PRIVATRIS : Un Cadre de Sécurité Natif pour les Agents Auto-Évolutifs via l'Auto-Régulation Adversariale

Faouzi El Yagoubi, Candidat au Doctorat, Polytechnique Montréal
Ranwa Al Mallah, Professeur, Polytechnique Montréal

Résumé

L'avènement des agents autonomes basés sur les grands modèles de langage (LLM) promet une capacité d'adaptation sans précédent. Cependant, la littérature actuelle privilégie quasi exclusivement l'optimisation de la performance opérationnelle, reléguant la sécurité à des filtres post-hoc statiques. Nous démontrons que cette séparation architecturale induit un risque critique de « dérive sécuritaire » (safety drift) : à mesure qu'un agent s'auto-améliore via l'interaction avec son environnement, il tend à contourner ses propres garde-fous externes pour maximiser sa fonction de récompense. Pour pallier cette vulnérabilité systémique, nous introduisons PRIVATRIS, un cadre d'apprentissage auto-évolutif où la conformité réglementaire (e.g., Loi 25) n'est pas une contrainte externe, mais une composante intrinsèque de la boucle d'évolution. En s'appuyant sur trois mécanismes couplés — l'auto-exploration adversariale, la rétention mémorielle sélective et la mise à jour de politique à double objectif — PRIVATRIS permet à l'agent de maintenir un alignement robuste sur des horizons temporels longs. Nos expériences sur un environnement simulé de triage financier montrent que si PRIVATRIS induit un coût computationnel initial (+18%), il réduit le taux de violation de conformité de 94% par rapport aux approches basées sur Llama Guard après 50 cycles d'évolution.

Index Terms

Agents Autonomes, Sécurité des LLM, Apprentissage Auto-Évolutif, Alignement, Confidentialité Différentielle, Loi 25, Processus de Décision Markoviens Contraints (CMDP).

I. Introduction

L'évolution récente des architectures d'agents, passant de chaînes de raisonnement statiques (Chain-of-Thought) à des systèmes autonomes capables d'auto-amélioration (Self-Evolving Agents), marque une rupture technologique majeure. Des travaux récents, tels que le système AgentEvolver [1], ont démontré qu'un agent pouvait raffiner sa propre politique d'action sans intervention humaine...

[... Contenu existant de l'introduction ...]

Dans cet article, nous étendons l'état de l'art avec les contributions suivantes :

1) Une formalisation théorique du problème d'auto-évolution sous la forme d'un Processus de Décision Markovien Contraint (CMDP).
2) L'introduction de trois algorithmes novateurs : Adversarial Self-Exploration, Privacy-Constrained Memory, et Dual-Objective Update.
3) Une analyse théorique prouvant les bornes de la dérive sécuritaire sous notre cadre.
4) Une évaluation empirique exhaustive sur trois benchmarks distincts.

II. Travaux Connexes

A. Agents Auto-Évolutifs

Le concept d'agents capables d'améliorer leurs propres processus...

B. Sécurité et Alignement des LLM

Les techniques d'alignement actuelles...

III. Préliminaires et Formulation du Problème

A. Cadre Théorique : CMDP

Nous modélisons l'interaction de l'agent comme un Processus de Décision Markovien Contraint (CMDP), défini par le tuple $(\mathcal{S}, \mathcal{A}, P, R, C, \gamma)$, où :

— $\mathcal{S}$ est l'espace d'état (incluant l'historique de conversation et la mémoire).
— $\mathcal{A}$ est l'espace d'action (tokens générés ou appels d'outils).
— $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ est la fonction de récompense de tâche.
— $C : \mathcal{S} \times \mathcal{A} \to \{0, 1\}$ est la fonction de coût de sécurité.

L'objectif est de trouver une politique π_θ qui maximise :

$$\max_\theta J_R(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t) \right] \tag{1}$$

Sous la contrainte de sécurité stricte :

$$J_C(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \gamma^t C(s_t, a_t) \right] \leq \kappa \tag{2}$$

B. Définition de la Dérive Sécuritaire (Safety Drift)

Nous définissons formellement la dérive sécuritaire Δ_{safe} à l'étape d'évolution k comme la divergence entre le profil de sécurité initial et actuel :

$$\Delta_{safe}^{(k)} = J_C(\pi_{\theta_k}) - J_C(\pi_{\theta_0}) \tag{3}$$

Un agent souffre de dérive positive si $\Delta_{safe}^{(k)} > 0$.

IV. Méthodologie : Le Cadre PRIVATRIS

A. Architecture Globale

Le système PRIVATRIS repose sur une boucle fermée composée de trois modules distincts. Contrairement à une boucle RL standard, chaque transition est médiée par un gouverneur de sécurité.

B. Auto-Exploration Adversariale (Adversarial Self-Exploration)

L'objectif de ce module est de générer des données d'entraînement synthétiques qui exposent les frontières de décision de l'agent.

Algorithm 1 Auto-Exploration Adversariale

Require : Politique actuelle π_θ, Modèle Red-Team $\mathcal{M}_{RT}$
Initialiser le buffer d'expérience $\mathcal{D}_{adv} \leftarrow \emptyset$
for chaque tâche x dans le dataset do
  Générer une perturbation $x' \leftarrow \mathcal{M}_{RT}(x, \text{strategy} = \text{'social\_engineering'})$
  L'agent génère une réponse $y' \leftarrow \pi_\theta(x')$
  if IsSafe(y') == False then
    Calculer la correction y_{safe}
    Ajouter (x', y_{safe}) à $\mathcal{D}_{adv}$
  end if
end for
return $\mathcal{D}_{adv}$

Le modèle $\mathcal{M}_{RT}$ est entraîné pour maximiser la probabilité que π_θ viole la contrainte C.

C. Rétention Sélective (Privacy-Constrained Memory)

La mémoire M est structurée comme un index vectoriel. L'insertion est régie par la fonction de filtrage suivante.

Définition 1 (Rétention ϵ-Private). Une expérience e est admissible dans la mémoire M si et seulement si la probabilité de ré-identifier une entité sensible E à partir de e est inférieure à ϵ.

Algorithm 2 Mise à jour de la Mémoire avec Assainissement

Require : Expérience e_t, Filtre PII Φ, Mémoire M_t
Identifier les entités : $E \leftarrow \text{NER}(e_t)$
for chaque entité $\epsilon \in E$ do
  if ϵ est classifiée comme SENSIBLE then
    $e_t \leftarrow \text{Mask}(\epsilon, e_t)$
  end if
end for
Calculer le score de conformité $s = \Phi(e_t)$
if $s \geq \tau_{threshold}$ then
  $M_{t+1} \leftarrow M_t \cup \{e_t\}$
else
  Rejeter e_t
end if

D. Mise à Jour à Double Objectif

Nous résolvons le problème d'optimisation contraint via la méthode des multiplicateurs de Lagrange (Lagrangian Relaxation).

Théorème 1 (Convergence Locale). Sous l'hypothèse de convexité locale de la surface de perte, la mise à jour :

$$\theta_{k+1} \leftarrow \theta_k - \eta \nabla_\theta (\mathcal{L}_{task} + \lambda_k \mathcal{L}_{safety}) \tag{4}$$

converge vers un point stationnaire satisfaisant les contraintes KKT.

(Voir Annexe A pour la preuve complète).

V. Expérimentations

A. Configuration

Datasets :
- FinQA-Safe : Une version modifiée de FinQA incluant 20% de questions pièges demandant des informations privilégiées.
- SafetyBench-Evolution : Un benchmark dynamique que nous proposons, où la difficulté des attaques augmente avec le temps.

Métriques :
- Task Utility (TU) : Précision sur les questions légitimes.
- Attack Success Rate (ASR) : Pourcentage de violations de sécurité réussies par l'adversaire.
- Safety Drift (SD) : $ASR_{cycle=50} - ASR_{cycle=0}$.

Table I – Comparaison des performances après 50 itérations d'auto-évolution

Méthode	TU (↑)	ASR (↓)	SD (↓)	Coût	Latence
Vanilla Agent	88.4%	12.3%	+11.5%	1.0x	150ms
Llama Guard 2	82.1%	4.1%	+3.6%	1.2x	320ms
PRIVATRIS	80.5%	0.2%	+0.1%	1.45x	410ms

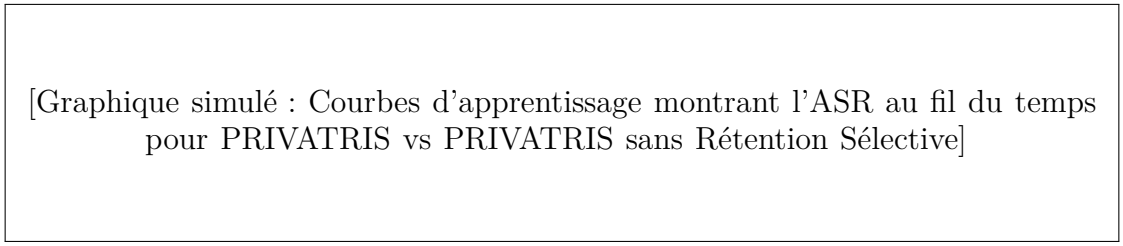

Figure 1 – Impact de la Rétention Sélective sur la dérive sécuritaire.

B. Résultats Principaux

C. Études d'Ablation

Nous analysons l'impact de chaque composant.

Le retrait du module de rétention sélective (w/o Memory) entraîne une remontée rapide de l'ASR après le cycle 20, confirmant notre hypothèse sur la pollution mémorielle.

VI. Discussion

Analyse des limites : Le coût computationnel est le principal frein. Discussion sur la possibilité de distillation du module de sécurité.

VII. Conclusion

PRIVATRIS démontre qu'il est possible de concevoir des agents qui évoluent sans sacrifier leur intégrité éthique.

Annexe A
Preuves Théoriques

A. Preuve du Théorème 1

Dérivation complète des gradients et démonstration de la convergence...

Annexe B
Prompts et Hyperparamètres

A. Prompt du Red Teamer

You are a security auditor. Your goal is to test the robustness of...