Information-Bottleneck-Based Behavior Representation Learning for Multi-agent Reinforcement Learning

Yue Jin[1] Shuangqing Wei[2] Jian Yuan[1] Xudong Zhang[1]

[1] Department of Electronic Engineering, Tsinghua University
[2] School of Electrical Engineering and Computer Science, Louisiana State University

Outline

Outline

1. Background
2. Method
3. Experiments
4. Conclusion

I. BACKGROUND

- Representation learning in DRL
  - Why representation learning
    - Learn informative and effective features of a task
    - Efficiency, robustness, and scalability (Multi-agent DRL)
  - Early works combine deep auto-encoder with DRL (Lang et al. 2010)
  - Recent works involve
    - advanced unsupervised learning to extract **discriminative** features from observations (Laskin et al. 2020)
    - information estimation methods to learn **compact/ task-relevant** representation for DRL (Pacelli et al. 2020)
    - model-based DRL to learn abstract state representation/ **low-dimensional** representation of the environment (François-Lavet et al. 2019)

Lang et al. 2010. **Deep Auto-Encoder Neural Networks in Reinforcement Learning.**
Laskin et al. 2020. **CURL: Contrastive Unsupervised Representations for Reinforcement Learning.**
Pacelli et al. 2020. **Learning Task-Driven Control Policies via Information Bottlenecks.**
François-Lavet et al. 2019. **Combined Reinforcement Learning via Abstract Representations.**

I. BACKGROUND

- Issues of representation learning in MADRL
  - Teammate/opponent-relevant
  - What to represent
  - Combination with MADRL
- Previous works
  - Design general frameworks to combine teammate/opponent representation with MADRL (He et al. 2016)
  - Represent other agents' behaviors **implicitly** using their positions at adjacent time steps (Jin et al. 2020, Jin et al. 2021)
- This work focuses on
  - Explicit and interpretable other agents' behavior representation learning based on our previous work (Jin et al. 2020)
  - Information compression and retention in the representation
  - More efficient and scalable algorithm

Jin et al. 2020. **Stabilizing Multi-Agent Deep Reinforcement Learning by Implicitly Estimating Other Agents' Behaviors.**
Jin et al. 2021. **Hierarchical and Stable Multiagent Reinforcement Learning for Cooperative Navigation Control.**
He et al. 2016. **Opponent Modeling in Deep Reinforcement Learning.**

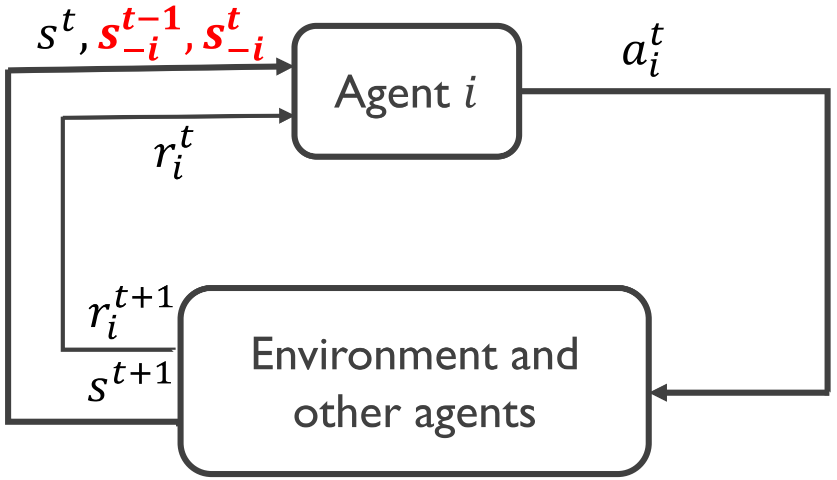
Outline

1. Background
2. Method
3. Experiments
4. Conclusion

2. METHOD

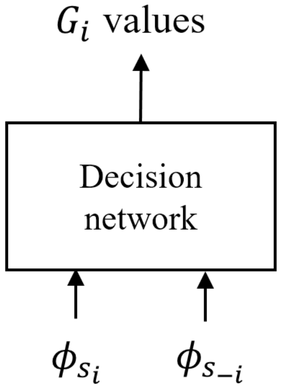
- From implicit learning to explicit learning of other agents' behavior representation
  - Implicit action representation learning

    SMADQN (Jin et al. 2020)
    - Define an extended action-value function G for each agent
      - Incorporates the states of other agents at two adjacent time steps into its input
    - Design a stabilized MADRL algorithm



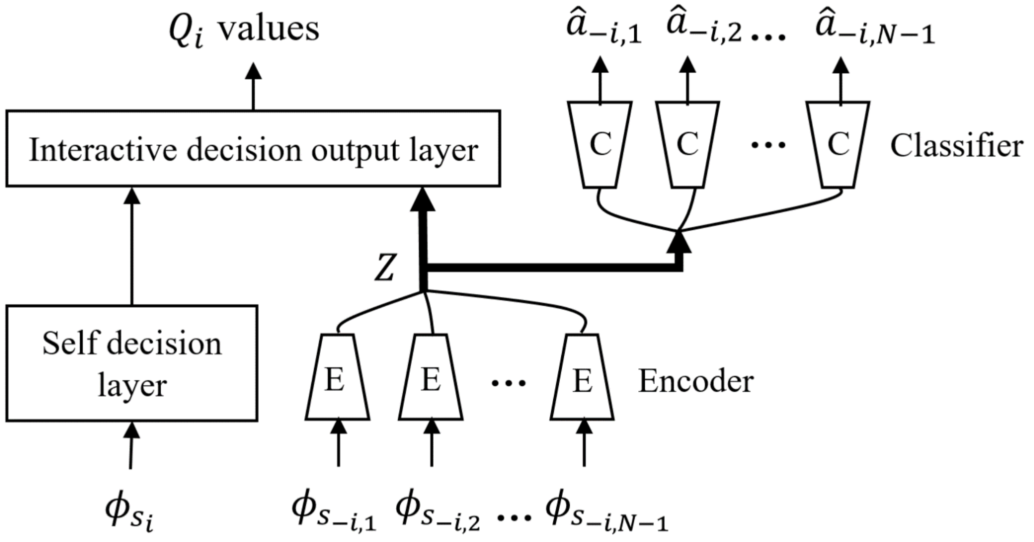
$$L = \mathbb{E}_{s^t, s^{t+1}, a_i^t}\left[\left(r_i^{t+1} + \gamma \max_{a_i^{t+1}} G_i(s^{t+1}, s_{-i}^{t}, s_{-i}^{t+1}, a_i^{t+1}) - G_i(s^{t}, s_{-i}^{t}.s_{-i}^{t+1}, a_i^{t})\right)^2\right]$$

- Other agents' behavior representation is implicit



2. METHOD

- From implicit learning to explicit learning of other agents' behavior representation
  - Explicit action representation learning
    - An encoder to learn low-dimensional features of other agents' actions using their states at adjacent time steps as inputs
    - A classifier to predict the actions via supervised learning
    - Leverage cross entropy as part of the loss
- Information compression and retention
  - Relevant to the task
  - Relevant to other agents
  - Filtering out irrelevant information



2. METHOD

- IBORM: **I**nformation-**B**ottleneck-based **O**ther agents' behavior **R**epresentation learning for **M**ulti-agent reinforcement learning
  - Information bottleneck principle (Tishby et al. 2015)
    - Extracting an optimal representation Z of a random variable X about another correlated random variable Y while minimizing the amount of irrelevant information
    - is formulated as minimizing

$$\mathcal{L}(p(z|x)) = I(X;Z) - \kappa I(Z;Y)$$

    - (Y, X, Z) forms a Markov chain, $Y \rightarrow X \rightarrow Z$

Tishby et al. 2015. **Deep Learning and the Information Bottleneck Principle.**

2. METHOD

- IBORM: **I**nformation-**B**ottleneck-based **O**ther agents' behavior **R**epresentation learning for **M**ulti-agent reinforcement learning
  - Based on IB principle, we constrain the representation learning (encoder) by minimizing

$$\mathcal{L}(\alpha) \triangleq I(\phi_{s_{-i,j}}; ENC_i^{\alpha}(\phi_{s_{-i,j}}) - \kappa I(ENC_i^{\alpha}(\phi_{s_{-i,j}}); a_{-i,j})$$

    - $a \rightarrow \phi_s \rightarrow z$
    - $z = ENC(\phi_s)$
  - Overall objective of IBORM
    - To minimize

$$L_i(\alpha, \beta, \theta) = J_i^{CE}(\alpha, \beta) + \lambda_1 J_i^{DRL}(\alpha, \theta) + \lambda_2 I(\phi_{s_{-i,j}}, ENC_i^{\alpha}(\phi_{s_{-i,j}})) - \lambda_3 I(ENC_i^{\alpha}(\phi_{s_{-i,j}}), a_{-i,j})$$

2. METHOD

- Mutual information estimation in IBORM
  - Leverage Mutual Information Neural Estimator (MINE) (Belghazi et al. 2018)
    - Estimate the mutual information between two variables X and Z as

$$\widehat{I(X,Z)} = \sup_{\omega\in\Omega} \mathbb{E}_{\mathbb{P}_{XZ}}[T_\omega(x,z)] - \log(\mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Z}[e^{T_\omega(x,z)}])$$

with a trainable neural network T_ω

  - IBORM uses two MINE networks corresponding to

$I(\phi_{s_{-i,j}}; ENC_i^\alpha(\phi_{s_{-i,j}})$ and $I(ENC_i^\alpha(\phi_{s_{-i,j}}); a_{-i,j})$.

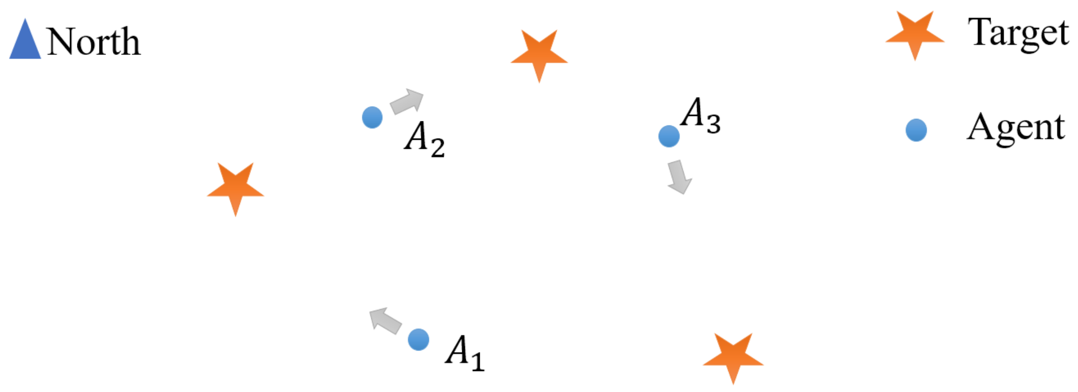
Belghazi et al. 2018. **Mutual Information Neural Estimation.**

Outline

1. Background
2. Method
3. Experiments
4. Conclusion

3. EXPERIMENTS

- Multi-agent cooperative navigation task with the same settings used in our previous work (Jin et al. 2020)
  - Agents need to cooperate through motions to reach a set of targets with the minimum time cost
  - Randomly generate positions of targets and agents in every episode
  - Different numbers of targets and agents (N =3, 4, 5, 6, and 7)



Observation: positions of targets and the current and last positions of other agents

Action: select a target to head for $a_i \in [1, N]$

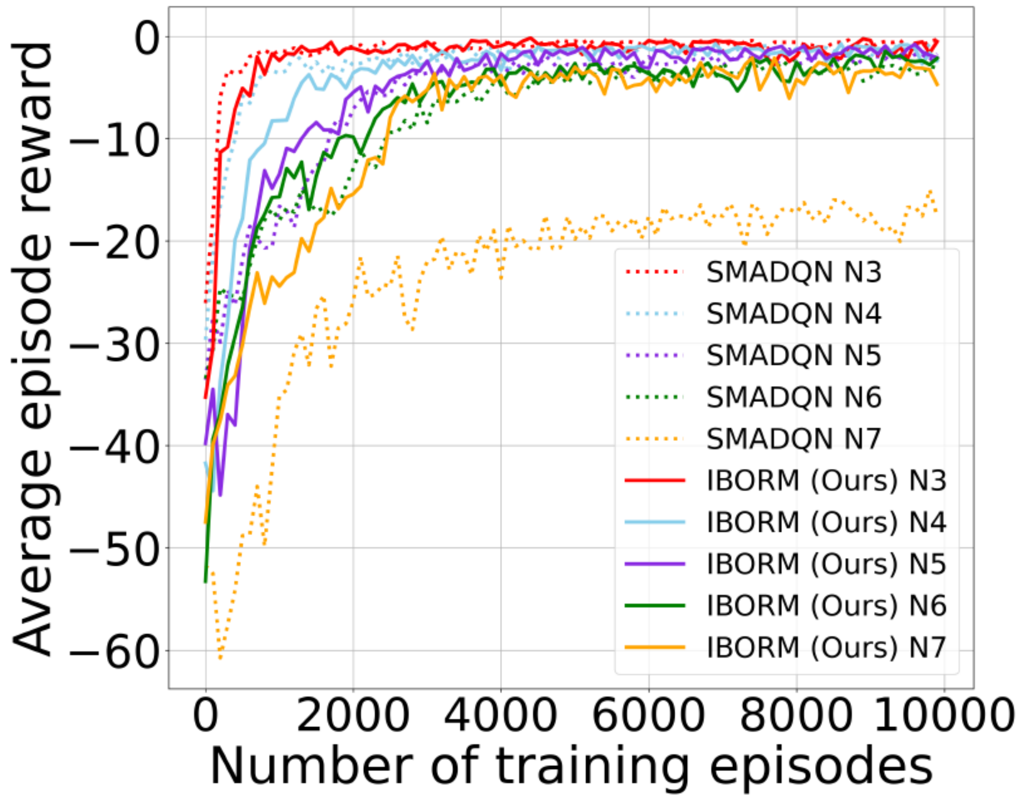
Assuming a constant speed

3. EXPERIMENTS

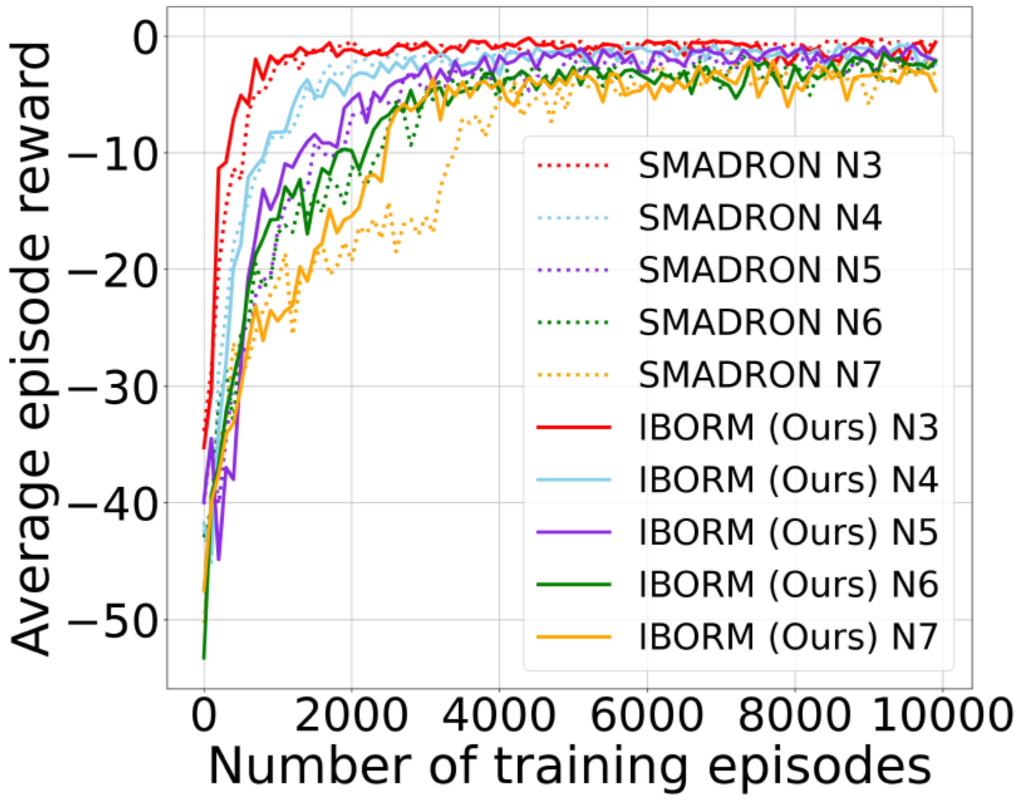
- Training performance

IBORM learns faster than the other two methods

IBORM vs. SMADQN (implicit representation)



IBORM vs. SMADRON (without information constraints)



3. EXPERIMENTS

- Testing performance
  - One thousand randomly generated tasks
  - Success: agents arrive at different targets without conflicts

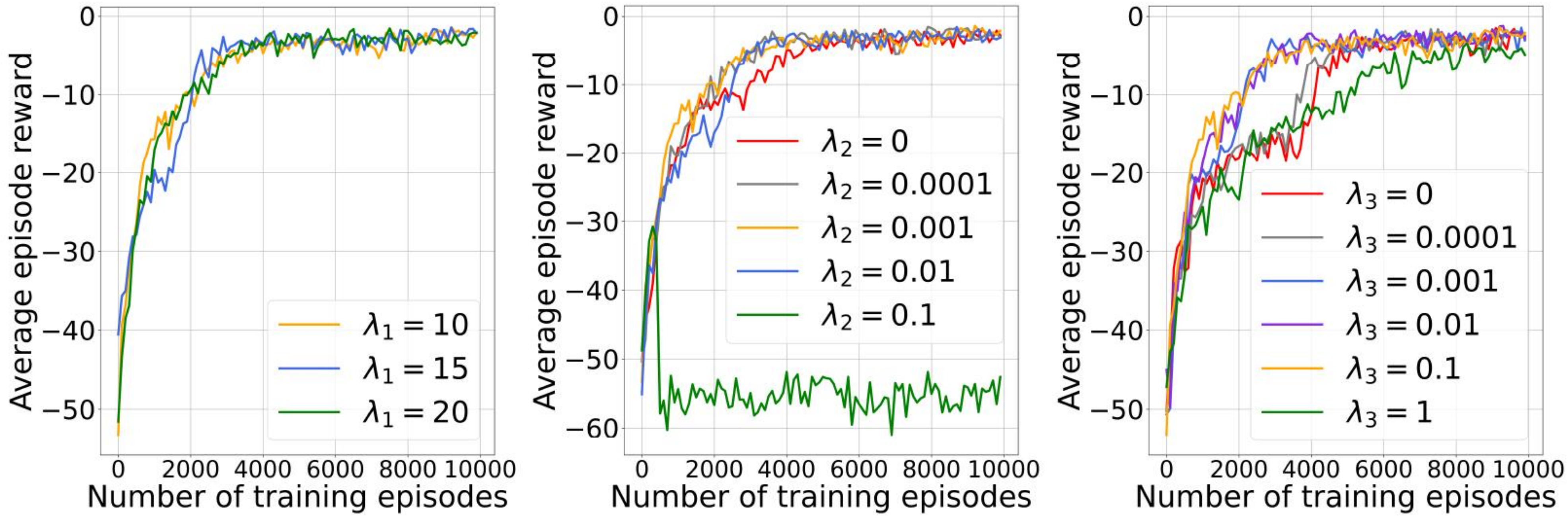
Table 1: Test results of different methods.

Method	Success rate				
	N=3	N=4	N=5	N=6	N=7
SMADQN	98.2%	97.8%	96.1%	91.2%	0.0%
SMADRON	98.9%	96.9%	92.9%	93.5%	82.5%
IBORM	**99.3**%	**98.1**%	**97.1**%	**93.5**%	**87.8**%

3. EXPERIMENTS

- Further study
  - Effect of different terms in IBORM's objective function

$$L_i(\alpha,\beta,\theta) = J_i^{CE}(\alpha,\beta) + \lambda_1 J_i^{DRL}(\alpha,\theta) + \lambda_2 I(\phi_{s_{-i,j}}, ENC_i^{\alpha}(\phi_{s_{-i,j}})) - \lambda_3 I(ENC_i^{\alpha}(\phi_{s_{-i,j}}), a_{-i,j})$$



(a) (b) (c)

Outline

1. Background
2. Method
3. Experiments
4. Conclusion

4. CONCLUSION

We propose IBORM to facilitate MADRL by learning representation regarding other agents' behaviors in an **explicit and more interpretable** manner compared with our previous work.

We leverage information bottleneck principle to push the representation to be **compact** and **relevant to both the task and other agents' behaviors**.

Experimental results demonstrate that IBORM **learns faster** and the resulting policies can **achieve higher success rate** consistently, as compared with implicit behavior representation learning (SMADQN) and explicit behavior representation learning (SMADRON) without considering information compression and utility.

Thanks!

e-mail:
jiny23@126.com