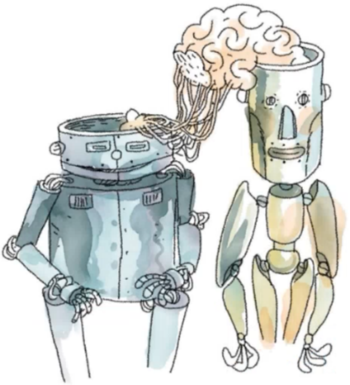
Transfert Learning in Robotics

Results of the PhD works of
M. Mounsif and S. Beaussant

Sébastien Lengagne

2024, April 4th

1. Context
2. Transfert in Robotics
3. The Universal Notive Network
4. The Latent-Space UNN
5. The Delay-Aware UNN
6. Conclusions and Perspectives



Context

This work was performed during the PhD of :

- Mehdi Mounsif (2017-2020) : *Exploration of Teacher-Centered and Task-Centered paradigms for efficient transfer of skills between morphologically distinct robots* under the supervision of L. Adouane, S. Lengagne et B. Thuilot

Context

This work was performed during the PhD of :

- Mehdi Mounsif (2017-2020) : *Exploration of Teacher-Centered and Task-Centered paradigms for efficient transfer of skills between morphologically distinct robots* under the supervision of L. Adouane, S. Lengagne et B. Thuilot
- Samuel Beaussant (2020-2023) *Transfert de compétences par abstraction d'état.* (Skill Transfert using state abstraction) under the supervision O. Stasse, S. Lengagne et B. Thuilot.

Context

This work was performed during the PhD of :

- Mehdi Mounsif (2017-2020) : *Exploration of Teacher-Centered and Task-Centered paradigms for efficient transfer of skills between morphologically distinct robots* under the supervision of L. Adouane, S. Lengagne et B. Thuilot
- Samuel Beaussant (2020-2023) *Transfert de compétences par abstraction d'état.* (Skill Transfert using state abstraction) under the supervision O. Stasse, S. Lengagne et B. Thuilot.

Goal : Design a learning framework for skill transfer in robotics and to validate it on real robots.

Transfert in Robotics

Motion generation for robots can be done using the following methods :

Transfert in Robotics

Motion generation for robots can be done using the following methods :

- analytics, based on models :

Transfert in Robotics

Motion generation for robots can be done using the following methods :

- analytics, based on models :
  - $++$: easy to transfer (just need model),

Transfert in Robotics

Motion generation for robots can be done using the following methods :

- analytics, based on models :
  - $++$: easy to transfer (just need model),
  - - - : not able to deal with complex motions, ...

Transfert in Robotics

Motion generation for robots can be done using the following methods :

- analytics, based on models :
  - ++ : easy to transfer (just need model),
  - - - : not able to deal with complex motions, ...
- using Learning methods (based on Neural Network)

Transfert in Robotics

Motion generation for robots can be done using the following methods :

- analytics, based on models :
  - ++ : easy to transfer (just need model),
  - - - : not able to deal with complex motions, ...
- using Learning methods (based on Neural Network)
  - ++ : deal with complex motions,

Transfert in Robotics

Motion generation for robots can be done using the following methods :

- analytics, based on models :
  - ++ : easy to transfer (just need model),
  - - - : not able to deal with complex motions, ...
- using Learning methods (based on Neural Network)
  - ++ : deal with complex motions,
  - - - : computation time/energy of the learning phase, ...

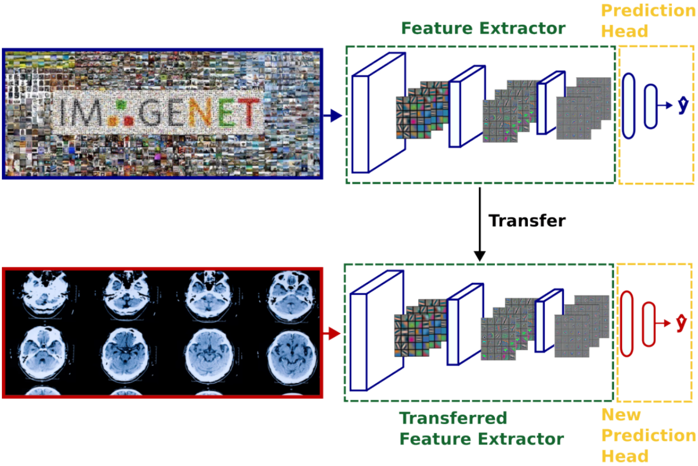
Transfert in Robotics

Motion generation for robots can be done using the following methods :

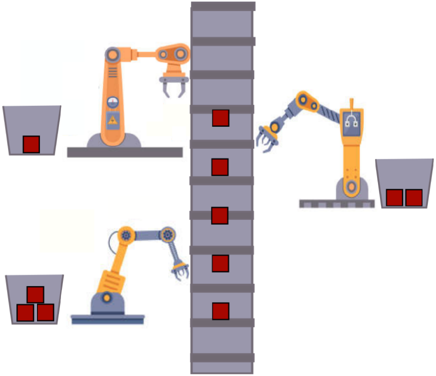
- analytics, based on models :
  - ++ : easy to transfer (just need model),
  - - - : not able to deal with complex motions, ...
- using Learning methods (based on Neural Network)
  - ++ : deal with complex motions,
  - - - : computation time/energy of the learning phase, ...

Transfert learning between several robots can avoid/decrease the time of the learning phases.

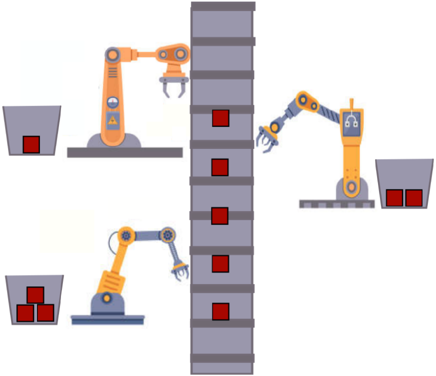
Transfert in vision



Transfert in robotics



Transfert in robotics



How can we deal with different robots ? With different structure ? With different number of dof ?

Definition of Transfert

The paper : *Transfer Learning in Robotics : An Upcoming Breakthrough ? A Review of Promises and Challenges*, from Jaquier et al.[1] presents the transfert in robotics through 3 questions :

1. https://arxiv.org/html/2311.18044v2

Definition of Transfert

The paper : *Transfer Learning in Robotics : An Upcoming Breakthrough ? A Review of Promises and Challenges*, from Jaquier et al.[1] presents the transfert in robotics through 3 questions :

- **When** is it interesting to transfer ?

1. `https://arxiv.org/html/2311.18044v2`

Definition of Transfert

The paper : *Transfer Learning in Robotics : An Upcoming Breakthrough ? A Review of Promises and Challenges*, from Jaquier et al. [1] presents the transfert in robotics through 3 questions :

- **When** is it interesting to transfer ?
- **What** part of the skill must be transfer ?

1. `https://arxiv.org/html/2311.18044v2`

Definition of Transfert

The paper : *Transfer Learning in Robotics : An Upcoming Breakthrough ? A Review of Promises and Challenges*, from Jaquier et al.[1] presents the transfert in robotics through 3 questions :

- **When** is it interesting to transfer ?
- **What** part of the skill must be transfer ?
- **How** can we transfer efficiently and safely ?

1. `https://arxiv.org/html/2311.18044v2`

Definition of Transfert

The paper : *Transfer Learning in Robotics : An Upcoming Breakthrough ? A Review of Promises and Challenges*, from Jaquier et al. [1] presents the transfert in robotics through 3 questions :

- **When** is it interesting to transfer ?
- **What** part of the skill must be transfer ?
- **How** can we transfer efficiently and safely ?

In this presentation, we focus on **how ?**.

1. `https://arxiv.org/html/2311.18044v2`

Reinforcement Learning in robotics

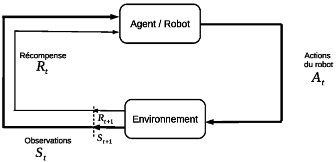
Lets define :

- A robot/ agent r (in real world or in simulation),
- that must perform a task $\mathcal{T}$

Reinforcement Learning in robotics

Lets define :

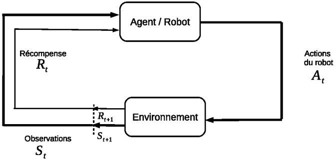
- A robot/ agent r (in real world or in simulation),
- that must perform a task $\mathcal{T}$



Reinforcement Learning in robotics

Lets define :

- A robot/ agent r (in real world or in simulation),
- that must perform a task $\mathcal{T}$

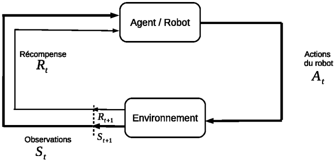


Using Reinforcement Learning, we want to found a policy $\pi_{r,\mathcal{T}}$ that, based on the observations S, produces the Actions A that maximise a Reward R.

Reinforcement Learning in robotics

Lets define :

- A robot/ agent r (in real world or in simulation),
- that must perform a task $\mathcal{T}$



Using Reinforcement Learning, we want to found a policy $\pi_{r,\mathcal{T}}$ that, based on the observations S, produces the Actions A that maximise a Reward R. We introduce the performance $\mathcal{P}_{r,\mathcal{T}}$ of the robot r to solve the task $\mathcal{T}$

Transfer formulation in robotics

Lets define :

- a World $\mathcal{W}_{r,\mathcal{T}}$, that contains a robot r that must perform a task $\mathcal{T}$,

Transfer formulation in robotics

Lets define :

- a World $\mathcal{W}_{r,\mathcal{T}}$, that contains a robot r that must perform a task $\mathcal{T}$,
- a Universe $\mathcal{U}$ is a set of n worlds composed from several robots and several tasks.

Transfer formulation in robotics

Lets define :

- a World $\mathcal{W}_{r,\mathcal{T}}$, that contains a robot r that must perform a task $\mathcal{T}$,
- a Universe $\mathcal{U}$ is a set of n worlds composed from several robots and several tasks.

Considering :

- a Source Universe $\mathcal{U}_s$ used during the learning phase,

Transfer formulation in robotics

Lets define :

- a World $\mathcal{W}_{r,\mathcal{T}}$, that contains a robot r that must perform a task $\mathcal{T}$,
- a Universe $\mathcal{U}$ is a set of n worlds composed from several robots and several tasks.

Considering :

- a Source Universe $\mathcal{U}_s$ used during the learning phase,
- and a Target Universe $\mathcal{U}_t$ with $\mathcal{U}_s \cap \mathcal{U}_t = \emptyset$

Transfer formulation in robotics

Lets define :

- a World $\mathcal{W}_{r,\mathcal{T}}$, that contains a robot r that must perform a task $\mathcal{T}$,
- a Universe $\mathcal{U}$ is a set of n worlds composed from several robots and several tasks.

Considering :

- a Source Universe $\mathcal{U}_s$ used during the learning phase,
- and a Target Universe $\mathcal{U}_t$ with $\mathcal{U}_s \cap \mathcal{U}_t = \emptyset$

We define the transfer as finding the policies that maximises the performances :

$$\forall \mathcal{W} \in \{\mathcal{U}_s \cup \mathcal{U}_t\} \quad \pi^*_{r,\mathcal{T}} = \underset{\pi_{r,\mathcal{T}}}{\operatorname{argmax}} \mathcal{P}_{r,\mathcal{T}}$$

Principle of the Universal Notive Network (UNN)

How can a German teach a recipe to a Spanish ?

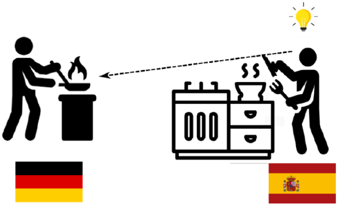
Principle of the Universal Notive Network (UNN)

How can a German teach
a recipe to a Spanish ?

They can speak English (or French)

Principle of the Universal Notive Network (UNN)

How can a German teach a recipe to a Spanish ?

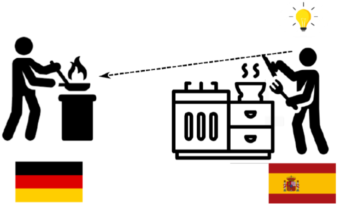


He can show him how to do.

They can speak English (or French)

Principle of the Universal Notive Network (UNN)

How can a German teach
a recipe to a Spanish ?

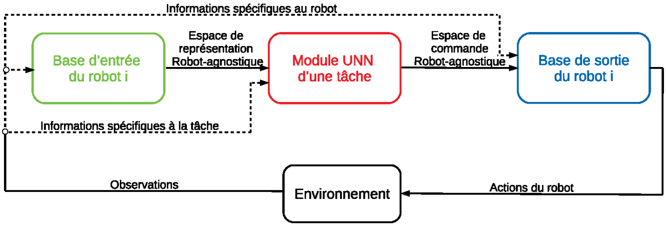


He can show him how to do.

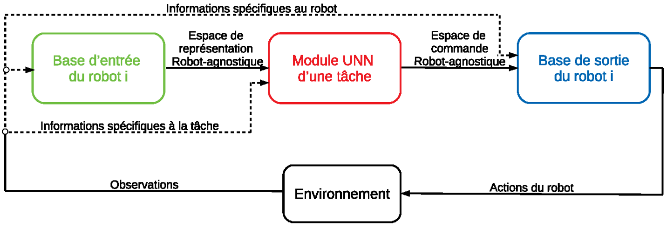
They can speak English (or French)

They need to have common representation of the actions to perform.

Principle of the Universal Notive Network (UNN)

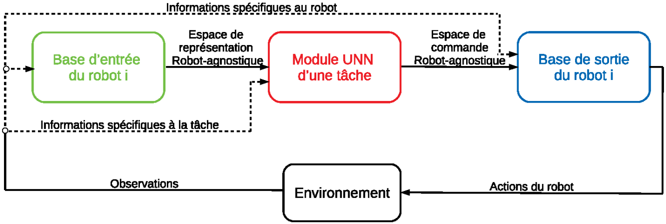


Principle of the Universal Notive Network (UNN)



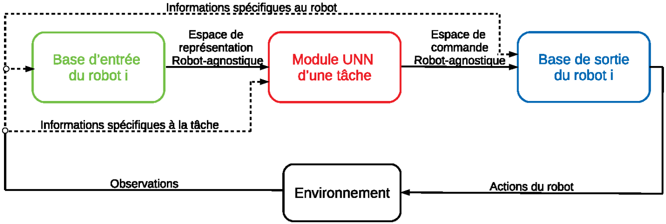
- The bases et the UNN module are (can be) Neural Networks,

Principle of the Universal Notive Network (UNN)



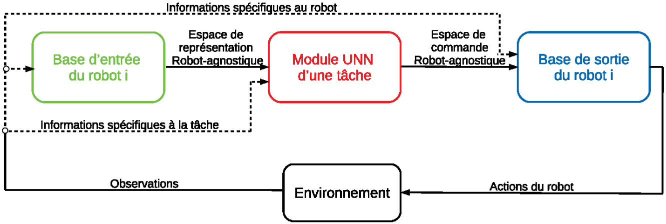
- The bases et the UNN module are (can be) Neural Networks,
- The robot-agnostic state is a common "language" to solve the task,

Principle of the Universal Notive Network (UNN)



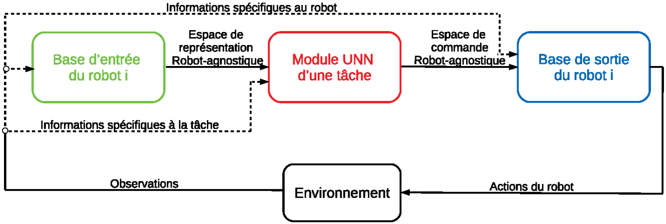
- The bases et the UNN module are (can be) Neural Networks,
- The robot-agnostic state is a common "language" to solve the task,
- The input and output bases are defined for a robot,

Principle of the Universal Notive Network (UNN)



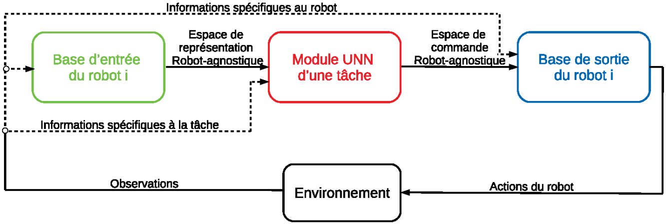
- The bases et the UNN module are (can be) Neural Networks,
- The robot-agnostic state is a common “language” to solve the task,
- The input and output bases are defined for a robot,
- The UNN module is dedicated to a given task,

Principle of the Universal Notive Network (UNN)



- The bases et the UNN module are (can be) Neural Networks,
- The robot-agnostic state is a common "language" to solve the task,
- The input and output bases are defined for a robot,
- The UNN module is dedicated to a given task, it is learned using one robot and can be used with an other robot.

Principle of the Universal Notive Network (UNN)



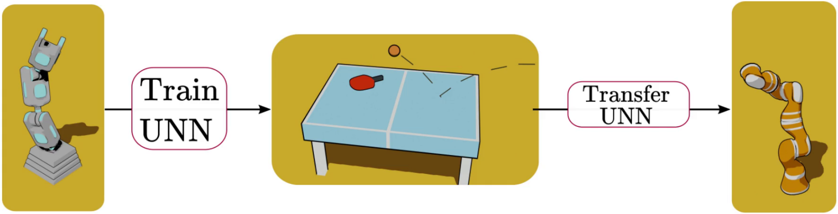
- The bases et the UNN module are (can be) Neural Networks,
- The robot-agnostic state is a common "language" to solve the task,
- The input and output bases are defined for a robot,
- The UNN module is dedicated to a given task, it is learned using one robot and can be used with an other robot.

Results

M. Mounsif considered this robot-agnostic state as the end-effector pose.

Universal Notice Networks: A broad panel of applications

Results: Tennis



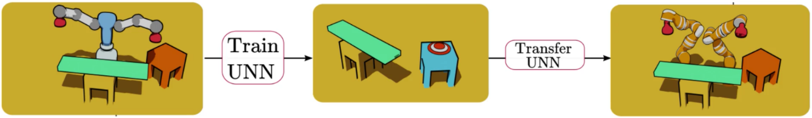
Mehdi Mounsif, Sebastien Lengagne, Benoit Thuilot, Lounis Adouane

Results

This framework can be extended to multi robot manipulation.

Universal Notice Networks: A broad panel of applications

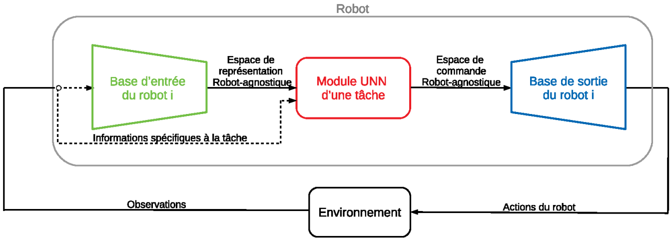
Results: Dual-Arm Manipulation



Mehdi Mounsif, Sebastien Lengagne, Benoit Thuilot, Lounis Adouane

The Latent-Space UNN

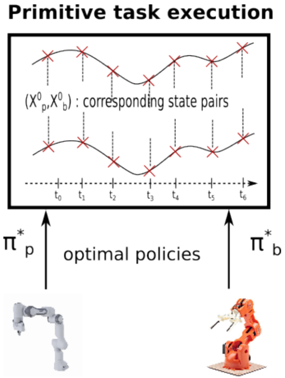
During his PhD, S. Beaussant investigate how to found the best robot-agnostic state.



We consider 4 steps.

The Latent-Space UNN : 1) pairing similar state

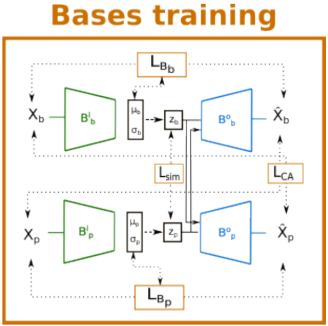
1) We build a database of paired states : two robot are performing the same task at the same speed.



The Latent-Space UNN : 2) Bases learning

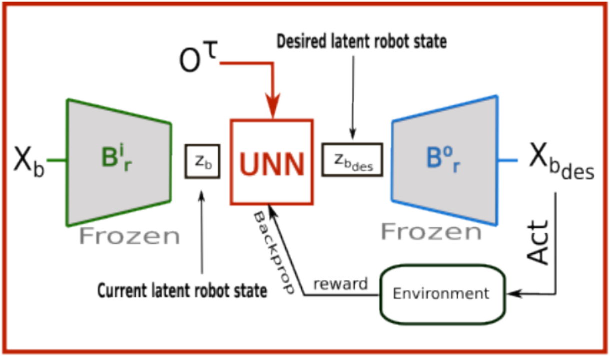
2) We learn the bases considering :

- the reconstruction (each robot encodes and decodes its states),
- the similarity (paired states must be encoded to close abstract state)
- the cross reconstruction
- KL divergence to improve generalization.



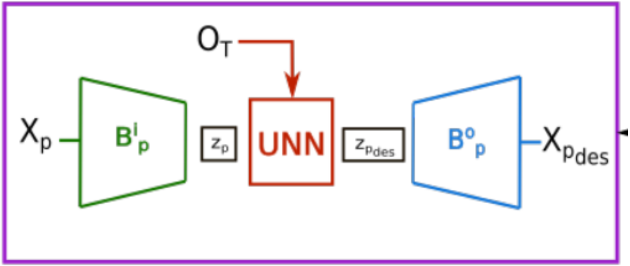
The Latent-Space UNN : 3) UNN learning

3) We learn the UNN module freezing the bases of the robot.

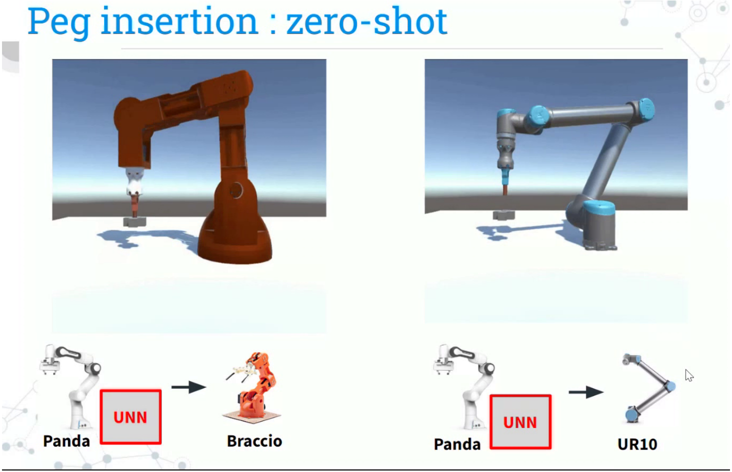


The Latent-Space UNN : 4) UNN transfer

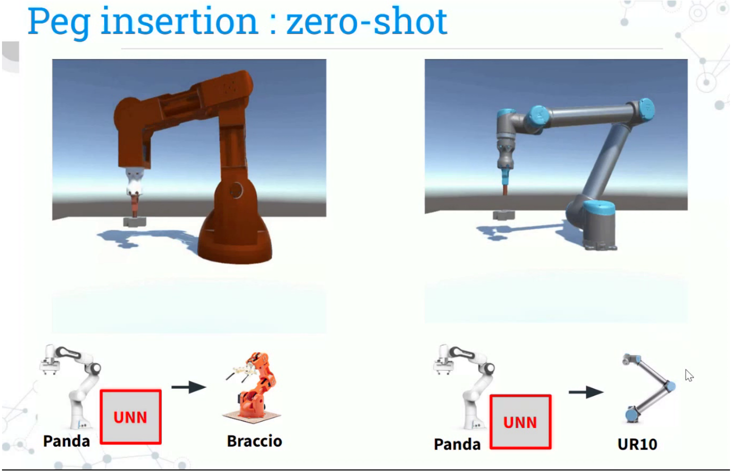
4) We transfer the UNN module to an other robot.



Results in simulation (precise motion)



Results in simulation (precise motion)



Some fails, a “fine-tuning” may be needed.

Comparing performances in simulation (precise motion)

Results before and after “fine-tuning”

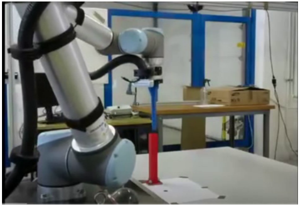
Critère de réussite : pointe du peg placé à une distance < δ_2

Source\Target	Braccio	Panda	UR10
Braccio	100	95 / 100	92 / 100
Panda	95 / 100	100	82 / 100
UR10	99 / 100	90 / 100	100

Percentage given for 1000 tests

Results on actual robots

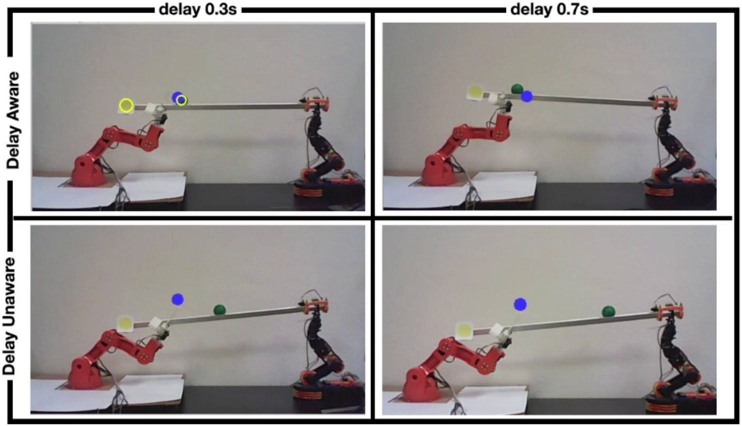
Panda robot with UNN trained on Braccio + FineTune (speed x4)



UR10 robot with UNN trained on Panda + FineTune (speed x4)

The Delay-Aware UNN

S. Beaussant proposes a modification of the UNN to deal with robot with different delay.



Conclusions and Perspectives

- Conclusion :
  Learning transfer is possible and reduces learning times and has been validated :

Conclusions and Perspectives

- Conclusion :
  Learning transfer is possible and reduces learning times and has been validated :
  - "fine-tuning" is from 10 to 25 % of the original learning computation time,

Conclusions and Perspectives

- Conclusion :
  Learning transfer is possible and reduces learning times and has been validated :
  - "fine-tuning" is from 10 to 25 % of the original learning computation time,
  - in simulation on complex tasks,

Conclusions and Perspectives

- Conclusion :
  Learning transfer is possible and reduces learning times and has been validated :
  - "fine-tuning" is from 10 to 25 % of the original learning computation time,
  - in simulation on complex tasks,
  - on actual robot with precise (but slow) task,

Conclusions and Perspectives

- Conclusion :
  Learning transfer is possible and reduces learning times and has been validated :
  - "fine-tuning" is from 10 to 25 % of the original learning computation time,
  - in simulation on complex tasks,
  - on actual robot with precise (but slow) task,
  - on actual robot with dynamic motion

Conclusions and Perspectives

- Conclusion :
  Learning transfer is possible and reduces learning times and has been validated :
  - "fine-tuning" is from 10 to 25 % of the original learning computation time,
  - in simulation on complex tasks,
  - on actual robot with precise (but slow) task,
  - on actual robot with dynamic motion
- Perspectives :
  - deal with different robot capabilities,

Conclusions and Perspectives

- Conclusion :
  Learning transfer is possible and reduces learning times and has been validated :
  - "fine-tuning" is from 10 to 25 % of the original learning computation time,
  - in simulation on complex tasks,
  - on actual robot with precise (but slow) task,
  - on actual robot with dynamic motion
- Perspectives :
  - deal with different robot capabilities,
  - deal with close loop,

Conclusions and Perspectives

- Conclusion :
  Learning transfer is possible and reduces learning times and has been validated :
  - “fine-tuning” is from 10 to 25 % of the original learning computation time,
  - in simulation on complex tasks,
  - on actual robot with precise (but slow) task,
  - on actual robot with dynamic motion
- Perspectives :
  - deal with different robot capabilities,
  - deal with close loop,
  - define the “transferability” of a task to a robot.

Conclusions and Perspectives

- Conclusion :
  Learning transfer is possible and reduces learning times and has been validated :
  - “fine-tuning” is from 10 to 25 % of the original learning computation time,
  - in simulation on complex tasks,
  - on actual robot with precise (but slow) task,
  - on actual robot with dynamic motion
- Perspectives :
  - deal with different robot capabilities,
  - deal with close loop,
  - define the “transferability” of a task to a robot.

Thank you for your attention

Vidéos are available using :
Simulation,
dynamic task,
precise task.