## **1. Basic Information**

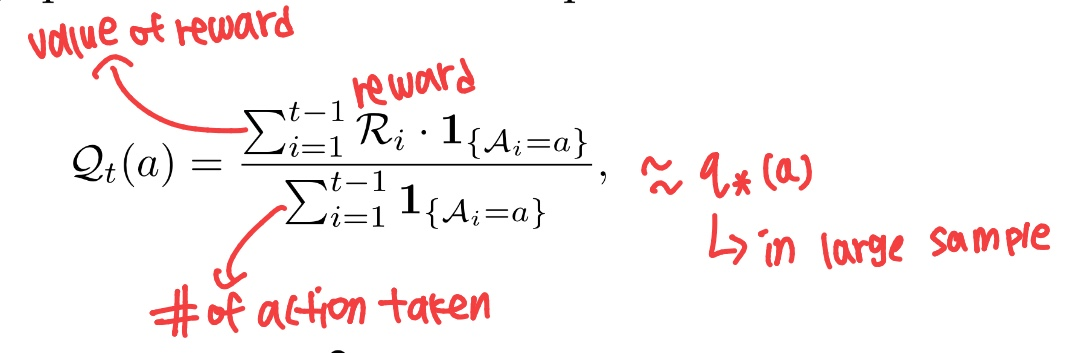
* **Title: Introduction to Reinforcement Learning**
* **Authors:Ghasemi, M., Ebrahimi, D.**
* **Published in:** 3 Dec 2024
* **Link:**
* **Group:**
* **Reviewer:** Kim Suah

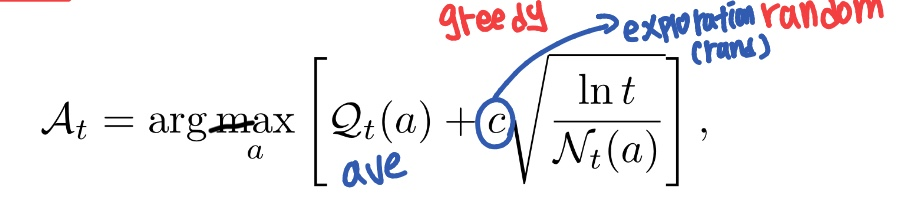
## **2. Motivation & Problem**

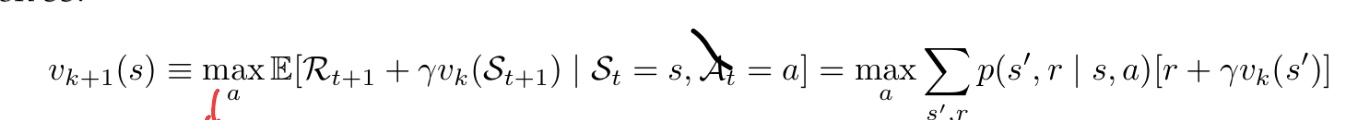
* 연구 동기: Understanding concepts about Reinforcement learning, which focuses on training by interacting with the environment, maximizing cumulative reward overtime.
* 문제 정의:

## **3. Methodology**

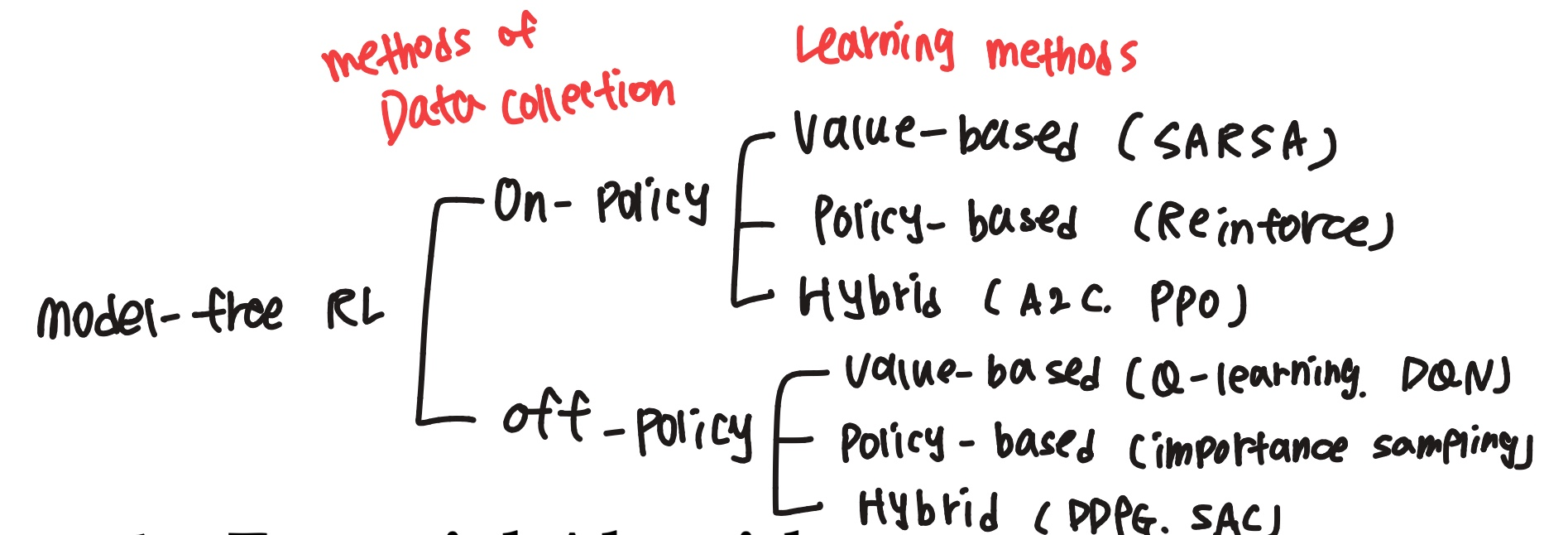
* Key words
  + State (s) : specific condition or configuration of the environment at a given time as perceived by the agent. / can be discrete, continuous
  + Actions (a) : set of possible moves or decisions an agent can make, interacing w. environment. / selected action -> to reach desired gole (optimal policy?)
  + Policy : guides the behavior of a learning agent by mapping states. Guidlline for desired gole / can be form of fuction, stochastic
    - Stochastic policy : prefer unknown, novel things. Increases diversity
    - Deterministic policy: known, increases efficiency.
  + Rewards (r) : defining local and global goals
    - Positive rewards
    - Negative rewards
    - No reward
    - Can be deterministic (R(s, r)=r) or stochastic (P(r|s,a))
  + Model
    - Transition dynamics : given current state -> define probability of reaching new state (P(s’|s,a))
    - Environment model : transition function / reward function -> MDP (M = (S, A, P, R, r))
* Key algorithems
  + Multi-Armed Bandit problems
    - Repeat choose from k action
    - Uses stationaty assumption -> reward possibility is constant
    - **Balancing** between exploration (choose random action -> immediate reward) and explotiation (greedy action -> high overall)
    - Sample-average method: used to update the action value



* + - Iteratively refine the value (incremental update rule)
    - NewEstimate <- OldEstimate + StepSize[Target - OldEstimate]
    - 1. Epcilon-greedy method
      * Choose random action (exploration) with given probability. Otherwise, select greedy action (explotiation)
      * All actions are sampled
    - 2. Upper confidence Bound (UCB)
      * Balancing by incorporation uncertainty into action selection 
      * Optimistic Initialization : assign high initial value -> make untried actions seem attractive. Hence, increased exploration even in greedy stratege, prevent converging
  + Markov Decision Process (MDPs)
    - Sequential decision making, actions affect immediate rewards, current states.
    - Aim to measure value of taking action a in state s. Discrete probablity distributions
    - Utilizes four-argument dynamics function p (s’, r, s, a)
    - Episodic problems: standard starting at each ends of episode -> able to return infinity
    - Discount rate (r) : return is finith when r <1.
      * ,r -> 0 : immidiate
      * ,r-> 1 future
* Policies and value functions
  + Value functions
    - State value functions (v)
      * Value in particular state
    - Action value functions (q)
      * Compare action potential in same state, action selection -> continuous action spaces
    - If 10 states each has 8 actions -> v requires 10 functions, q requires 80 functions (action!!)
  + Optimal value functions
    - Maximal value for each state-value function and action-value functions
    - Dynamic programming (DP) helps identify optimal values: via transforming bellman eq into update rules.
  + Policy evaluation and improvement
    - Select new action maximizes the action value function -> obtain new policy
    - Adapt greedy approahc based on value function
    - Iteration at MDP
      * Conveges to an optimal policy and value function
      * Cons: each iteration requires policy evaluation => value iteration



* 모델 구조 / 알고리즘 핵심 아이디어 (Core RL Methods)
  + Model-free & model-based methods
    - Model free methods



* + - * 1. methods of data collection
        + Cf. Behavior policy (b) : explore environment. Balance exploration, exploitation

Target policy (π) : improve performance via gathered experience

* + - * + On-policy methods: (b = π) stable learning, less efficient
        + Off-policy methods: (b != π) unstable learning, efficient.
      * 2. methods of learning
        + Value-based learning: learning action-value function

based on action / regardless of action

* + - * + Policy –based : learn policy directly, via expected reward
        + Hybrid (value-based, policy-based) : actor which updates policy, critic evaluates action-value function
    - Model based methods
      * Planing, decision making, make simulation possible
      * Uses new data to refine model.

## **4. Key Results**

* 주요 성과 (수치, 그래프, 비교 결과)
* baseline과의 성능 차이
  + RL refines policy, action decision algorithm continuously. Inhancing efficiency, agent at own
* 저자가 주장하는 기여점

## **5. Strengths**

* For model-based algorithm
  + Via simulation, faster convergence to optimal policy is possible
  + Adapt quickly to changing environment

## **6. Weaknesses / Limitations**

* Model-based algorithm requires high cost for operating......

## **7. Personal Insights**

* 내가 이해한 핵심 아이디어
* 향후 확장 가능성
* 실제 내 연구/프로젝트와의 연관성

## **8. References & Resources**

* 원 논문
* 관련 블로그/리뷰/코드 레포
* 발표 슬라이드 (있다면)