

Apprentissage par Renforcement pour la Manipulation Dextère

« Learning Dexterous In-Hand Manipulation » (OpenAI)

Présenté par:

BAZIÉ Dureel

BICABA Hawahoun Pauline

COULIBALY Cheick Ahmed

KOALA Valentin

TRAORÉ Soungalo

Sommaire

1. Problématique et Contexte
2. Méthode de l'article
3. Architecture Technique et Méthodologie
4. Résultats Expérimentaux et Analyse
5. Conclusion et Perspectives

Problématique et Contexte

Description du problème

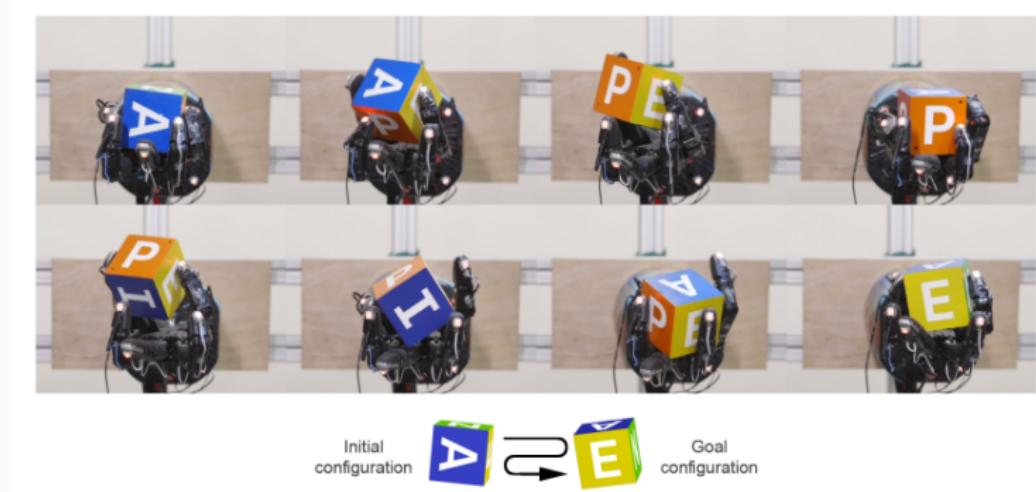


Figure 1: une main humanoïde à cinq doigts entraînée par apprentissage par renforcement manipulant un bloc d'une configuration initiale à une configuration cible en utilisant la vision comme capteur.

Main agile ShadowRobot



Figure 2: ShadowRobot

- elle est contrôlée par 20 moteurs à courant continu, actionnant sur des tendons (20), elle possède 24 DoF.

Suivi visuel PhaseSpace

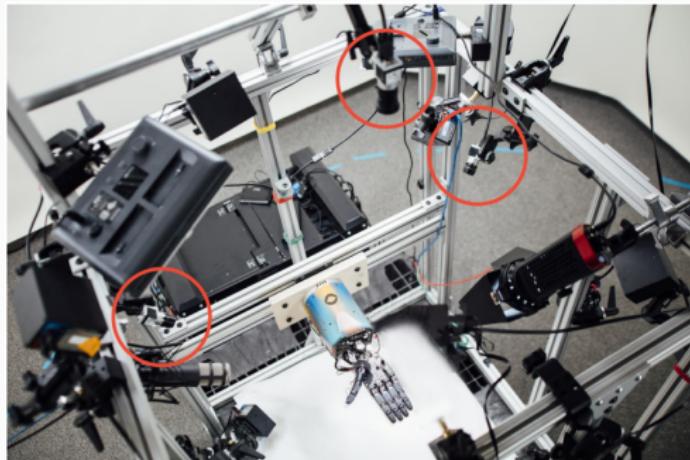


Figure 3: Estimation de l'état basée sur la vision

- Des marqueurs LED fixés sur la main émettent chacun sa propre lumière, qui sera captée par des caméras RVB spéciales (à infrarouges), qui identifie chaque marqueur comme un point (x, y, z), et une fois la position de tous les points connu on obtient un nuage de points 3D.

Méthode de l'article

Méthode

- **Approche** : apprentissage par renforcement profond (Deep RL).
- **Objectif** : combler le fossé simulation–réalité.
- **Technique clé** : randomisation en simulation :
 - Variation des paramètres : masse du cube, gravité, éclairage, couleurs.
 - Permet d'éviter une politique adaptée uniquement à une *simulation parfaite*.
 - Rend la stratégie robuste aux imprécisions et variations du monde réel.

Environnement simulé

- **Agent** : est une politique neuronale (réseau de neurones) qui prend en entrée des observations (vision et capteurs) et qui produit des actions (commandes des moteurs de la main). Il est entraîné pour maximiser une récompense liée à l'alignement du cube avec l'orientation cible
- **Simulation** : GYM, Goal-based robotics task, simulation physique réalisé avec MuJoCo (Multi-Joint Dynamics with Contact)

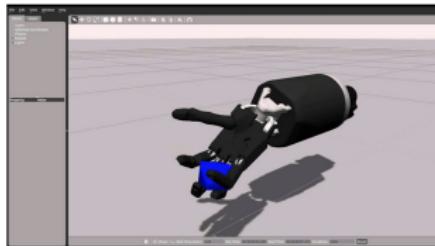


Figure 4: OpenAI Gym robotics environments

Environnement simulé

Enjeux techniques de la manipulation Dextère

- Complexité dimensionnelle (espace d'état/action)
- Coût prohibitif de l'apprentissage sur robot physique
- Modélisation imprécise de la physique réelle



Simulate Dexterous Hand

Figure 5: Problème du transfert simulation→réel

Randomisation



Figure 6: Simulations avec différentes apparences visuelles aléatoires. Les lignes correspondent aux rendus de la même caméra, et les colonnes correspondent aux rendus de trois caméras distinctes, alimentées simultanément dans le réseau neuronal.

Environnement simulé

Pour simuler toutes ces variantes possibles de l'environnement, ils ont construit un système qui exécute les processus de formation dans le cloud sur des milliers de machines = Rapid. Il a déjà été utilisé pour résoudre des jeux vidéo complexes.

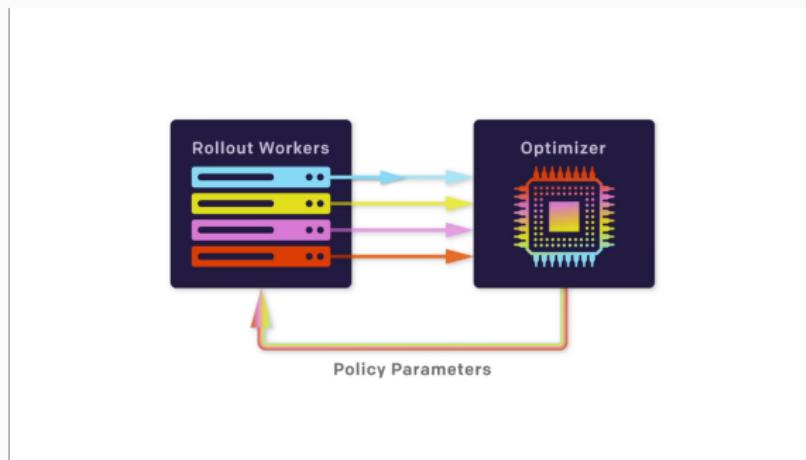


Figure 7: Architecture Rapid

Environnement simulé

- **États** : l'état du système est à 60 dimensions et comprend les angles et les vitesses de toutes les articulations du robot, ainsi que la position, la rotation et les vitesses (linéaires et angulaires) de l'objet. Les états initiaux sont échantillonnés en plaçant l'objet sur la paume de la main dans une orientation aléatoire et en appliquant des actions aléatoires pendant 100 étapes (l'essai est annulé si l'objet est lâché entre-temps).
- **Objectif** : est l'orientation souhaitée de l'objet représenté par un quaternion. Un nouvel objectif est généré une fois l'objectif actuel atteint avec une tolérance de 0,4 rad ($0.4 \text{ radian} \approx 23^\circ$).
- **Actions** : les actions correspondent aux angles souhaités des articulations de la main.

Environnement simulé

- **Récompenses** : la récompense donnée au pas de temps t est $r_t = d_t - d_{t+1}$, où d_t et d_{t+1} sont les angles de rotation entre les orientations souhaitée et actuelle de l'objet, respectivement avant et après la transition. Une récompense supplémentaire de 5 est attribuée lorsqu'un objectif est atteint avec une tolérance de 0,4 rad (c'est-à-dire $d_{t+1} < 0,4$) et une pénalité de -20 lorsque l'objet est lâché.
- **Durée** : chaque étape de l'environnement correspond à 80 ms de temps réel et se compose de 10 étapes MuJoCo consécutives, chacune correspondant à 8 ms. L'épisode se termine lorsque la stratégie atteint 50 objectifs consécutifs, qu'elle ne parvient pas à atteindre l'objectif actuel dans les 8 secondes de la simulation ou que l'objet est abandonné.

Architecture Technique et Méthodologie

Architecture RL : PPO avec Mémoire

PPO (Proximal Policy Optimization)

- Stabilité d'entraînement
- Bonnes performances empiriques
- Scalabilité distributionnelle

Fonction de Récompense

$$r_t = \underbrace{d_t - d_{t+1}}_{\text{Progrès}} + \underbrace{5 \cdot \mathbf{1}_{d_{t+1} < 0.4}}_{\text{Réussite}} - \underbrace{20 \cdot \mathbf{1}_{\text{échec}}}_{\text{Pénalité}}$$

où d_t est l'angle de rotation jusqu'à l'objectif

Architecture RL : PPO avec Mémoire

Architecture Réseau

- Entrée : 61 dimensions d'observation
- Couche cachée : 1024 unités (ReLU)
- **LSTM : 512 unités** (mémoire)
- Sortie : 20 actions discrètes (11 bins)



Figure 8: Architecture du réseau de politique

Randomisation de Domaine : Détails Techniques

Paramètres Physiques

Paramètre	Plage
Dimensions objet	$\times [0.95, 1.05]$
Masses	$\times [0.5, 1.5]$
Frottements	$\times [0.7, 1.3]$
Amortissement	$\log-[0.3, 3.0]$
Gains actionneurs	$\log-[0.75, 1.5]$
Limites articulaires	$\pm 0.15 \text{ rad}$
Gravité	$\pm 0.4 \text{ m/s}^2$

Bruit et Délais

Type	Valeur
Bruit observations	Gaussien
Corrélé (épisode)	$0(1\text{-}5\text{mm})$
Non-corrélé (step)	$0(1\text{-}2\text{mm})$
Délais action	80ms (50%)
Backlash	Modèle phys.
Forces aléatoires	0.1-10%

Randomisation Visuelle

Aspect	Variation
Positions caméras	$\pm 1.5\text{mm}$
Rotation caméras	$0 - 3^\circ$
FOV caméras	$\pm 1^\circ$
Textures objets	RGB $\pm 15\%$
Matériaux	Métal/brillance
Éclairage	4-6 sources
Contraste image	50-150%
Bruit pixel	$\pm 10\%$

Impact de la Randomisation

Sans randomisation : **aucun transfert** (médiane = 0 rotations)

Avec randomisation : **13 rotations réussies** (médiane)

Infrastructure d'Entraînement à Grande Échelle

Architecture Distribuée

Workers (Génération d'expérience) :

- 384 machines (6144 cœurs CPU)
- Génération de trajectoires
- Randomisation des environnements

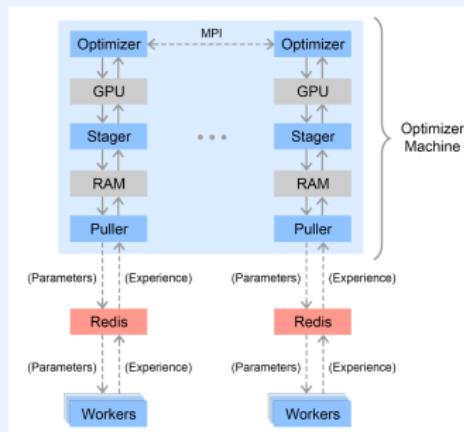


Figure 9: Architecture distribuée

Infrastructure d'Entraînement à Grande Échelle

Optimiseur (Apprentissage) :

- 1 machine avec 8 GPUs V100
- Mise à jour des paramètres
- Descente de gradient stochastique

Débit : 2 années d'expérience simulée par heure

Infrastructure d'Entraînement à Grande Échelle

Hyperparamètres PPO

Taux d'apprentissage	3×10^{-4}
Discount (γ)	0.998
Paramètre GAE (λ)	0.95
Clipping (ϵ)	0.2
Taille de batch	800k transitions
Régularisation entropie	0.01

Infrastructure d'Entraînement à Grande Échelle

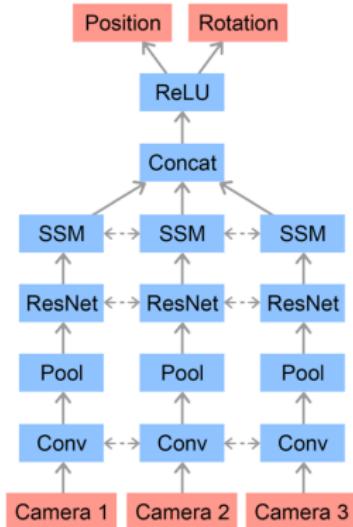


Figure 6: Vision network architecture. Camera images are passed through a convolutional feature stack that consists of two convolutional layers, max-pooling, 4 ResNet blocks [24], and spatial softmax (SSM) [19] layers with shared weights between the feature stacks for each camera. The resulting representations are flattened, concatenated, and fed to a fully connected network. All layers use ReLU [41] activation function. Linear outputs from the last layer form the estimates of the position and orientation of the object.

Figure 10: Réseau de vision

Résultats Expérimentaux et Analyse

Performance : résultats quantitatifs

Expériences sur le robot physique

- Échec fréquent : lâcher l'objet lors de l'inclinaison du poignet (*wrist pitch joint*).
- L'articulation verticale casse souvent car elle supporte la charge principale.
- Solution testée : verrouiller le poignet \Rightarrow meilleure robustesse et transfert.
- Résultat : manipulation plus stable et délibérée.
- Autres échecs : lâcher trop tôt ou objet coincé dans l'environnement.

Bloquer le poignet améliore la stabilité et la robustesse.

Performance : résultats quantitatifs

Rotations Consécutives Réussies (médiane)

Configuration	Simulation		Monde Réel	
	Médiane	Moyenne	Médiane	Moyenne
Bloc (état PhaseSpace)	50	43.4	13	18.8
Bloc (vision seulement)	33	30.0	11.5	15.2
Bloc (poignet bloqué)	50	44.2	28.5	26.4
Prisme octogonal	30	29.0	5	7.8

Performance : résultats quantitatifs

Succès Principaux	Limites Identifiées
<ul style="list-style-type: none">Transfert réussi simulation→réelPerformance visuelle proche de capteursRobustesse aux variations dynamiques	<ul style="list-style-type: none">Gap de performance simulationréelÉchec sur objets sphériquesSensibilité aux pannes matérielles

Impact de la Mémoire	Analyse de la Mémoire
<p>LSTM vs Feed-Forward :</p> <ul style="list-style-type: none">LSTM policy : 13 rotations (médiane)FF policy : 3 rotations (médiane)Mémoire essentielle pour l'adaptation	<p>État caché du LSTM prédit les propriétés de l'environnement :</p> <ul style="list-style-type: none">Taille de l'objet (80% de précision)Paramètres dynamiquesPropriétés physiques

Conclusion et Perspectives

Contributions et implications

Contributions Principales

1. Premier transfert réussi de politiques dextères complexes
2. Nouveau paradigme sim-to-real par randomisation massive
3. Architecture scalable avec mémoire et observations limitées
4. Validation expérimentale extensive

Implications pour la Recherche

- Preuve que le RL profond peut résoudre des problèmes robotiques complexes
- Importance cruciale de la randomisation pour le transfert
- Mémoire essentielle pour l'adaptation en ligne
- Validation de l'apprentissage purement simulé

Contributions et implications

Limites

- Calcul massif requis
- Calibration complexe
- Robustesse incomplète

Perspectives pour le Burkina Faso

Défis de RéPLICATION	Stratégies Adaptatives
Contraintes Techniques : <ul style="list-style-type: none">• Coût matériel élevé (robot, calcul)• Expertise spécialisée requise• Maintenance complexe	Approche Progressive : <ol style="list-style-type: none">1. Simulation seulement (formation)2. Robots simples (mains 3-4 doigts)3. Cloud computing pour l'entraînement4. Collaboration internationale
Contraintes Infrastructurelles : <ul style="list-style-type: none">• Accès limité au calcul haute performance• Connectivité internet insuffisante• Ressources financières limitées	Applications Prioritaires : <ul style="list-style-type: none">• Agriculture de précision• Artisanat assisté• Téléopération médicale

Merci pour votre aimable attention Démonstration

vidéo : <https://youtu.be/jwSbzNHGfLM>

Références

-  OpenAI et al.
Learning Dexterous In-Hand Manipulation
arXiv :1808.00177, 2019.
-  Schulman et al.
Proximal Policy Optimization Algorithms
arXiv :1707.06347, 2017.
-  Tobin et al.
Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World
arXiv :1703.06907, 2017.
-  Hochreiter et Schmidhuber
Long Short-Term Memory
Neural Computation, 1997.
-  Todorov et al.
MuJoCo : A physics engine for model-based control
IROS, 2012.