Q-learning bias

Off-policy Q-learning

Initialize $Q(s, a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize S
    Repeat (for each step of episode):
        Choose A from S using policy derived from Q (e.g., ε-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \alpha \big[\underbrace{\boxed{R + \gamma \max_a Q(S', a)}}_{\text{Q-target}} - Q(S, A) \big]$
        $S \leftarrow S'$;
    until S is terminal

Figure 1: Q-learning

Q value estimation

- Tabular method
  - Discrete state, action space
- Large/continuous state space
  - Function approximation
  - Deep learning
- Q-target: R+ \gamma *Q(s,a) is used in both methods.

Upward bias

- Originally referred as “overestimation” (Thrun and Schwartz, 1993)
- They used function approximation (polynomial).
- They described a systematic overestimation effect of Q-value
  - Back then they thought this was due to poor function approximation when used in recursive value estimation schemes.

Upward bias

- The same was found in a tabular setting (Hado van Hasselt, NeurIPS 2010).
- Experiment 1:
  - Roulette
    - One state, 170 actions (including betting on a number, on color etc).
    - Expected loss of $-0.053 on a dollar per bet.
    - One extra action: stop playing -> $0.

Experiment 1: Roulette

- Each trial consisted of a synchronous update of all 171 actions.

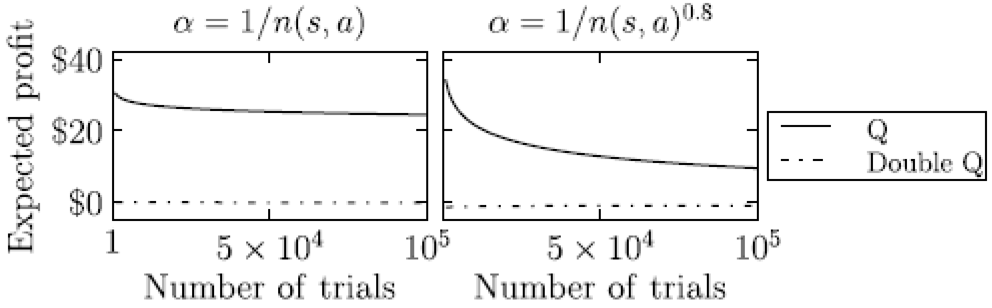
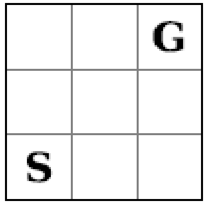


Figure 1: The average action values according to Q-learning and Double Q-learning when playing roulette. The 'walk-away' action is worth $0. Averaged over 10 experiments.

Experiment 2: Grid world

- 9 states 4 actions.
- If off the grid, the agent stays in the same state.
- Each non-terminating step, the agent receives a random reward of -12 or +10 with equal probability.
- Reaching the goal state yields +5.



Experiment 2

- Theoretical optimal value at the starting state S is at around 0.36.

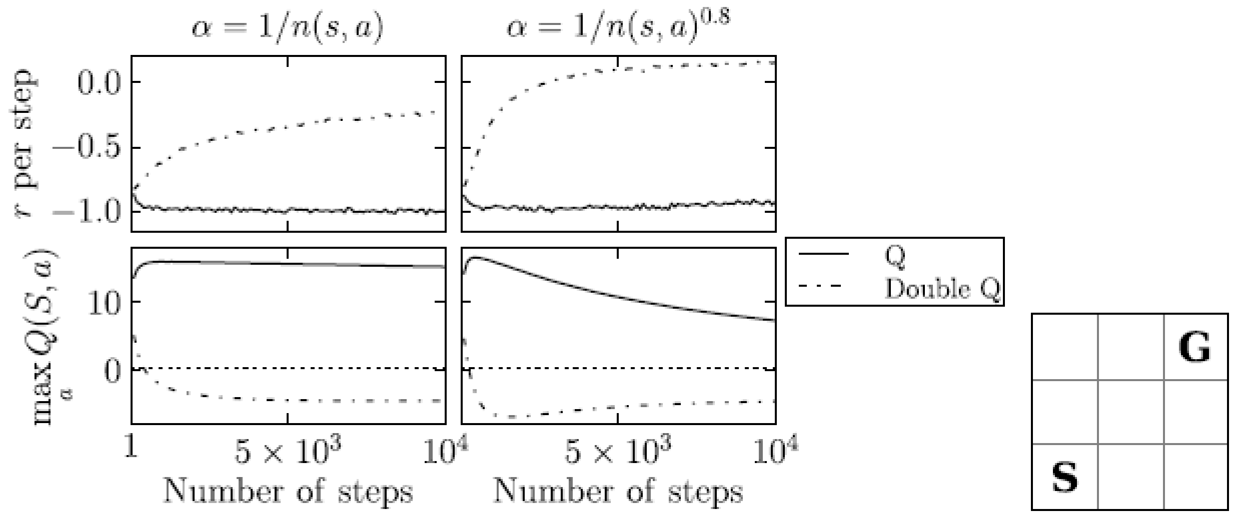


Figure 2: Results in the grid world for Q-learning and Double Q-learning. The first row shows average rewards per time step. The second row shows the maximal action value in the starting state S. Averaged over 10,000 experiments.

Upward bias

- Summarized and discussed in Hasselt, Guez & Silver (AAAI 2016)
- “Demonstrated more generally that estimation errors of any kind can induce an upward bias, regardless of whether errors are due to environmental noise, function approximation, non-stationarity, or any other source.”
- “This is important, because in practice any method will incur some inaccuracies during learning, **simply due to the fact that the true values are initially unknown.”**

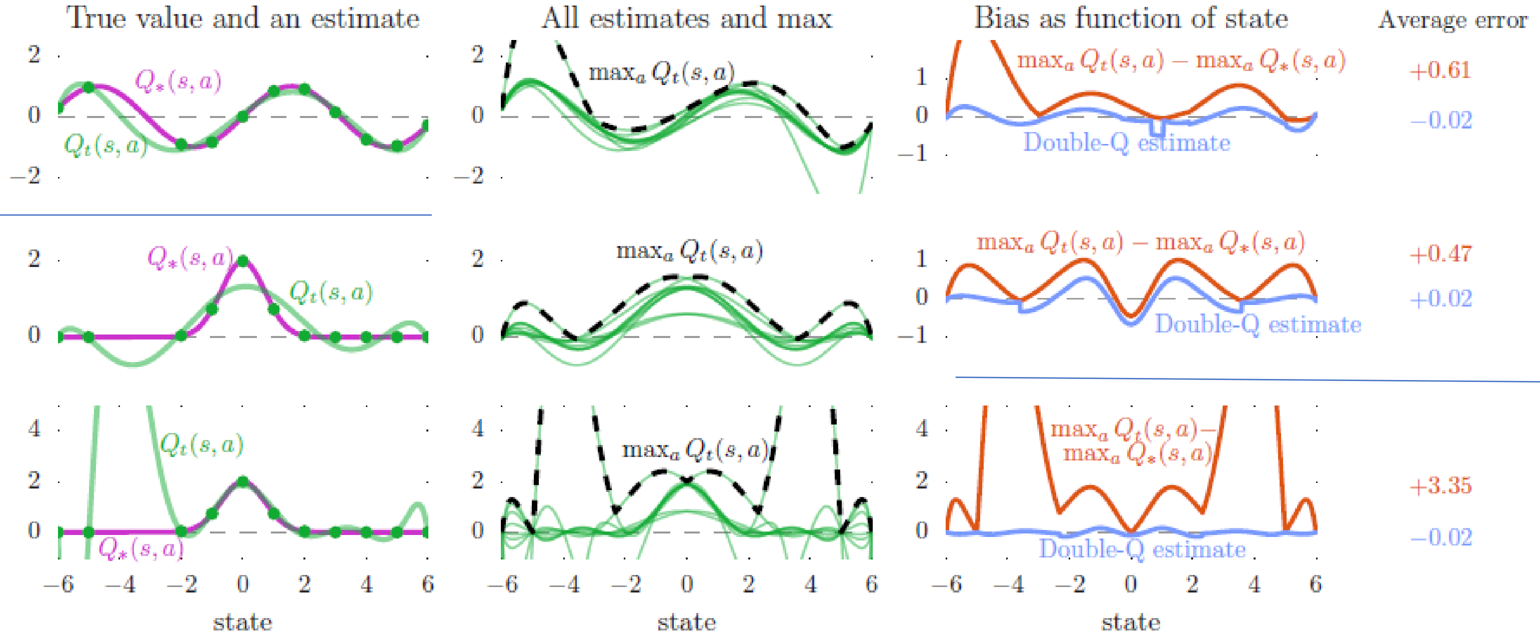
Example

- Consider a real-valued continuous state space with 10 discrete actions in each state.
- Assume the true optimal action values in this example depend only on state so that in each state all actions have the same true value.
  - Does not matter which action you take (in the long run)
- Three polynomial function approximations to estimate the true value.
  - With order (degree) d.

Example

True optimal:
$Q_*(s,a) = \sin(s)$

True optimal:
$Q_*(s,a) = 2\exp(-s^2)$



Polynomial estimation:
$d = 6$

Polynomial estimation:
$d = 9$

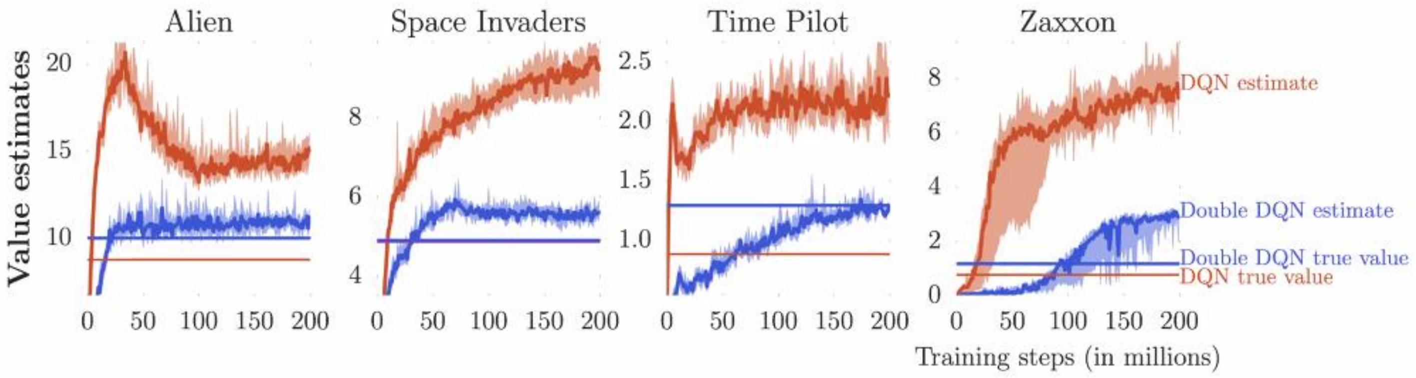
Figure 2: Illustration of overestimations during learning. In each state (x-axis), there are 10 actions. The **left column** shows the true values $V_*(s)$ (purple line). All true action values are defined by $Q_*(s,a) = V_*(s)$. The green line shows estimated values $Q(s,a)$ for one action as a function of state, fitted to the true value at several sampled states (green dots). The **middle column** plots show all the estimated values (green), and the maximum of these values (dashed black). The maximum is higher than the true value (purple, left plot) almost everywhere. The **right column** plots shows the difference in orange. The blue line in the right plots is the estimate used by Double Q-learning with a second set of samples for each state. The blue line is much closer to zero, indicating less bias. The three **rows** correspond to different true functions (left, purple) or capacities of the fitted function (left, green). (Details in the text)

Upward bias in Q learning

- In general we don't know the true Q value.
- However, Q-learning estimations/updates require $\max Q(s_{t+1}, a)$.
- If $\max Q(s_{t+1}, a)$ is overestimated for any reason, updated Q value will be overestimated as well (and further pass on to the future values).

How accurate are the Q-values?

RLChina 2020



- Overestimation
  - Target value: $r + \gamma \boxed{\max_{a'} Q(s', a'; \mathbf{w}^-)}$ This is the problem!
  - $\mathbb{E}[\max(X_1, X_2)] \geq \max\left(\mathbb{E}[X_1], \mathbb{E}[X_2]\right)$?
  - For bootstrapping (learning estimates from estimates), such overestimation can be problematic.

Double DQN

RLChina 2020

- DQN uses experience replay and target networks
  - Take action a_t according to ϵ-greedy policy
  - Store transition $(s_t, a_t, r_{t+1}, s_{t+1})$ in replay memory $\mathscr{D}$
  - Sample random mini-batch of transitions (s, a, r, s') from $\mathscr{D}$
  - Optimize MSE between Q-network and Q-learning targets

$$\mathscr{L}(\mathbf{w}) = \mathbb{E}_{s,a,r,s' \sim \mathscr{D}}\left[\left(r + \gamma Q(s', \underline{\arg\max_{a'} Q(s', a'; \mathbf{w})}; \mathbf{w}^-) - Q(s, a; \mathbf{w})\right)^2\right]$$

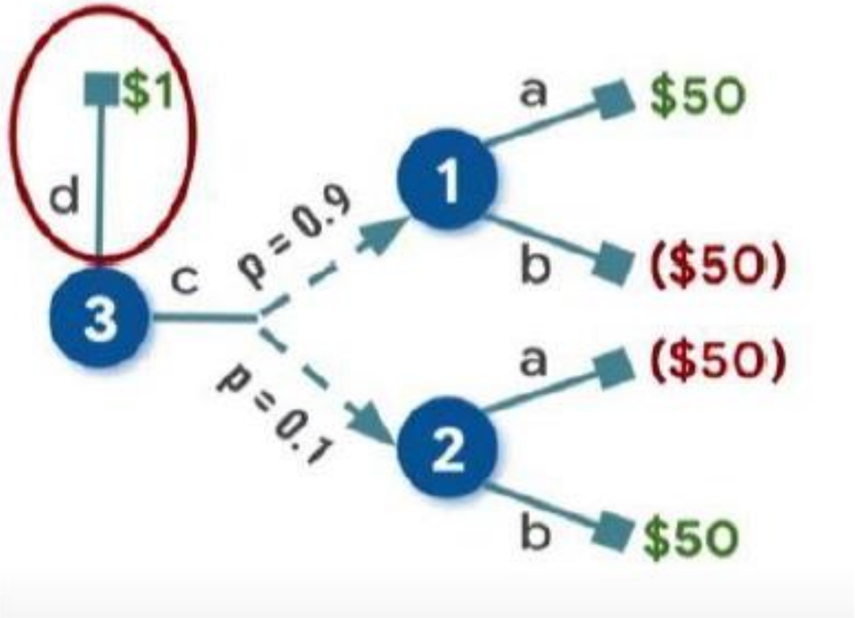
  - Using variant of SGD
  - $\mathbf{w}^- = (1-\tau)\mathbf{w} + \tau\mathbf{w}^-$

Delusional bias

- First appeared in Lu, Schuurmans and Boutilier (NeurIPS 2018)
- Further discussed in Su et. al (ICML 2020)
- “Delusional bias occurs whenever a backed-up value estimate is derived from action choices that are not realizable in the underlying policy class.”
- Greed policy limitations

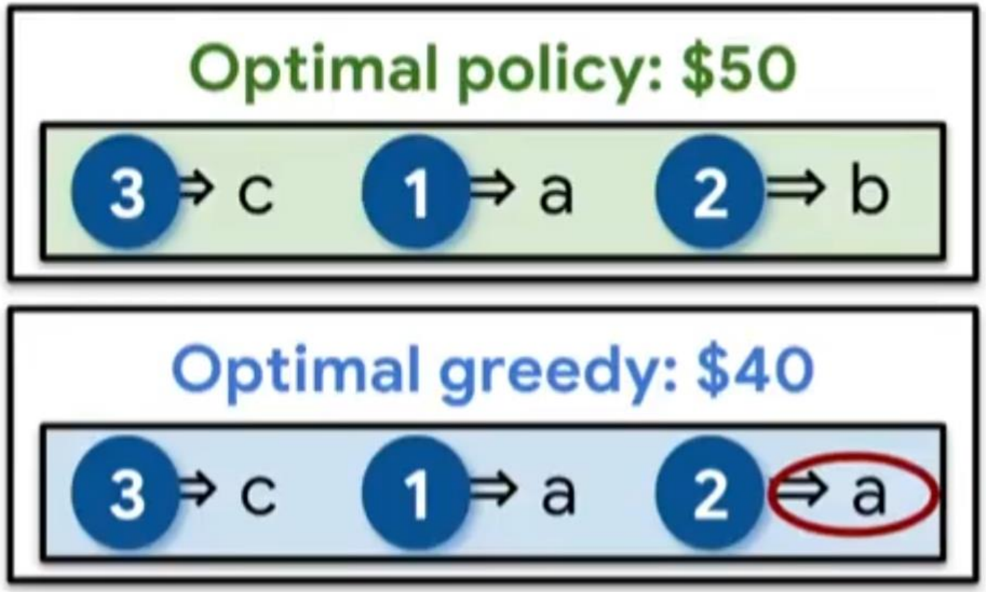
Example

- 3 states
- State 1 and 2: agents can perform two actions: a, b
- State 3: two actions: c, d
- Optimal policy (from human perspective): (3, c) -> (1, a), (2, b)

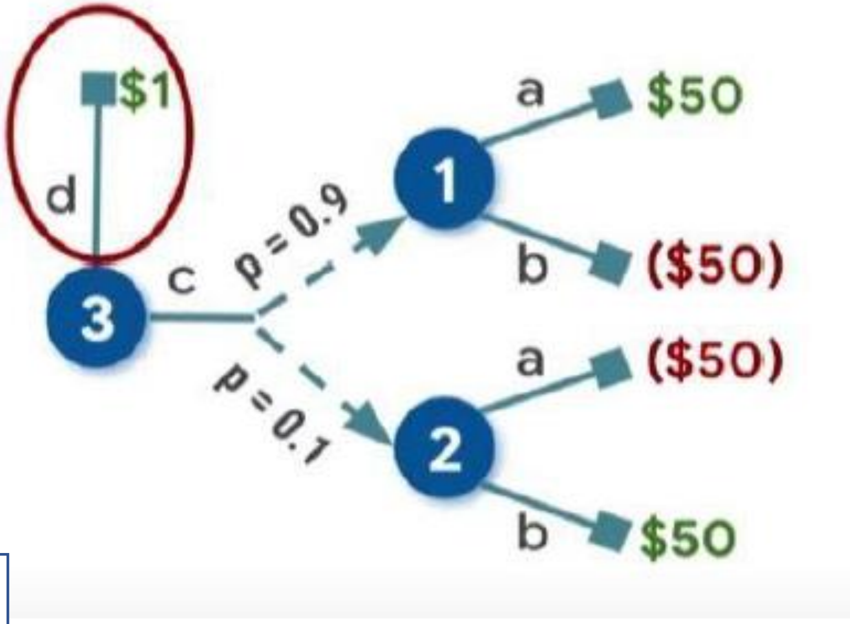


Example

- No greedy algorithm can figure out the second step optimal policy.

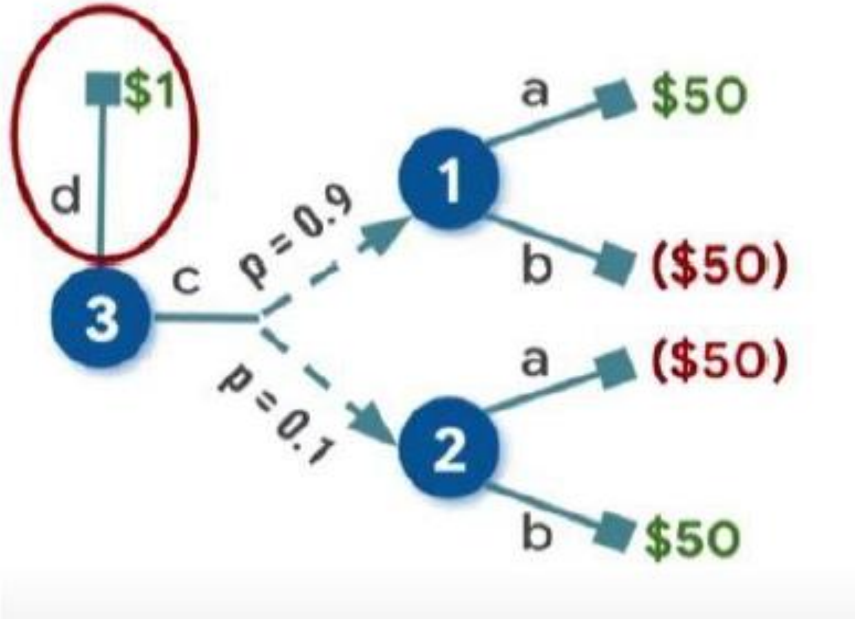


greedy algorithm



Q-learning problem

- If increase the value of a in state 1, we also increase the value of a in state 2, and we push down b value in both states.
- Likewise, if we increase value of b in state 1, we also increase the value of b in state 2, and we push down a value in both states.

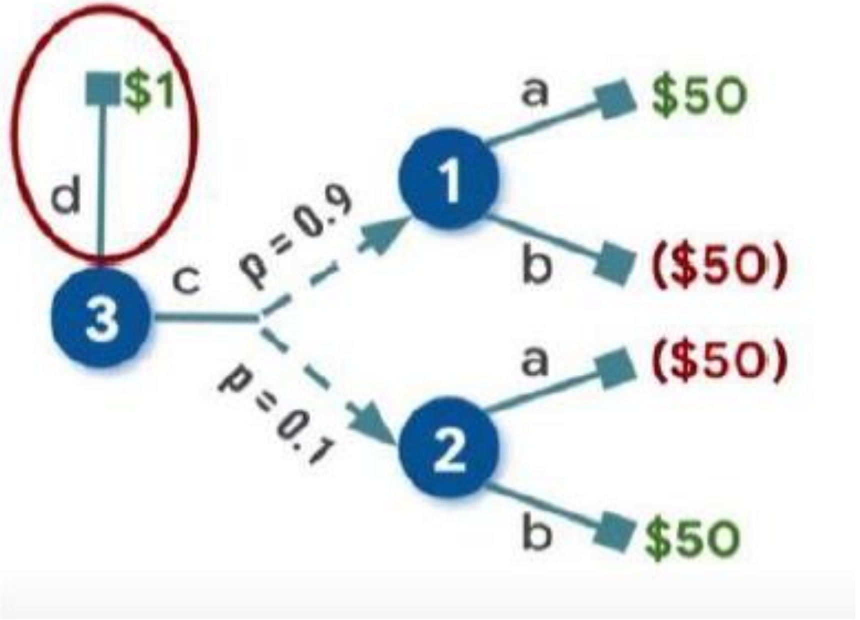


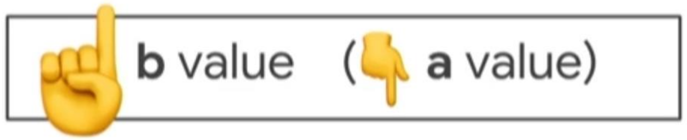
☝️ a value (👇 b value)

☝️ b value (👇 a value)

Q-learning problem

- Ultimately, upon convergence, $Q(1,a)=Q(1,b)=Q(2,a)=Q(2,b) = 0$
- Ended up taking action d in state 3.
- Q-learning averages values that are not jointly realizable
  - Averages values of a and b
- Converges to a poor policy, despite higher-valued policy available in representable greedy class.





Sources

- Delusional bias
  - https://www.youtube.com/watch?v=PSfJ44C3-sU

RL lectures

Date & Time	Course	Teacher
2020-07-27 19:00-19:10	Openning and Introduction	汪军
2020-07-27 19:10-20:50	Value-based Reinforcement Learning	卢宗青
2020-07-28 19:00-20:40	Policy-based RL and RL Theory	汪军
2020-07-29 19:00-20:40	Optimisation in Learning	Haitham
2020-07-30 19:00-20:40	Model-based Reinforcement Learning	张伟楠
2020-07-31 19:00-20:40	Control as Inference	朱占星
2020-08-01 19:00-20:40	Imitation Learning	俞扬
2020-08-03 19:00-20:40	Hierarchical Reinforcement Learning	郝建业
2020-08-04 19:00-20:40	Game Theory Basic	张海峰
2020-08-05 19:00-20:40	Multi-agent Systems	安波
2020-08-06 19:00-20:40	Deep Multi-agent Learning	张崇洁
2020-08-07 19:00-20:40	Advances in Multi-agent Learning	杨耀东
2020-08-08 19:00-20:40	Mean-field Games and Controls	徐任远
2020-08-08 20:40-21:10	Panel Discussion	全体导师