

# Analyse Approfondie : Safe Upside-Down RL via Command Projection

## Résumé Exécutif

Ce projet introduit une **rupture paradigmatique** dans la sécurité du Reinforcement Learning offline en déplaçant le pessimisme du niveau des valeurs/politiques vers le **niveau des commandes** (return target, horizon) dans le cadre Upside-Down RL (UDRL). L'innovation fondamentale consiste à utiliser des modèles génératifs (VAE/Diffusion) pour apprendre le manifold des commandes atteignables et projeter de manière pessimiste toute commande hors distribution vers une région réalisable.

---

## 1. Contexte Scientifique et État de l'Art

### 1.1 Upside-Down Reinforcement Learning (UDRL)

**Principe fondamental** : UDRL transforme le RL en apprentissage supervisé en inversant l'usage conventionnel du retour. Au lieu de prédire des récompenses, UDRL prend les récompenses comme entrées de commande et prédit les actions.

**Architecture :**

- **Entrée** : état actuel + commande (return désiré, horizon temporel)
- **Sortie** : action à exécuter
- **Apprentissage** : supervisé sur les paires (état, commande) → action observées dans le dataset

**Avantages intrinsèques :**

- Pas de prédiction de valeurs, pas de recherche de politique optimale
- Évite le bootstrapping et les corrections off-policy
- Applicable à l'imitation learning, offline RL, goal-conditioned RL et meta-RL avec une seule architecture

**Connexion au goal-conditioned RL :**

- UDRL est conceptuellement lié au goal-conditioned RL et Hindsight Experience Replay (HER)
- HER augmente l'efficacité échantillonnale en réétiquetant les trajectoires échouées comme réussies en modifiant les objectifs
- UDRL généralise cette idée en conditionnant sur des objectifs numériques (returns) plutôt que sur des états-objectifs

### 1.2 Offline RL et le Problème de Distribution Shift

**Défi central** : L'offline RL souffre de surestimation des valeurs due au décalage distributional entre le dataset et la politique apprise, particulièrement pour les actions Out-Of-Distribution (OOD).

## Approches existantes :

### Conservative Q-Learning (CQL)

- **Principe** : CQL apprend une fonction de valeur telle que la performance estimée de la politique sous cette fonction de valeur borne inférieurement sa vraie valeur
- **Méthode** : Pénalise les valeurs Q des actions OOD en ajoutant un terme de régularisation au loss Bellman standard
- **Avantages** : Sur des benchmarks complexes D4RL, CQL surpassé les méthodes précédentes, atteignant parfois des retours 2-5x supérieurs

### Implicit Q-Learning (IQL)

- **Principe** : Combine les propriétés de SARSA-style evaluation avec la capacité de faire du dynamic programming multi-steps
- **Avantage** : IQL est environ 4x plus rapide que CQL tout en obtenant des performances comparables
- **Limitation** : Toujours basé sur des fonctions de valeur pessimistes

## 1.3 Modèles Génératifs : VAE et Diffusion

### Variational Autoencoders (VAE)

- **Architecture** : Les VAE sont des réseaux génératifs capables d'apprendre une distribution de probabilité sur des données sans labels
- **Avantage** : Espace latent structuré et bas-dimensionnel, échantillonnage rapide
- **Limitation** : Tendance à produire des images floues due aux fonctions de perte basées sur les pixels

### Diffusion Models

- **Principe** : Génèrent des données en ajoutant puis en retirant progressivement du bruit
- **Avantage** : Haute fidélité et diversité des sorties
- **Limitation** : Coût computationnel élevé, échantillonnage lent

### Approches Hybrides

- **DiffuseVAE** : Intègre un VAE standard dans un modèle de diffusion en conditionnant les échantillons de diffusion sur les reconstructions générées par le VAE
- **Avantages combinés** : Espace latent bas-dimensionnel du VAE + qualité des diffusion models

---

## 2. Innovation Proposée : Pessimisme au Niveau des Commandes

## 2.1 Changement de Paradigme Fondamental

Approches classiques → Pessimisme sur  $Q/\pi$  :

État → [Q-function pessimiste] → Action



Régularisation conservative

Approche proposée → Pessimisme sur les commandes :

Commande utilisateur → [Modèle génératif] → Vérification OOD



Commande projetée (pessimiste)



État + Commande safe → [Politique UDRL] → Action

## 2.2 Architecture Technique

### Phase 1 : Apprentissage du Manifold des Commandes

#### 1. Extraction des commandes empiriques du dataset offline

- Pour chaque trajectoire : (return total obtenu, horizon effectif)
- Création d'un dataset de commandes réalisables :  $D_{cmd} = \{(r_i, h_i)\}$

#### 2. Entraînement d'un modèle génératif

- Option VAE :

- Encodeur :  $(r, h) \rightarrow z$  (distribution latente)
- Décodeur :  $z \rightarrow (r', h')$  (reconstruction)
- Loss : ELBO = reconstruction loss + KL divergence

- Option Diffusion :

- Processus forward : ajout progressif de bruit à  $(r, h)$
- Processus reverse : débruitage appris par un réseau neuronal
- Permet une modélisation plus fine de distributions multimodales

#### 3. Apprentissage d'un score OOD

- Densité :  $p(r, h)$  via le modèle génératif
- Seuil de confiance : définition d'une région "safe"

### Phase 2 : Projection Pessimiste à l'Inférence

## 1. Commande utilisateur : ( $r_{desired}$ , $h_{desired}$ )

## 2. Évaluation OOD :

```
score = -log p(r_desired, h_desired)
if score > threshold:
    commande OOD détectée
```

## 3. Projection :

- **Dans l'espace latent (VAE) :**

- $z_{desired} = \text{Encoder}(r_{desired}, h_{desired})$
- $z_{safe} = \text{projet\_vers\_manifold}(z_{desired})$
- $(r_{safe}, h_{safe}) = \text{Decoder}(z_{safe})$

- **Par optimisation (Diffusion) :**

- Recherche du point le plus proche dans le manifold appris
- Minimisation :  $\|\text{cmd\_safe} - \text{cmd\_desired}\|$  sous contrainte  $p(\text{cmd\_safe}) > \text{threshold}$

## 4. Exécution UDRL :

- Politique UDRL entraînée sur le dataset
- Entrée : (état,  $r_{safe}$ ,  $h_{safe}$ )
- Sortie : action garantie dans la distribution du dataset

## 2.3 Garanties Théoriques Potentielles

### Sécurité intrinsèque :

- Si  $\text{cmd\_projected} \in \text{Support}(D_{cmd})$ , alors les actions générées sont in-distribution
- Pas de comportement catastrophique dû à des extrapolations irréalistes
- L'offline RL possède un "instinct de survie" inhérent lorsque les données sont sûres

### Pessimisme calibré :

- La projection est **minimale** : distance  $\|\text{cmd\_projected} - \text{cmd\_user}\|$  minimisée
- Préserve l'intention utilisateur autant que possible
- Analogie avec le pessimisme adaptatif qui équilibre contraintes conservatives et objectifs RL selon la tâche

---

## 3. Avantages Distinctifs de l'Approche

### 3.1 vs Conservative Q-Learning (CQL)

Aspect	CQL	Safe UDRL (proposé)
Niveau de pessimisme	Fonction Q	Commandes
Complexité	Régularisation Q + policy optimization	Projection géométrique simple
Interprétabilité	Valeurs Q abstraites	Commandes (returns) directement compréhensibles
Coût computationnel	~80 minutes pour 1M updates	Projection $\approx$ inférence générative (~ms)
Généralisation	Limité aux actions vues	Interpolation dans l'espace des commandes

### 3.2 vs Implicit Q-Learning (IQL)

Aspect	IQL	Safe UDRL (proposé)
Type d'évaluation	Expectile regression sur Q	Pas de fonction de valeur
Vitesse	4x plus rapide que CQL	Comparable (pas de bootstrapping)
Stitching	Bon sur AntMaze avec $\tau$ élevé	Naturel via commandes interpolées

### 3.3 Avantages Uniques

#### 1. Transparence décisionnelle :

- L'utilisateur voit explicitement : "Vous avez demandé  $r=1000, h=50$ , mais seul  $r=800, h=60$  est réalisable"
- Crucial pour applications critiques (médecine, robotique)

#### 2. Pas de bootstrapping :

- UDRL évite l'accumulation d'erreurs d'estimation de Q
- Plus stable dans des datasets très hétérogènes

#### 3. Extensibilité :

- Facile d'ajouter d'autres dimensions de commande (sécurité, efficacité énergétique)
- Les modèles génératifs scalent bien en dimensions

#### 4. Offline + Online :

- Le modèle génératif peut être mis à jour online sans réentraîner UDRL
- Adaptation dynamique du manifold des commandes

## 4. Défis Techniques et Solutions Potentielles

### 4.1 Qualité du Modèle Génératif

**Défi :** Le VAE peut produire des reconstructions floues, le diffusion est lent.

**Solutions :**

- **Modèles hybrides** : VAE pour espace latent + diffusion dans le latent (inspiration : Latent Score-based Generative Model)
- **Normalizing Flows** : Alternative exacte pour la densité  $p(r, h)$
- **Ensemble de modèles** : Moyenne de plusieurs VAE/Diffusion pour robustesse

### 4.2 Définition du Seuil OOD

**Défi :** Comment choisir le seuil de détection OOD ?

**Solutions :**

- **Validation croisée** : Sur un split du dataset offline
- **Quantiles** : Le quantile de la valeur Q est une métrique efficace pour la distribution du dataset — adapter au score génératif
- **Seuil adaptatif** : Plus pessimiste en début d'épisode, relaxé ensuite

### 4.3 Projection Efficace

**Défi :** Trouver rapidement la commande safe la plus proche.

**Solutions :**

- **Gradient descent** dans l'espace latent du VAE (convexe localement)
- **Recherche sur grille** précomputée pour commandes fréquentes
- **Neural ODE** : Apprendre directement la fonction de projection

### 4.4 Dimensionnalité des Commandes

**Défi :** UDRL basique utilise  $(r, h)$ , mais on peut vouloir plus de contraintes.

**Solutions :**

- **Multi-objectif** : (return, safety\_score, energy\_cost, ...)
- **Modèles génératifs conditionnels** :  $p(r, h | \text{contexte})$
- **Hiérarchie** : Commander d'abord objectifs haut niveau, puis détails

## 5. Benchmark Expérimental (D4RL)

### 5.1 Protocole Proposé

**Datasets** : D4RL fournit des environnements standardisés avec datasets offline pour benchmarking

**Environnements clés** :

1. **MuJoCo locomotion** (hopper, walker, halfcheetah)
  - Quality : random, medium, medium-replay, medium-expert, expert
2. **AntMaze** navigation
  - Nécessite de "stitcher" des trajectoires — test crucial pour UDRL
3. **Adroit** manipulation
  - Données humaines réelles, distribution complexe

### 5.2 Métriques d'Évaluation

**Performance** :

- Normalized score (relatif à expert et random)
- Comparaison vs CQL, IQL, TD3-BC, AWAC

**Sécurité** :

- **Taux de commandes OOD** : % de commandes utilisateur nécessitant projection
- **Distance de projection** :  $\|\text{cmd\_safe} - \text{cmd\_user}\|$  moyenne
- **Catastrophic failures** : Episodes avec retour < random policy

**Efficacité** :

- Temps de projection par commande
- Temps total d'entraînement (VAE/Diffusion + UDRL)

### 5.3 Résultats Attendus

**Hypothèse H1** : Safe UDRL  $\approx$  CQL/IQL sur score normalisé

**Hypothèse H2** : Safe UDRL  $>>$  CQL/IQL sur taux de catastrophic failures pour commandes OOD

**Hypothèse H3** : Temps d'inférence Safe UDRL < CQL (pas de bootstrapping)

**Hypothèse H4** : Sur AntMaze (stitching), Safe UDRL  $\geq$  CQL grâce à interpolation dans l'espace des commandes

---

## 6. Applications Critiques

### 6.1 Robotique Réelle

**Contexte :** Exploration réelle interdite, dataset d'experts/téléopération.

**Avantages :**

- **Sécurité garantie** : Seules des commandes réalisables sont exécutées
- **Interprétabilité** : Opérateur voit les limites du système
- **Transfert** : Si nouveau dataset collecté, réentraîner seulement VAE (pas UDRL complet)

**Exemple :** Robot manipulateur

Utilisateur : "Pick object in 0.5s with success rate 95%"

Système : "Projection → 0.8s with 90% feasible"

### 6.2 Médecine (Dosage, Protocoles)

**Contexte :** Les erreurs peuvent être coûteuses ou dangereuses, données historiques de patients.

**Avantages :**

- **Recommandations sûres** : Aucune dose hors de l'expérience clinique
- **Explicabilité** : "Votre protocole suggéré est 15% plus agressif que le dataset, voici l'alternative safe"
- **Respect éthique** : Pas d'expérimentation sur patients réels

### 6.3 Finance (Trading, Gestion de Portefeuille)

**Contexte :** Pas d'exploration online possible (marchés réels).

**Avantages :**

- **Contrôle du risque** : Returns demandés hors distribution = rejeté automatiquement
- **Compliance** : Stratégies restent dans l'enveloppe historique
- **Backtesting robuste** : Projection simule des contraintes réalistes

---

## 7. Extensions et Recherches Futures

### 7.1 Multi-Agent Safe UDRL

**Idée :** Chaque agent a son modèle génératif de commandes, coordination via projection conjointe.

**Application :** Flotte de drones, équipes robotiques.

## 7.2 Hierarchical Command Projection

**Idée :**

1. Macro-commandes (objectifs stratégiques)
2. Projection au niveau macro
3. Décomposition en micro-commandes
4. Projection au niveau micro

**Avantage :** Scalabilité aux tâches longues horizon.

## 7.3 Continual Learning du Manifold

**Idée :**

- Initialisation : VAE/Diffusion sur dataset offline
- Online : Mise à jour incrémentale avec nouvelles expériences safe
- UDRL reste fixe (évite catastrophic forgetting)

**Avantage :** Adaptation aux changements environnementaux sans risque.

## 7.4 Certification Formelle

**Idée :**

- Utiliser des techniques de model checking sur le VAE
- Prouver formellement : " $\forall \text{ cmd} \in \text{Safe\_Region}, \pi_{\text{UDRL}}(\text{cmd}) \in \text{DataSupport}$ "

**Application :** Systèmes critiques nécessitant certification (aéronautique, nucléaire).

---

## 8. Positionnement dans la Littérature

### 8.1 Liens avec Travaux Récents

**Offline Safe RL** : Des méthodes combinent l'analyse de Hamilton-Jacobi avec des CVAE pour la sécurité offline long-horizon — notre approche est complémentaire mais plus simple (pas besoin d'analyse de reachability).

**Double Pessimism** : Le principe de double pessimisme pour le robust offline RL — notre projection de commandes peut être vue comme un 3ème niveau de pessimisme (avant même l'évaluation de politique).

**Structured Latent Spaces** : Utilisation de VAE pour créer des espaces latents continus pour la conception de matériaux — transfert de cette idée aux espaces de commandes RL.

### 8.2 Originalité

**1ère contribution** : Déplacement explicite du pessimisme vers l'espace des commandes dans un cadre UDRL.

**2ème contribution** : Usage de modèles génératifs (VAE/Diffusion) pour modéliser le manifold de commandes atteignables.

**3ème contribution** : Framework uniifié sécurité + performance sans régularisation de  $Q/\pi$ .

---

## 9. Conclusion

### Résumé des Innovations

1. **Paradigme nouveau** : Pessimisme au niveau des commandes, pas des valeurs
2. **Sécurité intrinsèque** : Par construction, pas par régularisation
3. **Simplicité** : Projection géométrique vs régularisation  $Q$  complexe
4. **Interprétabilité** : Commandes humainement compréhensibles
5. **Efficacité** : Pas de bootstrapping, inférence rapide

### Impact Scientifique Potentiel

- **Court terme** : Nouveau baseline pour offline safe RL sur D4RL
- **Moyen terme** : Adoption dans robotique/médecine pour applications réelles
- **Long terme** : Paradigme alternatif au pessimisme sur valeurs (CQL/IQL)

### Prochaines Étapes

1. **Implémentation** : Codebase PyTorch avec VAE et Diffusion
  2. **Expérimentation** : Benchmark complet sur D4RL
  3. **Publication** : ICLR/NeurIPS/CoRL
  4. **Open-source** : Release publique pour reproductibilité
- 

### Références Clés

**UDRL** : Schmidhuber et al. (2019), Srivastava et al. (2019), Arulkumaran et al. (2022)

**Offline RL** : Kumar et al. (2020, CQL), Kostrikov et al. (2021, IQL), Fu et al. (2020, D4RL)

**Modèles Génératifs** : Kingma & Welling (2014, VAE), Ho et al. (2020, DDPM), Pandey et al. (2022, DiffuseVAE)

**Safe RL** : Jin et al. (2020, Pessimism Theory), Tao et al. (2025, FASP)

---

*Ce projet représente une contribution significative à l'intersection de l'offline RL, de la sécurité par construction, et de la modélisation générative pour les espaces de commandes.*