基于强化学习的智能体决策

动态复杂情景下决策问题研究及应用

蚂蚁集团人工智能部 - 动态博弈

熊君武

2022.09.03

关于我

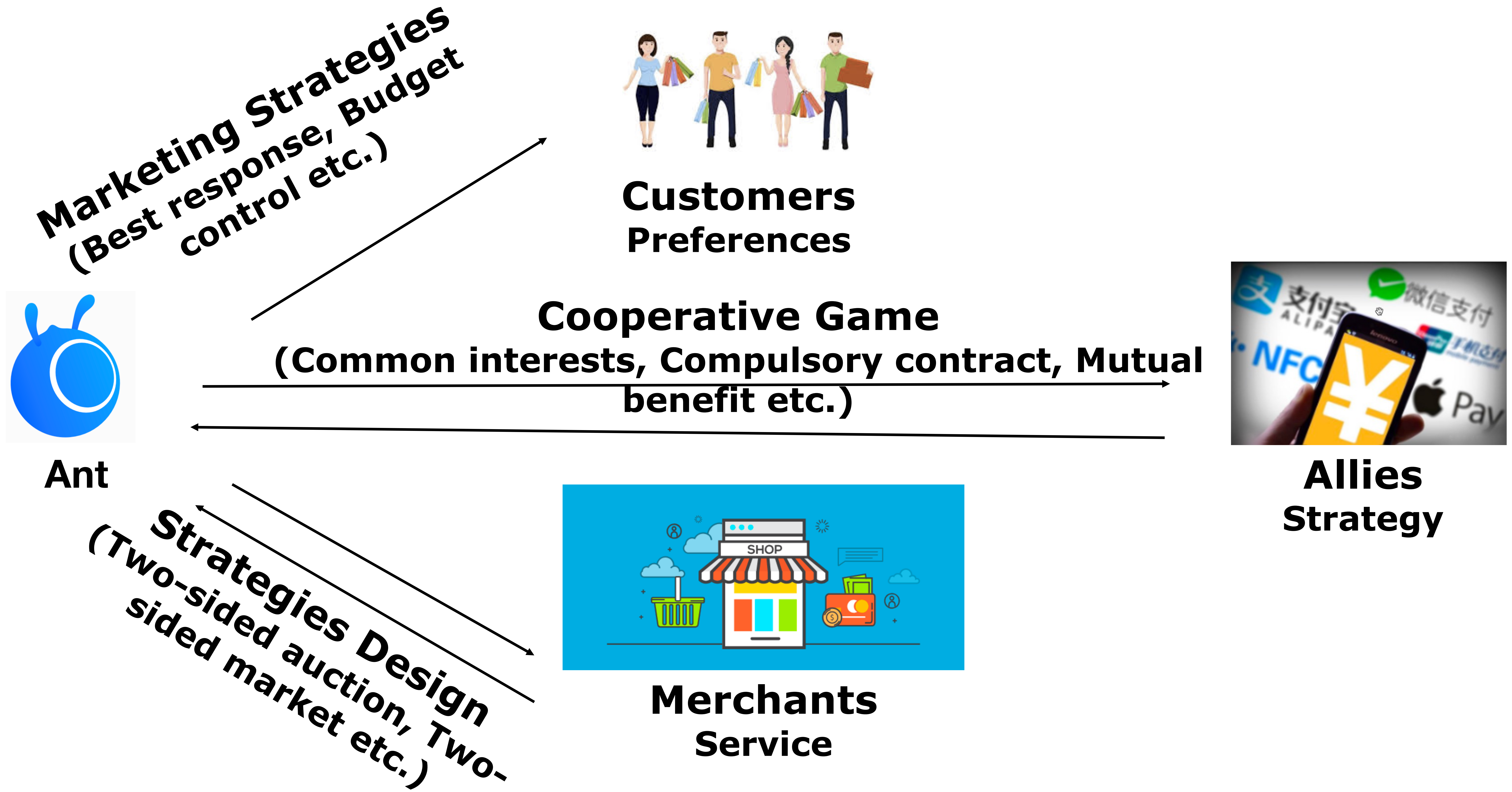
- 最近
  - 2016-现在：蚂蚁集团-高级算法专家-人工智能部-动态博弈
  - 2014-2016：阿里巴巴-算法专家-推荐平台-推荐与投放算法
- 过往
  - 2011.08-2014.02 人人应用研究中心-高级算法工程师-SocialGraph
  - 2010.11-2011.07 百度-社区搜索研发部实习-用户UGC挖掘算法
  - 2008.09-2011.03 北航-计算机ACT硕士-IOT能耗优化算法
  - 2008.04-2008.09 City HK(SZ)未来网络中心-助理研究员-流媒体优化算法
- 在ICML、NeurIPS等国际会议发表多篇文章、拥有多项专利，多个顶会审稿人。

Outline

1. Agent Decision Making in Dynamic Complex Context
2. Digital Life: Customer Lifecycle Marketing On the Internet
3. Green AI: Cloud Resource Scheduling Management
4. Agent Based Reinforcement Learning(RL):
   Algorithm Library, Dataflow Framework and System Platform
5. What's Ongoing & Next

1. Agent Decision Making in Dynamic Complex Context: Marketing in the Open Internet Ecosystem

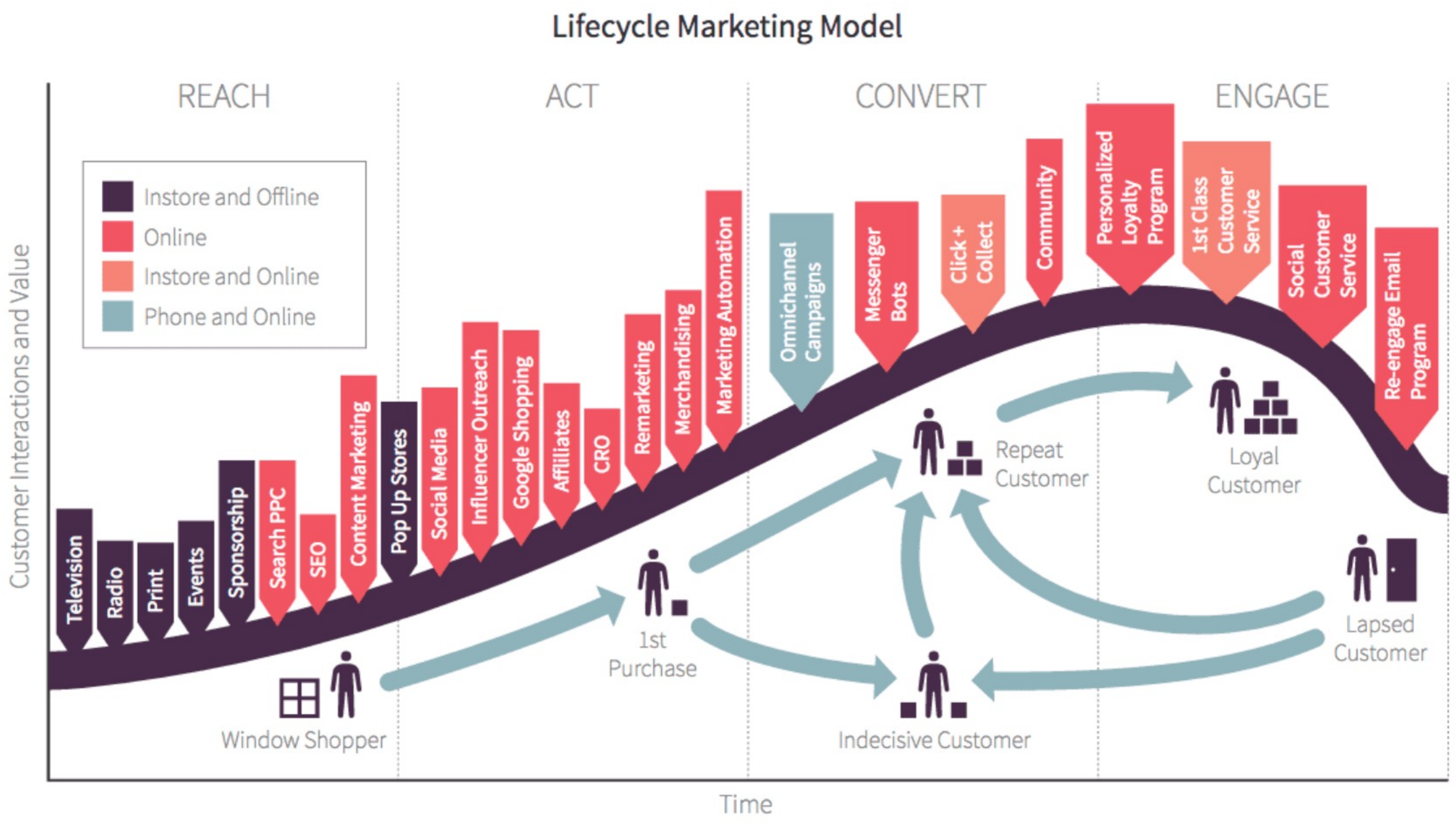
Agent Decision Making in Dynamic Complex Context: Marketing in the Open Internet Ecosystem

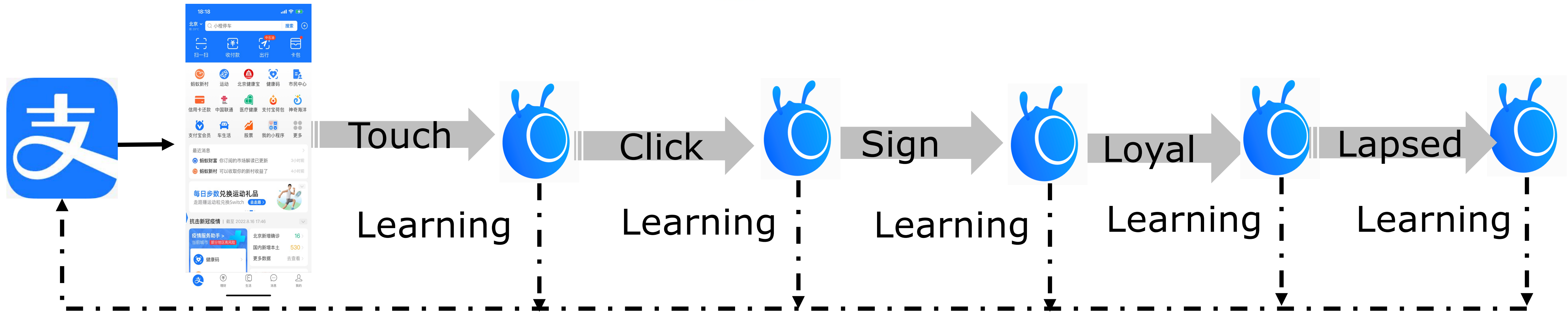


2. Digital Life: Customer Lifecycle Marketing on the Internet

Customer Lifecycle Marketing on the Internet(1)

- **Who**
  - Different customers with different needs in different periods of lifetime
    - ✓ Behavioral economics, demographics
    - ✓ Aesthetic fatigue, behavior psychology
- **Where & When**
  - User's context: Customer activity, Precise delivery time, Scene orientation
    - ✓ Partial observed or Unobserved
    - ✓ Uncertain, dynamic and multi-dimensional
    - ✓ Immediate and precision decision making
- **What & Why**
  - Products or service: Keep simple towards complex business flow
  - $\propto \sum Scale$, Customer Lifetime Value(CLTV) (over the user's trajectory), Service Rate, Efficiency, etc.
    - ✓ Frequency, Duration/stickiness: Uplift/CTR/CVR
    - ✓ Resources turnover cycle, Asset Utilization, Revenues: ROI (Return On Investments), GMV,AUM

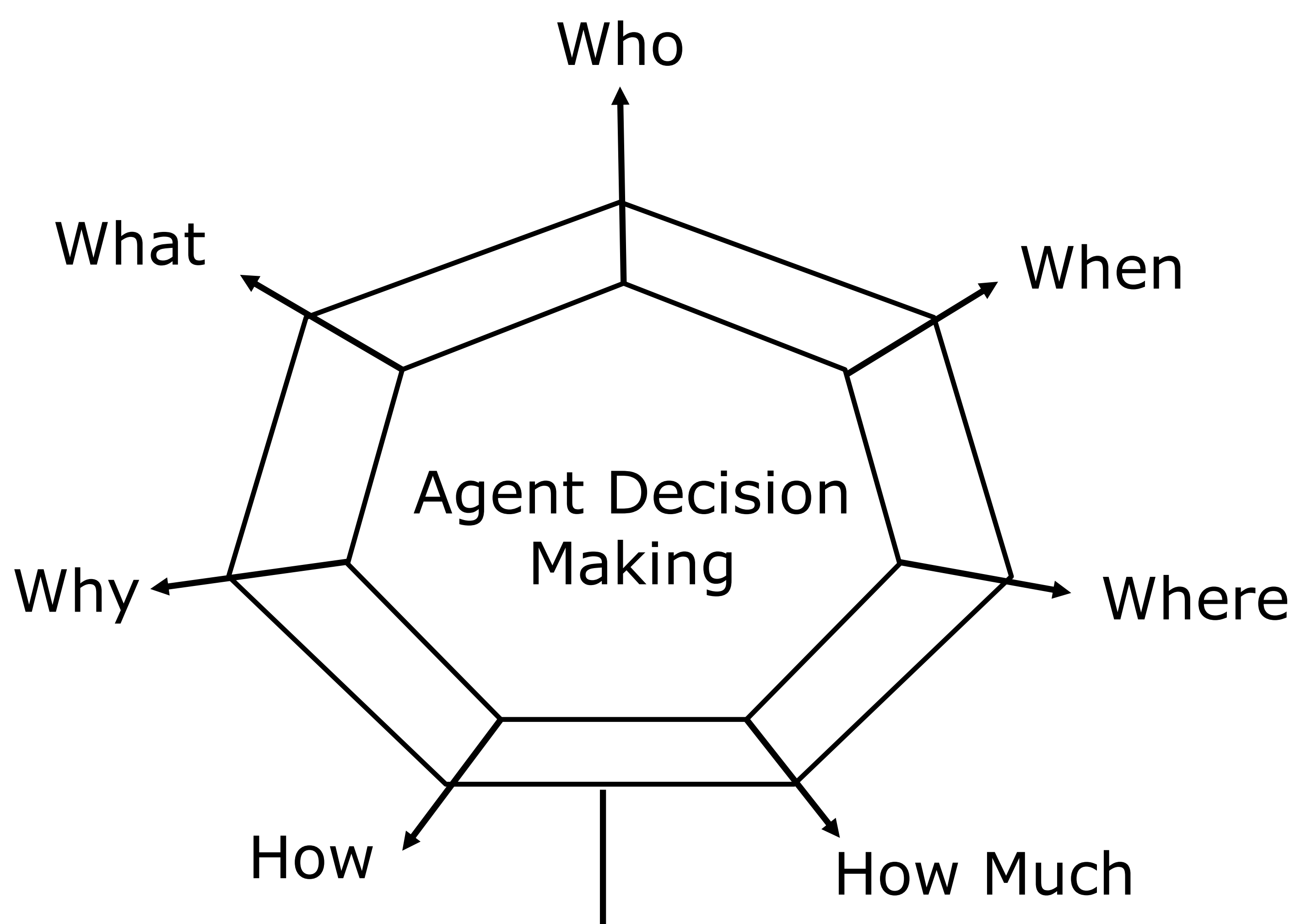


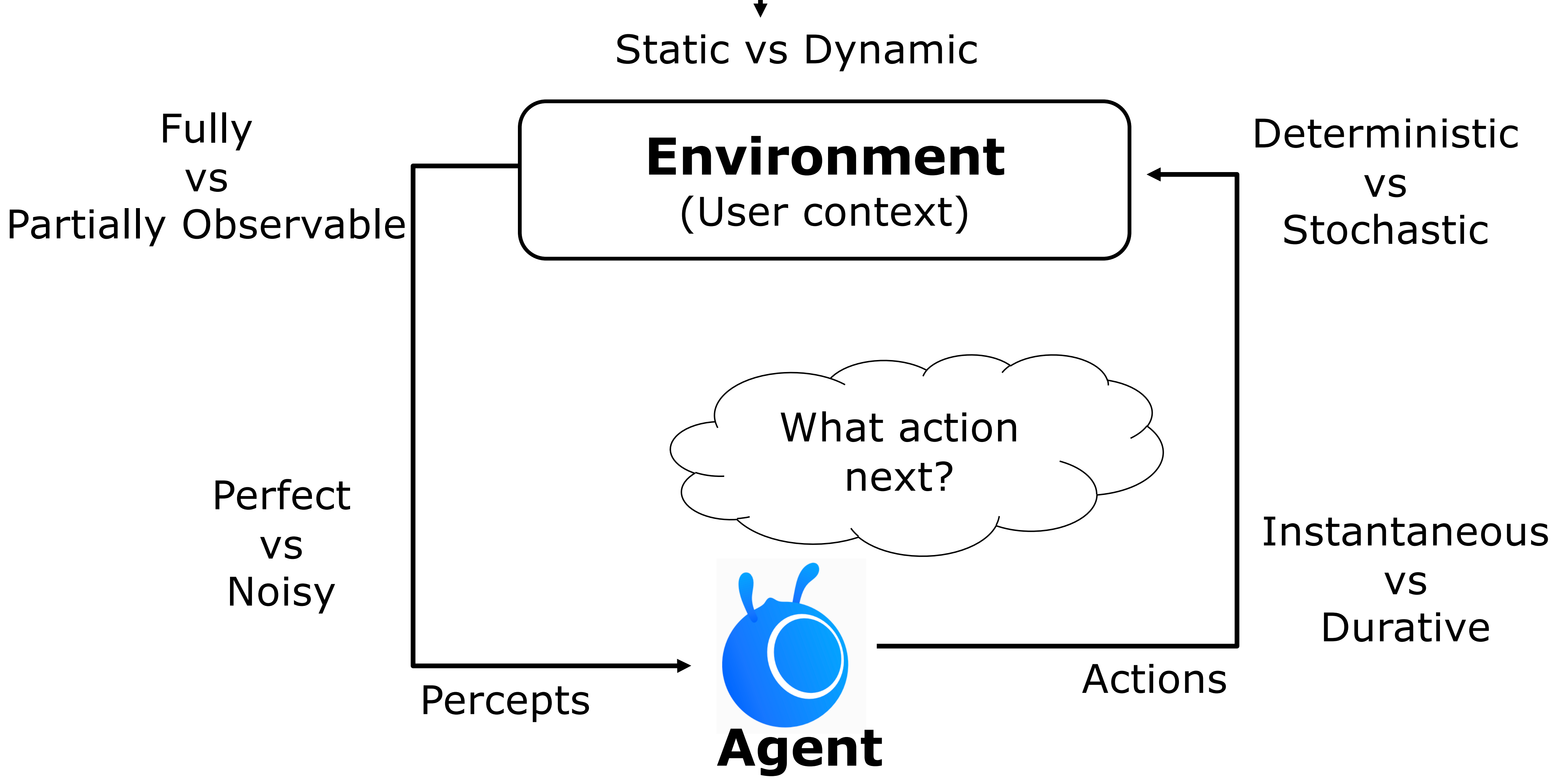


- **A complete behavioral paths in Ant marketing ecosystem**

Customer Lifecycle Marketing on the Internet(2)

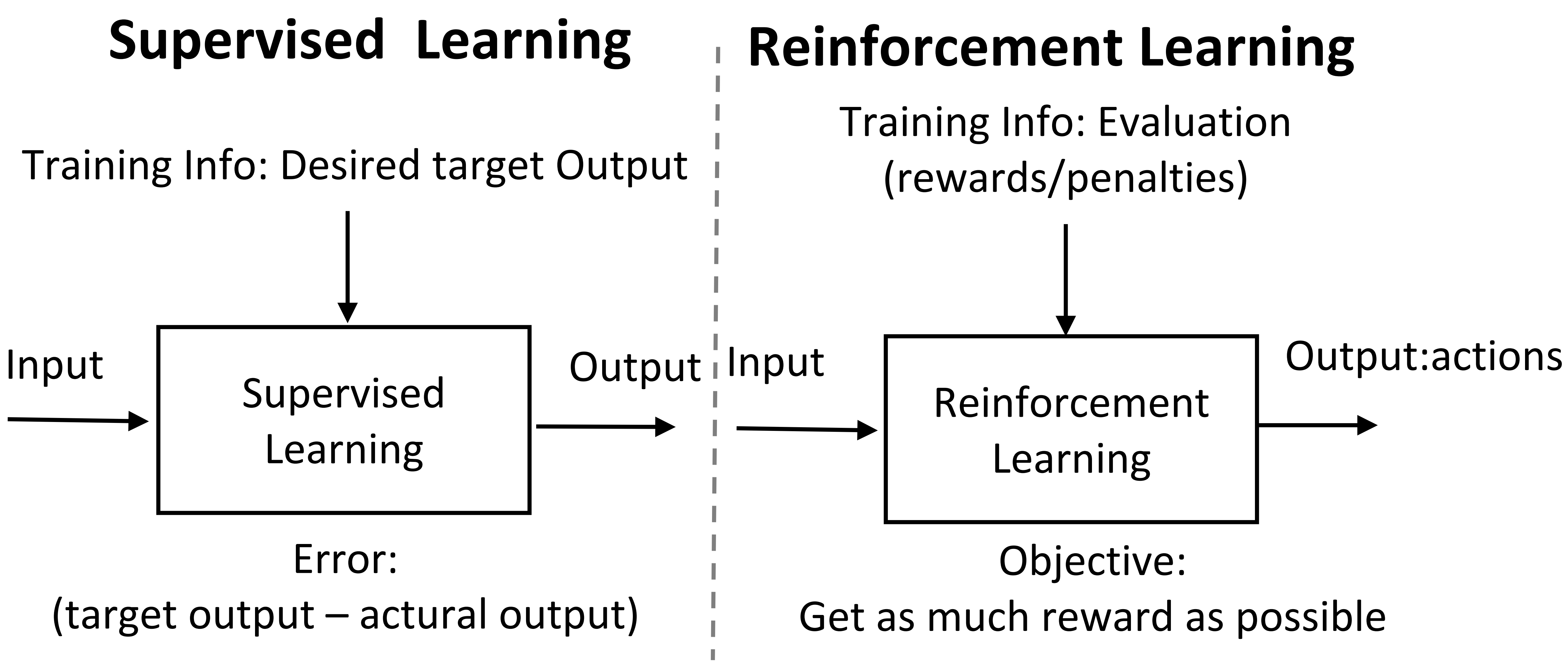
- How
  - Business flow: Promote activity, Frequency period, Time decay, Superposition/Mutual exclusion
  - Channel to touch targeted users
    - ✓ Push matching: Messages, SMSs, Phone calls etc.
- How much
  - User benefits: Coupons, Cash back, Red packet, Discounted rate etc.
  - Budget Constraint/Limit: macro control and micro optimization
    - ✓ Overall budget constraints, Maximum Capital Limit etc.



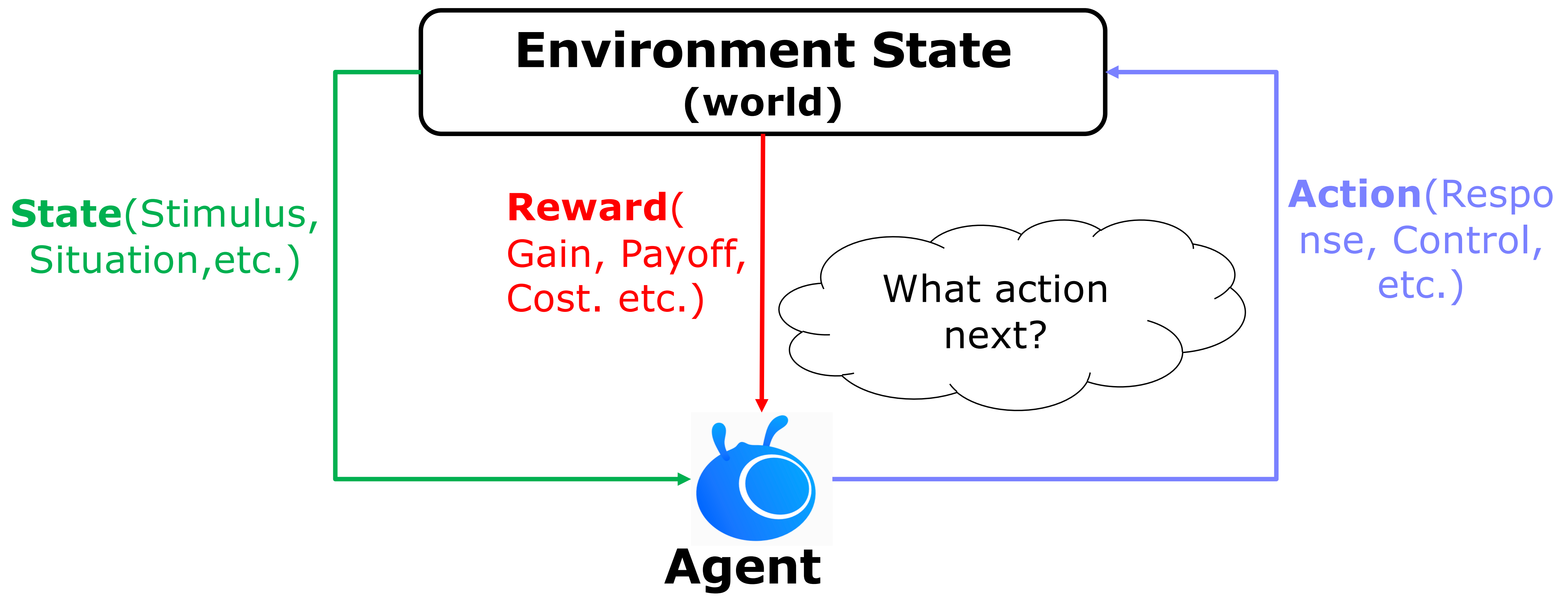


Customer Lifetime Value(CLTV) Modeling(1)

- Reinforcement Learning(RL) VS Supervised Learning(SL)
  - RL learning from interactions: Agent learns a policy mapping states to actions
    - ✓ Impractical to obtain examples of desired behavior that are both correct and representative of all the situations
    - ✓ Trade-off between exploration and exploitation
    - ✓ Delayed reward
    - ✓ Learn from its own experience
  - SL learning from examples
    - ✓ Provided by a knowledgeable external supervisor



Supervised Learning vs Reinforcement Learning

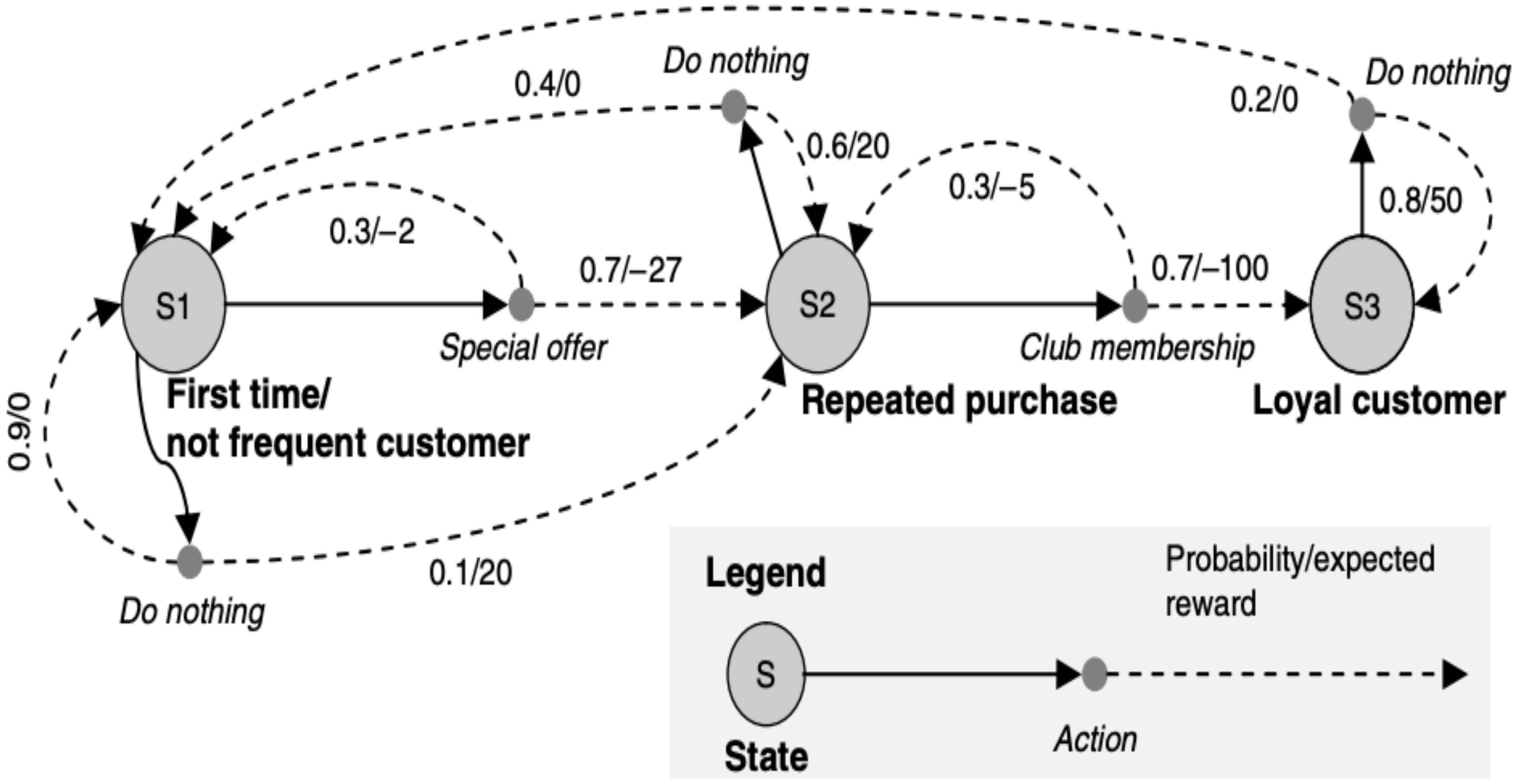


CLTV Modeling(2): Through Agent Decision Making Based on RL

- RL seems to provide a very promising solution framework
  - Interactive and sequence decision learning: Interactive behavior sequences
  - A general end-to-end decision-making framework
    - ✓Explicitly considers the whole problem of a goal-directed agent interacting with an uncertain environment
    - ✓Seeking to maximize its cumulative reward in the long run
    - ✓Multi-objective decision making
    - ✓A unified, automatic and real-time intelligent decision making
- RL with deep learning or DRL
  - Apply deep learning to RL
    - ✓Use deep neural network approximation to opt value function/policy/model end-to-end

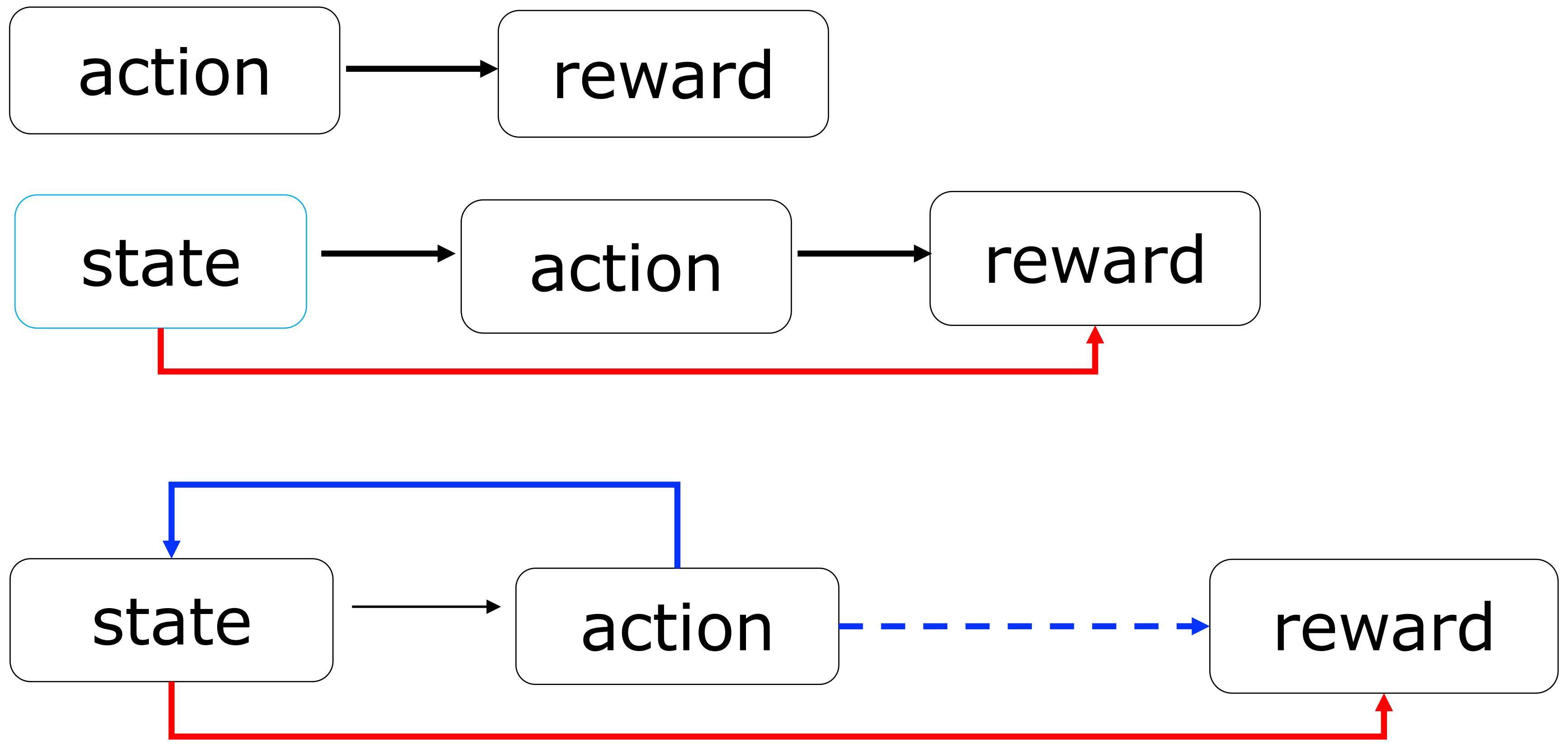
CLTV Modeling(3): Through Agent Decision Making Based on RL

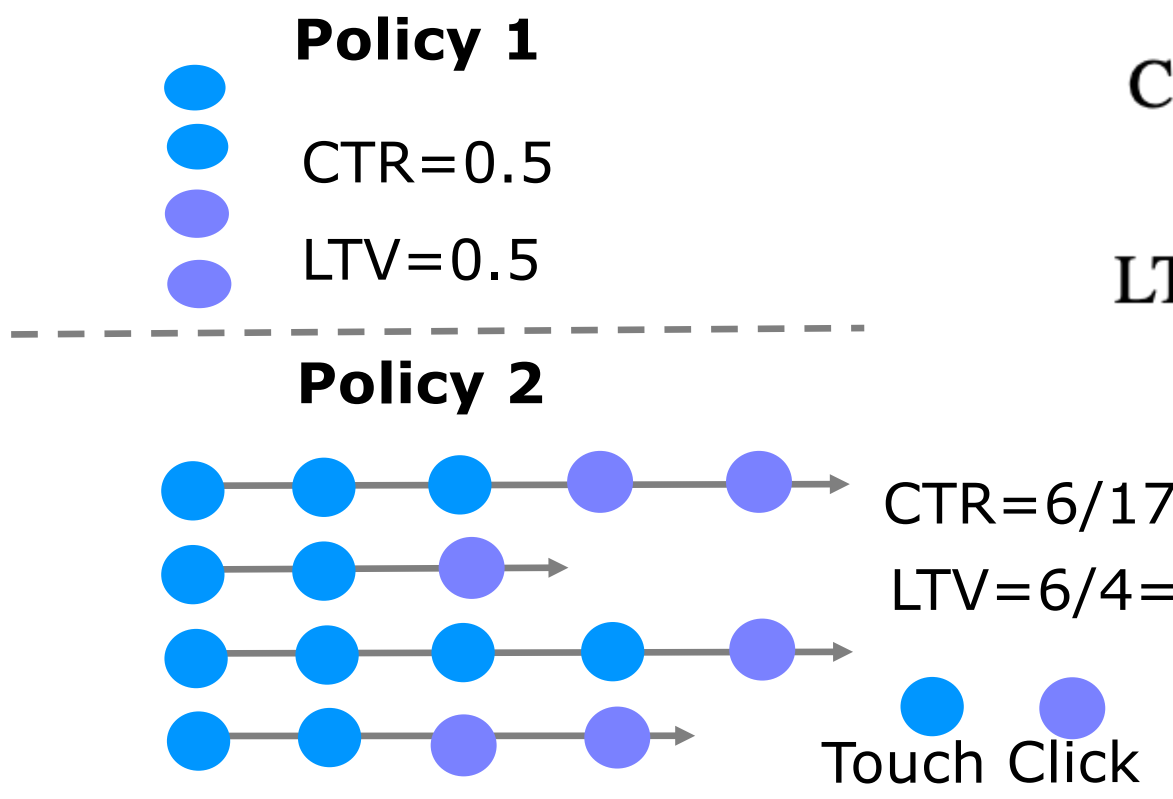
- Is it possible for an ensemble modeling framework adaptive to different business time-scales?



Customer Dynamics Modeling Using an MDP

- Multi-armed Bandit, Context Bandit, Full RL Problem





$$\text{CTR} = \frac{\text{Total \# of Clicks}}{\text{Total \# of \textbf{Visits}}} \times 100,$$

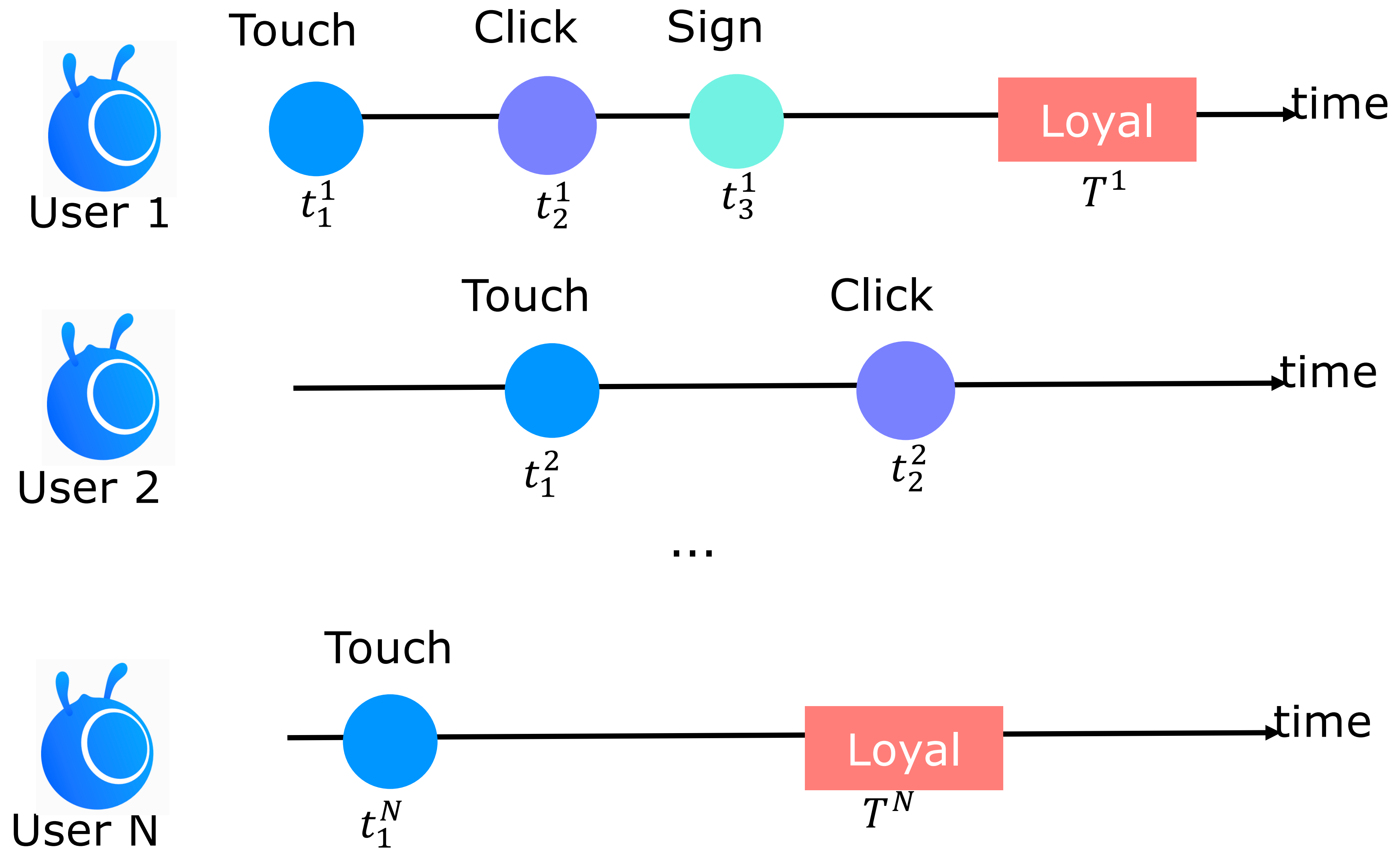
$$\text{LTV} = \frac{\text{Total \# of Clicks}}{\text{Total \# of \textbf{Visitors}}} \times 100.$$

LTV is potentially a better metric than CTR[1]

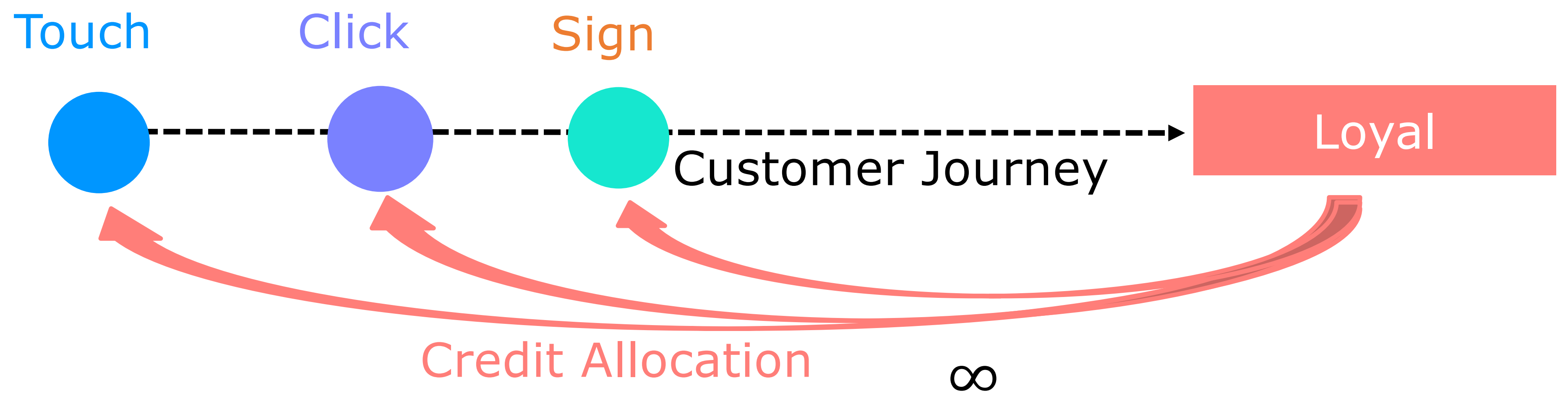
[1]Personalized Ad Recommendation Systems for Life-Time Value Optimization with Guarantees, IJCAI, 2015

CLTV RL: Algorithm Design(1)

- **Context**
  - Customer life cycle marketing, essential to customer life value
  - Most active users are loyal and the rest are hard-to-convert users
- **Goal**
  - Through different marketing activities to touch users repeatedly and change marketing strategy according to users' behavior feedback
- **RL model design**
  - Repeated touch sequences for reinforcing decision, each marketing activity as an episode, N days for delivery cycle



- **The possible behavioral paths in Ant marketing ecosystem. Each such path consists of the chronological sequence of a user's interactions with different channel.**



$$G_t = R_{t+1} + \gamma R_{t+2} += \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

ANT GROUP 蚂蚁集团

CLTV RL: Algorithm Design(2)

- RL model design
  - Actor-critic Deep RL

$$\nabla_\theta J(\theta) = \mathrm{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) \mathrm{A}^{\pi_\theta}(s,a)]$$

✓ Here,

$$\mathrm{A}^{\pi_\theta}(s,a)] = \mathrm{Q}^{\pi_\theta}(s,a) - \mathrm{V}^{\pi_\theta}(s,a)]$$

- State
  - Feature embedding through DL models
- Action
  - Compounded decisions
- Reward
  - Combined multiples goals through reward function and tuning

- Here:
- AC : Actor-Critic
  - Use Q to reduce variance
  - Actor aims at improving policy (adaptive search element)
  - Critic evaluates the current policy (adaptive critic element)
  - Learning is based on the TD error t
  - Reward only known to the critic
  - Critic should improve as well
- A2C
  - Advantage Actor-Critic
- A3C[1]
  - Asynchronous Advantage Actor-Critic
  - Efficient/Independent training
  - Experience replay, parallel actor-critic learners
  - Discrete or continuous contexts

[1] Asynchronous Methods for Deep Reinforcement Learning, ICML, 2016

CLTV RL: Algorithm Design(3)

- Q-function

high values = pleasure

high values = pleasant anticipation

$$Q_{t+1}(s_t,a_t) = \underbrace{Q_t(s_t,a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t,a_t)}_{\text{learning rate}} \times \left[\overbrace{\underbrace{R_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_a Q_t(s_{t+1},a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q_t(s_t,a_t)}_{\text{old value}} \right]$$

low values = pain

low values = fear

- AC and A2C

$$\begin{aligned} \nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ Q\ (s,a)\right] && \text{Q Actor-Critic} \\ &= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ A\ (s,a)\right] && \text{Advantage Actor-Critic} \end{aligned}$$

  - Here, K-Step advantages:

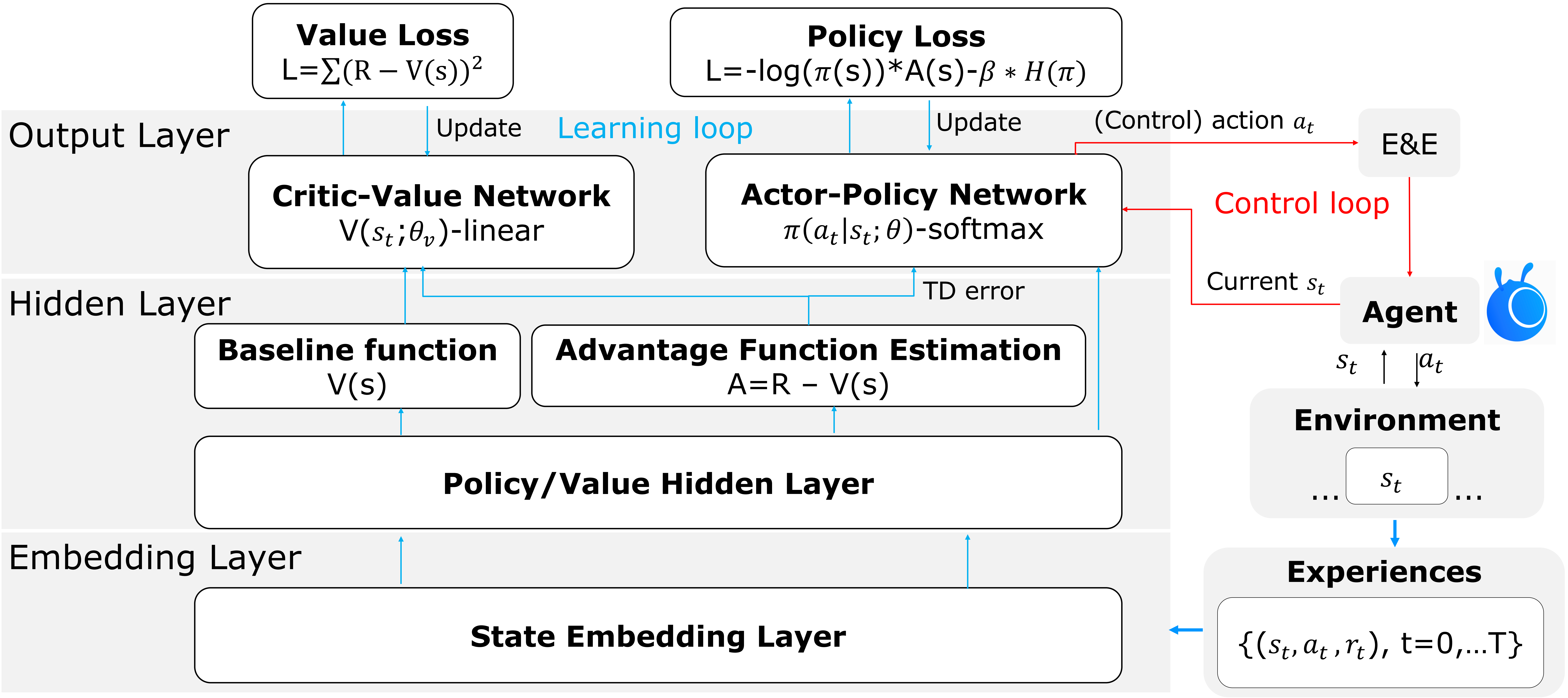
$$A(s_t,a_t) = \underbrace{\sum_{i=0}^{k-1} \gamma^i R_{t+i} + \gamma^k V(s_{t+k})} - V(s_t)$$

Reward obtained

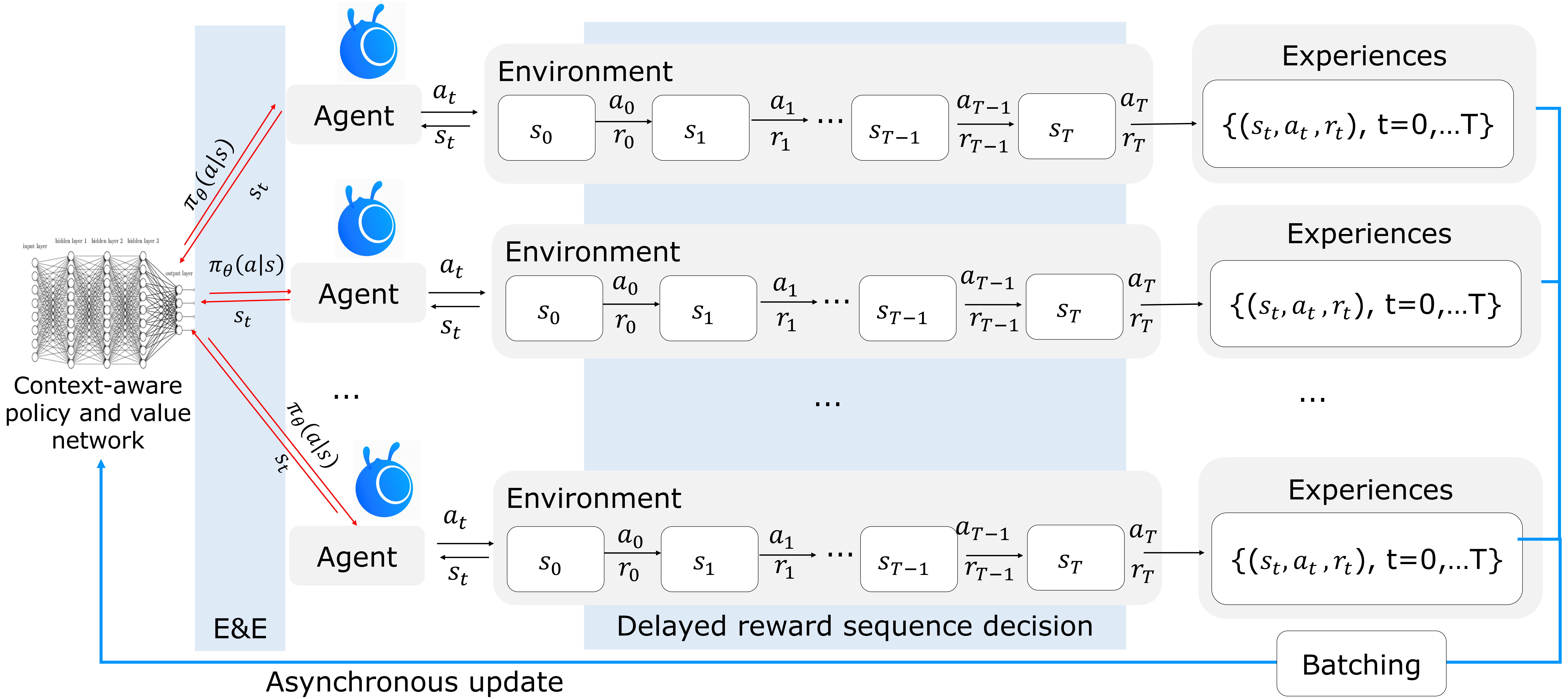
Estimate @ future time step

Baseline return

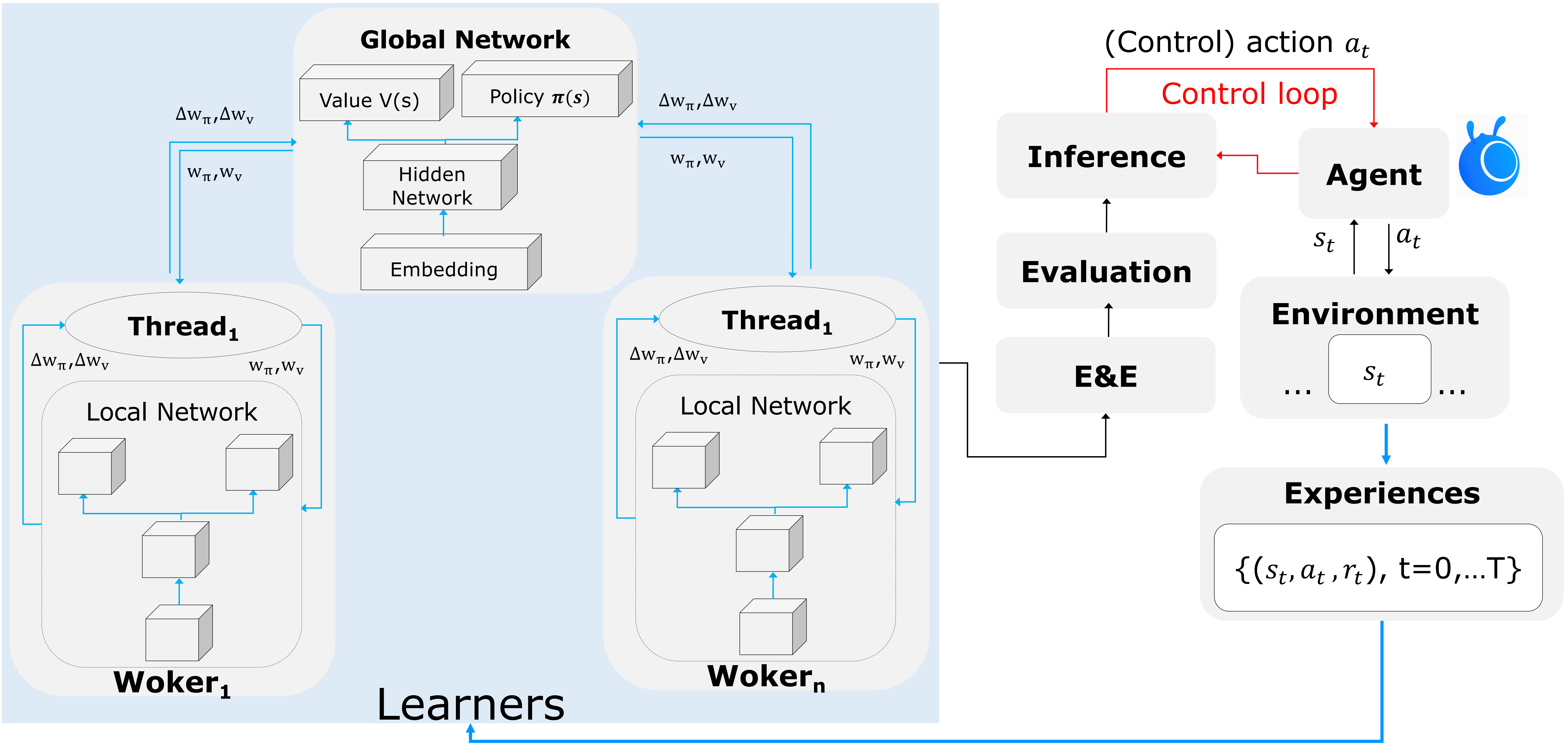
CLTV RL: Algorithm Design(4)



CLTV RL: Algorithm Implementation(1)



CLTV RL: Algorithm Implementation(2)



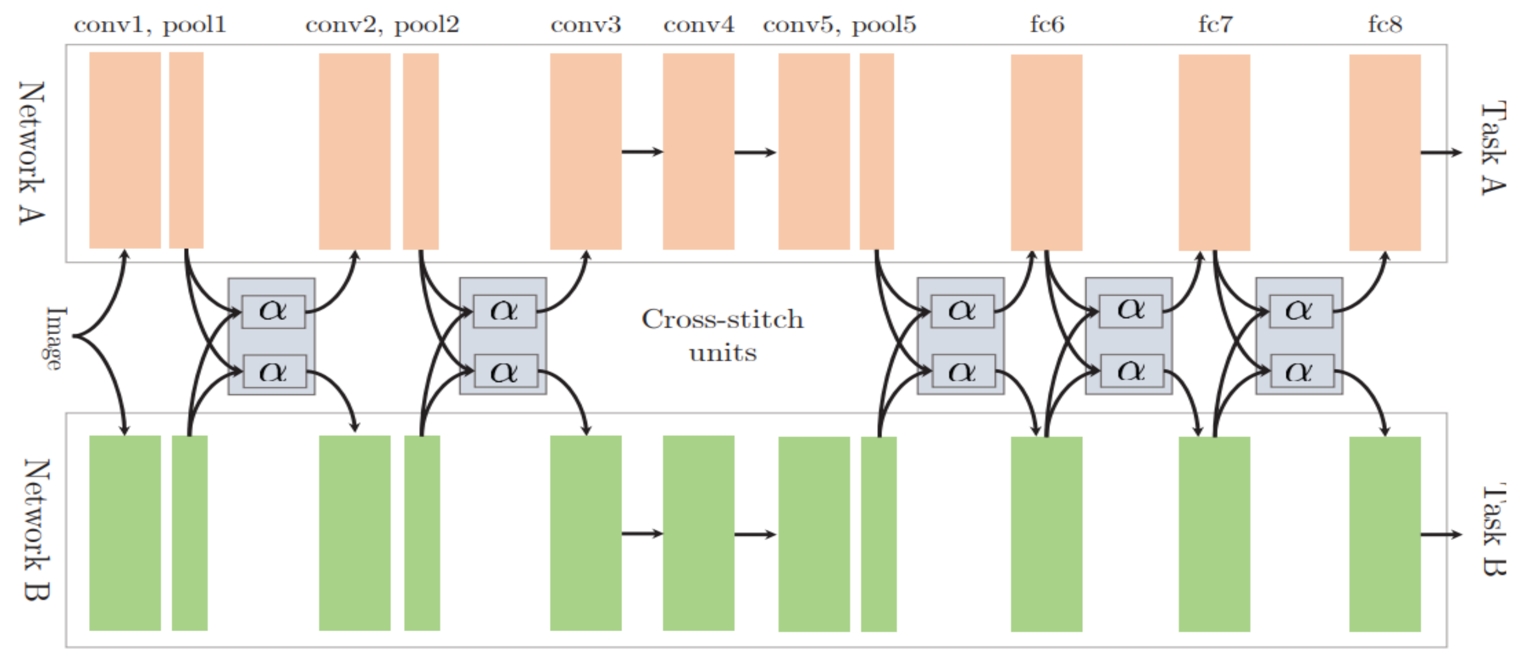
CLTV RL:Experimental Design for ABTest(1)

- The problem was formulated as a classification problem
  - Sign and click object and separately build two models
  - Given an user, the models predict the action that can make the user sign or click with max probability
- Performance among DRL , MTL methods and single DNN method were compared , especially for DRL with multi-task/multi-View/multi-Object supervised learning
  - Tensor Factorization for MTL through tensor trace norm[1] and Cross-Stitch MTL[2] methods were choosed
  - Tensor Trace Norm MTL
  - Cross Stich MTL

$$\textbf{(Tensor Trace Norm) Tucker} \quad ||\mathcal{W}||_* = \sum^{N} \gamma_i ||\mathcal{W}_{(i)}||_*$$

$$\textbf{(Tensor Trace Norm) TT} \quad ||\mathcal{W}||_* = \sum_{i=1}^{N-1} \gamma_i ||\mathcal{W}_{[i]}||_*$$

$$\textbf{(Tensor Trace Norm) Last Axis Flattening} \quad ||\mathcal{W}||_* = \gamma ||\mathcal{W}_{(N)}||_*$$



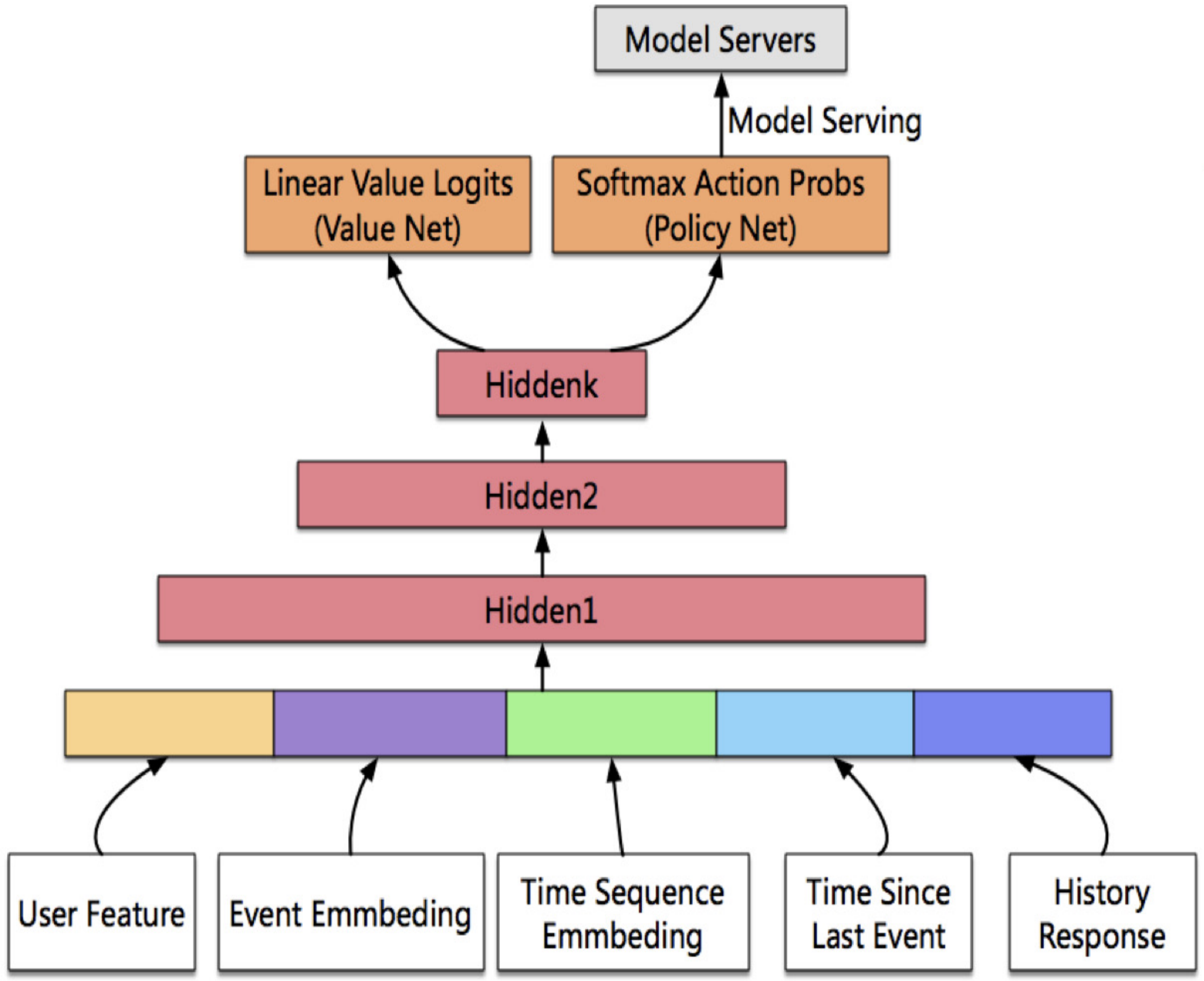
- **Using cross-stitch units to stitch two AlexNet networks**

[1] Trace Norm Regularised Deep Multi-Task Learning, ICLR, 2017

[2Cross-stitch networks for multi-task learning[C], CVPR, 2016

CLTV RL:Experimental Design for ABTest(2)

- DRL model settings
  - Discount factor = 0.99
  - The policy network is a classification network with 3 hidden layers:
    - ✓The number of each layer: [256,256,256]
    - ✓Activation function: tanh
    - ✓Learning rate: 0.00025
    - ✓Loss function: cross-entropy
  - The value networks is a regression network with 3 hidden layers:
    - ✓The number of each layer: [256,256,256]
    - ✓Activation function: tanh
    - ✓Learning rate: 0.00025
    - ✓Loss function: squared difference



CLTV RL:Experimental Design for ABTest(3)

- Trace norm MTL(Fig.1)
  - $Loss = L1(X1,Y1)+L2(X2,Y2)+Loss\_trace\_norm(W)$
  - $Loss\_trace\_norm$: The multitask regularization term with tensor trace norm constraint (LAF, Tucker, TT)
  - The weight of trace norm term: 0.0005
- Cross Stitch MTL(Fig.2)
  - $Loss = L1(X1, Y1) + L2(X2, Y2)$
  - The cross-stitch unit is used to learning task relationship
- Model setting
  - Left network learns the sign model and the right network learns the click model
  - X1, X2: User's feature (880).
  - Y1, Y2: The labels of different users (6).
  - W: The parameters of the two networks.
  - $L1$: The cross-entropy loss function of the sign model.
  - $L2$: The cross-entropy loss function of the click model.
  - The number of each layer: [125,125,125]
  - Activation function: sigmoid
  - Learning rate: 0.001
  - Batch size: 100

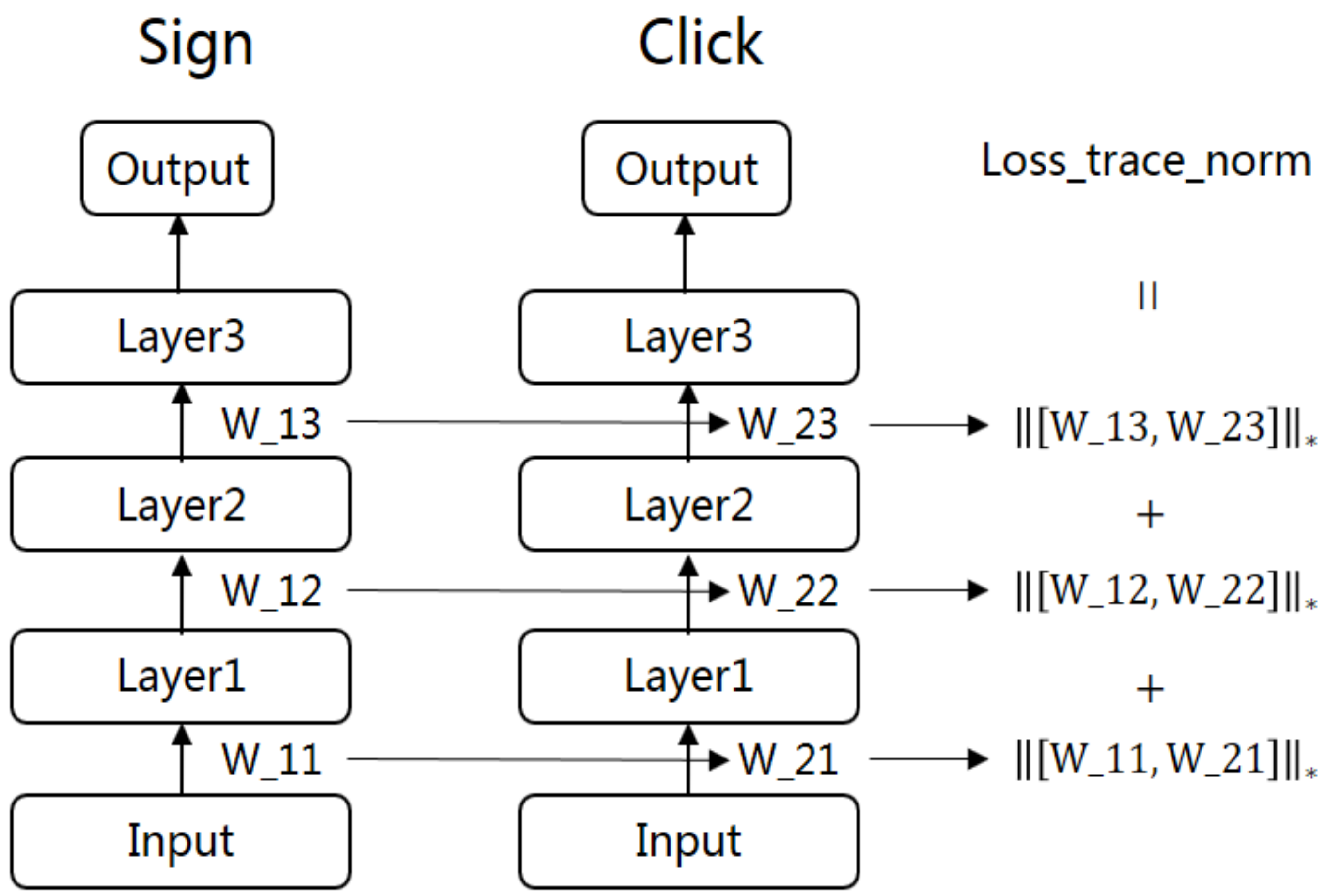


Fig. 1 Trace Norm MTL

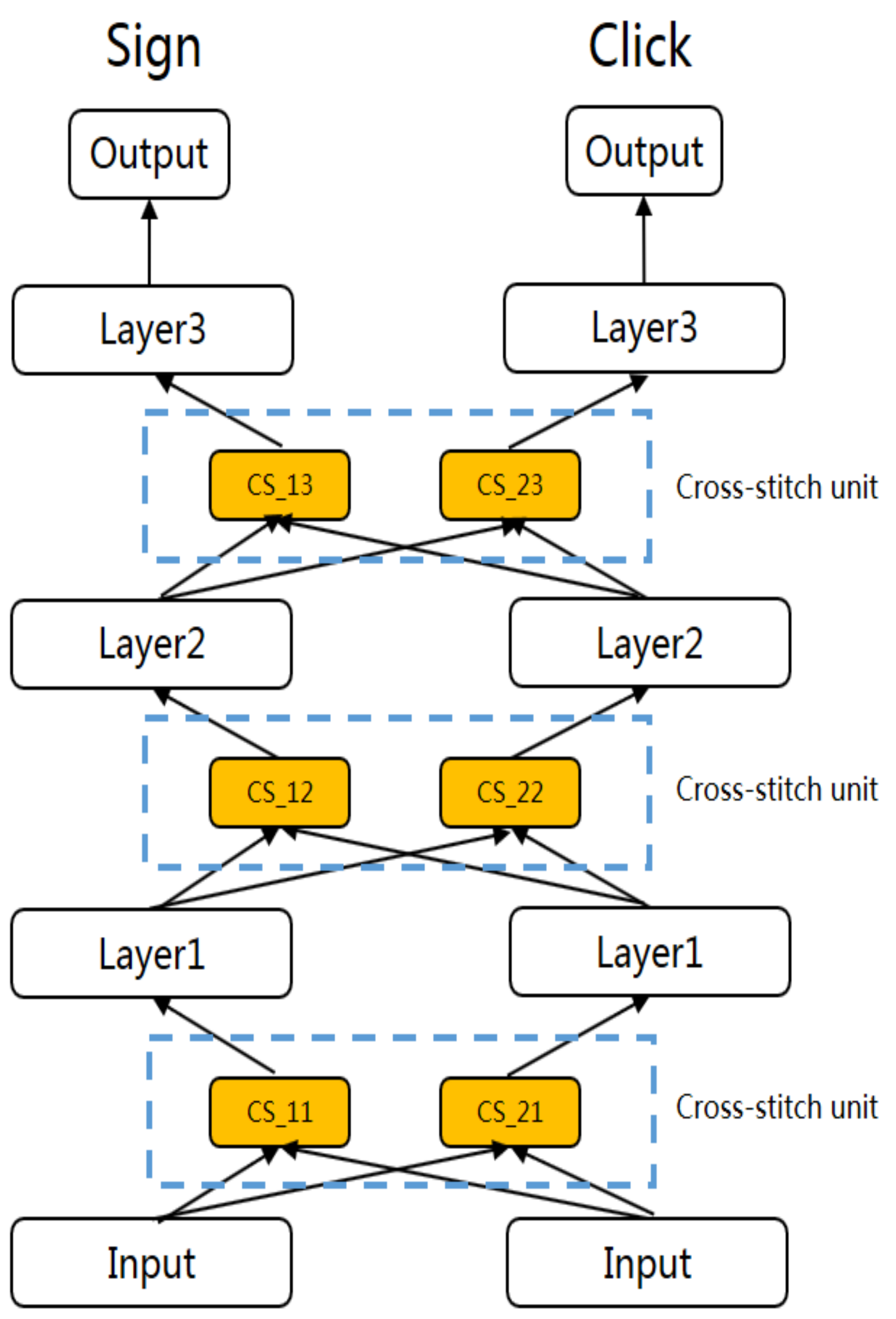


Fig. 2 Cross Stitch MTL

蚂蚁集团 ANT GROUP

CLTV RL:Experimental Design for ABTest(4)

- Comparison DRL with MTL with BPI(Business Performance Index)

Methods	convRateLift	avgHitConvCost	avgAllConvCost
MTL-TN-TT	-10.53%	3.80	4.15
MTL-TN-Tucker	-15.84%	3.96	4.15
MTL-TN-LAF	-18.26%	3.92	4.15
MTL-CS-125	-18.34%	3.72	4.15
MTL-CS-256	-20.55%	3.92	4.15
MTL-CS-525	-19.10%	3.99	4.15

$$Lift_{bpi}(\pi) = \frac{ConvRate(C) - ConvRate(B)}{ConvRate(B)}$$

s.t.

$$A = \{s \in U \mid a = \pi_\theta(s)\}$$
$$B = \{s \in U \mid a = actual\_offer(s)\}$$
$$C = \{s \in U \mid a = \pi_\theta(s) \ \& \ \pi_\theta(s) = actual\_offer(s)\}$$
$$|C| \geqslant \gamma |B|, \qquad \gamma \leqslant 1$$

- It shows that the performance DRL method better than this two type of MTL methods

- For our other related work, please refer to the following papers:

[1] Reinforcement Learning for Uplift Modeling, arxiv:1811.10158, 2018(Cooperated with Prof Xiaotie Deng)

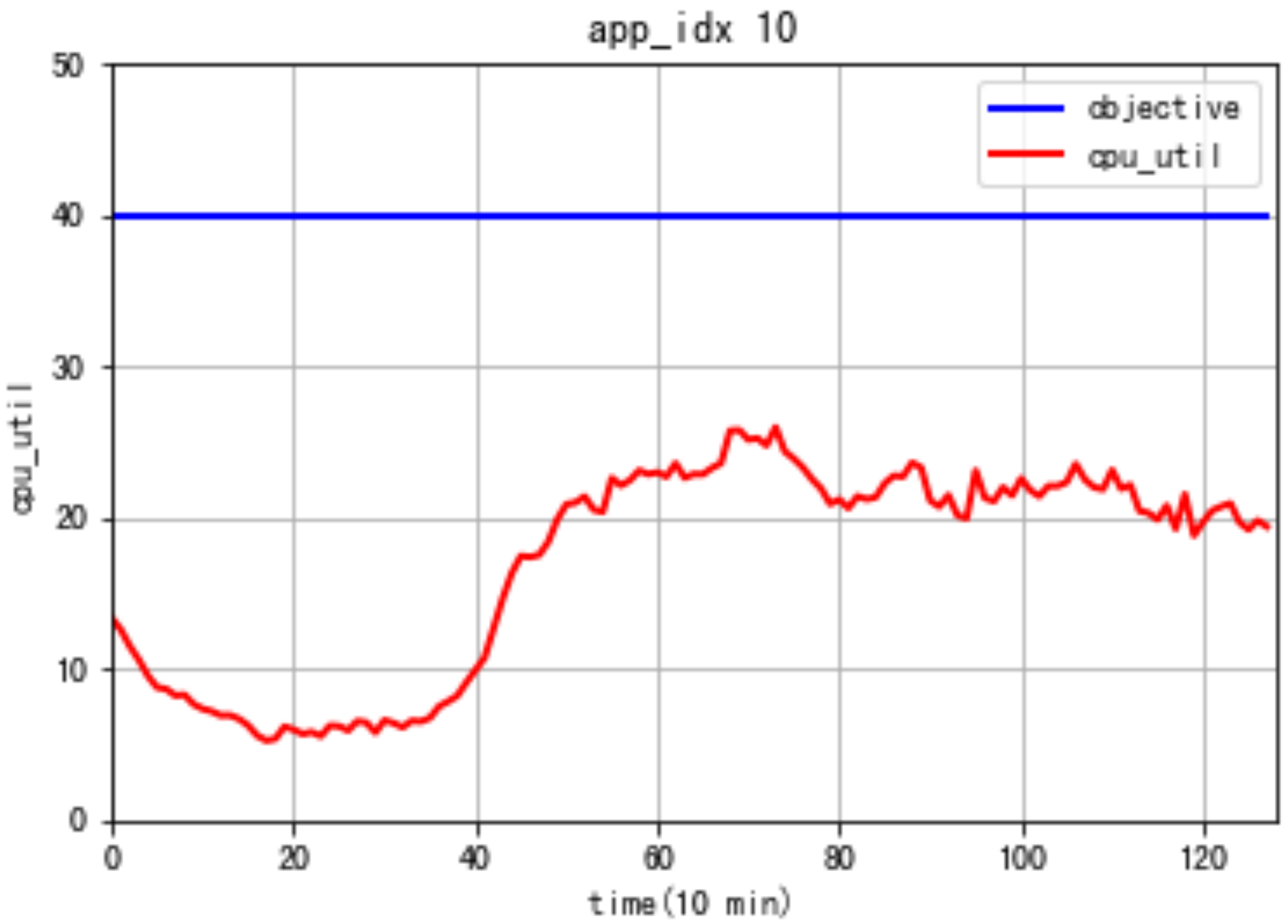
[2] Latent Dirichlet Allocation for Internet Price War, AAAI, 2019 (Cooperated with Prof Xiaotie Deng)

[3] Cost-Effective Incentive Allocation via Structured Counterfactual Inference, AAAI, 2020 (Cooperated with Prof Michael I. Jordan, Le Song)

3. Green AI: Cloud Resource Scheduling Management

Cloud Resource Scheduling Management (CRSM)

- Background



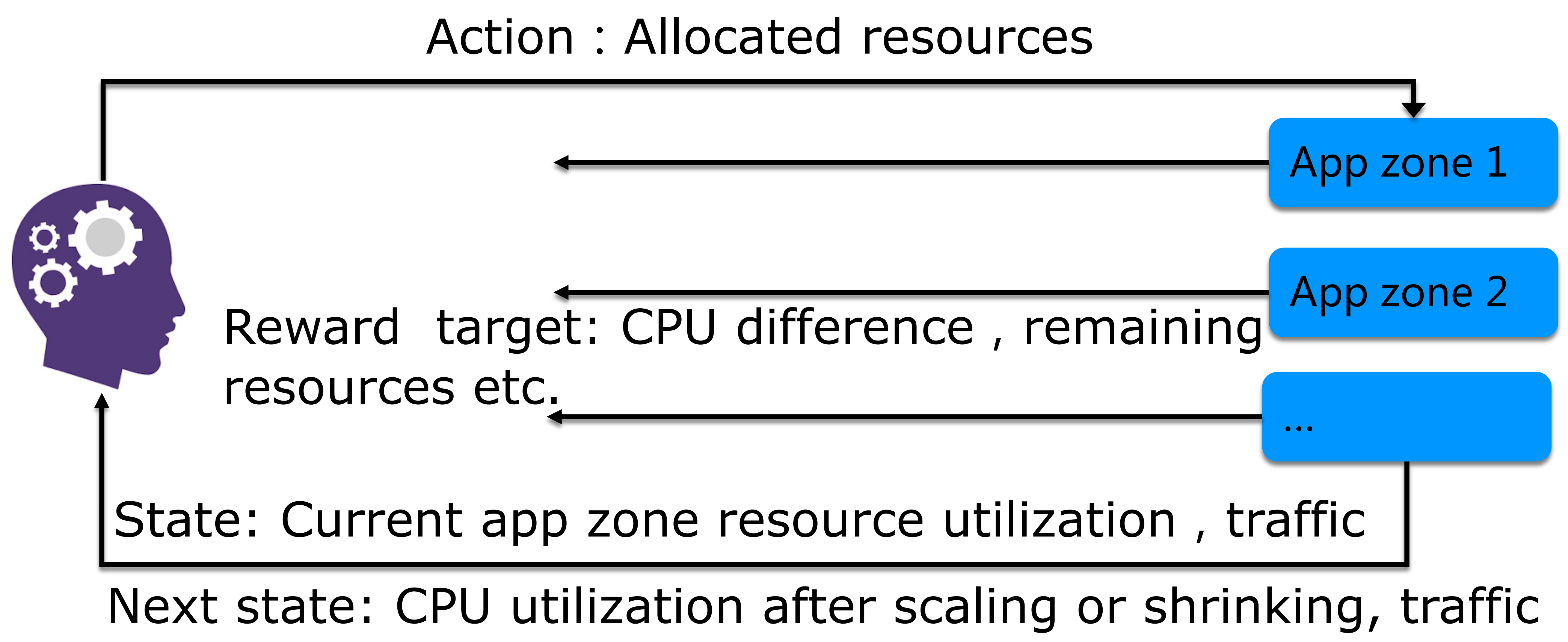
- Problem
  - Low Computing resource utilization
  - Great variations of the CPU utilization at different times
  - Huge differences among different apps and zones
- Goal
  - Automatic allocation(scaling or shrinking) of machines to each app and zone with CPU utilization high enough but stable
  - More flexible cloud services and user configuration policies
  - Intelligent procurement strategy, carbon neutral
- Benchmarks
  - Amazon EC2[1], Google cloud autopilot[2]

[1] https://docs.aws.amazon.com/autoscaling/index.html

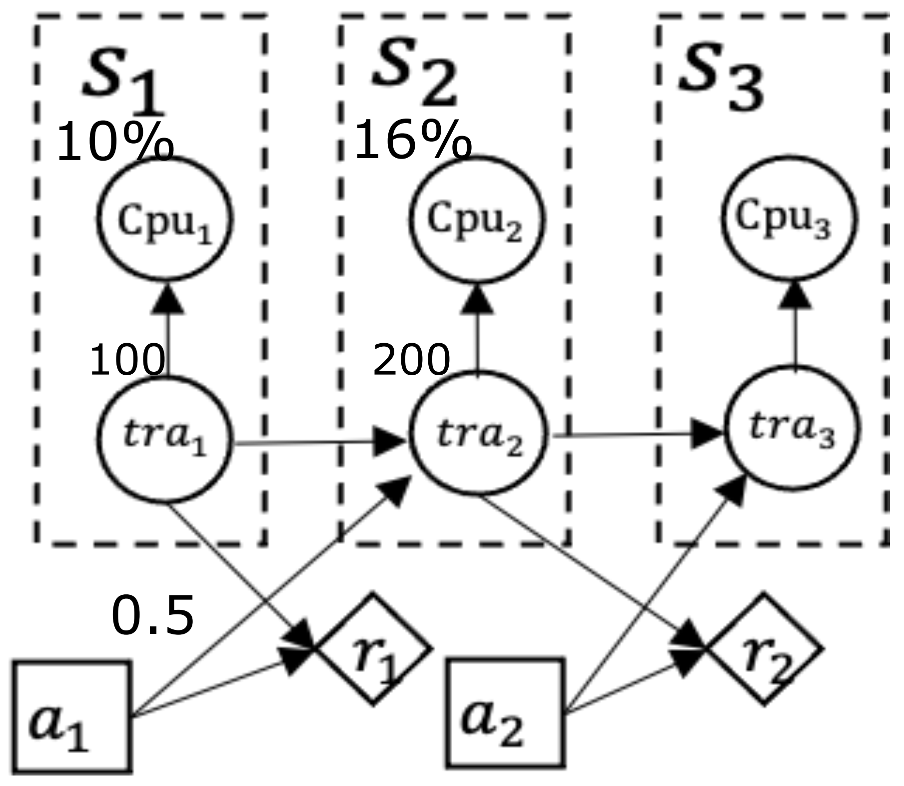
[2]Autopilot: Work autoscaling at Google, EuroSys, 2020

CRSM Modeling: Through Agent Decision Making Based on Meta-RL

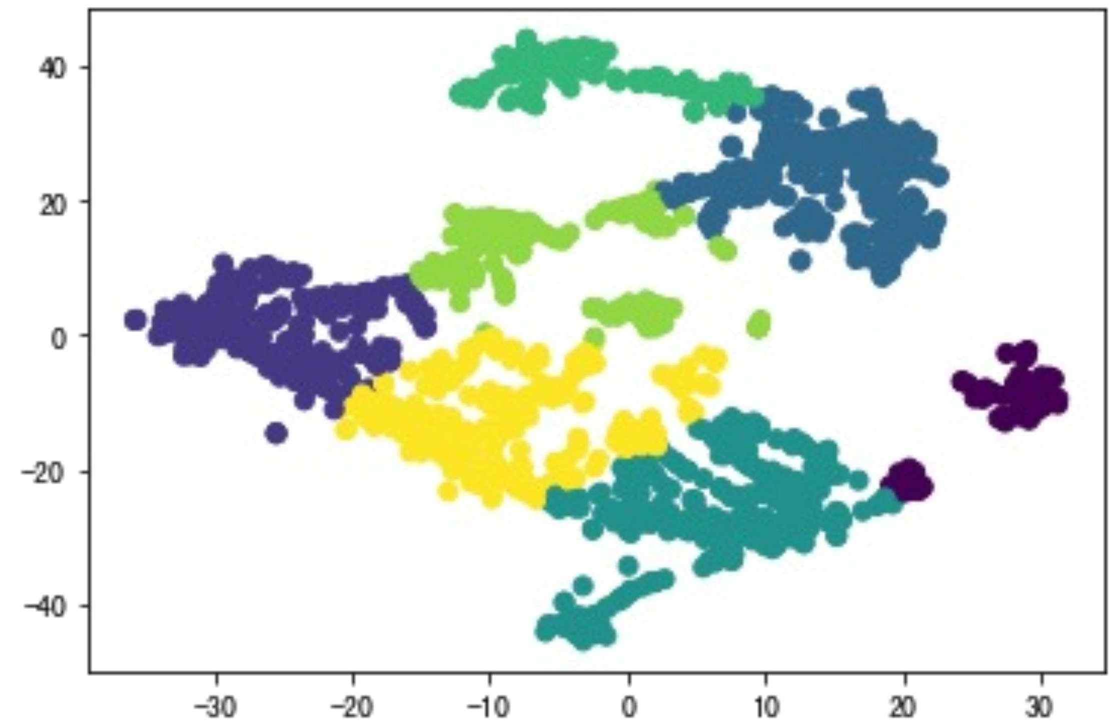
- Challenges
  - No resources changes ever occurred online(No historical data)
  - More than 30000 app zones and impossible to model each one individually
  - Risky online assessment strategies
  - RL ?



- Solutions: Meta model-based RL
  - Formulate individual app zone and its allocated resources with the business logic into the dynamic model
  - Uniformly model thousands of app zones with meta learning
  - Offline evaluation the accuracy of the model



- Traffic transition and CPU utilization fitting



- Build thousands of tasks into several large clusters
- model thousands of app zones with meta learning uniformly
- Visualizing Data using t-SNE[1]

[1] Visualizing Data using t-SNE, JMLR, 2008

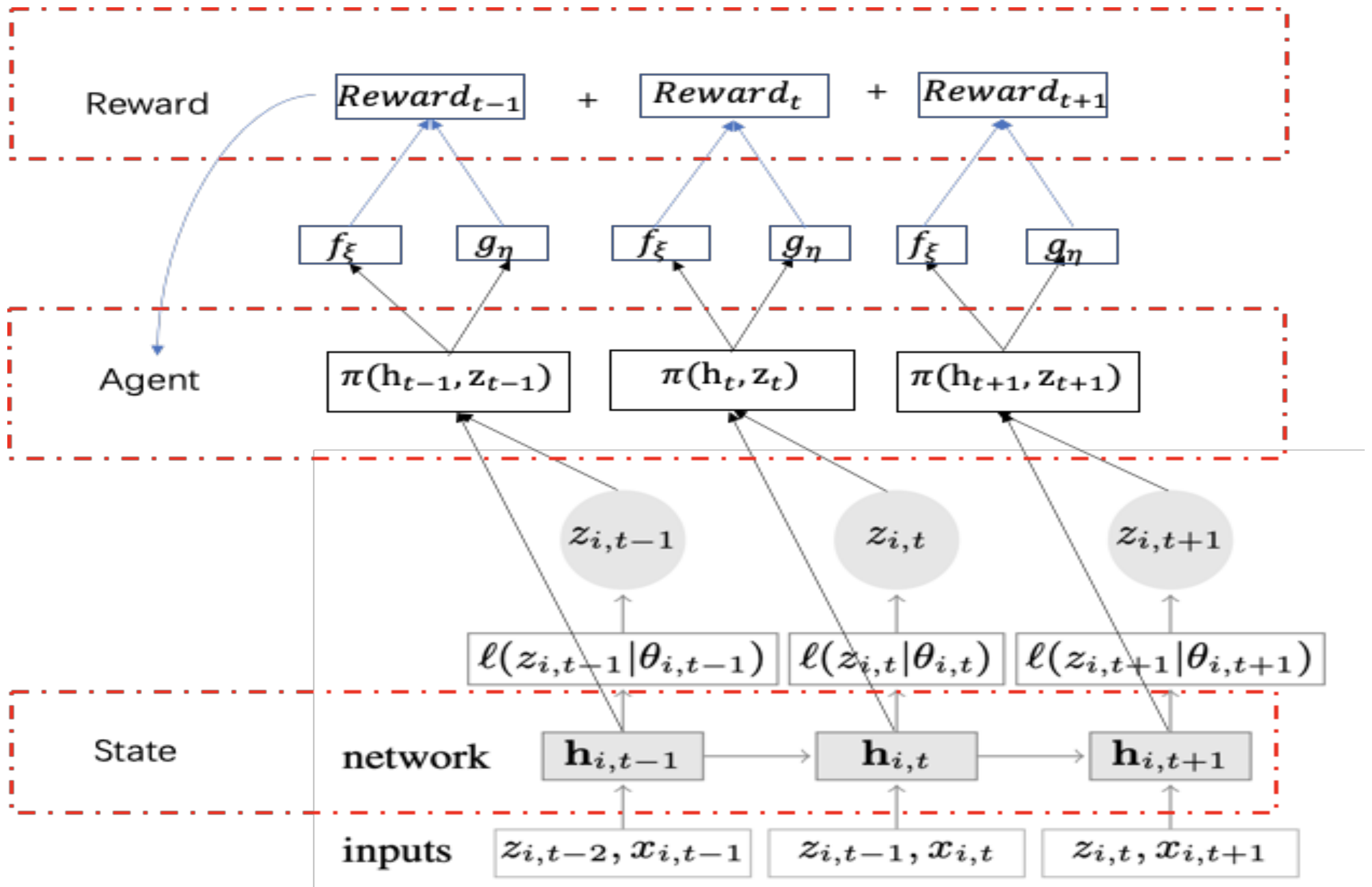
蚂蚁集团 ANT GROUP

CRSM Meta-RL: Algorithm Design(1)

- Model-based RL
  - Few opportunities to interact with online and the interaction is high risk
  - Transitions and rewards are partially defined by fixed logic, and the whole process can be differentiable
  - Environment model and CPU utilization updated by new policy can be partially evaluated offline
- RL model design
  - State: Predicted traffic information, CPU utilization, etc.

    s=($h_{i,t}$,predicted_qps)
  - Reward: Difference between current CPU utilization ratio and ideal utilization ratio, penalty term, reward function:

    $r(s,a) = -||cpu_{target} - s_{cpu}||_2^2 + \delta$
  - Action: Allocation(scaling or shrinking) ratio



- Embedding layer(Deep autoregressive model[1])

  $h_{i,t} = h(h_{i,t-1}, z_{i,t-1}, \Theta)$

  Here, likelihood factor: $\prod_{i=0}^{T} l\ (z_{i,t}|\theta(h_{i,t}, \Theta))$
- CPU Utility: $CPU_{util} = f_\xi(qps, h_i, action)$
- SLO(Service Level Objective)[2] Utility:

  $SLO = g_\eta(qps, memory, action)$

[1]A Spatial–Temporal Attention Approach for Traffic Prediction, T-ITS, 2021
[2]FIRM: An Intelligent Fine-Grained Resource Management Framework for SLO-Oriented Microservices, 2020, OSDI

CRSM Meta-RL: Algorithm Design(2)

- RL model design
  - Transition: Fixed allocation rule; CPU utilization, decided by traffic and transition learning[1]:

$$\left(s'_{traf}, s'_{cpu}\right) = \left(\frac{s_{traf}}{a}, ANP\left(\frac{s_{traf}}{a}\right)\right) \doteq g(s_{traf}, a)$$

  - Policy: A neutral network with input s and task embedding e_{task}, $a = \pi(s, e_{task})$, here, task embedding is learned through attentive neutral process(Maximizing the following evidence lower bound(ELBO)[1]):

$$\mathrm{Max}_{\theta,\phi} E_{q(z|s_T)}[\log p_\theta(y_T|x_T, r_c, z) - D_{KL}(q_\phi(z|s_T)||q_\phi(z|s_c)\,)]$$

Policy training[2,3,4]: $V(s,z) = r(s,a,z) + \gamma V(g(s, \pi_\theta(a|s,z)), z))$

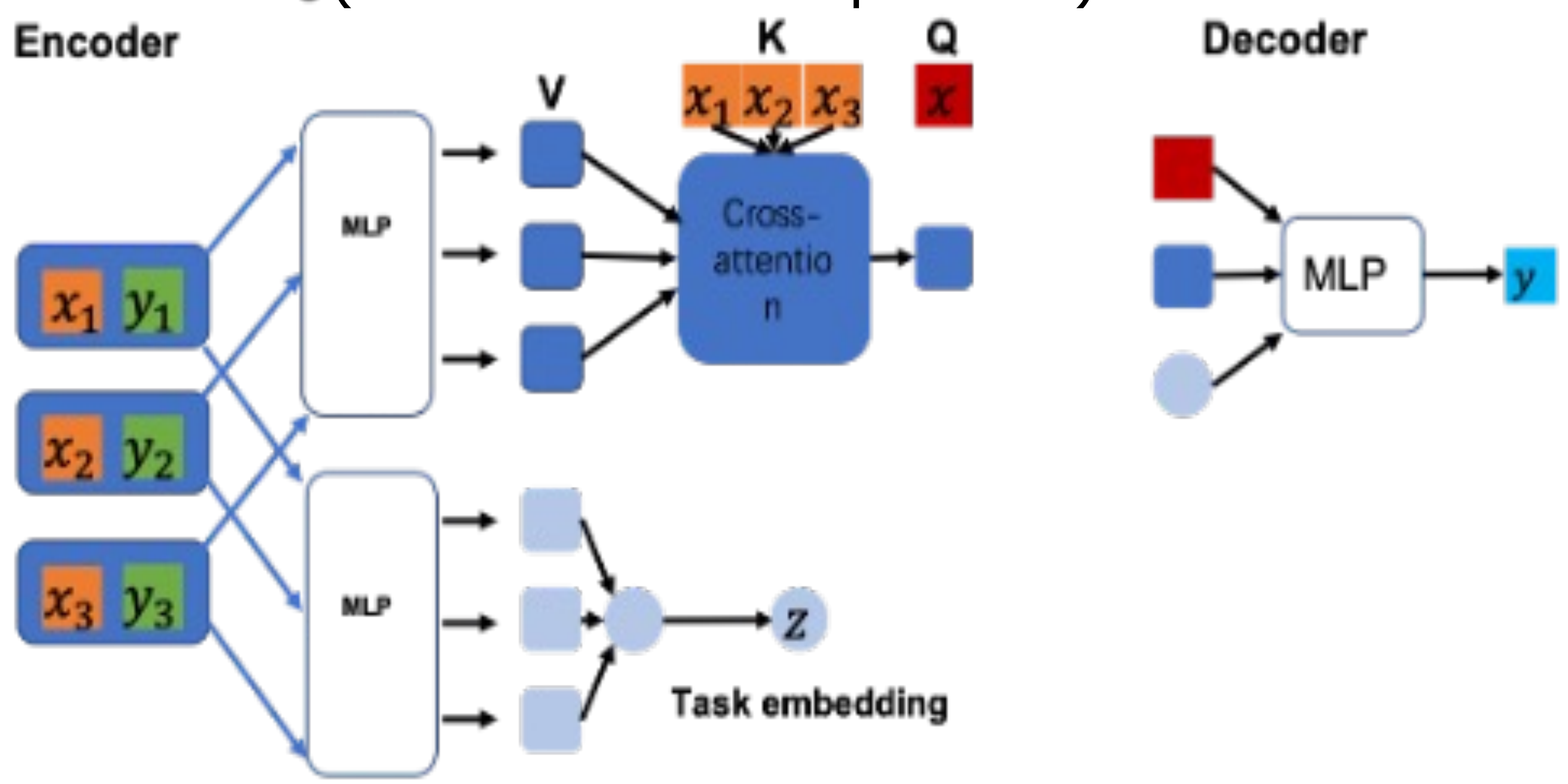
$$\theta \leftarrow \theta + \beta \frac{\partial}{\partial \theta} V(s,z)$$

Training Loss: $\mathrm{Min}_\pi \sum_0^T (CPU_{util} - CPU_{ideal})^2 + \lambda * SLO_t$

[1] Attentive neural process, ICLR 2019
[2] Model Embedding Model-Based Reinforcement Learning, arxiv:2006.09234, 2020
[3] Learning Continuous Control Policies by Stochastic Value Gradients, neurips, 2015
[4] Dream to control: Learning behaviors by latent imagination, ICLR, 2019

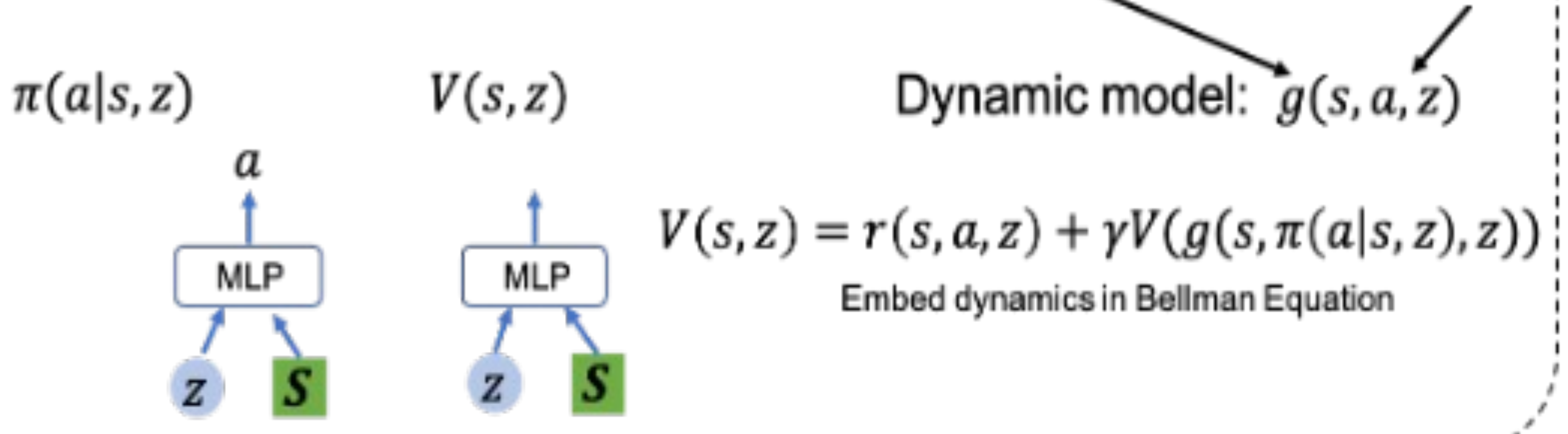
CRSM Meta-RL: Algorithm Design(3)

Task learning (Attentive neural process)



Policy learning

Traffic formulation

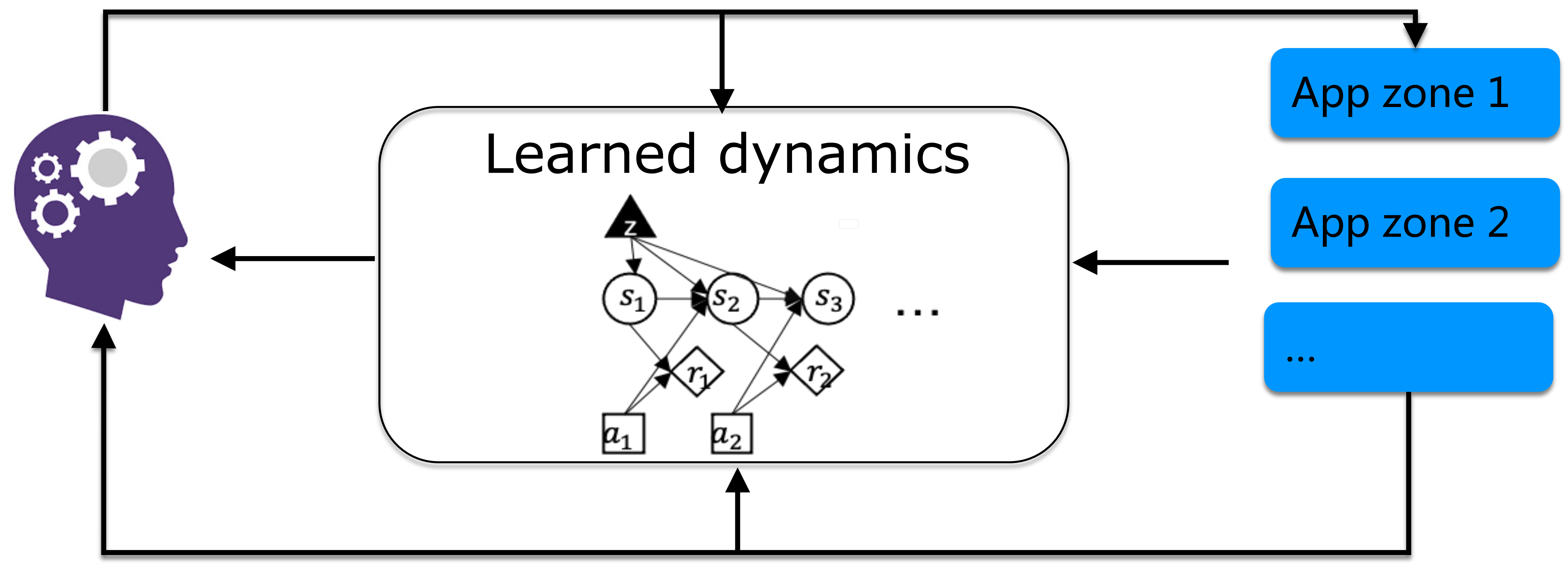


$$V(s,z) = r(s,a,z) + \gamma V(g(s,\pi(a|s,z),z))$$

Embed dynamics in Bellman Equation

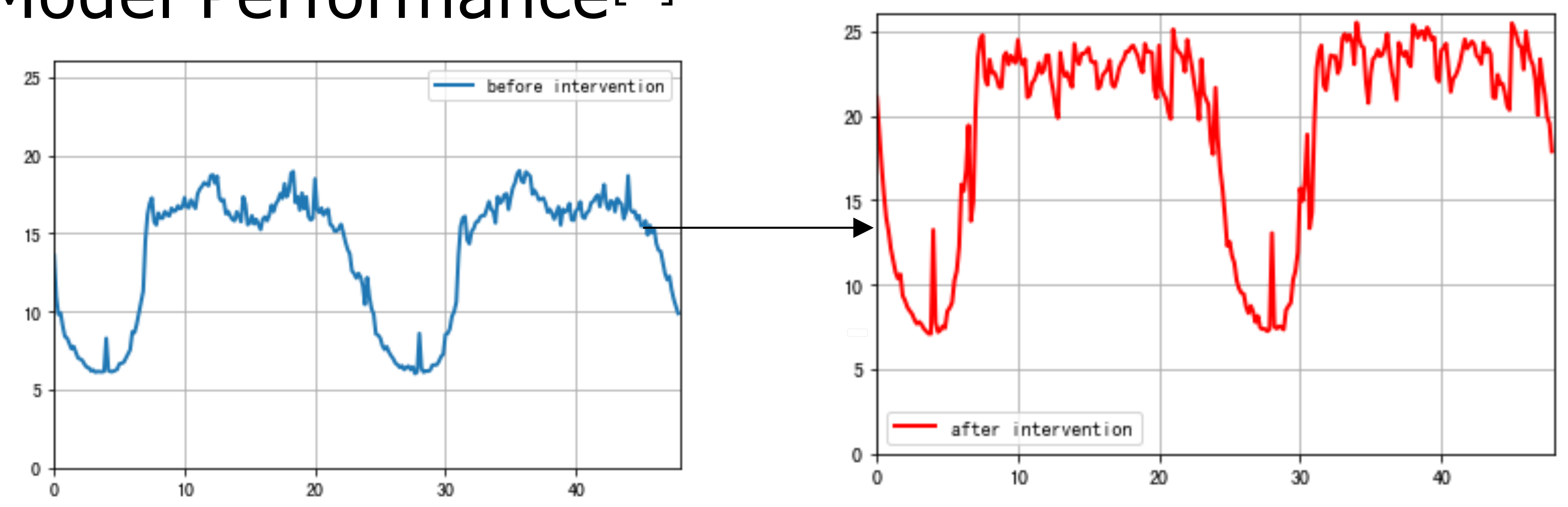
Meta model-based RL

Action : Allocated resources



Learn from imagination

Model Performance[1]



- Save resources 50% daily
- One model covers 1000 app zones

- Automatic Offline evaluation: Automatically filter apps that can be scaled according to model metric

[1] A Meta Reinforcement Learning Approach for Predictive Autoscaling in the Cloud, KDD, 2022

4. Agent Based Reinforcement Learning(RL): Algorithm Library, Dataflow Framework and System Platform

蚂蚁集团
ANT GROUP

Agent Based RL: Algorithm Library

Technique Problems

- Complex Embedding
- Offline Training
- Sample Efficiency
- Multi-Goal/Multi-Task
- Hybrid actions
- Offline Evaluation
- Dynamic Updating
- Overall Scheduling

Algorithms Library

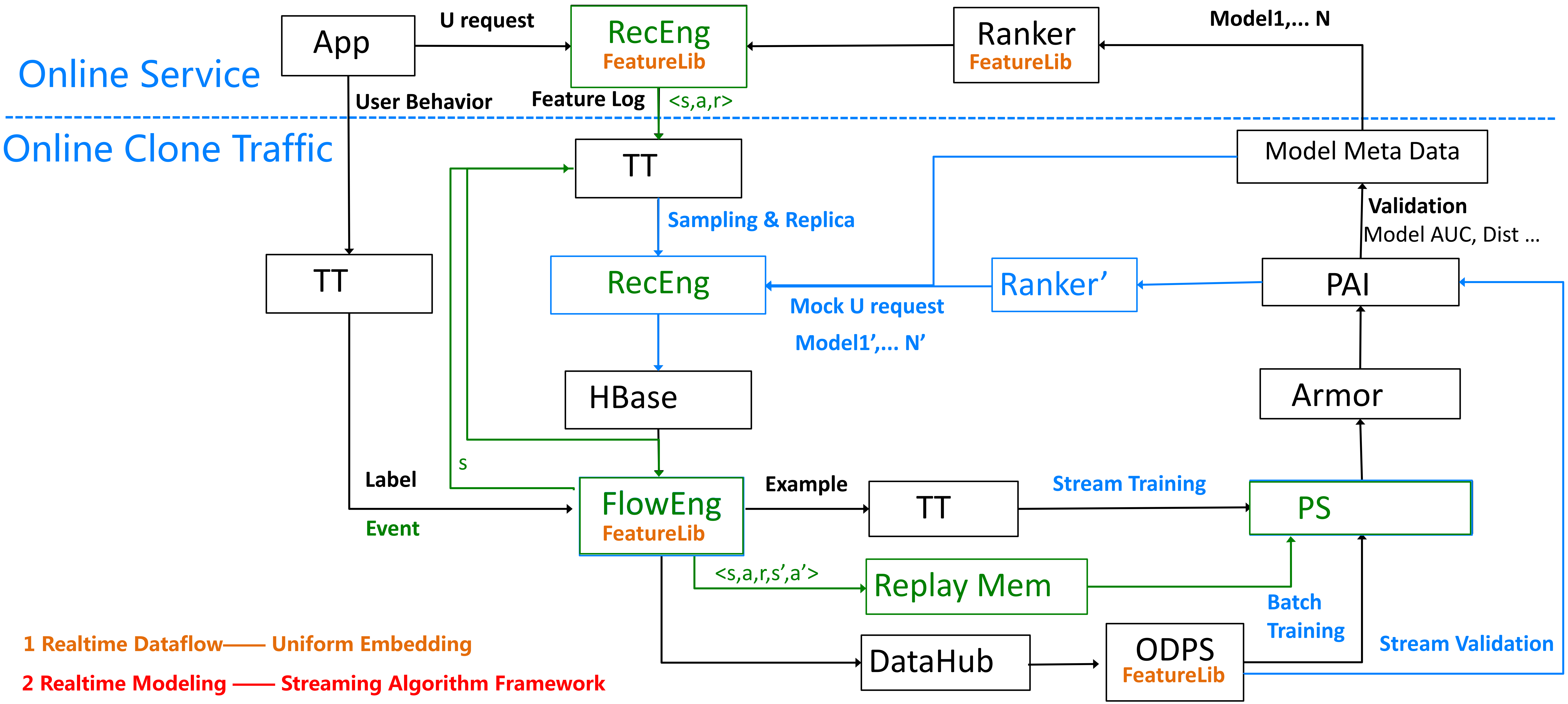
- Hyperbolic Embedding[1]
- Off-policy, model-based RL, and Batch RL[2]
- Goal conditioned RL, Meta RL and Lifelong RL
- Hybrid actions RL and Combinatorial RL
- Off-policy RL
- Online Learning
- Multiagent RL[3,4]

Algorithm Models

- HAN, HECO
- DQN, QR-DQN
- DDPG, TD3
- SAC, Hybrid Action SAC
- MEMB
- D4PG, QR-D4PG, BCQ
- UVFA, HER
- PEARL
- Double Robust, Dual Dice, OPC
- Contextual Bandit, Bandit with Knapsack
- Online DL/On-policy RL
- MADDPG, QMIX

[1] Unit Ball Model for Embedding Hierarchical Structures in the Complex Hyperbolic Space, arxiv:2105.03966, 2021(Cooperated with Prof Yangqiu Song)
[2] Model Embedding Model-Based Reinforcement Learning, arxiv:2006.09234, 2020
[3] Variational Policy Propagation for Multi-agent Reinforcement Learning, arxiv:2004.08883, 2020
[4] Value Propagation for Decentralized Networked Deep Multi-agent Reinforcement Learning, NeurIPS 2019

Agent Based RL: Dataflow Framework



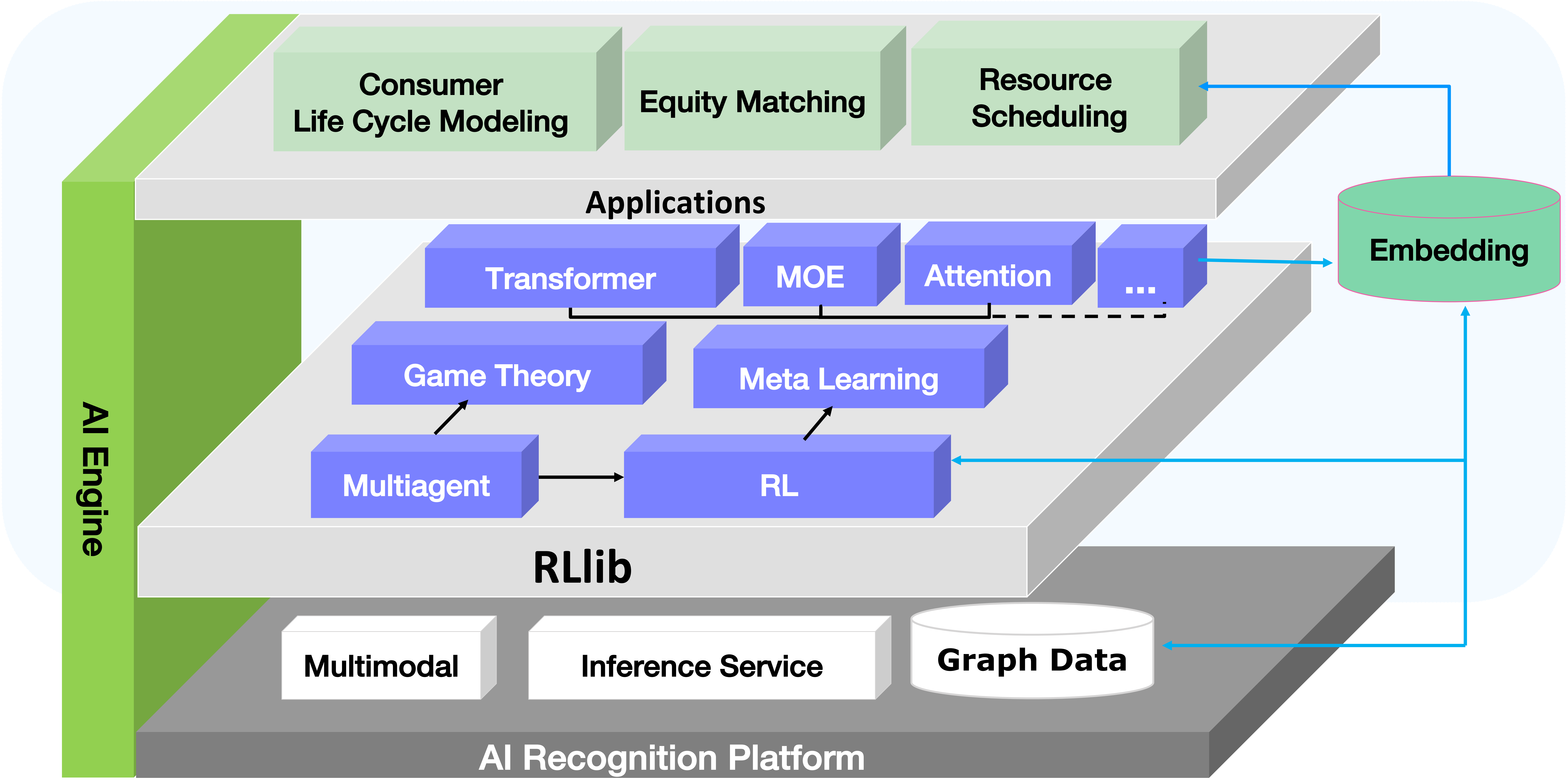
1 Realtime Dataflow—— Uniform Embedding

2 Realtime Modeling —— Streaming Algorithm Framework

3 Real-time Learning Evaluation - Streaming Learning Framework & Streaming Model Evaluation

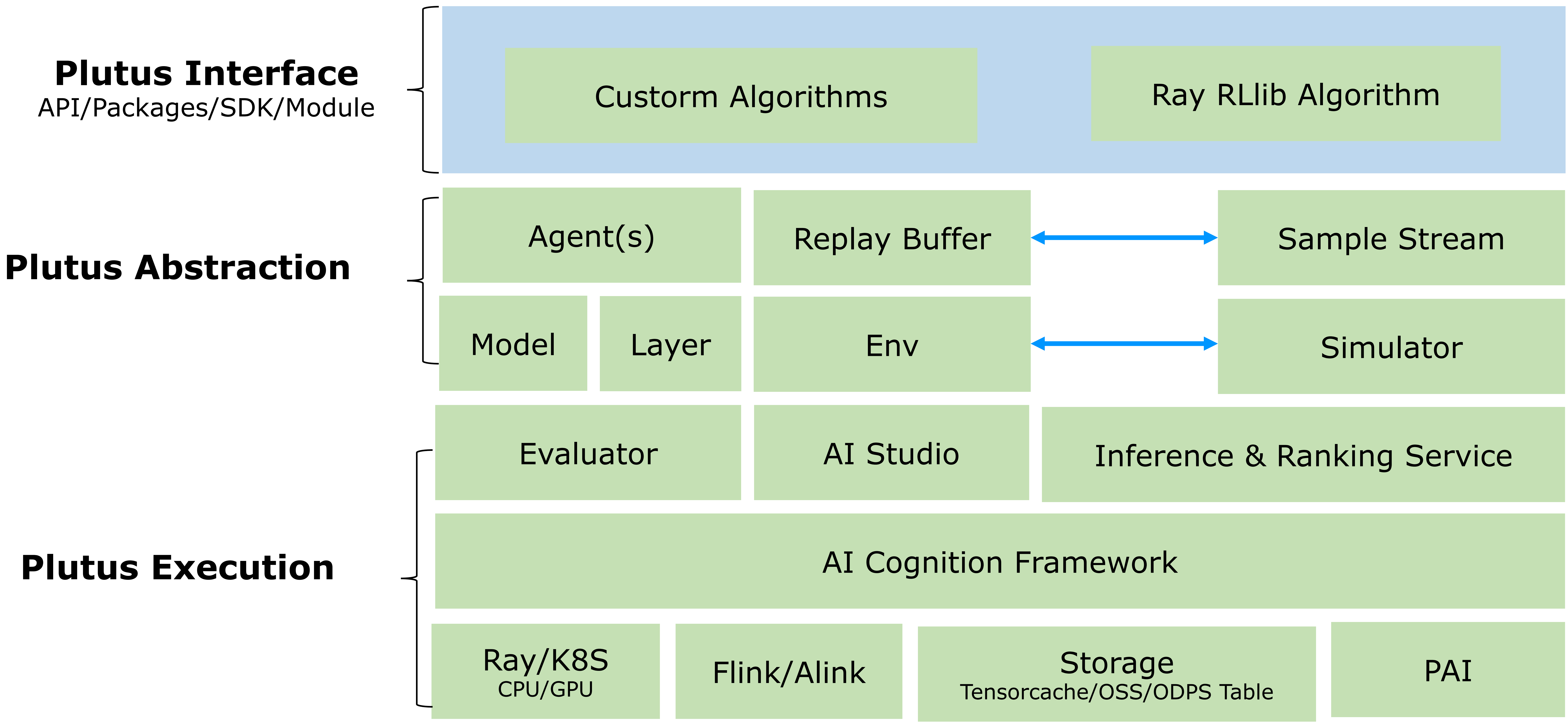
4 Real-time Decision Making - Streaming Online Reinforcement Learning

Agent Based RL: System Platform



5. What’s ongoing & next

Agent Decision Making: Agent Based RL Development Toolkit—Plutus



蚂蚁集团
ANT GROUP

Agent Decision Making: One-step Service

产品服务
- 一站式建模服务
- 端到端建模能力
- 一流动态博弈技术
- 开放的算法生态

动态决策
- 统一多模态&Graph Embedding
- 交互动态感知
- 长序列决策
- 多目标决策
- 多任务学习
- 多智能体博弈
- 云端一体

感知交互

智能体状态(State)
- To-C/B/I画像 场景定向 经济状况
- 产品/服务权益 权益激励 服务价值
- 交互行为 浏览点击交易等

智能体决策(Action：5W+2H)
- 择对象 人(群)机构等
- 择时 早中晚...
- 择地 场景
- 择行为 权益
- 择途径 渠道
- 择代价 Cost
- 择理由 创意 解释

智能体反馈(Reward)
- LTV
- ROI

算法库
- 强化生命周期 A3C ...
- 强化Uplift RLift ...
- 强化图概率/强化博弈 RL-PGM / GameTheory(NE, CE)
- 强化时间序列 RL-TimeSeq ...
- 强化多目标/多任务/多智能体 MetaRL/Multi-agents ...
- 强化组合控制 RL-Combined Control ...
- RayRLlib ...

AI引擎
- Simulator：Sim2Real(Interactive)
- Evaluator(Flow-based)
- Inference Service

计算平台
- Ray
- Flink
- PAI

Agent Decision Making: Inclusive & Green AI

基于用户导向的资金资源资产全生命周期、全链路绿色效能

蚂蚁集团
ANT GROUP

Q&A

为世界带来微小而美好的改变

Bring small and beautiful changes to the world

DingTalk: 劲鸾
Email: junwu.xjw@antgroup.com