



RL and Optimization Problem

-
- 가치함수 $V(s)$ (직관력)을 높이는 (최적화) 학습 방법

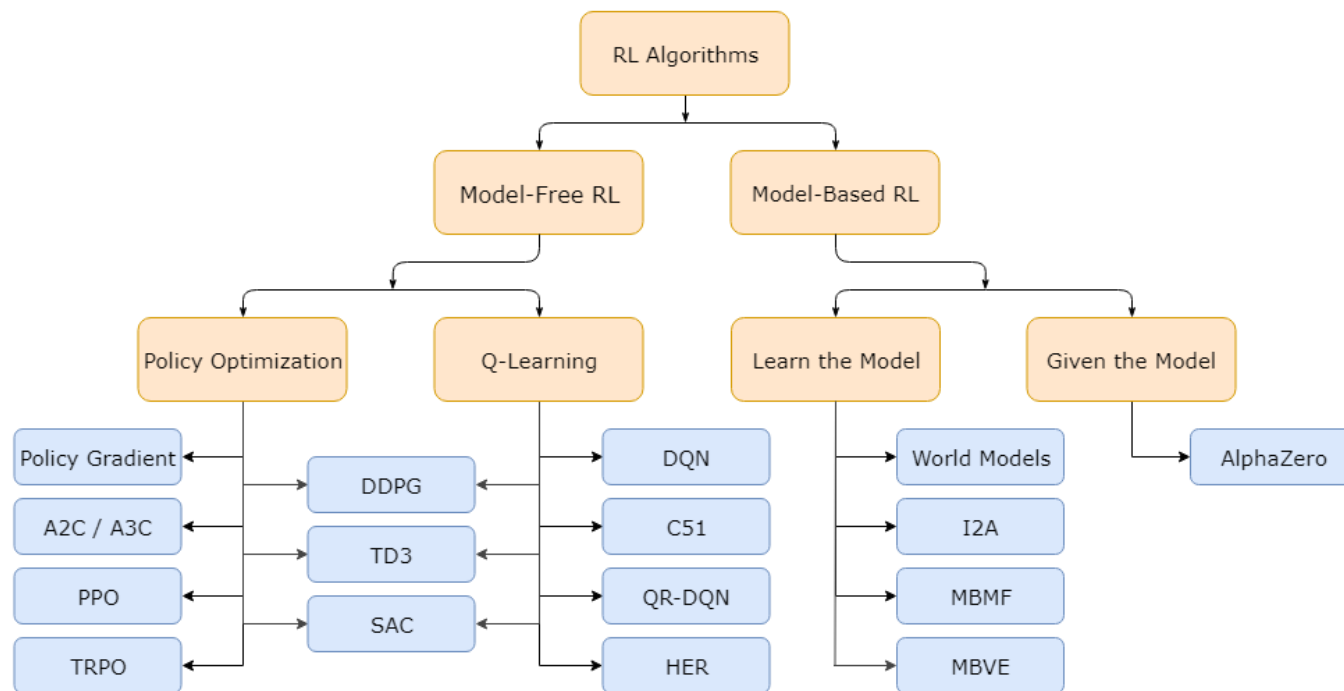
MDP

- MDP로 풀기에는 현실 문제의 복잡도가 너무 높다

RL and Optimization

- Traditional (Combinatorial) Optimization
 - Scheduling
- RL
 - Dynamic Optimization
 - Real-time decision

강화학습의 종류



Experience replay

■ Experience replay memory

- **Issue 1.** 성공적인 Deep Learning applications는 hand-labelled training data set을 요하는데, Reinforcement Learning에서는 오로지 reward를 통해 학습이 이루어지고, 그 reward도 sparse하고 noisy 심지어는 delay되어 주어진다.
- **Issue 2.** Deep Learning에서는 data sample이 i.i.d. 분포를 가정하지만, Reinforcement Learning에서는 현재 state가 어디인지에 따라 갈 수 있는 다음 state가 결정되기 때문에 state간의 correlation이 크다. 즉, data간의 correlation이 크다.

Experience replay

이전에 살펴본 algorithm들과 같이 본 논문이 나오기 전에는 environment와 상호작용하며 얻어진 on-policy sample들,

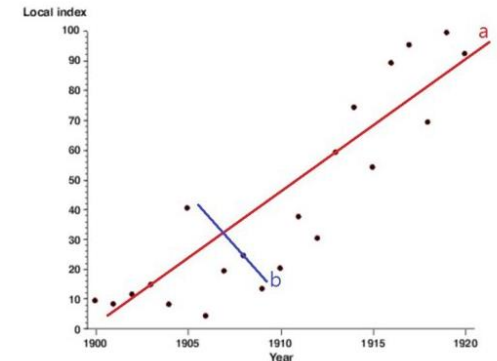
$$s_t, a_t, r_t, s_{t+1}, a_{t+1}$$

을 통해 parameter를 update하였습니다. 하지만, 이와 같은 방법은 on-policy sample을 통해 update하기때문에 sample에 대한 의존성이 커서 policy가 converge하지 못하고 oscillate할 수 있습니다. 이러한 문제를 해결하고자, 착안한 아이디어가 **Experience replay**입니다. 각 time-step별로 얻은 sample(experience)들을 아래와 같은 tuple형태로 data set에 저장해두고, randomly draw하여 mini-batch를 구성하고 update하는 방법입니다.

$$e_t = (s_t, a_t, r_t, s_{t+1}), \quad D = e_1, e_2, \dots, e_N$$

이때, data set은 무한히 저장할 수 없기 때문에 N 으로 고정하고, FIFO방식으로 저장합니다. 또한 input으로써 arbitrary length를 받는데에 어려움이 있기 때문에 function ϕ 를 정의하여 pre-processing하여 state의 length를 fixed시킵니다. Experience replay를 이용하면 아래와 같은 이점을 얻을 수 있습니다.

- data sample을 한번 update하고 버리는 기존의 방법과 달리 random sampling을 함으로써 data usage efficiency를 도모할 수 있습니다.
- RL에서 발생할 수 있는 state간의 high correlation 문제(위에서 언급한 Issue 2. correlation문제)를 해결 할 수 있습니다. 현재 update된 parameter를 통해 다음 training의 대상이 되는 sample을 어느정도 determine할 수 있습니다.
- 현재의 policy에 따라 argmax action을 고르고 난 뒤, 그 다음 training sample은 이 action에 따라 결정됩니다. 이를 current sample에 의해 다음 training sample이 dominate되었다고 말합니다. 본 논문에 따르면, 이렇게 되면 불필요한 feedback loops를 돌거나, parameter들이 local minimum에 빠지거나 diverge하는 경우를 찾아낼 수 있어서, 이를 방지할 수 있다고 합니다. 주의할 점은, experience replay를 사용할 땐, 반드시 off-policy로 learning해야하는데, 이는 current parameter가 update하는 sample들과 다르기때문입니다.

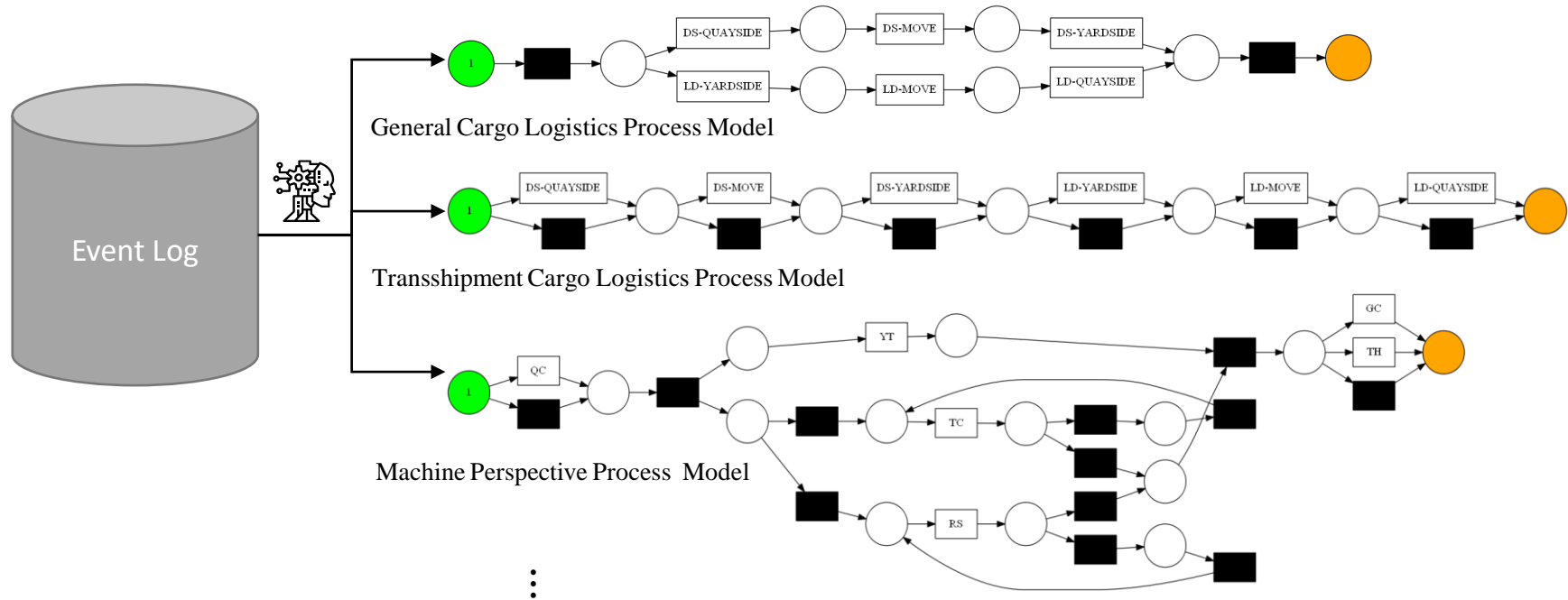


출처: <https://sumniya.tistory.com/18>

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory \mathcal{D} to capacity N
Initialize action-value function Q with random weights
for episode = 1, M **do**
 Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
 for $t = 1, T$ **do**
 With probability ϵ select a random action a_t
 otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
 Execute action a_t in emulator and observe reward r_t and image x_{t+1}
 Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
 Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in \mathcal{D}
 Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from \mathcal{D}
 Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
 Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
 end for
end for

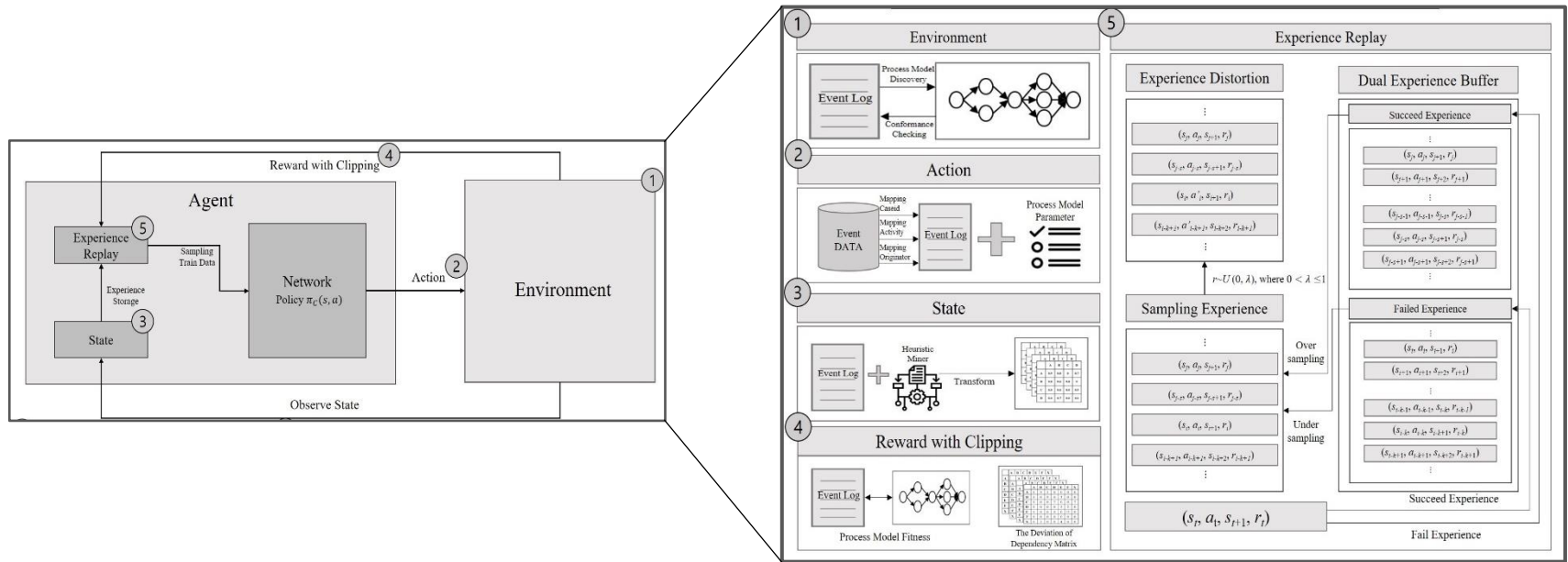
Automatic discovery of process model



Let W be an event log over T , and $a, b \in T$:

- $|a >_W b|$ is the number of times $a >_W b$ occurs in W ,
- $a \Rightarrow_W b = \left(\frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$

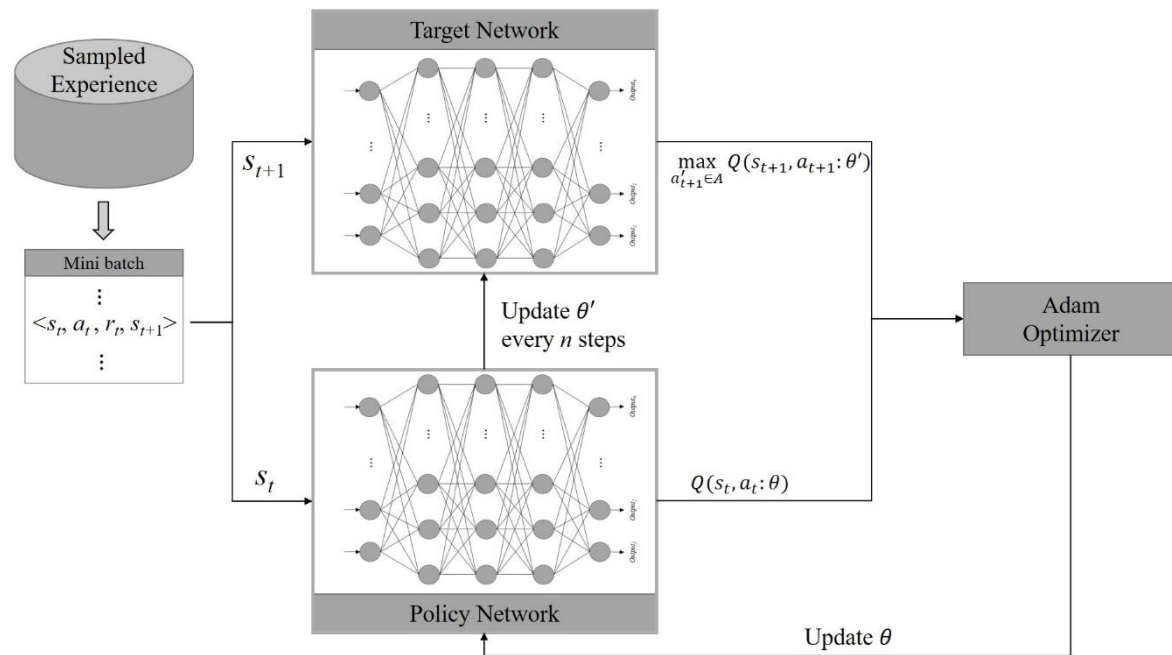
Deep RL for process model discovery



Research Results

Key Components

- ✓ Action space
- ✓ State
- ✓ Reward
- ✓ Environments
- ✓ Q-Networks
- ✓ Experiences Replay



2. Research Contents

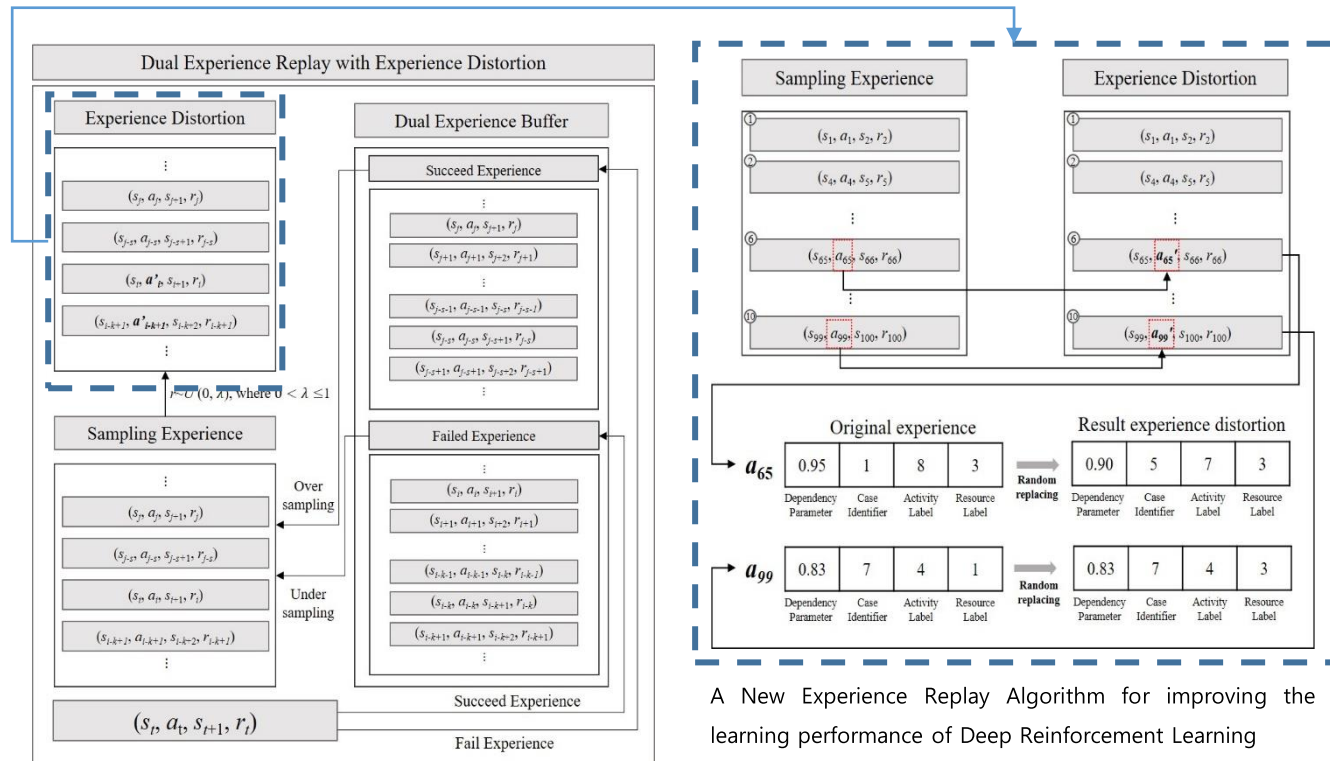
04. [AutoPD] Automatic Discovery Multi-perspective Process Model



Research Results

Key Components

- ✓ Action space
- ✓ State
- ✓ Reward
- ✓ Environments
- ✓ Q-Networks
- ✓ Experiences Replay



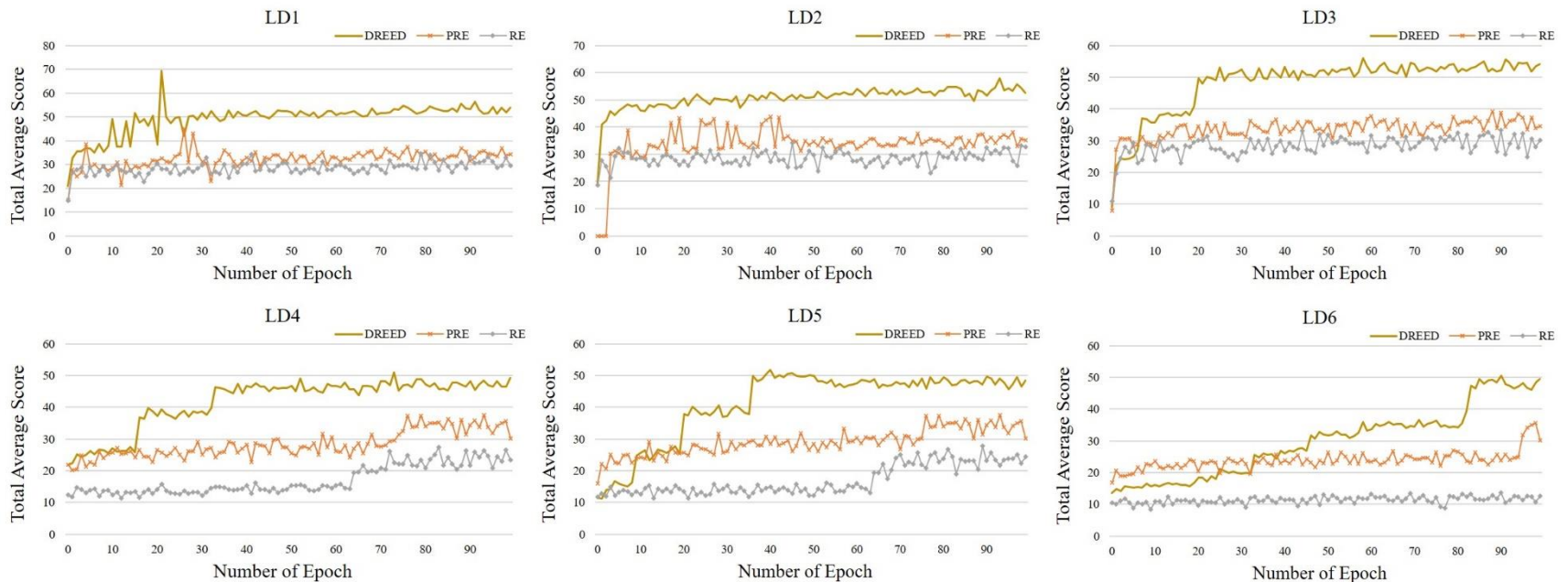
2. Research Contents

04. [AutoPD] Automatic Discovery Multi-perspective Process Model



Research Results

[Experiment 4-1] The overall average reward received by the agent while learning is in progress



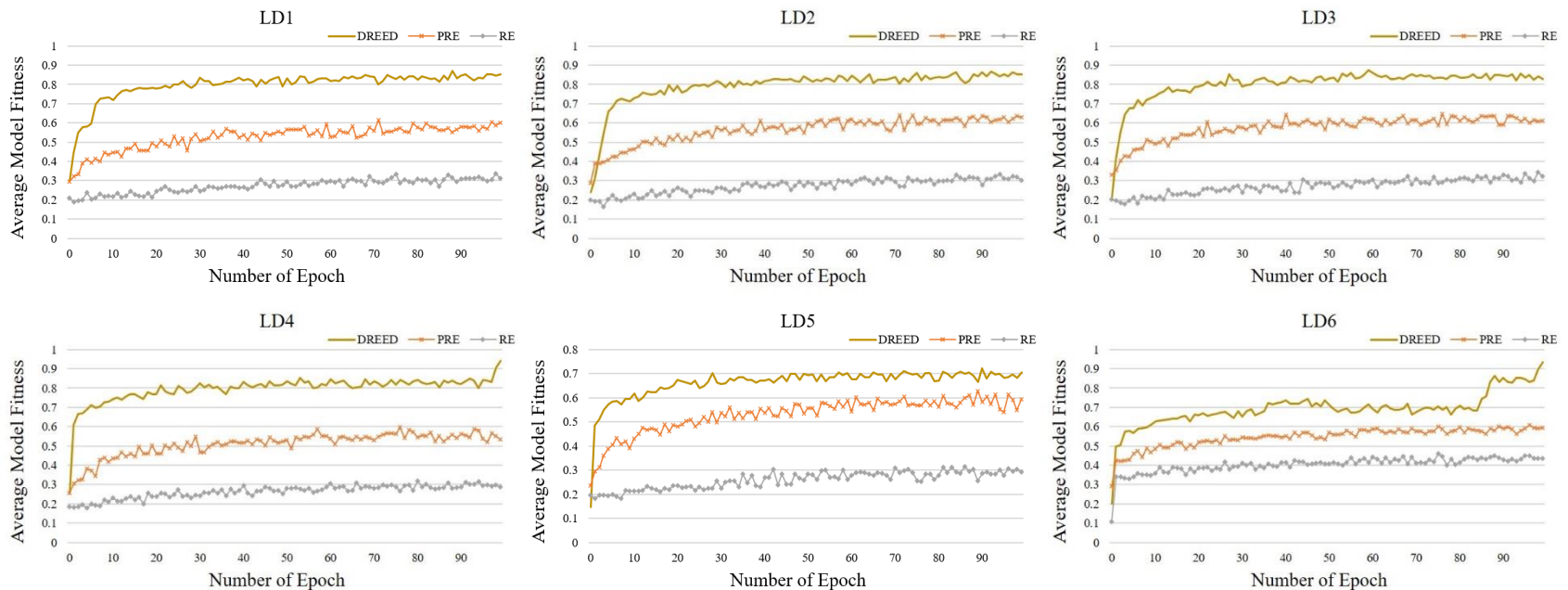
2. Research Contents

04. [AutoPD] Automatic Discovery Multi-perspective Process Model



Research Results

[Experiments 4-2] The average fitness of the process model derived by the agent during training



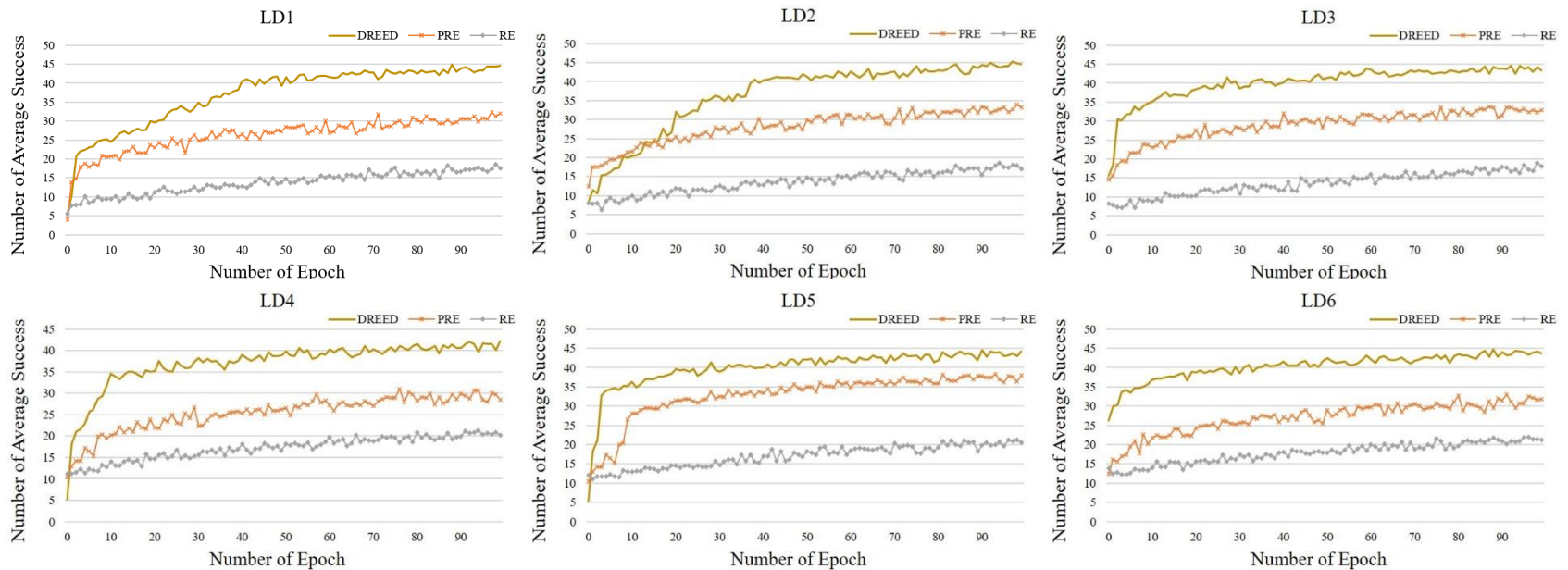
2. Research Contents

04. [AutoPD] Automatic Discovery Multi-perspective Process Model



Research Results

[Experiments 4-2] The average number of times an agent discovering a process model with the fitness of 0.7 or more during the learning process



2. Research Contents

04. [AutoPD] Automatic Discovery Multi-perspective Process Model



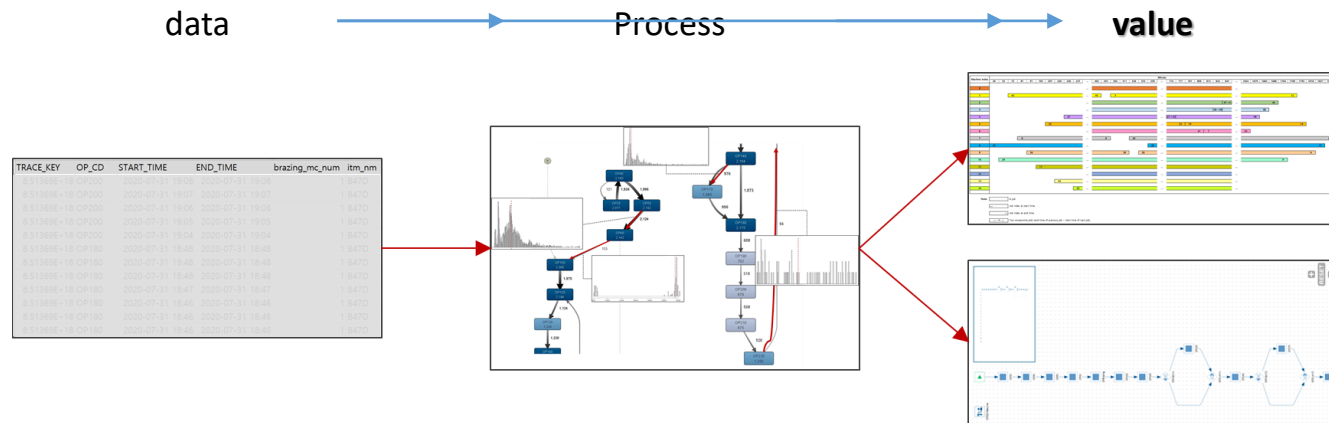
Contribution

- ✓ Proposed a **new perspective process discovery** algorithm that can derive process models from various perspectives at once
- ✓ A new ER method called DERED is proposed to improve the performance of reinforcement learning.
- ✓ Using 6 actual event logs, we secured nearly twice the learning performance improvement in the experiment compared to the existing ER method.
- ✓ Through the proposed method, process model discovery, conformance checking, and process model improvement are automatically performed to make it easier to provide process models to users.

3. Research Summary

01. Research Summary

- Through this study, we propose an **automated** process mining method that allows business experts to **easily** apply process mining.



-
- <https://sumniya.tistory.com/18>