CURIOSITY-DRIVEN EXPLORATION BY SELF-SUPERVISED PREDICTION

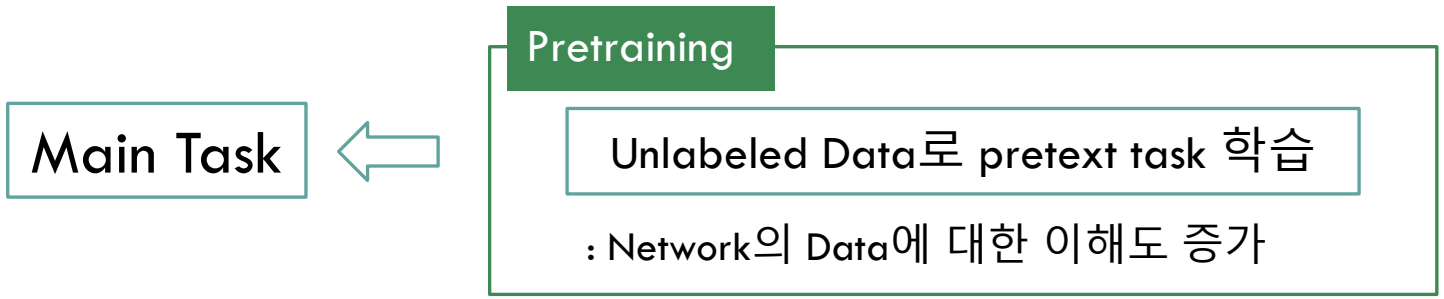
D. Pathak et al. (2017)

2020.6.6

최하늘 (Haneul Choi) caelum02@snu.ac.kr

BACKGROUND SELF-SUPERVISED LEARNING

Pretext task : Network의 **Data 이해도**를 높이기 위해 설정한 문제



Self-Supervised Learning의 목표 : **Feature Extraction!**

SPARSE REWARD PROBLEM – IN REAL-WORLD

Atari Breakout

: Dense Reward

Real-World Scenarios

: Extremely Sparse Rewards

Sparse Reward Problem

- Real world scenario 에서는 reward가 sparse한 경우가 많음
- 많은 경우, 확실한 action에 대해서만 reward를 주기 때문에 sparse

Random Exploration에서 reward를 얻기 쉽지 않음

효율적인 Exploration Policy 필요!

https://gym.openai.com/envs/Breakout-v0/

INTRINSIC REWARD

Intrinsic ~ 내재적

사람은 reward가 없어도 '호기심'이라는
내재적인 동기가 Exploration 유도

- Ryan et al. (2000), Silvia (2012)

RL에도 agent의 intrinsic reward 도입하려 시도 해왔음

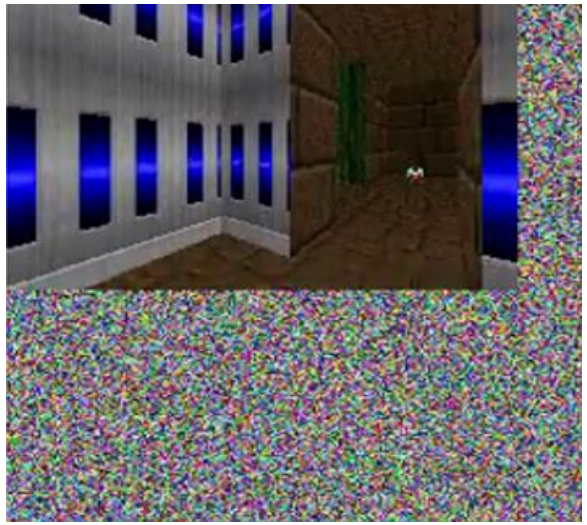
1. State Novelty *Bellemare et al. (2016), Lopes et al. (2012), Poupart et al. (2006)*
   - 방문횟수 적은 State를 방문하도록 유도
2. Prediction Error *Houthooft et al. (2016), Mohemed & Rezende (2015) and more …*
   - Action에 의한 State의 변화에 대해 prediction error 최소화

$$r_{intrinsic} = ||\hat{s}_{t+1} - s_{t+1}||$$

두 방식 모두 **새로운 Observation을 incentivize**

Difficulties

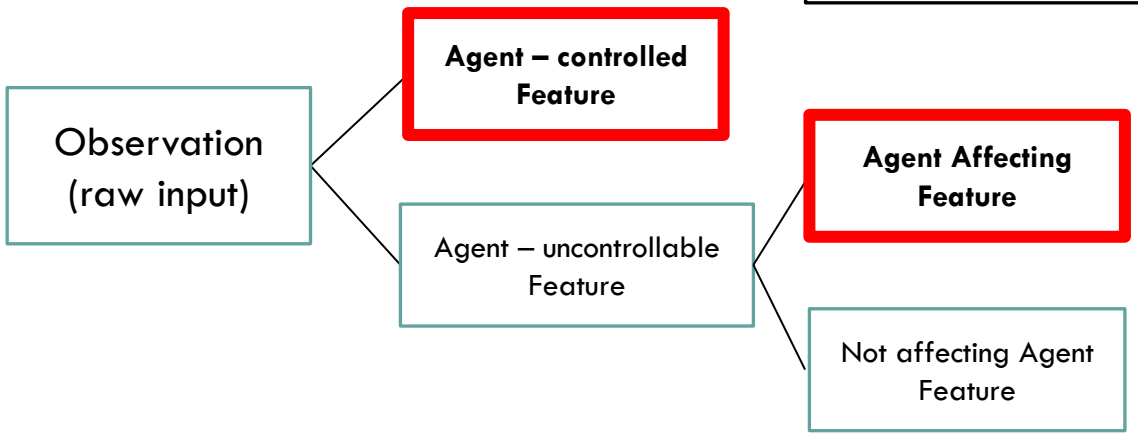
- High Dimensional Observation 다루기 힘듦
- Environment Dynamics의 Stochasticity에 영향 받음



- Every state is Novel
- Impossible to reduce Prediction Error

➡ Agent에게 유의미한 Feature만 Predict 하자!

: agent related feature



논문의 Key Idea

$\phi(S) :=$ *state* S 의 Agent-Related Feature Vector

$$r_{intrinsic} = \left\| \hat{s}_{t+1} - s_{t+1} \right\| \Rightarrow \left\| \hat{\phi}(s_{t+1}) - \phi(s_{t+1}) \right\|$$

Feature Extraction? : Self-Supervised Learning!

Also helps **Generalization**!

INTRO SUMMARY - KEY IDEAS OF PAPER

$$r_t = r_t^i + r_t^e$$

r^i: intrinsic reward
r^e: extrinsic reward

$$r_t^i := \frac{\eta}{2} ||\hat{\phi}(s_{t+1}) - \phi(s_{t+1})||_2^2$$

➡ Agent-related Feature $\phi(s)$만 Predict!

Feature Prediction

Forward Model f with parameter θ_F ;

$$\hat{\phi}(s_{t+1}) = f(\phi(s_t), a_t; \theta_F)$$

$$L_F\left(\phi(s_{t+1}), \hat{\phi}(s_{t+1})\right) = \frac{1}{2} ||\hat{\phi}(s_{t+1}) - \phi(s_{t+1})||_2^2$$

Feature Extraction

- Self-Supervised Learning
- Pretext task : s_t, s_{t+1}에서 a_t 추론

➡ 일반적인 agent-related feature extraction 방법론 제안!

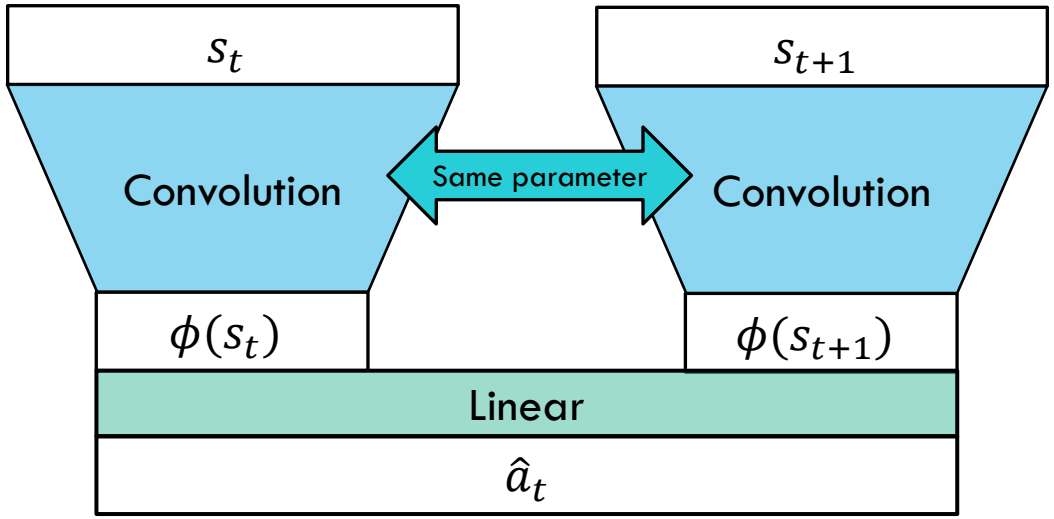
FEATURE EXTRACTION WITH SELF-SUPERVISION

Pretext task : s_t, s_{t+1}에서 a_t 추론

Inverse Dynamics Model g with parameter θ_I ;

$$\hat{a}_t = g(s_t, s_{t+1}; \theta_I)$$

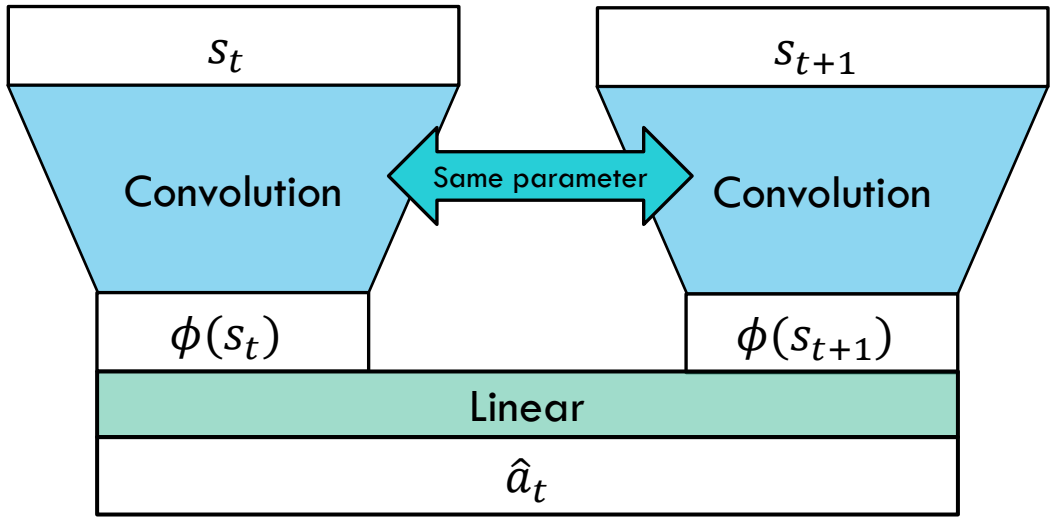
$$\min_{\theta_I} L_I(\hat{a}_t, a_t)$$

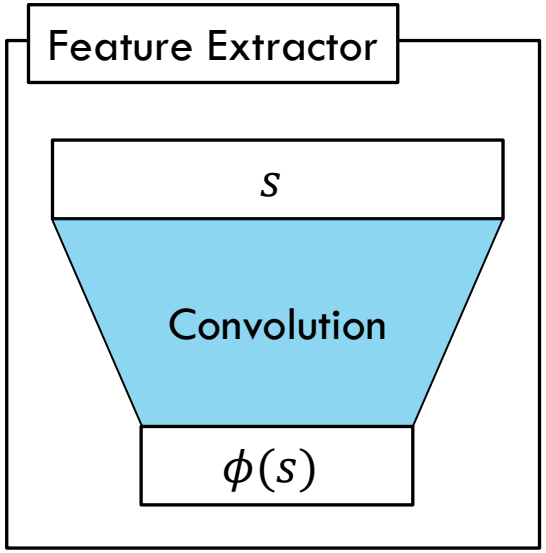


Convolution Layer : raw observation에서 **feature extract**

Linear Layer (Fully Connected Layer) : extract된 feature로 **각 action에 대한 logit estimate**

FEATURE EXTRACTION WITH SELF-SUPERVISION

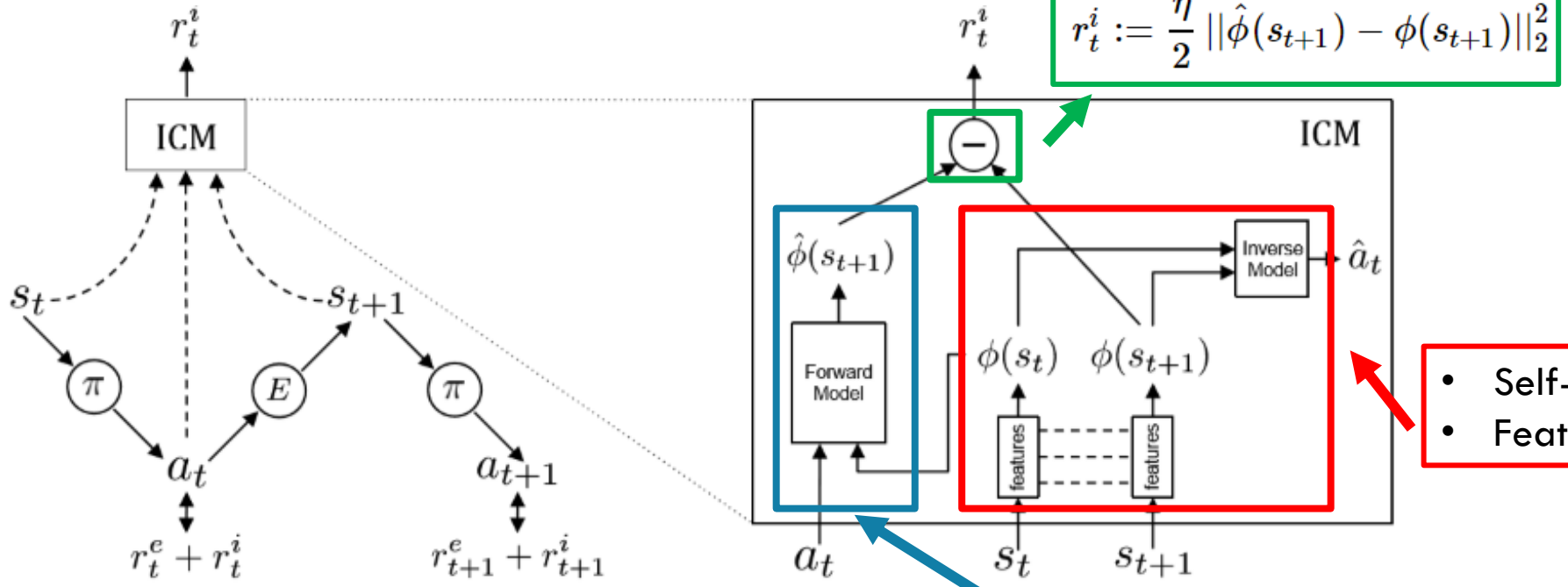




s_t, s_{t+1}에서 a_t를 잘 추측하기 위해 Network가 action과 관계 있는 feature를 학습!

pretext task를 해결함으로써 feature extraction 가능

INTRINSIC CURIOSITY MODULE (ICM)



$$r_t^i := \frac{\eta}{2} ||\hat{\phi}(s_{t+1}) - \phi(s_{t+1})||_2^2$$

- Self-supervised learning
- Feature Extractor 학습

Feature Prediction

- Forward model과 Feature Extractor 동시에 optimize
- Intrinsic reward (curiosity)에 의해 Policy가 forward model이 예측 못하는 (s, a)를 방문하도록 incentivize

OVERALL OPTIMIZATION PROBLEM

- λ : Policy gradient loss 가중치
- β : Forward Model과 Inverse Model Loss 가중치

실험에서는 $\lambda = 0.1, \beta = 0.2$ 사용

Agent의 Environment에 대한 정보 습득

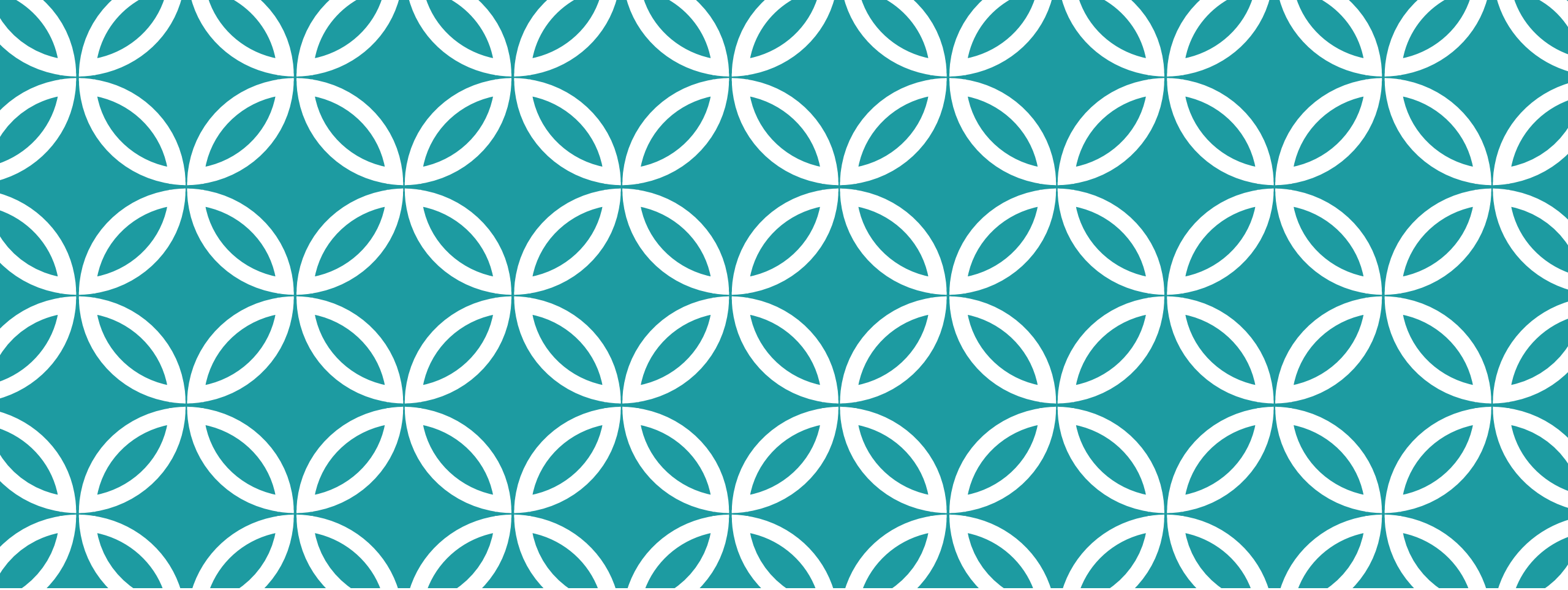
이미 잘 아는 (s, a)에 대해 curiosity 줄임

$$\min_{\theta_P,\theta_I,\theta_F} \left[-\lambda \mathbb{E}_{\pi(s_t;\theta_P)}[\Sigma_t r_t] + (1-\beta) L_I + \beta L_F \right]$$

Policy는 Intrinsic Reward (curiosity) maximize

→ 새로운 state explore하도록 Policy 학습

Agent-related Feature 더욱 잘 추출하도록 학습



EXPERIMENTS

ICM의 효과

1. Sparse Reward Problem에 도움
   - Environment에서 reward가 없어도 학습을 지속할 수 있도록 함
2. Agent가 효율적인 Exploration Policy를 학습하도록 유도
   - Agent가 Environment에 대한 새로운 지식을 습득하도록 유도
3. Generalization에 도움

논문에서 ICM의 세가지 효과를 실험적으로 증명해 냄

THREE EXPERIMENTS

1. Sparse Extrinsic Reward Setting
2. No Extrinsic Reward Setting

   (Only Intrinsic Rewards)
3. Generalization

Used Environments

1. OpenAI gym DoomMyWayHome-v0 (VizDoom)
2. Super Mario Bros.

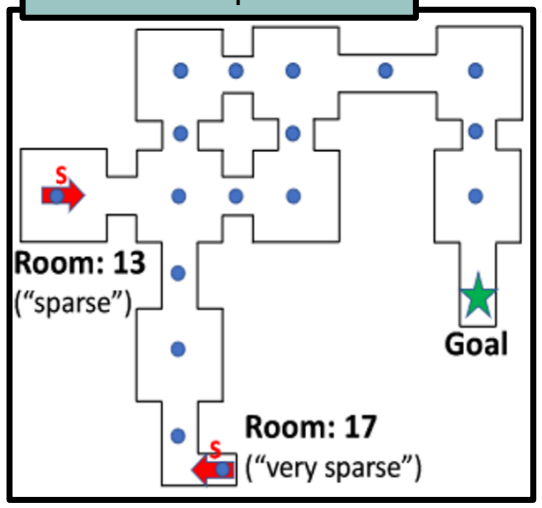
Used Agents

1. A3C with ε-greedy exploration
2. ICM + A3C
3. ICM-pixels*+ A3C

* ICM-pixels : feature 이용하지 않고 raw input으로 intrinsic reward 생성

SPARSE REWARD SETTINGS – EXPERIMENT SETUP

VizDoom Map Scenario



Reward Settings

- **"Dense"** reward : 파란 점에서 시작
- **"Sparse"** reward : 13번 방에서 시작
  - Optimal Policy 기준 goal 까지 **270 step**
- **"Very-Sparse"** reward : 17번 방에서 시작
  - Optimal Policy 기준 goal 까지 **350 step**

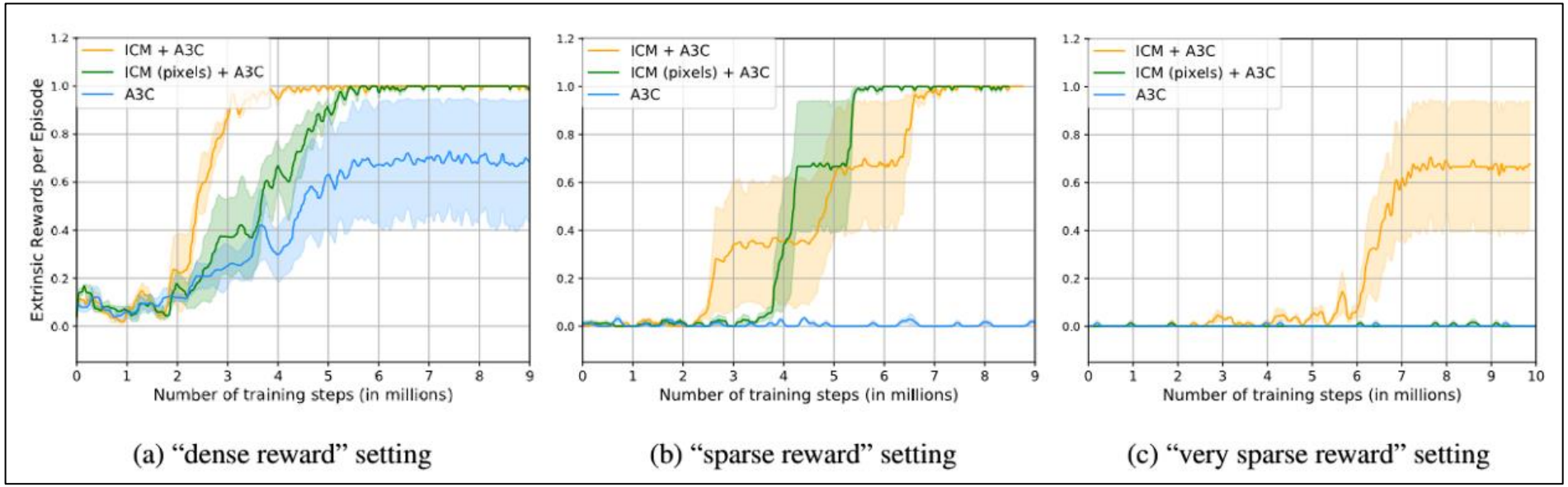
Environment Setting

$$r_t^e = \begin{cases} 1 & \text{if } s_{t+1} \text{is goal state} \\ 0 & \text{else} \end{cases}$$

goal state에 도달하거나 2100 step 후 episode 종료

SPARSE REWARD SETTINGS – EXPERIMENT RESULT

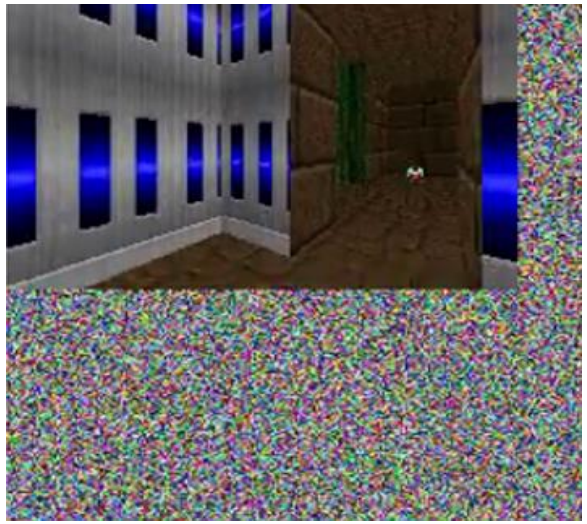
(mean ± standard error)



- Curious agent가 ε-greedy exploration 보다 잘 학습함
- "Very Sparse" Reward setting에서는 ICM+A3C만 학습 성공

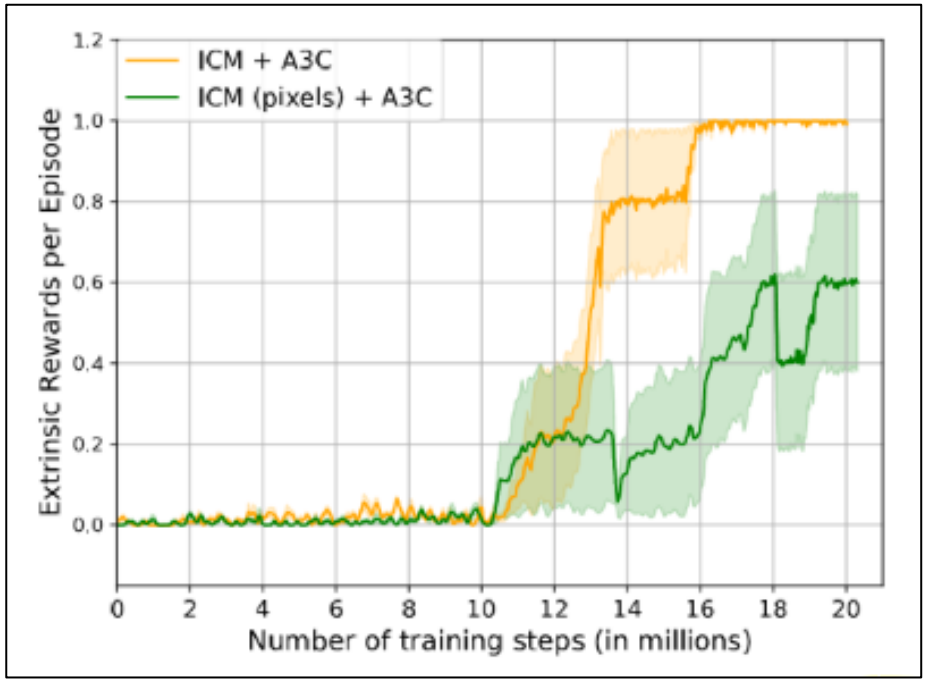
SPARSE REWARD SETTINGS – ROBUSTNESS OF ICM

Input with white noise



+

"Sparse" reward setting



ICM은 Agent와 무관한 feature에 robust함!

SPARSE REWARD SETTINGS – COMPARISON TO TRPO-VIME

VIME : 2017년 당시 state-of-the-art exploration method
TRPO : A3C보다 Sample Efficient한 method

Method ("sparse" reward setup)	Mean (Median) Score (at convergence)
TRPO	26.0 % (0.0 %)
A3C	0.0 % (0.0 %)
VIME + TRPO	46.1 % (27.1 %)
ICM + A3C	**100.0** % (100.0 %)

Sparse reward setting에서 ICM이 기능을 한 것 맞는가?

↳ r^i 대신 random noise(uniform, gaussian, laplacian)로 실험

↓

전부 goal 도달 못함

NO REWARD SETTINGS – EXPERIMENT SETUP

Good Exploration Policy ≈ Visit as many states as possible without goals

➡ 각 환경에서 **extrinsic reward 없이** agent가 방문하는 **state 수 비교** (comparison to random exploration)

VizDoom

- 2100 step 이후 종료
- 전체 방 중 방문한 방의 비율 비교

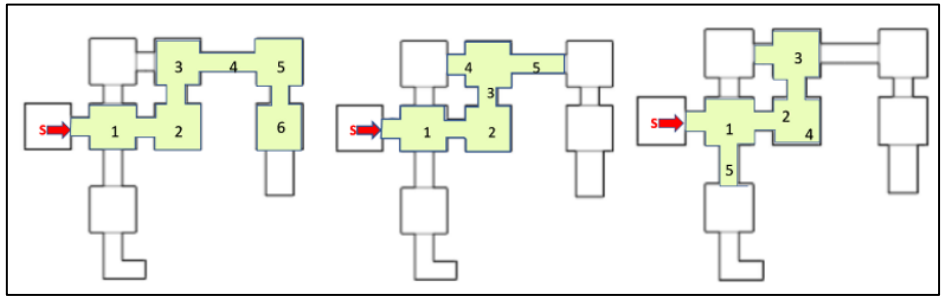
Mario

- extrinsic reward 없이 맵 진행률

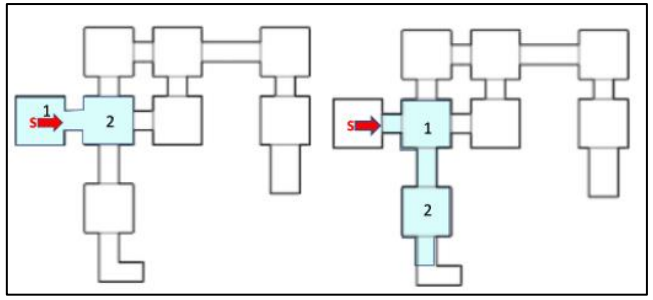
NO REWARD SETTINGS – RESULTS

Exploration Patterns in VizDoom

ICM



Random Exploration



→ External reward 없이 useful한 exploration policy 학습

Mario

- 외부 보상 전혀 없이 Level-1의 30%가량 진척 성공
- 적을 죽이거나, 도망가는 방법 스스로 학습

죽지않고 계속 progress하는 것이 Curiosity를 더욱 키우기 때문

GENERALIZATION – EXPERIMENT SETUP

Agent가 generalized skill을 습득하는 것인지, 단지 training set을 memorize하는 것인지 확인

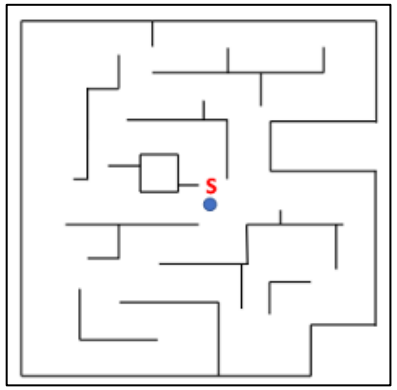
Curiosity 만으로 Pre-train한 policy를 3가지 방법으로 적용

- **"as is"** : pre-trained policy 그대로 적용한 결과
- Fine-tuning with / without extrinsic rewards

Mario

Level-1에서 Intrinsic Reward 만으로 pre-train 이후 Level-2, 3에 적용

(as is , Fine-tuning with curiosity)



Train map for VizDoom

VizDoom

Train map에서 pre-train 후

"Very Sparse" setting에서 evaluate

GENERALIZATION – MARIO

Level Ids	Level-1	Level-2				Level-3			
Accuracy Iterations	Scratch 1.5M	Run as is 0	Fine-tuned 1.5M	Scratch 1.5M	Scratch 3.5M	Run as is 0	Fine-tuned 1.5M	Scratch 1.5M	Scratch 5.0M
Mean ± stderr	711 ± 59.3	31.9 ± 4.2	466 ± 37.9	399.7 ± 22.5	455.5 ± 33.4	319.3 ± 9.7	97.5 ± 17.4	11.8 ± 3.3	42.2 ± 6.4
% distance > 200	50.0 ± 0.0	0	64.2 ± 5.6	88.2 ± 3.3	69.6 ± 5.7	50.0 ± 0.0	1.5 ± 1.4	0	0
% distance > 400	35.0 ± 4.1	0	63.6 ± 6.6	33.2 ± 7.1	51.9 ± 5.7	8.4 ± 2.8	0	0	0
% distance > 600	35.8 ± 4.5	0	42.6 ± 6.1	14.9 ± 4.4	28.1 ± 5.4	0	0	0	0

As is

Level-3는 texture가 비슷해서 매우 잘됨

Level-2는 texture 많이 다름 (night world)

Fine-tuning

Level-2 : Fine tuning한 결과 texture 달라도 일반화 성공

많은 iteration으로 처음부터 학습시킨 agent보다 성능 좋음

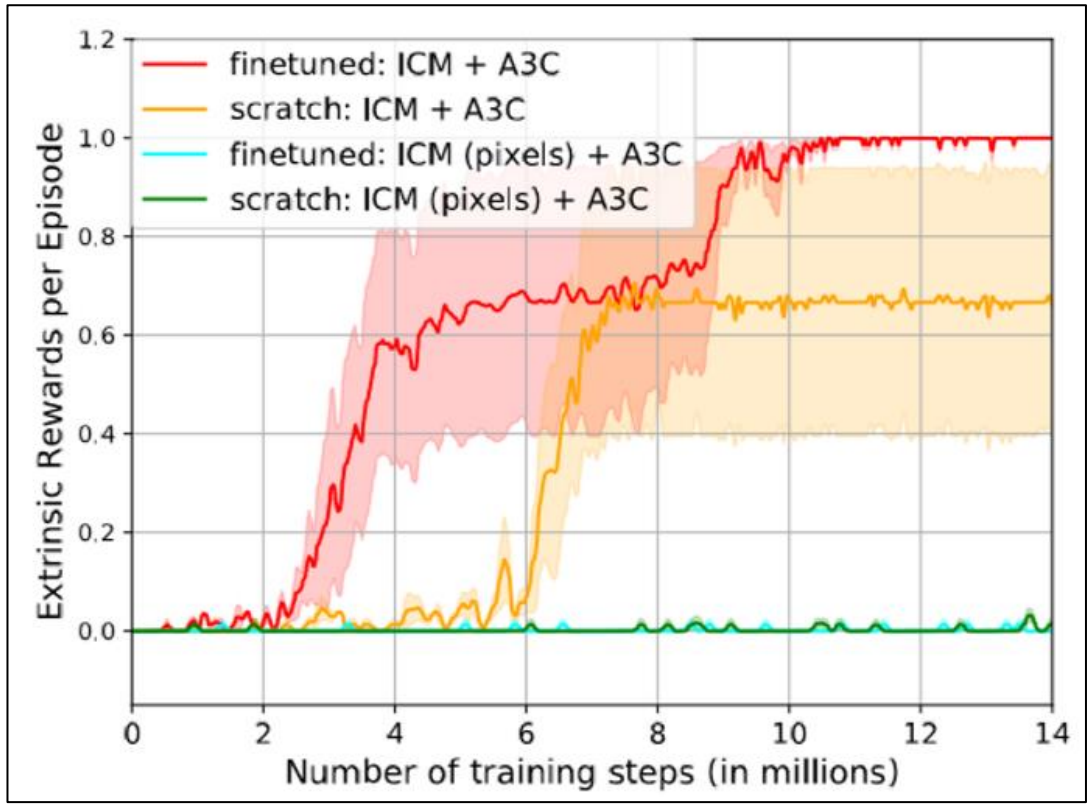
Level-3에서 fine-tune 결과 성능 퇴화

Level-3에 curiosity만으로는 넘어가기 힘든 어려운 구간 존재

이 구간을 넘어가지 못하면서 curiosity 0에 수렴 (boredom)

GENERALIZATION – VIZDOOM

Pretrain 후 새로운 map과 texture로 generalization 되었는지 실험



- ICM-pixel은 전혀 성공 못함
- Pre-train된 ICM+A3C agent가 더 빨리 학습

  → generalizable한 exploring policy를 학습하였음

SUMMARY

1. ICM architecture를 도입, curiosity-driven intrinsic reward를 생성

Sparse reward problem 해결, 좋은 exploration policy 학습 가능

2. Generalization

Self-supervised Learning을 이용한 Feature Extractor로 agent-related feature만 extract 할 수 있었음

실험적으로 generalization이 잘 되었음을 입증

REFERENCE

- D. Pathak et al. (2017) *Curiosity-driven Exploration by Self-supervised Prediction.*
- 이호성 (2019) *Unsupervised Visual Representation Learning Overview : Toward Self-Supervision.* Retrieved from https://hoya012.github.io/blog/Self-Supervised-Learning-Overview/
- 박건영 (n.d.) *Curiosity-driven Exploration 리뷰.* Retrieved from https://parkgeonyeong.github.io/Curiosity-driven-Exploration-%EB%A6%AC%EB%B7%B0/