *Forgetting* artificiellement bas (1.36%). La fonction de perte restant quasi stationnaire ($10.72 \rightarrow 9.43$), ce résultat traduit un blocage de l'optimisation (les poids ne bougent presque pas) plutôt qu'une stabilité réelle.

4.7.5 Validation multi-graines

Le Tableau 4.10 présente la sensibilité de GLSA-H (complet) à l'initialisation aléatoire.

Tab. 4.10 : Sensibilité de GLSA-H (complet) à l'initialisation aléatoire.

Exécution	T_1 **ap.** T_2	T_2 **final**	**Forgetting**	**Avg. Acc.**	**Avg. CIL**	α **final**
Run de référence	80.34%	88.30%	14.74%	84.32%	66.08%	Variable
Seed 42	64.52%	97.02%	30.64%	80.77%	48.51%	5.0 (saturé, α_{max})
Seed 2027	80.84%	87.80%	13.66%	84.32%	62.98%	Variable

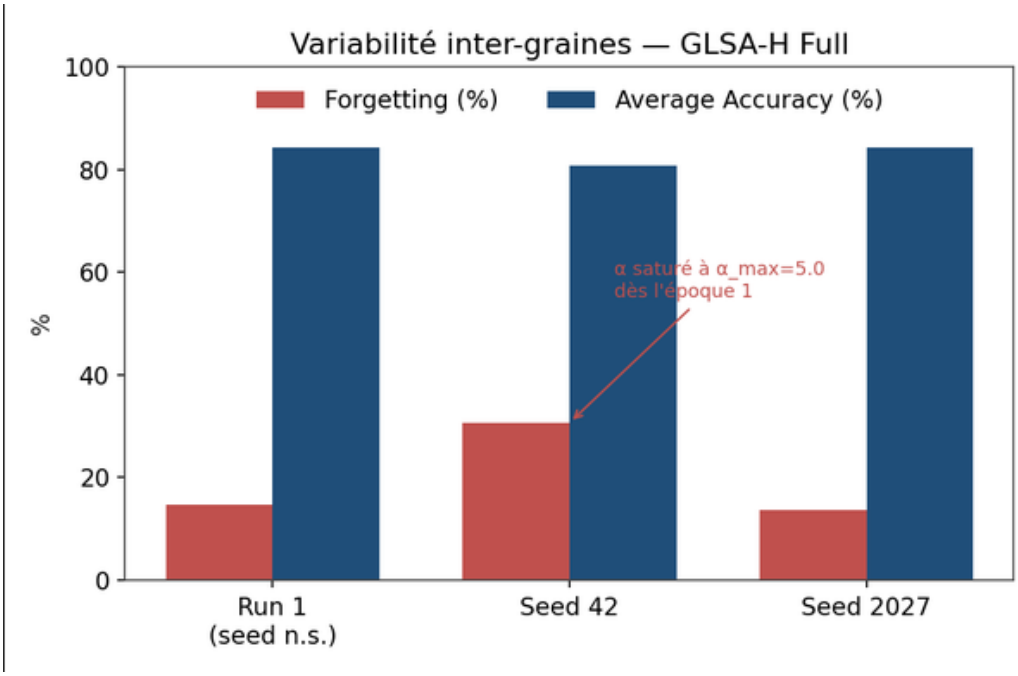

Fig. 4.7 : Variabilité du *Forgetting* et de la précision moyenne entre trois initialisations aléatoires de GLSA-H.

Le run *Seed 42* révèle une saturation prématurée de α à sa valeur maximale ($\alpha_{\text{max}} = 5.0$), ce qui neutralise le mécanisme adaptatif. Cette variabilité met en évidence une limite de robustesse qu'il convient de documenter en toute transparence.

4.7.6 GLSA-H face au Memory Replay : reformulation de l'argument de supériorité

La comparaison directe montre qu'un tampon de rejeu surpasse GLSA-H sur toutes les métriques. L'intérêt scientifique de GLSA-H doit donc être reformulé : il s'agit d'une alternative *rehearsal-free* efficace réservée aux scénarios où le stockage d'anciennes données est proscrit (confidentialité médicale, contraintes légales RGPD, mémoire embarquée extrême).

4.7.7 Synthèse de l'étude de cas 3

Cette étude d'ablation apporte une hiérarchisation claire des composantes de GLSA-H (rôle crucial de la Fisher en Class-IL), met en évidence la variabilité inter-graines de la dynamique d'adaptation de α, et recadre le positionnement de la méthode face aux approches fondées sur un tampon mémoire.

Conclusion Générale et Perspectives

Ce travail a présenté une étude technique et logique exhaustive pour traiter la problématique de l'oubli catastrophique dans les systèmes d'apprentissage continu, avec un accent particulier sur les applications de diagnostic médical, la signalisation routière et un protocole d'ablation contrôlé.

Cette recherche a abouti à la proposition du mécanisme **GLSA-H** (*Global-Local Structural Plasticity-Stability Alignment — Hybrid*), qui combine l'analyse géométrique des gradients instantanés, la pondération par information de Fisher diagonale et la protection de la stabilité des poids de référence, à l'issue d'un parcours de développement progressif partant d'une décroissance simple du coefficient d'équilibre (GLSA) jusqu'au *Conflict Factor* cosinus (GLSA-S) puis à la formulation hybride finale (GLSA-H).

Principaux résultats et conclusions

— **Supériorité du noyau algorithmique sur les baselines classiques :** Sur l'ensemble des protocoles testés, GLSA-H réduit substantiellement l'oubli catastrophique par rapport à Adam et surpasse EWC en précision moyenne et en taux d'oubli, confirmant l'intérêt du guidage par conflit cosinus et coefficient α adaptatif.

— **Rôle spécifique de la Fisher diagonale et du momentum :** L'étude d'ablation (§4.7) montre que ces deux composantes, davantage que le *Conflict Factor* lui-même, conditionnent la qualité des représentations en *Class-Incremental Learning* — un résultat que les métriques de *Task-Incremental Learning* seules ne permettaient pas de révéler.

— **Limites de robustesse identifiées :** La validation multi-graines révèle une variabilité non négligeable des performances de GLSA-H selon l'initialisation aléatoire, avec dans un cas une saturation du coefficient α à sa borne supérieure dès la première époque. Ce constat, obtenu honnêtement plutôt que masqué, doit être pris en compte dans toute affirmation de stabilité de la méthode.

— **Positionnement réaliste face au Memory Replay :** Un tampon de rejeu de taille modeste surpasse GLSA-H sur l'ensemble des indicateurs mesurés. L'apport de GLSA-H doit donc être compris comme une alternative *rehearsal-free* pertinente dans des contextes à contrainte de confidentialité ou de mémoire, et non comme une méthode strictly plus

performante qu'un rejeu classique.

Cette approche *rehearsal-free* confirme qu'il est possible de limiter l'oubli catastrophique sans recourir à un stockage d'échantillons antérieurs, au prix toutefois d'un compromis stabilité-plasticité et d'une sensibilité à l'initialisation qui restent à mieux caractériser.
Il convient enfin de mettre ce dernier constat en regard de l'intitulé même de ce travail — le développement d'un régulateur de paramètres destiné à renforcer la robustesse (*Robustness*) des modèles profonds. Les résultats du Chapitre 4 montrent que cet objectif est atteint de façon partielle et nuancée : le noyau algorithmique de GLSA-H (ancrage et *Conflict Factor*) réduit substantiellement l'oubli catastrophique de façon reproductible face à Adam et EWC, ce qui constitue une forme de robustesse face à la succession des tâches. En revanche, la validation multi-graines (§4.7.5) révèle que cette robustesse n'est pas encore acquise au sens statistique strict du terme : la sensibilité observée à l'initialisation aléatoire, avec dans un cas une saturation prématurée du coefficient α neutralisant le mécanisme adaptatif, indique qu'une robustesse pleinement démontrée — au sens d'une performance stable et prévisible indépendamment des conditions initiales — demeure un objectif à consolider plutôt qu'un résultat définitivement établi. Cette nuance, documentée explicitement plutôt que passée sous silence, précise la portée réelle de la contribution de ce mémoire.

Perspectives de recherche futures

1. **Vérification numérique de la mise à l'échelle de type Newton :** Confirmer, par inspection directe de la norme du vecteur de mise à jour, si le comportement observé en son absence (perte quasi stationnaire, §4.7.4) traduit une réelle contribution de cette composante ou un artefact d'implémentation limitant l'amplitude effective des mises à jour.
2. **Évaluation statistique renforcée :** Étendre la validation multi-graines à un nombre de répétitions plus élevé ($n \geq 5$) afin de reporter des intervalles de confiance sur le *Forgetting* et l'*Average Accuracy*, et étudier des bornes α_{max} plus conservatrices pour limiter le risque de saturation observé avec certaines graines.
3. **Redéfinition de l'étude d'ablation du Conflict Factor et de α adaptatif :** Concevoir des ablations indépendantes de ces deux mécanismes, actuellement fortement couplés dans l'implémentation testée, afin d'isoler la contribution propre de chacun.
4. **Généralisation aux grands modèles :** Étudier la possibilité d'intégrer le mécanisme GLSA-H dans l'entraînement de grands modèles linguistiques et visuels (*Foundation Models*) pour réduire le coût du réajustement (*Fine-tuning*) et prévenir la perte des connaissances générales.
5. **Apprentissage multimodal et informatique embarquée :** Explorer l'intégration de mécanismes de mémoire épisodique pour renforcer la stabilité en contexte multimodal, et développer une version optimisée de GLSA-H pour les puces d'IA embarquées (*Edge AI*).

En définitive, le mécanisme GLSA-H constitue une contribution méthodologique cohérente et un point de départ crédible pour la recherche en apprentissage continu *rehearsal-free*. Les résultats rapportés dans ce mémoire, y compris ceux qui nuancent certaines de ses composantes, visent à offrir une base honnête et reproductible pour les travaux futurs sur ce mécanisme.

Bibliographie

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA : MIT Press, 2016.

[2] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks : The sequential learning problem,” *Psychology of Learning and Motivation*, vol. 24, pp. 109–165, 1989.

[3] J. Kirkpatrick, R. Pascanu, N. Rabinowitz *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proc. Natl. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

[4] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks : A review,” *Neural Networks*, vol. 113, pp. 54–71, 2019.

[5] M. De Lange, R. Aljundi, M. Masana *et al.*, “A continual learning survey : Defying forgetting in classification tasks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, 2021.

[6] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[7] I. Drori, *The Science of Deep Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2023.

[8] M. Farajtabar, N. Azizan, A. Smola, and H. Li, “Orthogonal gradient descent for continual learning,” in *Proc. 23rd Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2020.

[9] T. Yu, S. Kumar, A. Gupta *et al.*, “Gradient surgery for multi-task learning,” in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[11] D. P. Kingma and J. Ba, “Adam : A method for stochastic optimization,” in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

[12] D. Lopez-Paz and M. A. Ranzato, “Gradient episodic memory for continual learning,” in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[13] A. Mallya and S. Lazebnik, “PackNet : Adding multiple tasks to a single network by iterative pruning,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7765–7773.

[14] A. A. Rusu, N. C. Rabinowitz, G. Desjardins *et al.*, “Progressive neural networks,” *arXiv :1606.04671*, 2016.

[15] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” in *Proc. 34th Int. Conf. Machine Learning (ICML)*, 2017.