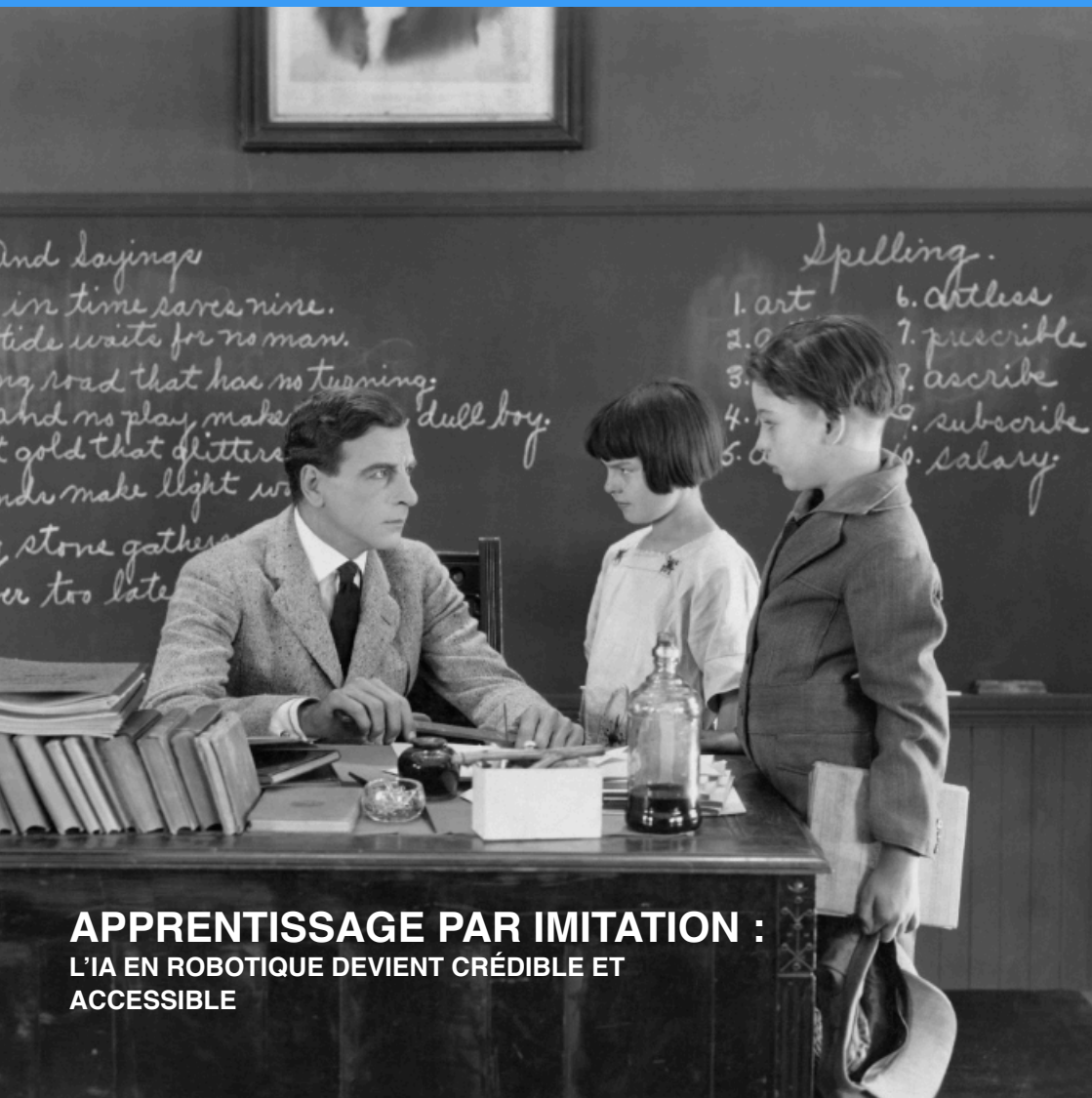




Datalchemy

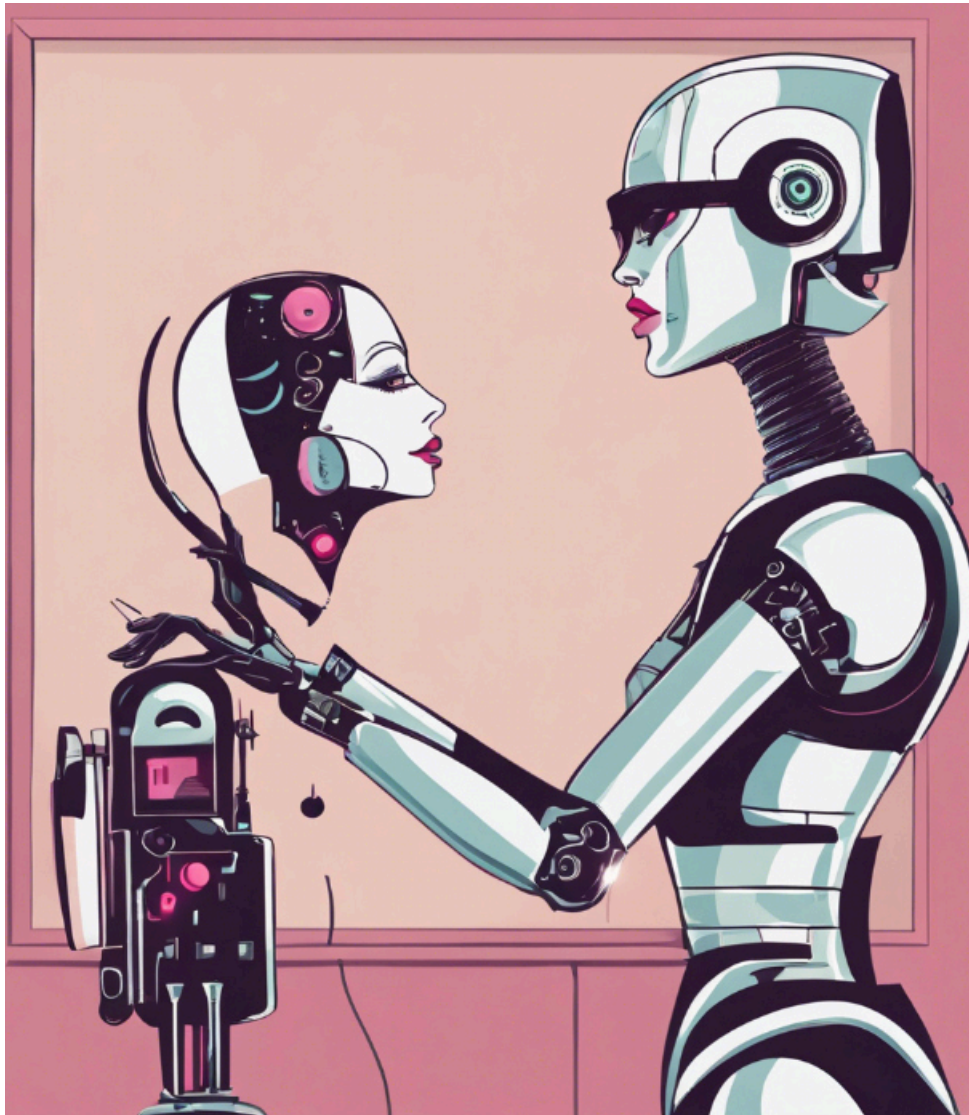
# ECHOS

DE LA RECHERCHE #12



**APPRENTISSAGE PAR IMITATION :**  
L'IA EN ROBOTIQUE DEVIENT CRÉDIBLE ET  
ACCESSIBLE

# TL;DR ?



## Cinq mot-clés

#Robotique, #DeepReinforcementLearning, #ImitationLearning, #ALOHA, #DiffusionPolicies

## Pourquoi lire cette publi peut vous être utile concrètement ?

Si vous travaillez en contrôle robotique, vous n'avez pas le droit d'ignorer la révolution en cours sur l'imitation. Et au-delà du contrôle robotique, tout problème d'optimisation modélisant un agent devant prendre des décisions peut s'inspirer de ces approches.

## Ce que vous pouvez en dire à un collègue ou à votre boss ?

Datalchemy, en partenariat avec Kickmaker, fait le tour des toutes nouvelles stratégies d'entraînement en imitation learning, et même si cela semble encore un peu magique, cela fonctionne avec une robustesse inattendue, et surtout rend enfin ces entraînements abordables, et presque simples à mettre en oeuvre.

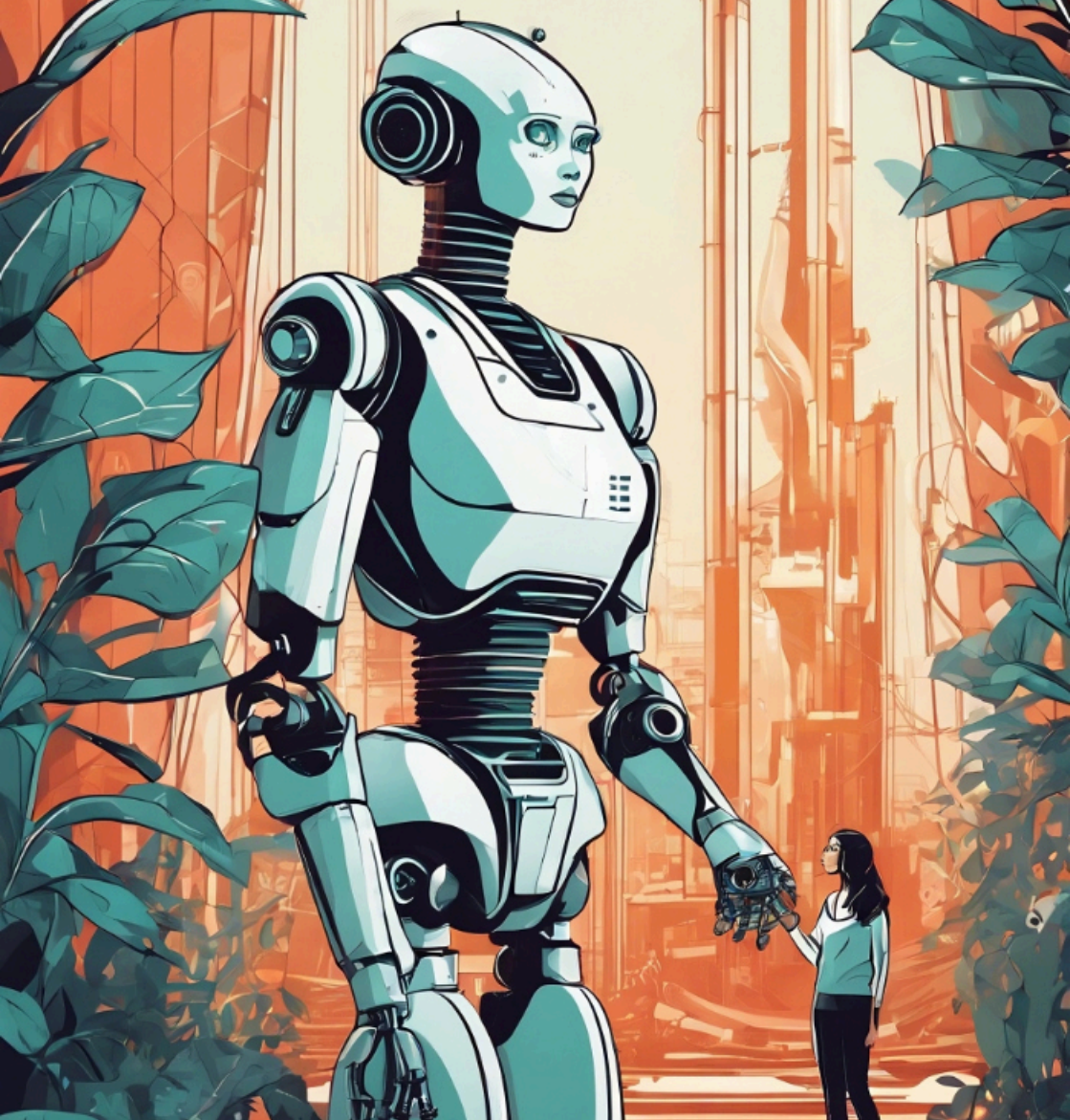
## Quels process métier seront probablement modifiés sur la base de ces recherches ?

C'est quasiment toute la chaîne qui est impactée par ce changement de paradigme. Mais plus spécifiquement, la partie simulation pour recréer les conditions d'entraînement pourrait se voir grandement réduite, puisqu'on se contente désormais de l'enregistrement des actions de l'opérateur humain qui vont servir de dataset d'entraînement.

## Quelle phrase mettre dans un mail pour envoyer cet écho de la recherche à un.e ami.e et lui donner envie de le lire ?

C'est fou ! Datalchemy et Kickmaker s'appuient sur les résultats du MIT et Toyota pour montrer qu'en un temps très restreint et un budget dérisoire, on peut mettre en place de l'imitation learning pour entraîner un robot.





## Les cas d'usage que nous avons développé pour des clients qui touchent au sujet de cet écho de la recherche ?

Entraînement d'un bras robot pour saisir une pièce et la placer sur une cible, quelle que soit sa position dans un espace donné.

## Si vous n'avez qu'une minute à consacrer à la lecture maintenant, voici le contenu essentiel en 6 points ou en 6 phrases)

1. Apparition très récente de nouvelles méthodes d'entraînement robotique (diffusion policies).
2. Assez miraculeusement, on passe du deep reinforcement learning (extrêmement long et coûteux, en plus d'avoir un résultat parfois aléatoire) aux diffusion policies et ALOHA (beaucoup plus simple et abordable, pour un résultat très robuste)
3. Miraculeusement car, jusqu'à aujourd'hui, le soubassement théorique de cette petite révolution est encore à étayer.
4. Ces deux approches permettent d'entraîner le robot à effectuer des tâches potentiellement complexes (complexes dans le sens où recréer une simulation complète de ces tâches relève quasiment de l'impossible) avec un dataset ridiculement petit (quelques centaines d'exemples de tâches correctement exécutées par un opérateur humain.
5. Les diffusion policies (dont une partie de l'architecture s'appuie donc sur la même diffusion que l'IA générative) montrent une robustesse assez incroyable aux variations lors de l'entraînement (une main qui passe devant le capteur vidéo, l'objet à manipuler malicieusement déplacé, etc. .)
6. Une évolution extrêmement récente d'ALOHA rend tout le processus portable, permettant d'imaginer l'entraînement d'un robot à domicile pour aider les personnes âgées ou en situation de handicap dans les tâches de la vie quotidienne.

# IA & ROBOTIQUE ?

Un peu de changement ! Datalchemy a l'honneur d'accompagner l'entreprise Kickmaker depuis bientôt deux ans sur des sujets d'intelligence artificielle notamment appliqués à la robotique. Et nous avons donc été aux premières loges pour observer l'évolution impressionnante de ce domaine. Si les applications IA à la robotique existent depuis les travaux de Mnih et al et l'invocation du Deep Reinforcement Learning, force est de reconnaître que ces travaux sont longtemps restés inaccessibles, ceci pour deux grandes raisons :

- Les entraînements en DRL (Deep Reinforcement Learning) sont naturellement beaucoup plus coûteux en puissance GPU que les entraînements Deep Learning classiques. Les projets qui nous faisaient de l'œil en 2018 (comme l'entraînement d'une main robotique par OpenAI, cf. schéma ci-dessous) nécessitaient un minimum de 8 GPUs de compétition pour espérer obtenir un agent. Beaucoup de nos clients auraient adoré tester ce genre d'approche, mais n'étaient pas prêts à se ruiner pour se forger une opinion.

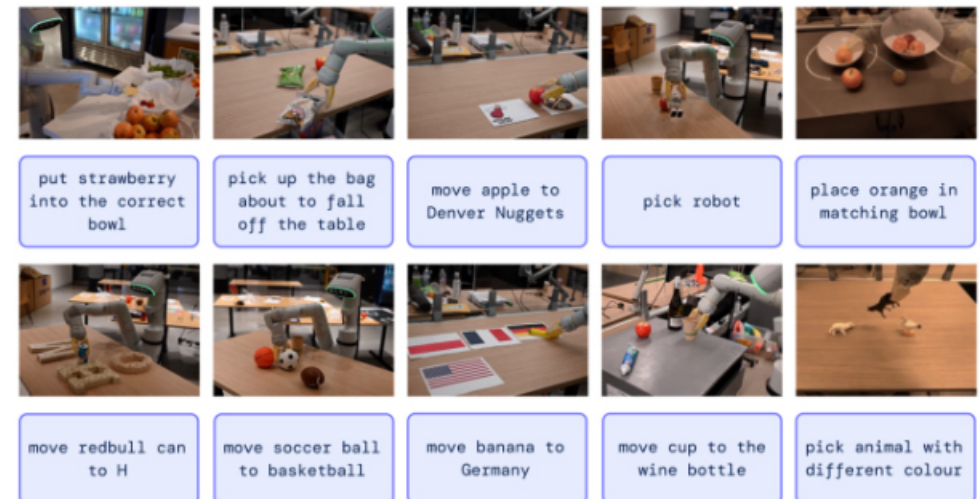
- Ce domaine, le DRL, s'il est extrêmement plaisant dans sa formulation (nous entraînons un agent logiciel à maximiser une récompense arbitraire), était très frileux aux applications réelles. Autant, dans un simulateur classique, un modèle pouvait apprendre à jongler d'une manière experte assez facilement, autant le transfert à la réalité était voué à l'échec, la réalité se chargeant de créer des conditions trop déconnectées de la simulation. Quant à apprendre directement dans la réalité, considérant le coût d'un robot, la chose était impossible.



Figure 1: A five-fingered humanoid hand trained with reinforcement learning manipulating a block from an initial configuration to a goal configuration using vision for sensing.

Le domaine est donc resté l'apanage des « grands » acteurs du Deep Learning, comme OpenAI ou Google. Google qui par ailleurs a marqué un grand coup ces dernières années avec les modèles RT-1, puis RT-2. En effet, Google a présenté au monde un système robotique robuste qui exploite les derniers avancements des Large Language Models (encore eux...) et des cross embeddings (permettant de travailler images et textes dans un espace mathématique commun) pour contrôler un bras robotique sur une grande quantité de tâches. Si cet entraînement a dû être particulièrement lourd, la promesse laisse rêveur. En effet, contrôler un bras robotique suppose une maîtrise extrêmement fine

de chaque position dans l'espace, et pour pouvoir par exemple saisir un objet, il devient nécessaire de localiser exactement cet objet. Une fois cette localisation (peu triviale) réalisée, le geste robotique doit lui aussi être calculé exactement en respectant ses contraintes mécaniques... Ici, l'idée d'avoir juste un texte de définition pour exécuter une action semble une disruption très forte de ce domaine, avec une variance impressionnante, le texte pouvant servir autant à décrire le geste que l'objet cible (ci-dessous, entre autres exemples : « pick up the bag about to fall off »)



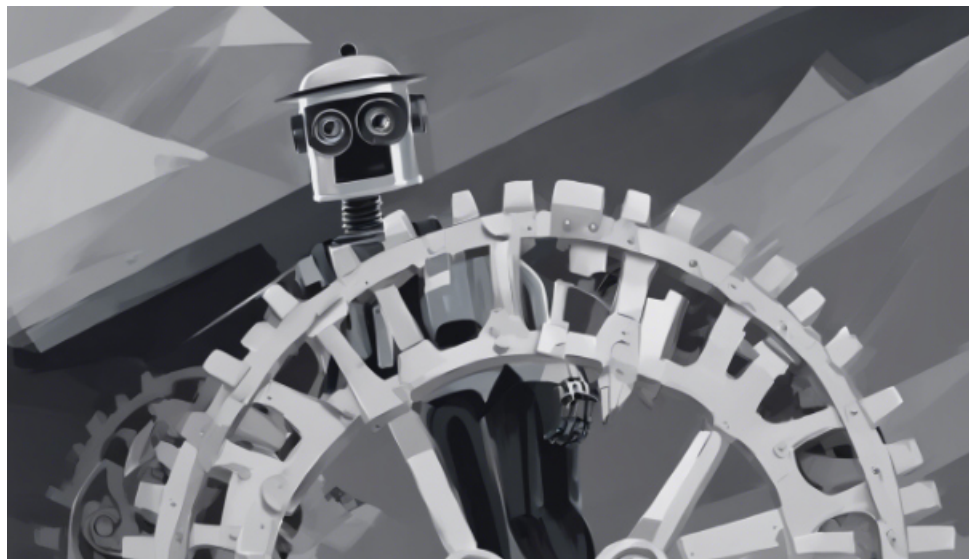


Mais alors, Google a-t-il révolutionné le domaine de la robotique ? Les acteurs du domaine se sont-ils saisis de ces travaux pour les appliquer à leurs propres cas de figure et révolutionner la branche industrielle ? Hélas, non. En effet, (et Google nous avait habitués à mieux), si la publication est libre, le code nécessaire pour entraîner ou utiliser ces modèles est, à date, introuvable. Et considérant que ces entraînements supposent autant l'entraînement du LLM que celui de l'acteur, leurs coûts cumulés empêchent tout travail de reproduction.

Signalons néanmoins un autre point potentiellement complexe avec cette approche : le modèle RT2 est un modèle end to end : nous mettons en entrée un texte, et observons en sortie des actions contrôlant le robot. Hors, toute personne ayant travaillé un petit

peu avec les Large Language Models sait que le contrôle d'un modèle par un prompt est un art obscur, illogique et anxiogène, dans lequel des erreurs peuvent facilement apparaître sans prévenir. Il n'est pas impossible que RT2 souffre du même problème...

À ce stade, nous pourrions conclure tristement que l'intelligence artificielle de pointe en IA est réservée aux très grands acteurs qui, tôt ou tard, commercialiseront leurs travaux. Mais ces six derniers mois, le paysage scientifique a radicalement changé, et est devenu d'un coup beaucoup plus accessible...



# APPRENDRE PAR DES IMITATIONS

La surprise est venue d'un autre courant de recherche du DRL, qui jusqu'ici était resté plus une curiosité scientifique qu'une véritable solution : l'Imitation Learning.

Dans une approche « classique », nous allons entraîner un modèle IA à trouver les bonnes actions à réaliser afin d'obtenir la plus grande récompense possible. Le modèle sera confronté à un simulateur reproduisant (plus ou moins bien) la réalité cible, effectuera une quantité astronomique d'erreurs, pour finalement (avec un peu de chance, soyons honnêtes) devenir suffisamment « bon » pour réussir sa tâche.

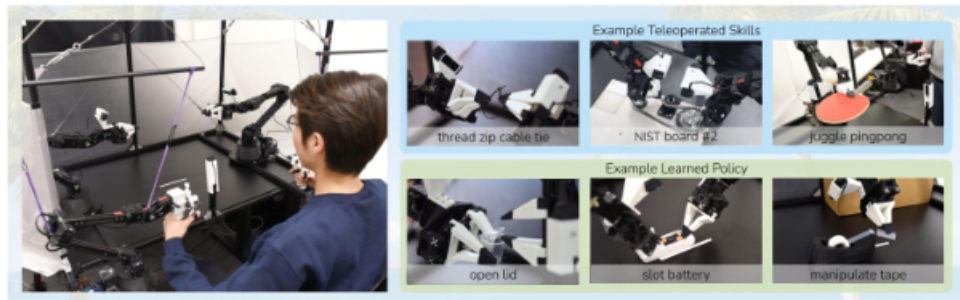
Dans les approches « Imitation Learning », nous n'allons pas utiliser un simulateur directement. Un expert humain va générer un certain nombre de démonstrations, chacune correspondant à une suite d'actions réussies. Une fois ces démonstrations accumulées, un modèle va être entraîné à généraliser à partir de ces démonstrations, pour pouvoir gérer le

plus grand nombre de cas de figure. Considérant que nous sommes en général sur quelques centaines de démonstrations, l'enjeu de généralisation est ici particulièrement ardu, car le modèle doit apprendre à gérer des cas qu'il n'a jamais vu en démonstration, sans avoir accès aux contraintes physiques du monde dans lequel il évolue. Pour cette raison principalement, l'Imitation Learning était resté un sujet plutôt de côté. Avec pourtant une promesse très intéressante : entraîner un modèle sur une tâche agnostique à partir d'exemples qui peuvent donc totalement correspondre à la réalité, sans besoin d'un simulateur très poussé. Plusieurs travaux ont révolutionné ce domaine récemment, des travaux qui sont aujourd'hui applicables à l'industrie et qui demandent un investissement économique modéré. Il est temps de présenter ces petites révolutions 🤖

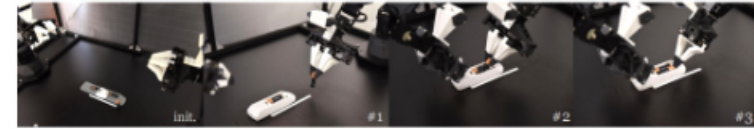
# ALOHA : LEARNING FINE-GRAINED BIMANUAL

Travail joint de Meta, Stanford et Berkeley, ALOHA est une bouffée de sens pratique et d'efficacité qui a proprement bouleversé le milieu pour notre plus grand plaisir. Fondamentalement, le point fort de cette approche est que les chercheurs ne se sont pas contentés de faire tourner un algorithme plus ou moins exotique sur un cluster GPU pour en extraire de jolies vidéos. Au contraire, les auteurs proposent ici une approche intégrale permettant de recréer la configuration d'enregistrement de

démonstrations en plus du code d'entraînement et d'inférence. Et cette configuration de manipulation robotique est, ici, « low cost », avec un coût de reproduction estimé à moins de 6000\$ (pour peu que l'on dispose d'une imprimante 3D correcte). ALOHA propose en effet de créer des interfaces permettant le contrôle direct des deux bras robot par un opérateur humain. Via cette approche, un expert peut donc enregistrer un certain nombre de démonstrations sur des tâches précises :



*Slide Ziploc:* Open the ziploc bag that is standing upright on the table. The bag is randomized along the 15cm white line. It is dropped from ~5cm above the table to randomize the deformation, which affects the height and appearance of the bag. The left arm first grasps the bag body (*Subtask #1 Grasp*) followed by the right arm pinching the slider (*Subtask #2 Pinch*). Then the right arm moves right to unzip the bag (*Subtask #3 Open*).



*Slot Battery:* Insert the battery into the remote controller. The controller is randomized along the 15cm white line. The battery is initialized in roughly the same position with different rotations. The right arm first grasps the battery (*Subtask #1 Grasp*) then places it into the slot (*Subtask #2 Place*). The left arm presses onto the remote to prevent it from sliding, while the right arm pushes in the battery (*Subtask #3 Insert*).

Cette approche « low cost » est déjà une petite révolution dans un monde où le moindre matériel coûte une petite fortune. Ici, l'ensemble du système coûte l'équivalent d'un bras robotique d'entrée de gamme. D'une manière plus générale, les avantages sont assez nombreux : versatilité avec une très grande gamme d'applications possibles, simple d'utilisation, facile à réparer comme à construire (tout du moins relativement au monde de la robotique). Il est déjà assez rare que des chercheurs posent la question de la performance et de la reproductibilité de leurs travaux à ce point, ici, tout est disponible pour reproduire ce système assez facilement. Dès lors que l'on arrivera via l'interface de contrôle à réaliser une tâche, on pourra espérer entraîner un modèle qui généralise cette tâche d'une manière satisfaisante. Les tâches accomplies, justement, ont ceci de particulier qu'elles vont largement au-delà des applications communément admises en renforcement. Nous parlons ici

d'actions extrêmement précises, comme ouvrir ou fermer un sac ziplock, attraper une carte de crédit dans un portefeuille (skynet aura besoin d'argent de poche, ne discutez pas), insérer un composant électronique, tourner les pages d'un livre, jusqu'à faire rebondir une balle de ping pong sur une raquette !

Ces tâches sont intéressantes, car elles seraient typiquement très dures (voir impossibles) à apprendre dans une approche classique, ne serait-ce que parce qu'elles sont très complexes à reproduire en simulation (imaginons un simulateur reproduisant la déformation physique du plastique). Ce sont aussi des tâches à haute précision. Elles sont ici possibles, précisément car on se base sur des démonstrations réalisées par des opérateurs humains qui eux, sauront réagir à l'évolution du problème pour trouver la bonne politique d'action. On pourrait postuler (d'une manière ambitieuse) que si une tâche peut être

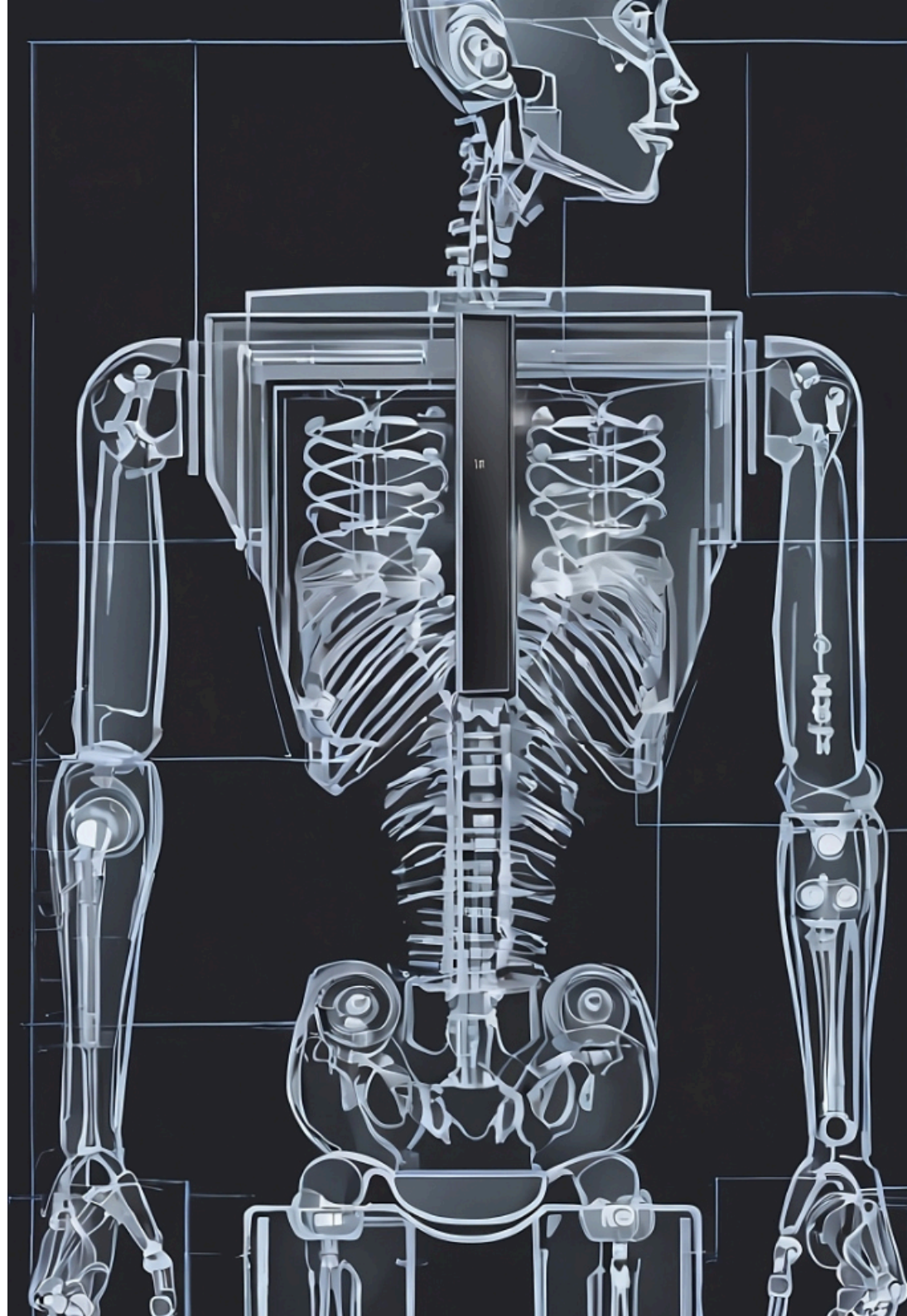


réalisée en démonstration, elle pourra être soumise à apprentissage...

Concernant l'architecture interne du modèle, quelques originalités techniques que nous laisserons ici de côté, pour noter que nous retrouvons l'éternel Transformer (note : l'éternité a débuté en 2017 pour les chercheurs en Deep Learning). Un encoder de type CNN va générer des représentations de haut niveau des images de chaque caméra (ici, quatre caméras), pour alimenter ensuite un encoder Transformer qui prendra aussi en entrée l'historique des commandes du

robot. En sortie, le decoder sera appelé pour générer la prochaine séquence d'action. Point d'intérêt, un conditional VAE est ici utilisé pour entraîner un autre Transformer encoder et lui permettre d'apprendre une compression pertinente de l'information en entrée. ALOHA a été le premier coup de canon avec des résultats nettement supérieurs à ce que l'on pouvait observer auparavant. Ci-dessous, le ACT de ALOHA est comparé aux approches précédentes sur deux tâches synthétiques et deux tâches réelles. Les résultats observés sont nettement plus convaincants.

	Cube Transfer (sim)			Bimanual Insertion (sim)			Slide Ziploc (real)			Slot Battery (real)		
	Touched	Lifted	Transfer	Grasp	Contact	Insert	Grasp	Pinch	Open	Grasp	Place	Insert
BC-ConvMLP	34   3	17   11	1   10	5   10	1   10	1   10	0	0	0	0	0	0
BeT	60   16	51   13	27   11	21   10	4   10	3   10	8	0	0	4	0	0
RT-1	44   4	33   2	2   10	2   10	0   10	1   10	4	0	0	4	0	0
VINN	13   17	9   11	3   10	6   10	1   10	1   10	28	0	0	20	0	0
<b>ACT (Ours)</b>	<b>97   82</b>	<b>90   60</b>	<b>86   50</b>	<b>93   76</b>	<b>90   66</b>	<b>32   20</b>	<b>92</b>	<b>96</b>	<b>88</b>	<b>100</b>	<b>100</b>	<b>96</b>



# DIFFUSION POLICIES : VISUOMOTOR POLICY

## LEARNING VIA ACTION DIFFUSION, CHI ET AL

Un autre travail est apparu en juin 2023 en s'imposant comme une nouvelle référence du domaine de l'Imitation Learning. Porté par le MIT, Toyota et Columbia University, les Diffusion Policies se sont imposées comme une approche très efficace et polyvalente, devenue aujourd'hui pour nous une quasi baseline à tester face à un nouveau problème.

Nous restons dans le même paradigme que précédemment, soit, l'entraînement d'un modèle à résoudre un problème en utilisant un nombre fini de démonstrations. Cette approche s'est notamment distinguée par son approche architecturale qui la rend extrêmement efficace. Notamment :

Une modélisation permettant un fonctionnement en boucle fermée, en prenant sans cesse en compte l'historique des dernières actions du robot pour réajuster les futurs mouvements produits par le modèle. L'idée est que le modèle va pouvoir continuellement adapter ses actions, notamment en cas d'événement soudain : occultation de la caméra, déplacement d'un objet par un acteur tiers, etc. Cette robustesse, que nous

avons pu observer dans nos tests, est un avantage très fort des Diffusion Policies.

Un conditionnement visuel efficace. Un sous-modèle est spécialisé pour extraire d'une suite d'images (l'historique des actions du robot vues par une caméra) des représentations qui, ensuite, alimenteront le processus de diffusion qui génère les actions du robot (nous y revenons très vite). Cette extraction n'étant faite qu'une seule fois par image, indépendamment de la diffusion, le modèle résultant peut tourner en temps réel.

Une nouvelle architecture ad-hoc optimale pour effectuer de la diffusion (je vous ai promis que nous allons y revenir, cette promesse sera tenue !) avec un Transformer.

Et promesse faite promesse tenue, profitons-en pour parler de la diffusion 😊

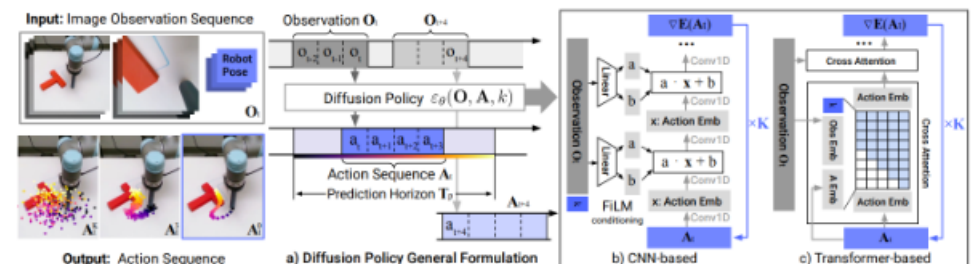
Si vous suivez le Deep Learning un minimum, vous n'avez pas raté cette nouvelle approche qui s'est imposée depuis deux ans, notamment dans le cas des IAs génératives comme Stable

Diffusion. L'idée de la diffusion est que l'on va entraîner un modèle à produire l'information qui nous intéresse, mais en rajoutant une contrainte à cette génération : partir d'une information totalement bruitée et, progressivement, par débruitage, obtenir le résultat obtenu. Nous avons déjà parlé de ces nouvelles approches dans notre revue de la recherche, notamment car si elles donnent des résultats impressionnants, elles sont trop récentes pour que nous comprenions pourquoi elles fonctionnent si bien.

Comme d'accoutumée en Deep Learning, une approche qui marche très bien dans un domaine sera testée dans tous les autres domaines. Ce n'est pas toujours une très bonne idée (combien d'architecture visant à « optimiser » l'attention du Transformer ont fini dans les limbes ?), mais pour la diffusion, force est de reconnaître que cette approche globale semble s'appliquer à d'autres sujets que la génération d'images sans peine.

Ici, nous découvrons que ces modèles

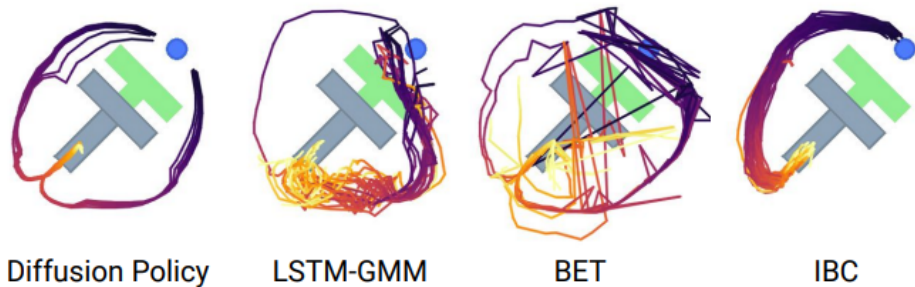
de diffusion sont un excellent outil pour apprendre à générer des informations de contrôle d'un bras robot, conditionnées sur un historique d'actions et d'images. Nous n'essaierons même pas de tenter une justification de cette réussite. Mais il semble que cette approche réussit à réellement généraliser de nouvelles actions à partir de démonstrations, et ce d'une manière très efficace. Le schéma ci-dessous représente l'architecture globale. Nous avons (en haut à gauche) un historique d'actions et d'images. Ces observations vont être exploitées pour générer, par diffusion, la séquence d'actions (schéma a) Diffusion Policy General Formulation). Cette séquence est générée en modélisant les images soit par un bon vieux CNN (schéma b), soit un Transformer (schéma c). On voit en bas à gauche une modélisation de la diffusion avec, tout à gauche, une série de positions totalement bruitée, qui par diffusion va finir par (troisième image) être raffinée en une série de positions valables pour pousser le « T » au bon endroit.





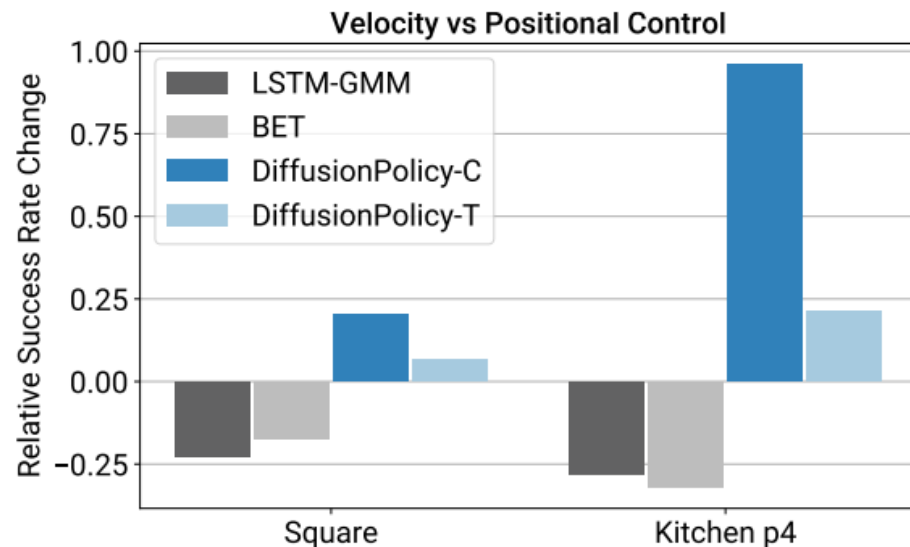
Plusieurs points d'intérêt sont ici à relever : Les modèles de diffusion sont multi-modaux. Ils apprennent une véritable variance dans les stratégies d'actions et restent très stables contrairement aux approches

précédentes, comme exposé sur le schéma ci-dessous, où l'IBC n'apprend qu'une seule direction (comme quasiment le LSTM-GMM), et où BET/LSTM-GMM sont très, voir trop bruités :

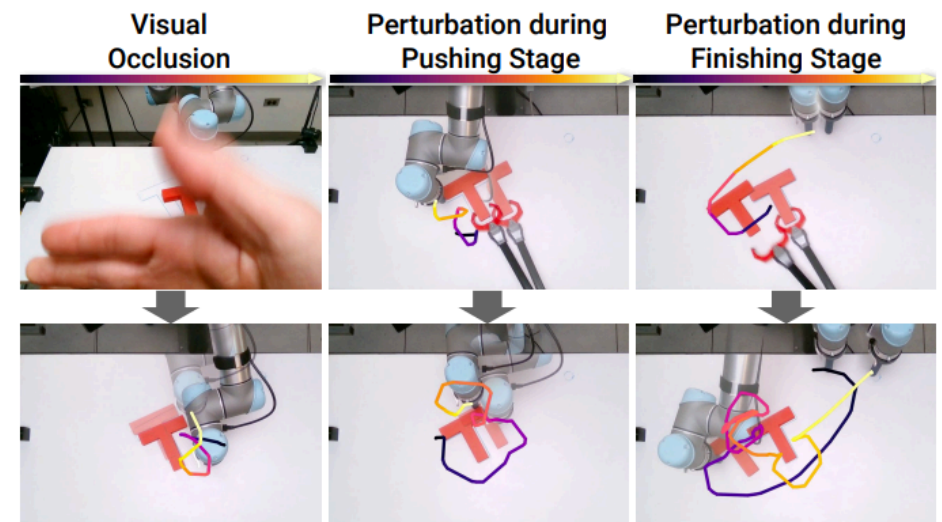


Diffusion Policy travaille en positions. La majorité des approches de contrôle robotique par imitation modélisent le déplacement du bras

robot comme un contrôle de vitesse, et non de position. Ici, d'une manière surprenante, le contrôle par position fonctionne beaucoup mieux.



- **Diffusion Policy est robuste face à la latence de réception.** Une force de cette approche est de proposer une méthode efficace, et donc robuste à certaines perturbations. Les auteurs ont introduit une latence allant jusqu'à 10 timesteps pour une perte négligeable de résultats. Nous observons ici une forme de généralisation appréciable qui nous permet de nous projeter correctement vers une utilisation pratique de cet outil.
- **Diffusion Policy est stable à l'entraînement.** Nous avons pu l'observer personnellement lors de nos tests, là où le choix des hyperparamètres relève souvent de la tentative désespérée pouvant totalement invalider un entraînement, Diffusion Policy est extrêmement stable. Et c'est une excellente nouvelle, car d'ordinaire, il est nécessaire d'itérer sur ces hyperparamètres pour espérer trouver la bonne combinaison, ce qui est autant de temps et d'argent perdus.
- **Diffusion Policy est stable face aux perturbations.** Ce point est impressionnant. Lors de l'entraînement d'un robot en environnement réel, les auteurs vont régulièrement occulter la caméra, ou déplacer un élément que le robot doit manipuler. Le modèle adapte sa stratégie en temps réel sans tomber en erreur. Là encore, nous avons un argument fort si nous nous intéressons à des approches concrètes et efficaces. Fusce condimentum turpis vel dolor. Ut blandit. Sed elementum justo quis sem. Sed eu orci eu ante iaculis accumsan. Sed suscipit dolor quis mi. Curabitur ultrices nonummy lacus. Morbi ipsum ipsum, adipiscing eget,

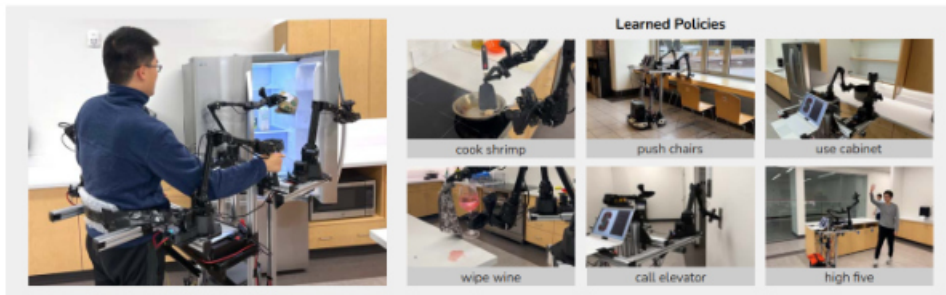


# MobileALHOA :

## Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation, Fu et al

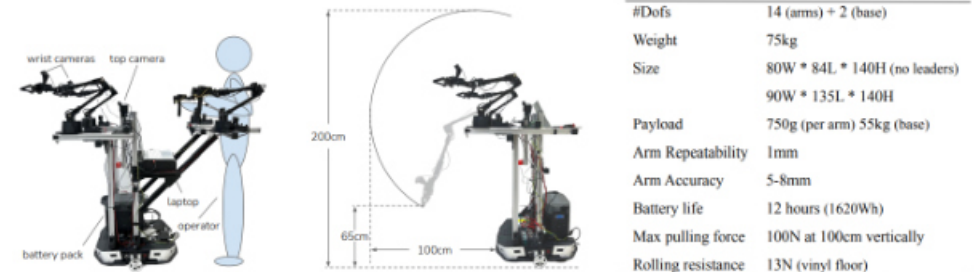
Ce dernier travail est beaucoup plus récent, mais mérite d'être remonté. Ici, pas vraiment d'innovation mathématique Deep Learning, mais une lucarne ouverte sur l'avenir de la robotique à partir de ces approches. Les auteurs ont repris l'approche ALOHA et l'ont fait évoluer jusqu'à obtenir un environnement de

démonstration qui soit portable et puisse être appliqué n'importe où, notamment au domicile de quelqu'un. Cela permet de récolter des démonstrations dans un cadre ouvert, pour ensuite entraîner un modèle à interagir avec des objets courants via un robot complet :



Le plus impressionnant dans cette affaire n'est pas la polyvalence des actions réalisées, mais le fait que le système reste accessible financièrement (environ 30K\$) et ouvre la porte à de nombreuses applications intéressantes. Nous avons là un

système qui est rapide à l'exécution (comparable à un humain en déplacement normal), stable y compris pour la manipulation d'objets lourds, et auto-suffisant en terme d'alimentation énergétique (batterie intégrée).



Un autre point d'intérêt est que les auteurs ont réutilisé les anciennes démonstrations d'ALOHA pour entraîner le modèle conjointement avec ces nouvelles applications, et ont observé une réelle amélioration des résultats. Cela implique que chaque problème n'est pas totalement cloisonné, et qu'un modèle peut apprendre à transférer d'une situation à l'autre. Il devient imaginable de

disposer d'un dataset global jouant un rôle de fondation, et permettant d'apprendre de nouvelles tâches très efficacement. Notons aussi que MobileALOHA, ici, se contente d'une cinquantaine de démonstrations pour apprendre une tâche, un peu en dessous des diffusion policies qui elles nécessitent plusieurs centaines... Ci-dessous, quelques exemples de tâches apprises et reproduites par le modèle :



**Wipe Wine:** The robot base is initialized within a square of 1.5m x 1.5m with yaw up to 30°. It first navigates to the sink and picks up the towel hanging on the faucet (#1). It then turns around and approaches the kitchen island, picks up the wine glass (randomized in 30cm x 30cm), wipes the spilled wine (#2), and puts down the wine glass on the table (#3). Each demo has 1300 steps or 26 seconds.



**Cook Shrimp:** The robot is randomized up to 5cm and all objects up to 2cm. The right gripper first pours oil into the hot pan (#1) followed by raw shrimp (#2). With left gripper lifting the pan at an angle, the right gripper grasps the spatula and flips the shrimp (#3). The robot then turns around and pours the shrimp into an empty bowl (#4) before placing the pan on the table. Each demo has 3750 steps or 75 seconds.





## Conclusions ?

Si nous devons conclure, nous pouvons déjà observer que nous sommes à un instant clé dans le domaine de la robotique augmentée par intelligence artificielle. Un domaine qui était extrêmement coûteux d'accès devient d'un coup beaucoup plus accessible (les entraînements diffusion policy, par exemple, se font en deux/trois jours sur un gpu classique). Les domaines d'application explosent aussi en faisant disparaître des tonnes de problèmes que nous avons, en résumant d'une manière un peu caricaturale par « si nous pouvons contrôler le robot et récolter des démonstrations, nous pourrons espérer entraîner un agent ».

Après, attention ! Ces travaux sont encore très récents, et comme d'accoutumée en Deep Learning, entraîner un modèle est une chose,

mais le cadrer et le contrôler pour éviter de mauvais cas de figure reste un travail à part entière à ne pas sous-estimer, parfois plus complexe que l'entraînement lui-même. Il est donc nécessaire de contrôler au fur et à mesure les actions envoyées pour éviter les chocs (quand cela est possible !), ou à minima pour empêcher les modes de fonctionnement non désirés. Tester un tel système est aussi une gageure, notamment si nous avons entraîné un robot réel directement... Il n'empêche que la robotique semble prête à connaître un nouvel âge d'or. Là où la programmation d'un tel système était d'une lourdeur incroyable, accumuler des démonstrations est beaucoup plus rapide et adresse directement par le biais de l'opérateur les spécificités du problème visé. Affaire à suivre, comme d'habitude 😊