Intrinsic Motivated Multi-Agent Communication

Chuxiong Sun, Bo Wu, Rui Wang, Xiaohui Hu, Xiaoya Yang and Cong Cong

Nouveau nom d'équipe !!! : SyCoSMA

Intrinsic Motivated Multi-Agent Communication

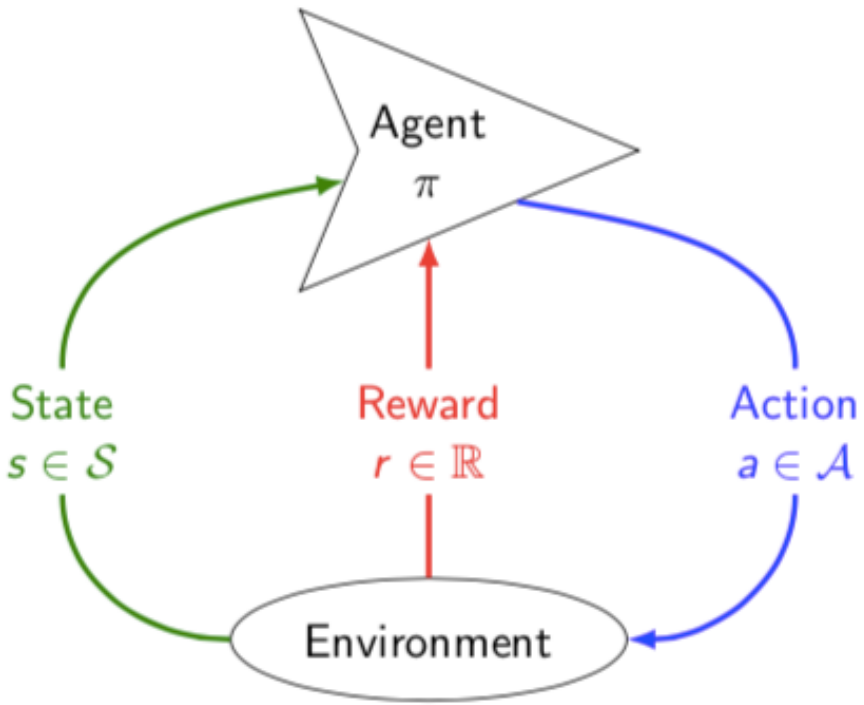
Accepté à AAMAS 2021 (papier court)

- 612 soumissions
- 152 article long (25%)
- 94 résumé étendu (15%)

Intrinsic Motivated Multi-Agent Communication (IMMAC)

- problème de prise de décisions séquentielles, multi-agents, coopératif, PO
  $\rightarrow$ méthodes deep pour MARL
- apprendre quoi, à qui, quand communiquer
  $\rightarrow$ apprentissage de la communication en deep MARL avec mécanismes d'attention
- communication motivée intrinsèquement

Modèle Mono-Agent



Markov Decision Process (MDP)

$$(\mathcal{S}, \mathcal{A}, P, R, P_0, T)$$

$\mathcal{S}$	State set	$\mathcal{A}$	Action set
P	Transition func.	R	Reward func.
P_0	Initial distrib.	T	Horizon

Policy

which action to choose according to current state

$$\pi : \mathcal{S} \to \mathcal{A}$$

Objective

find policy maximizing expected cumulated reward

$$\max_\pi J(\pi) = \mathrm{E}_\pi \left[\sum_{t=0}^{T-1} r_t \right]$$

R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. Ed. by Bradford Books. MIT Press, 1998

Modèle Mono-Agent partiellement observable

Partially Observable MDP (POMDP)

$$(\mathcal{S}, \mathcal{A}, \mathcal{Z}, P, \Omega, R, P_0, T)$$

$\mathcal{Z}$ Observation set | Ω Observation func.

History

state unavailable $\Rightarrow$ gather information

$$h_t = (z_0, a_0, \ldots, z_t) \in \mathcal{H}$$

Policy

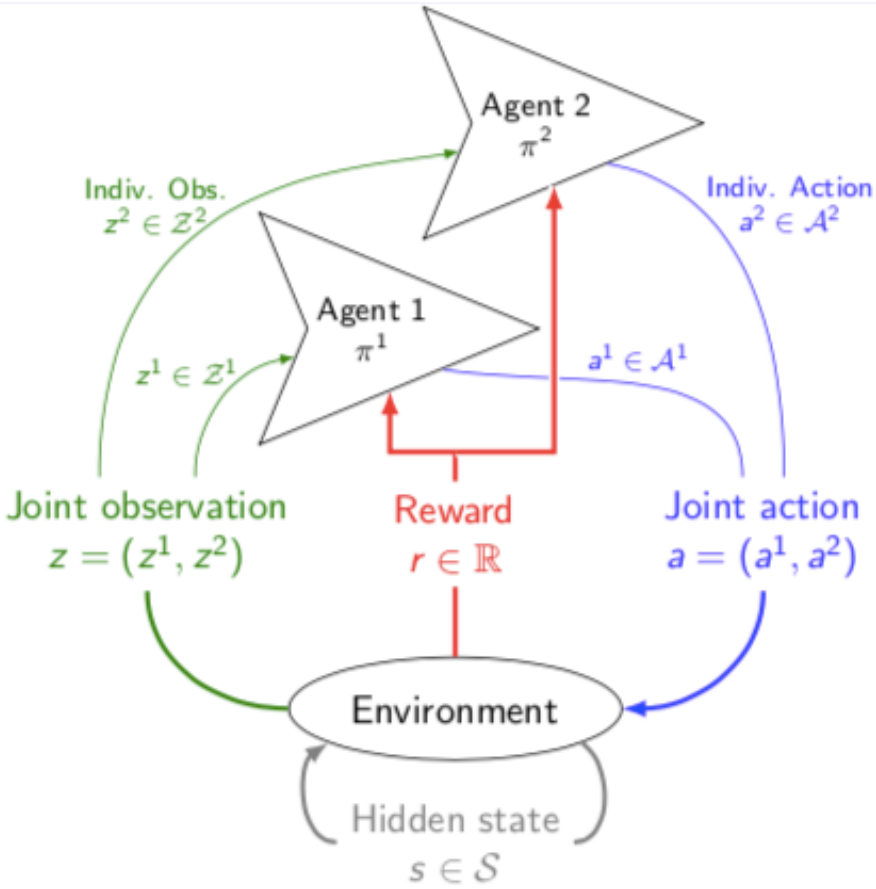
which action to choose according to available info.

$$\pi : \mathcal{H} \to \mathcal{A}$$

Fonction d'approximation utilisant des NN récurrents : deep recurrent Q network (DRQN) [1] approxime $Q(z_t, h_{t-1}, a_t)$ avec un LSTM.

1. Matthew Hausknecht et Peter Stone (2017). *Deep Recurrent Q-Learning for Partially Observable MDPs*. arXiv : 1507.06527 [cs.LG].

Modèle Multi-Agents coopératif partiellement observable



Decentralized POMDP (DecPOMDP)

$$(\mathcal{I}, \mathcal{S}, \mathcal{A}, \mathcal{Z}, P, \Omega, R, P_0, T)$$

$\mathcal{I}$	Agent set		
$\mathcal{A}$	Joint action set	$= \times_i \mathcal{A}^i$	Indiv. action set
$\mathcal{Z}$	Joint obs. set	$= \times_i \mathcal{Z}^i$	Indiv. obs. set

Indiv. histories and policies

Agent i collects **local** info. $h_t^i = (z_0^i, a_0^i, \ldots, z_t^i) \in \mathcal{H}^i$
and chooses indiv. actions according to $\pi^i : \mathcal{H}^i \to \mathcal{A}^i$.

- Coopératif : récompense globale à maximiser, la même pour tous les agents, mais elle dépend de l'action jointe $R : \mathbf{S} \times \mathbf{A} \to \mathbb{R}$
- Multi-agent credit assignment challenge : tous les agents apprennent et explorent en même temps, difficile pour un agent d'estimer l'impact de son action individuelle sur la récompense globale obtenue.

Paradigmes d'apprentissage multi-agents

Deux phases

- phase d'entrainement : les agents peuvent partager de l'information (*free com*) et ont accès à des informations *extra* (l'état complet s par ex.)
- phase de contrôle/d'exécution : les agents exécutent leur politique individuelle

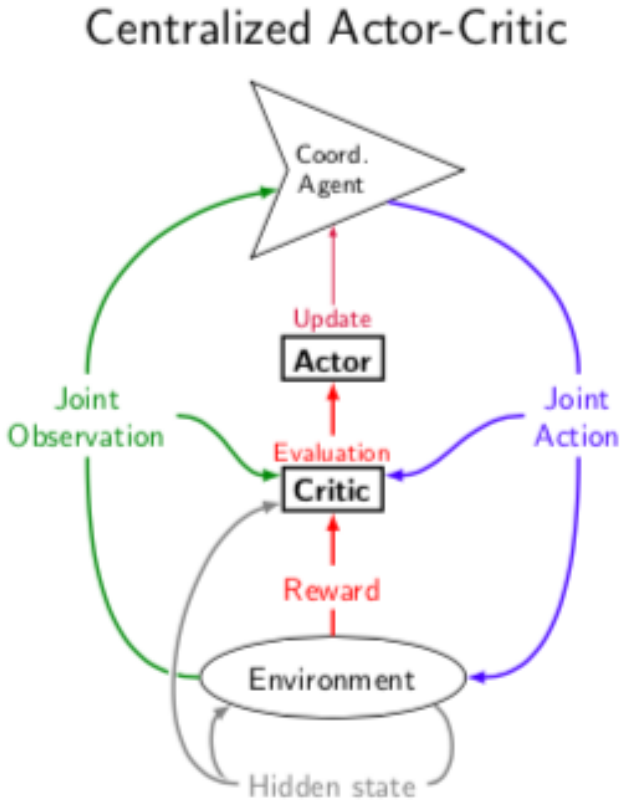
Paradigmes d'apprentissage multi-agents

Entrainement et contrôle centralisés :

Politique et Q-fonction centralisées :

$$Q^C(\boldsymbol{\tau}, \mathbf{a}), \pi^C(\mathbf{a}|\boldsymbol{\tau})$$

- $\mathbf{A}$ espace d'actions jointes exponentiel selon le nombre d'agents
- suppose *full and free* communication à l'exécution



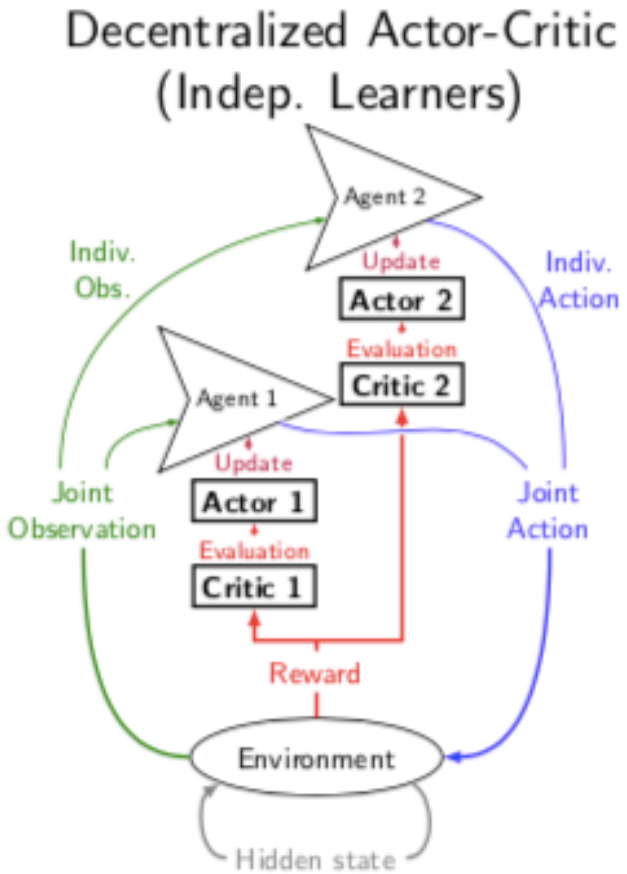
✓ learn collectively
x cannot act independently

Paradigmes d'apprentissage multi-agents

Entrainement et contrôle décentralisés :

Politique et Q-fonction décentralisées :
pour l'agent i : $Q^i(\tau_i, a_i), \pi^i(a_i|\tau_i)$

- pas de coordination
- instabilité car problème de non stationnarité



✗ do not learn collectively
✓ act independently

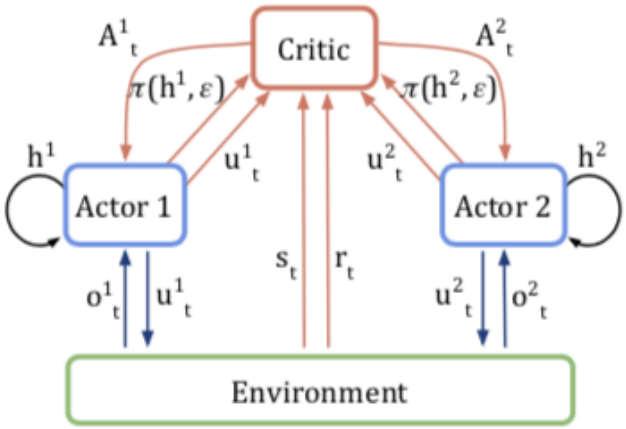
Independant Q-Learning with DQN (IDQN)[2].

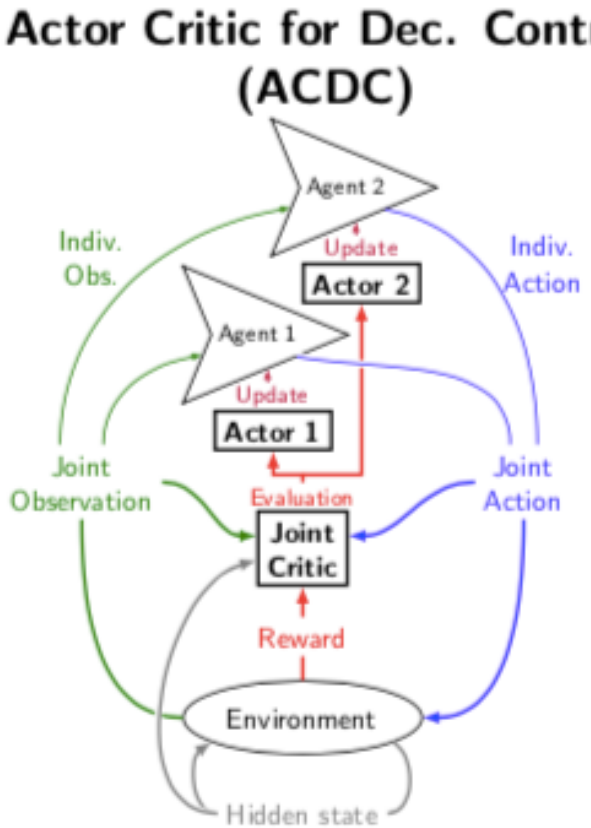
2. Ardi Tampuu et al. (2015). ≪ Multiagent Cooperation and Competition with Deep Reinforcement Learning ≫. In : *CoRR* abs/1511.08779. arXiv : 1511.08779.

Paradigmes d'apprentissage multi-agents

Entrainement centralisé et contrôle décentralisé :

- Q-fonction centralisée utilisée pendant l'apprentissage : $Q^C(\boldsymbol{\tau}, \mathbf{a})$
- politique décentralisée : $\pi^i(a_i|\tau_i)$ pour l'agent i
- Comment extraire/apprendre des π^i décentralisées ?





✓ learn collectively
✓ act independently

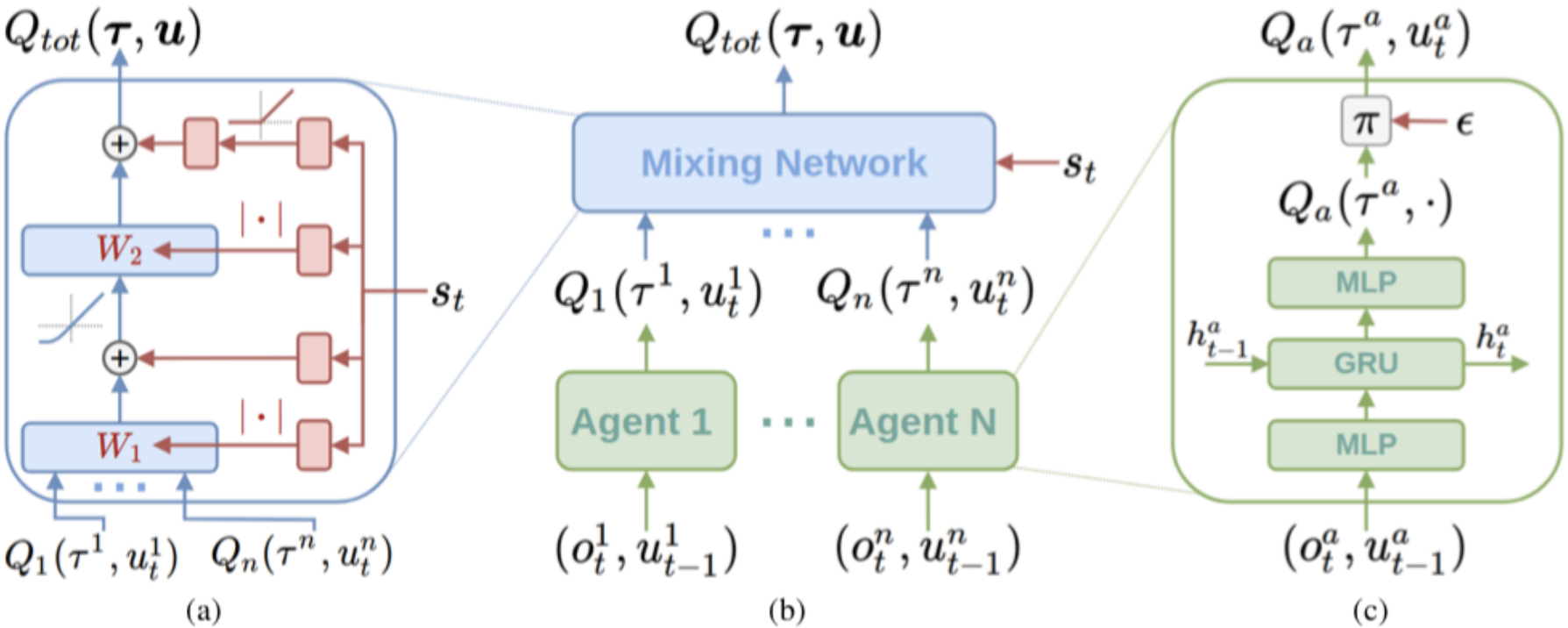
Counterfactual Multi-Agent Policy Gradients (COMA)[3].

3. Jakob N. Foerster et al. (2018). ≪ Counterfactual Multi-Agent Policy Gradients ≫. In : *AAAI*, p. 2974-2982.

Comment extraire des pi décentralisées d'une Q valeur centralisée ?

Apprendre une factorisation de Q^C pour pouvoir extraire des politiques décentralisées :

- VDN [4] : $Q^C(\boldsymbol{\tau}, \mathbf{a}) = \sum_{i=1}^{n} Q^i(\tau^i, a^i; \theta^i)$
- QMIX [5] : *monotonicity constraint* sur lien entre Q^C et Q^i



4. Peter Sunehag et al. (2018). ≪ Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward ≫. In : *AAMAS*, p. 2085-2087.
5. Tabish Rashid et al. (2018). ≪ QMIX : Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning ≫. In : *ICML*. T. 80, p. 4292-4301.

Communication autorisée à l'exécution

Apprentissage *end-to-end* du protocole de communication :

- Paradigme ACDC avec communication autorisée (mais limitée) à l'exécution.
- Améliore la prise de décision avec le partage d'informations observées

- A chaque pas t, l'agent i décide :
  - une action d'environnement a_i^t
  - une action de communication m_i^t, qui sera reçue par les autres agents à $t+1$.
  - multi-round communication

RIAL et DIAL [6], CommNet [7] : Broadcast

6. Jakob N. Foerster et al. (2016). ≪ Learning to Communicate with Deep Multi-Agent Reinforcement Learning ≫. In : *NeurIPS*, p. 2137-2145.
7. Sainbayar Sukhbaatar, Arthur Szlam et Rob Fergus (2016). ≪ Learning Multiagent Communication with Backpropagation ≫. In : *NeurIPS*, p. 2244-2252.

Communication autorisée à l'exécution

Communication ciblée :

- *quand ?* : décider s'il faut communiquer (ATOC, IC3Net [8]) : gating mechanism
- *à qui ?* : sélectionner agents destinataires d'un message (TarMAC) : Utilisation de mécanismes d'attention

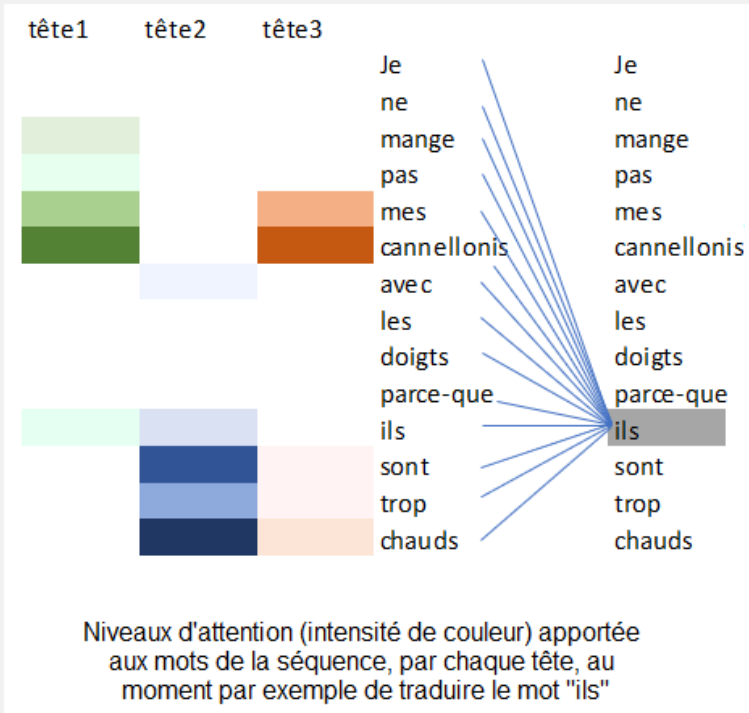
	Decentralized Execution	Targeted Communication	Multi-Round Decisions	Reinforcement Learning
DIAL (Foerster et al., 2016)	Yes	No	No	Yes (Q-Learning)
CommNet (Sukhbaatar et al., 2016)	Yes	No	Yes	Yes (REINFORCE)
VAIN (Hoshen, 2017)	No	Yes	Yes	No (Supervised)
ATOC (Jiang & Lu, 2018)	Yes	No	No	Yes (Actor-Critic)
IC3Net (Singh et al., 2019)	Yes	No	Yes	Yes (REINFORCE)
TarMAC (this paper)	Yes	Yes	Yes	Yes (Actor-Critic)

Table 1: Comparison with previous work on collaborative multi-agent communication with continuous vectors.

8. Amanpreet Singh, Tushar Jain et Sainbayar Sukhbaatar (2019). ≪ Learning when to Communicate at Scale in Multiagent Cooperative and Competitive Tasks ≫. In : *ICLR*.

Mécanismes d'attention

- NN apprend à se concentrer sur des parties spécifiques de données complexes (image, phrase, ...).
- Transformer [9], BERT (Bidirectional Encoder Representations from Transformers) [10]

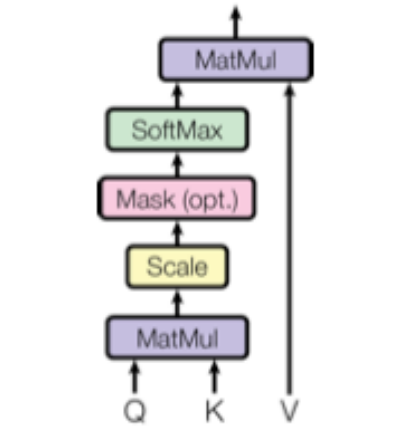


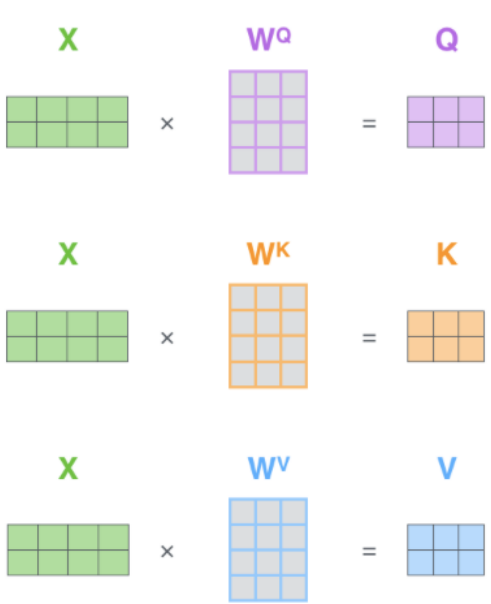
Niveaux d'attention (intensité de couleur) apportée aux mots de la séquence, par chaque tête, au moment par exemple de traduire le mot "ils".

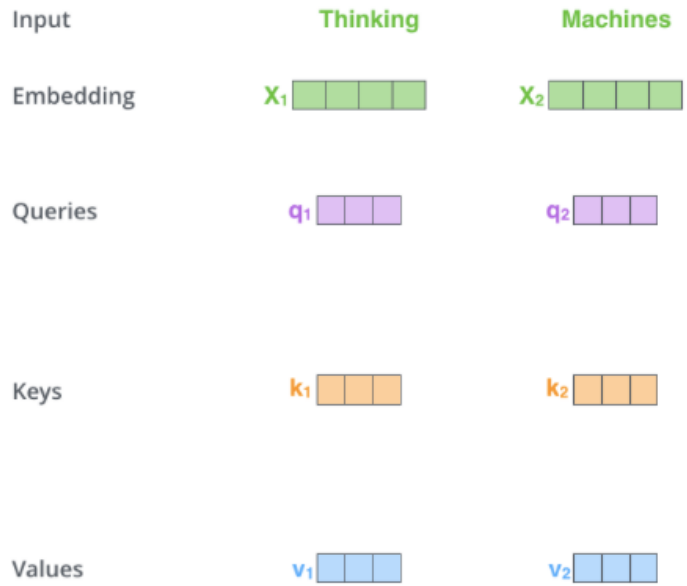
9. Ashish Vaswani et al. (2017). ≪ Attention is All you Need ≫. In : *NeurIPS*.
10. https ://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

Mécanismes d'attention [11]

3 types d'éléments en entrée : Querys, Keys et Values.







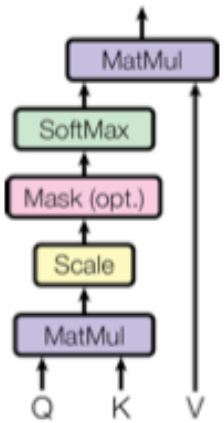
11. Ashish Vaswani et al. (2017). ≪ Attention is All you Need ≫. In : *NeurIPS*.

Mécanismes d'attention [12]

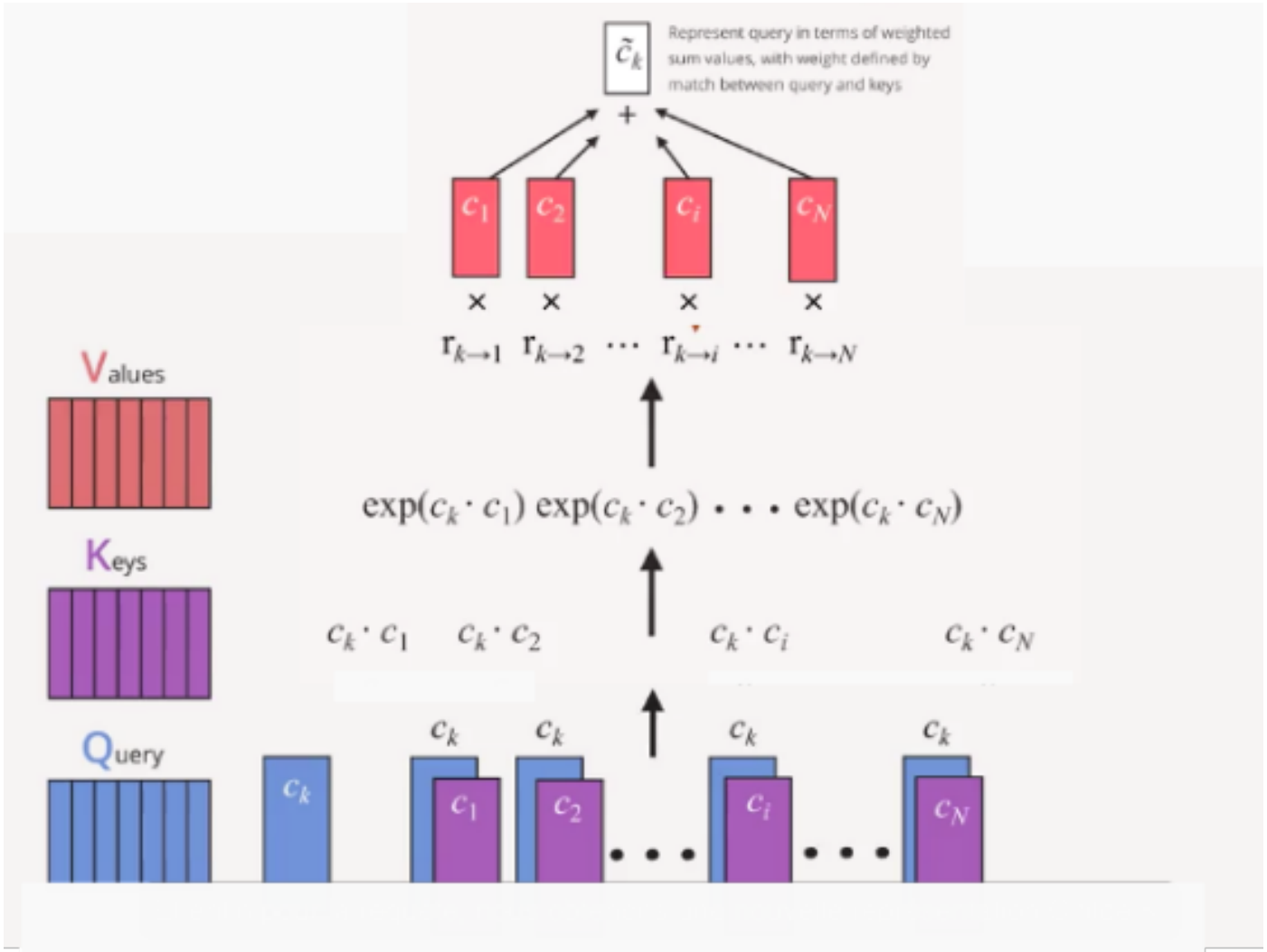
Détail pour une query c_k :

- $r_{k,i}$ poids d'attention/similarité entre query c_k et clé c_i
- calcule $\tilde{c}_k$

Scaled Dot-Product Attention



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



TarMAC : Mécanisme d'attention pour diriger les messages

- Attention générée conjointement par émetteur et receveur
- Les destinataires des messages distinguent les informations importantes dans tous les messages reçus.

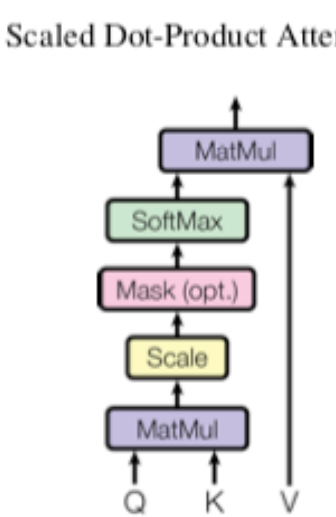
TarMAC [13] : Mécanisme d'attention pour diriger les messages

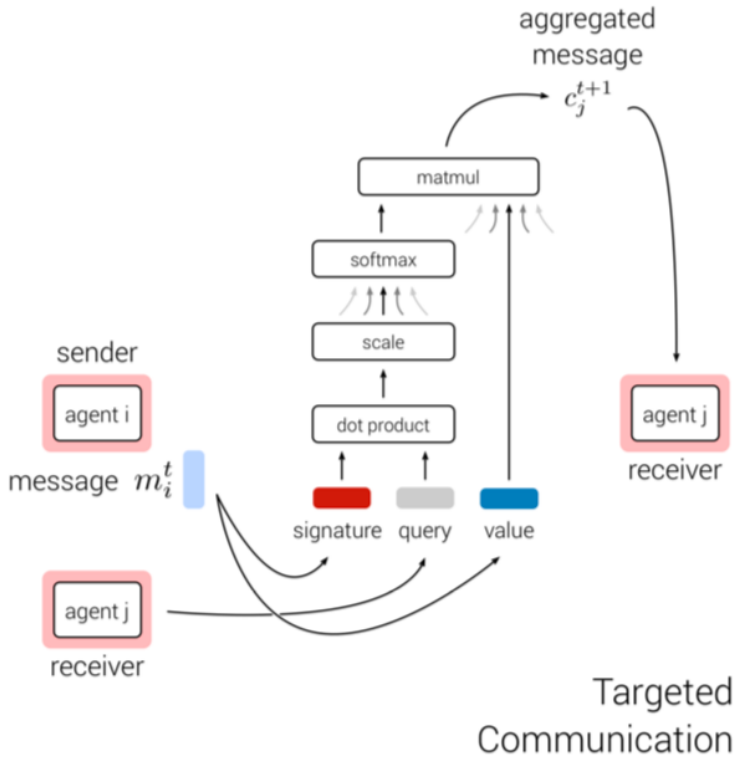
- émetteur i calcule message $m_i = (cle, valeur)$:
- receveur j prédit une query q_j et calcule vecteur des score d'attention α_{ji} à accorder à chaque message reçu
- receveur j calcule un message agrégé c_j qui pondère les valeurs des messages reçus avec leur score d'attention :

$$m_i^t = [\overbrace{k_i^t}^{\text{signature}} \quad \underbrace{v_i^t}_{\text{value}}]$$

$$\alpha_j = \text{softmax}\left[\frac{q_j^{t+1^T} k_1^t}{\sqrt{d_k}} \dots \underbrace{\frac{q_j^{t+1^T} k_i^t}{\sqrt{d_k}}}_{\alpha_{ji}} \dots \frac{q_j^{t+1^T} k_N^t}{\sqrt{d_k}}\right]$$

$$c_j^{t+1} = \sum_{i=1}^{N} \alpha_{ji} v_i^t$$

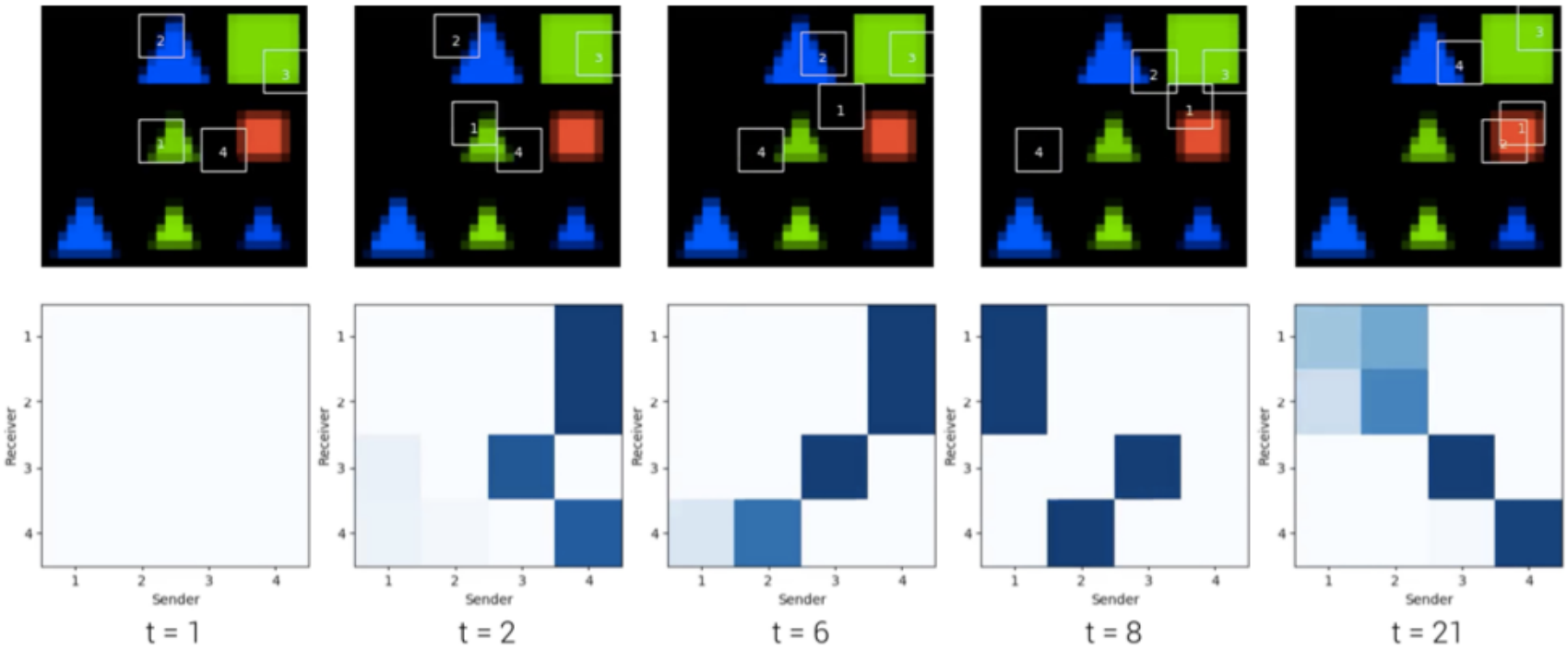




13. Abhishek Das et al. (2019). ≪ TarMAC : Targeted Multi-Agent Communication ≫. In : *ICML*. T. 97. Proceedings of Machine Learning Research, p. 1538-1546.

TarMAC : Résultats

- Objectif $[red, red, green, blue]$ respectivement.
- observations : 5×5 et coordonnées, 4 actions de déplacement
- $r_t = \frac{nbagentsongoal}{nbagents}$



- clé du message : information liée à l'observation de l'émetteur qui est pertinente pour les receveurs et leurs query
- valeur du message : information permettant d'atteindre l'objectif

TarMAC : Résultats

	30 × 30, 4 agents, find[red]	50 × 50, 4 agents, find[red]	50 × 50, 4 agents, find[red, red, green, blue]
No communication	95.3±2.8%	83.6±3.3%	69.1±4.6%
No attention	**99.7**±0.8%	**89.5**±1.4%	82.4±2.1%
TarMAC	**99.8**±0.9%	**89.5**±1.7%	**85.8**±2.5%

Table 2: Success rates on 3 different settings of cooperative navigation in the SHAPES environment.

	Easy	Hard
No communication	84.9±4.3%	74.1±3.9%
CommNet (Sukhbaatar et al., 2016)	99.7±0.1%	78.9±3.4%
TarMAC 1-round	**99.9**±0.1%	84.6±3.2%
TarMAC 2-round	**99.9**±0.1%	**97.1**±1.6%

Table 3: Success rates on traffic junction. Our targeted 2-round communication architecture gets a success rate of 97.1±1.6% on the 'hard' variant, significantly outperforming Sukhbaatar et al. (2016). Note that 1- and 2-round refer to the number of rounds of communication between actions (Equation 4).

Motivation intrinsèque pour la communication multi-agent [14]

IMMAC vs TarMAC

- IMMAC : *Communicate what surprises you*
- TarMAC : *Communicate what rewards you*
- TarMAC combiné à gate fonctionne mal [a]

a. Tonghan Wang et al. (2019). « Learning Nearly Decomposable Value Functions Via Communication Minimization ». In : *CoRR* abs/1910.05366. arXiv : 1910.05366.

- paradigme ACDC basé sur QMIX avec communication à l'exécution
- combine *quand* et *à qui* communiquer :
  - attention pour pondérer les messages reçus
  - filtrage pour communiquer uniquement les informations utiles
  - basé sur une mesure intrinsèque de la nouveauté des observations locales
- combinaison de récompenses intrinsèques et extrinsèques (TarMAC)

14. Chuxiong Sun et al. (2021). « Intrinsic Motivated Multi-Agent Communication ». In : *AAMAS*.

Random Network Distillation [16]

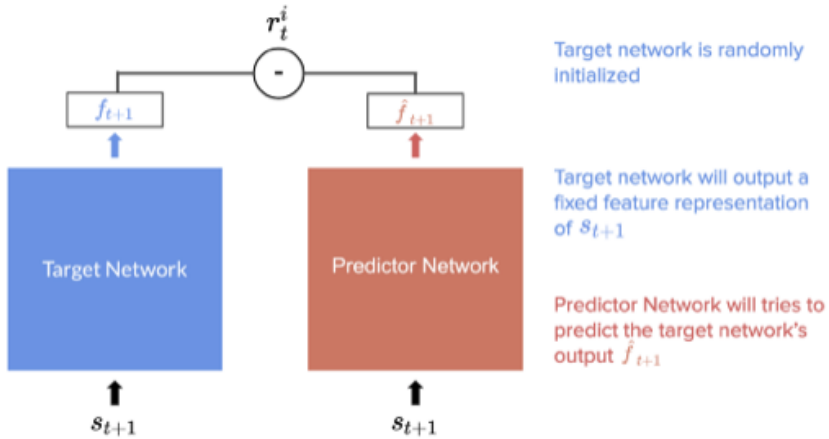
- Récompense intrinsèque r^i : bonus d'exploration mesurant la nouveauté des états

- Utilise 2 NNs :
  - NN cible figé
  - NN prédictif prédit la sortie du NN cible
  - r^i mesure l'erreur de prédiction

Knowledge acquisition
Exploration
Prediction error
State novelty
Novelty as discrepancy towards other states
Information gain
Empowerment
Learning a relevant state representation
State space as a measure of distance
One feature for one object of interaction

Skill learning
Skill abstraction
Building the goal space from the
Mutual information between goa
Curriculum learning
Goal sampling
Multi-armed bandit
Adversarial training

Table 2: Classification of the use of IMs in DRL.



Erreur de prédiction élevée pour les états peu rencontrés.
15

15. Arthur Aubret, Laetitia Matignon et Salima Hassas (2019). *A survey on intrinsic motivation in reinforcement learning*. arXiv : 1908.06976 [cs.LG].
16. Yuri Burda et al. (2019). ≪ Exploration by random network distillation ≫. In : *ICLR*.

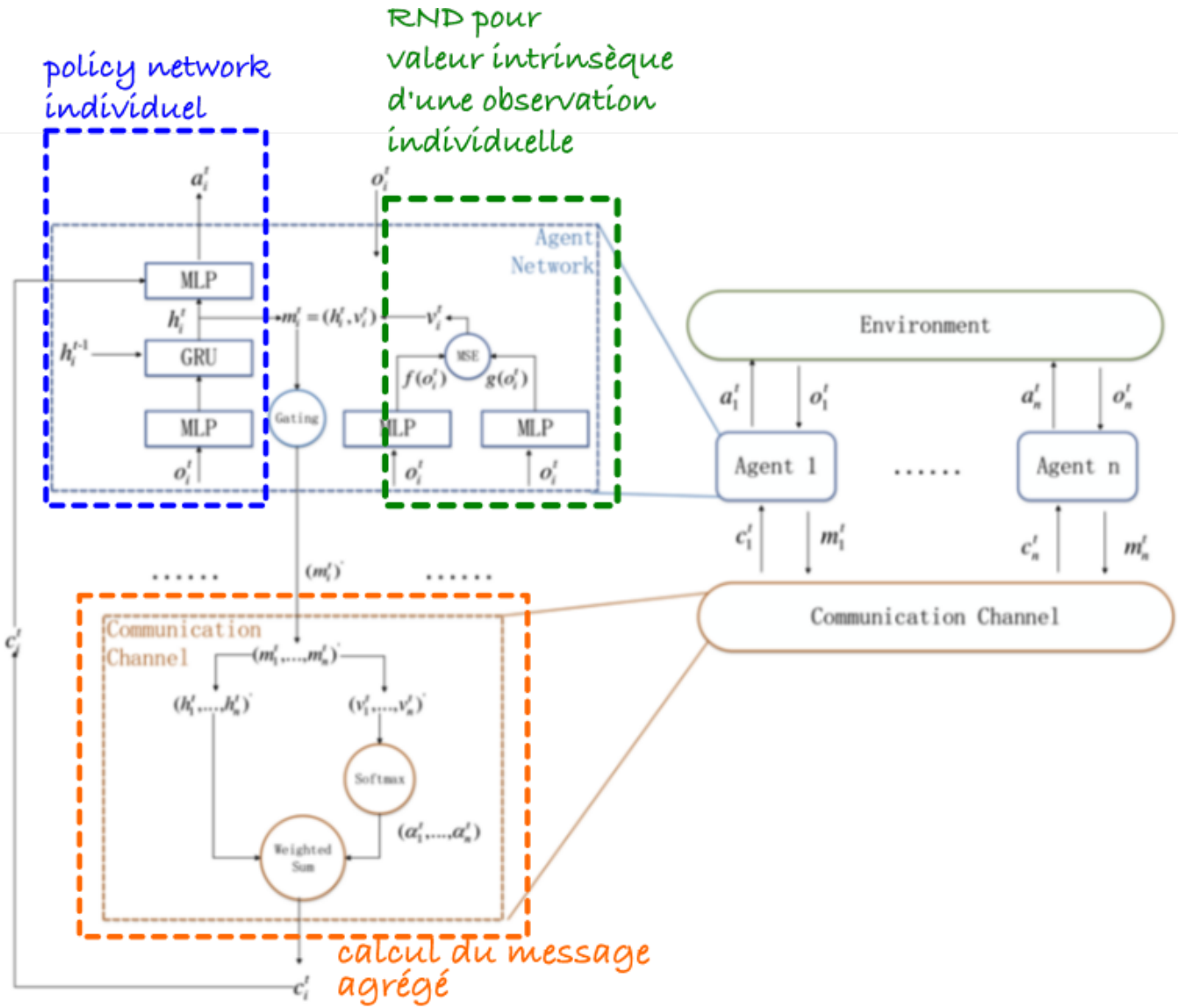
Random Network Distillation

Utilisation d'un NN cible fixe et déterministe réduit les autres sources d'erreurs de prédiction :

In general, prediction errors can be attributed to a number of factors:

1. *Amount of training data*. Prediction error is high where few similar examples were seen by the predictor (epistemic uncertainty).
2. *Stochasticity*. Prediction error is high because the target function is stochastic (aleatoric uncertainty). Stochastic transitions are a source of such error for forward dynamics prediction.
3. *Model misspecification*. Prediction error is high because necessary information is missing, or the model class is too limited to fit the complexity of the target function.
4. *Learning dynamics*. Prediction error is high because the optimization process fails to find a predictor in the model class that best approximates the target function.

Motivation intrinsèque pour la communication multi-agent



- Gate : $v_i > \delta$

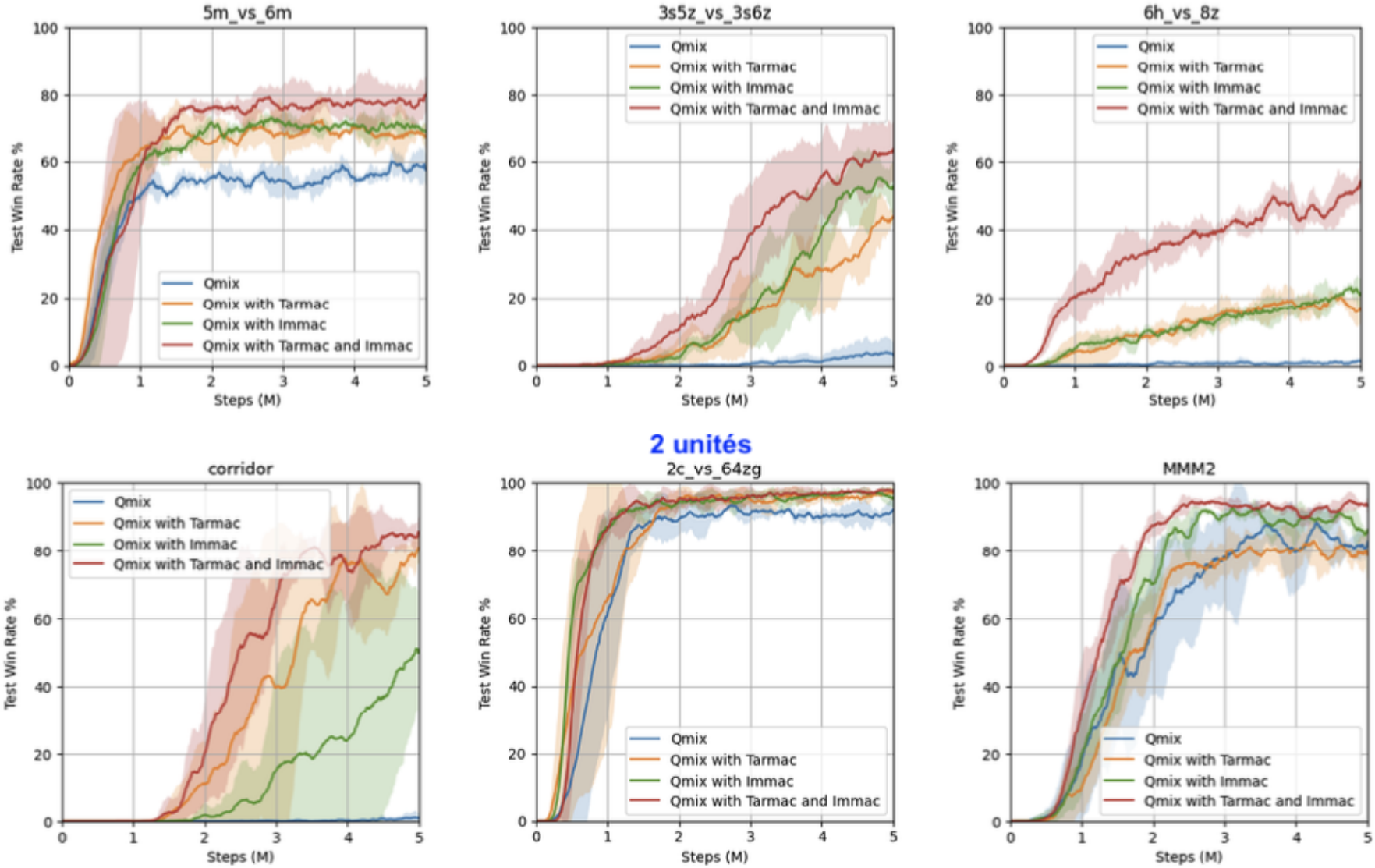
Motivation intrinsèque pour la communication multi-agent

Comparaisons

- pas de communication (QMIX)
- communication basée sur récompense extrinsèque et ciblée (TarMAC)
- communication basée sur récompense intrinsèque sans filtrage (QMIX + IMMAC , $\delta = 0$)
- communication basée sur récompense intrinsèque et extrinsèque sans filtrage (QMIX + IMMAC + TarMAC , $\delta = 0$)

Motivation intrinsèque pour la communication multi-agent

StarCraft Multi-Agent Challenge (SMAC) :



Motivation intrinsèque pour la communication multi-agent

- rôle du filtrage δ (QMIX + IMMAC)

Table 2: Communication rate %

	$\delta = 0$	$\delta = 0.005$	$\delta = 0.008$	$\delta = 1$
5m_vs_6m	100.0	56.5	44.8	0.0
3s5z_vs_3s6z	100.0	34.1	27.5	0.0
6h_vs_8z	100.0	58.8	45.1	0.0
corridor	100.0	46.5	35.6	0.0
2c_vs_64zg	100.0	45.9	40.5	0.0
MMM2	100.0	35.9	27.1	0.0

Table 3: Test win rate of the last 25×10^4 steps %

	$\delta = 0$	$\delta = 0.005$	$\delta = 0.008$	$\delta = 1$
5m_vs_6m	70.7	76.5	77.8	58.2
3s5z_vs_3s6z	53.8	65.1	54.7	3.7
6h_vs_8z	21.7	29.4	27.2	1.2
corridor	48.3	47.6	39.4	0.8
2c_vs_64zg	96.0	95.8	98.4	91.5
MMM2	86.9	84.8	92.4	82.2