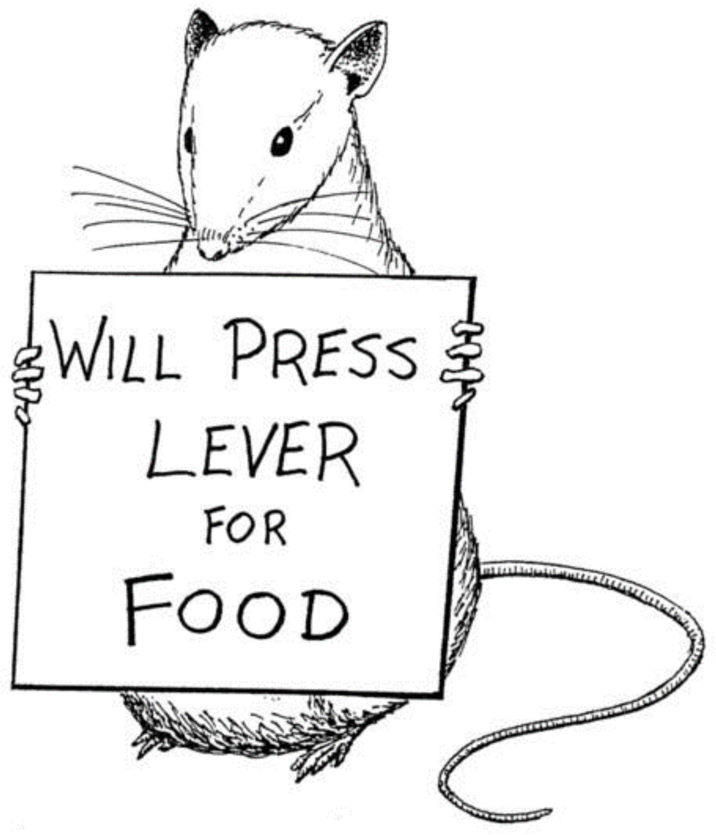
University of California, Irvine

CS 277: Control and Reinforcement Learning

Winter 2024

Lecture 20: Open Questions

Roy Fox
Department of Computer Science
School of Information and Computer Sciences
University of California, Irvine



Logistics

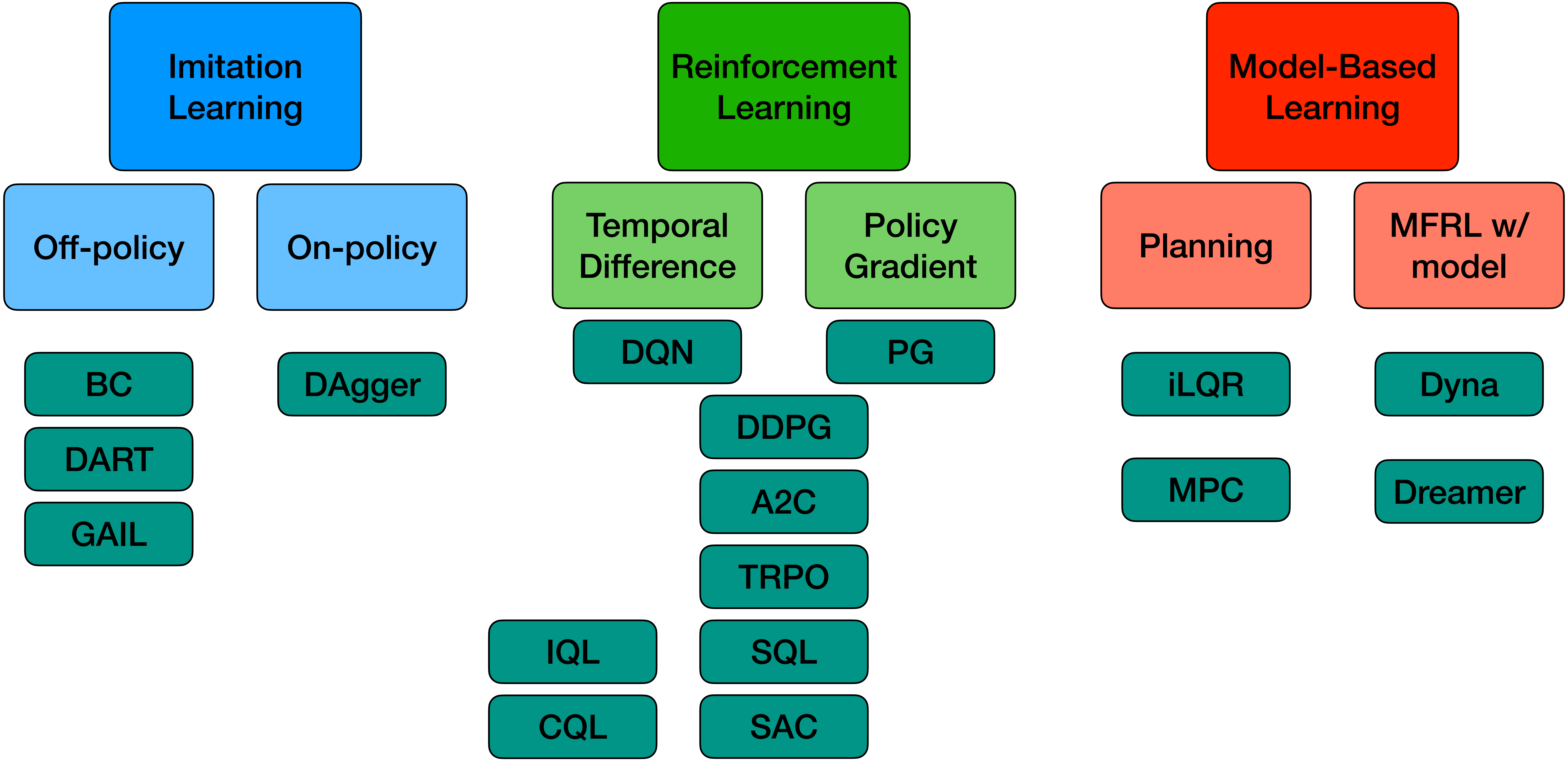
evaluations

- Course evaluations due end of the week

assignments

- Exercise 5 due next Monday
- Exercises 1–3 graded

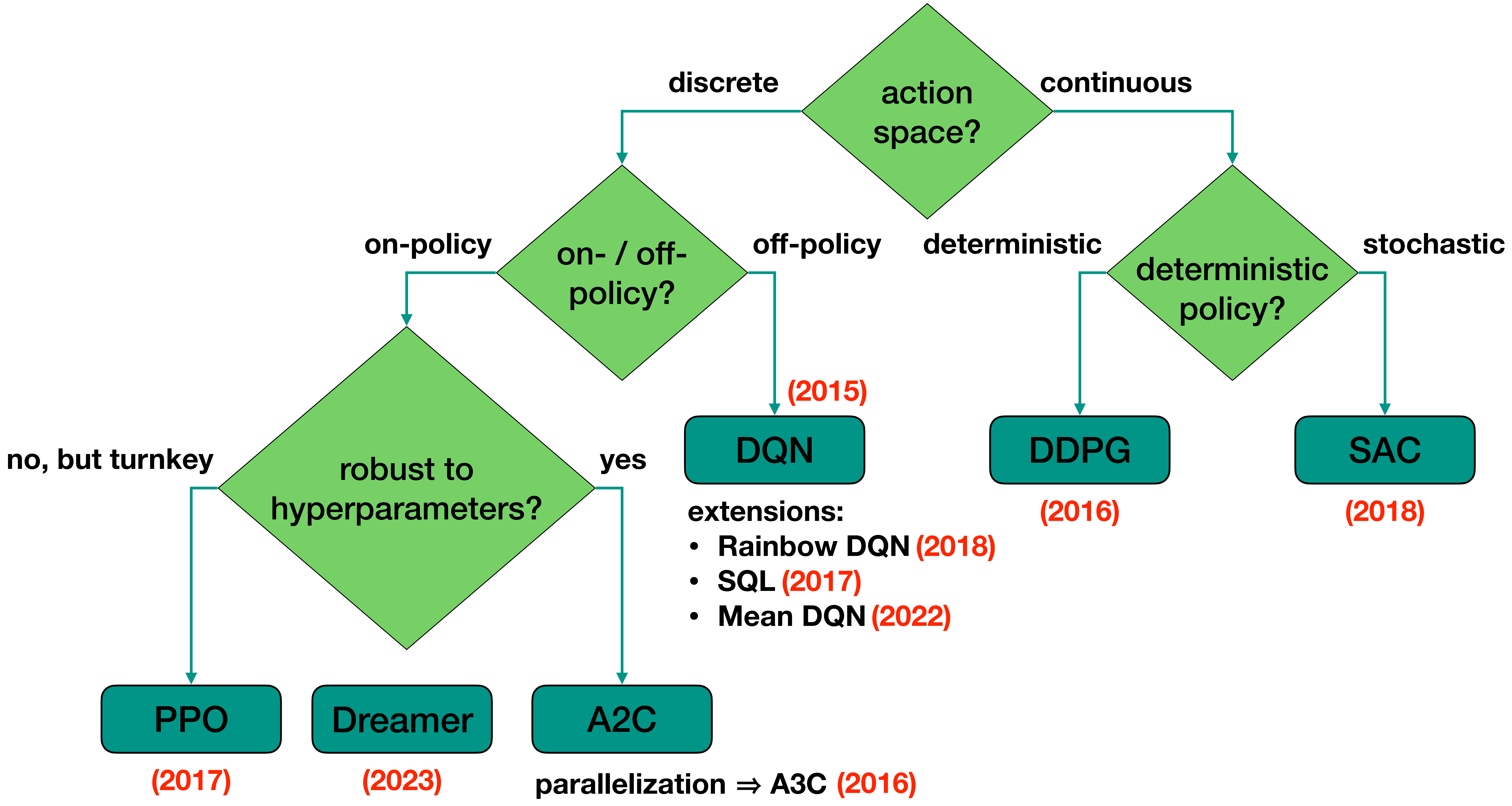
Taxonomy


Imitation Learning
Reinforcement Learning
Model-Based Learning
Off-policy
On-policy
Temporal Difference
Policy Gradient
Planning
MFRL w/ model
BC
DART
GAIL
DAgger
DQN
PG
DDPG
A2C
TRPO
IQL
SQL
CQL
SAC
iLQR
MPC
Dyna
Dreamer

Why so many algorithms?

- We may have different modeling assumptions
  - Is the environment stochastic or deterministic?
  - Is the state / action space continuous or discrete?
  - Is the horizon episodic or infinite?
- We may care about different tradeoffs
  - Sample efficiency? Computational efficiency while learning / executing? Succinct representation?
  - Algorithmic stability, reproducibility, ease of use (existing code), ease of adaptation
- Different difficulty to represent or learn in different domains
  - Represent / learn a policy or a model?
  - Discover structure? Memory? Transfer / share with other tasks?

Flowchart: which algorithm to choose?



On- or off-policy data?

- The faster our simulator ⇒ the faster we can refresh our data
  - And still keep sufficient diversity for training
- Fast enough ⇒ can use on-policy data
  - No need for replay buffer
  - No train→test distributional mismatch (= covariate shift)
  - Can still use off-policy algorithms with on-policy data
- Extremely slow simulator ⇒ not even off-policy, just offline RL

Topics we covered

- Imitation learning
- Policy evaluation + improvement
  - Monte-Carlo vs. Temporal Difference
  - On- vs. off-policy
- Policy Gradient
  - Advantage estimation, Actor–Critic
- Exploration
- Optimal control
- Planning, model-based learning
- Partial observability
- Inverse RL
- Bounded RL
- Structured control
- Offline RL
- Multi-task learning
- Multi-agent learning

Topics we didn't cover

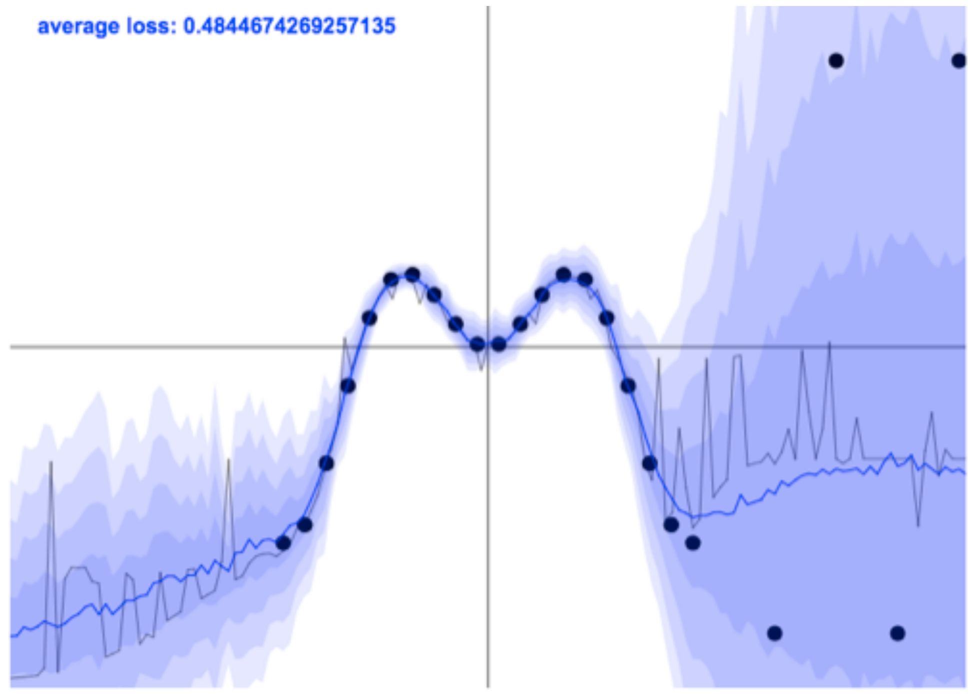
- Hindsight Experience Replay (HER)
- Eligibility traces
- Generalized Value Functions (GVF)
  - Successor representation
- Value Iteration / Prediction Nets (VIN / VPN)
- Natural policy gradient
  - Mirror descent
- Distributional RL
- Bayesian RL
- Hyperparameter tuning
- Distributed RL
- Robot learning
- Safety
- Curiosity / diversity / empowerment
- Preference elicitation
- 3rd-person imitation
- Meta-learning
- Lifelong learning

Trends and open questions in ML

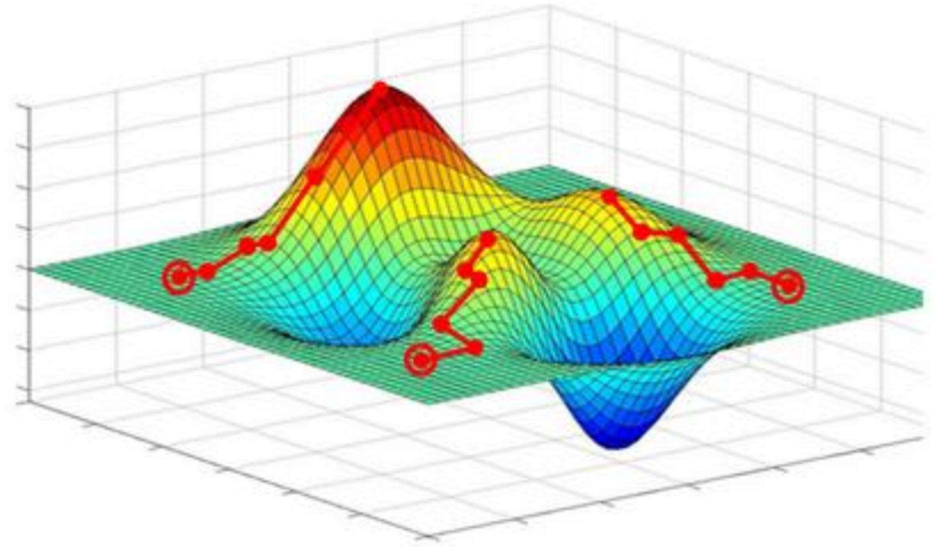
- Bayesian Deep Learning
- Optimization theory
- Neuro-symbolic AI
- Meta-learning / learning to learn
- Lifelong learning
- Causality
- Interpretability, explainability
- AI ethics: fairness, debiasing, alignment

Bayesian RL

- Two kinds of uncertainty
  - Aleatoric = things I haven't seen / haven't happened yet: $p(s_t \mid m_t)$, $p(r_{t+k} \mid m_t)$, ...
  - Epistemic (= model uncertainty) = things I haven't modeled / learned yet: $\hat{p}$, π_θ, Q_ϕ, ...
- Standard RL already considers aleatoric uncertainty
  - "Overtake truck quickly, to reduce time with partial observability, probability of crash"
- Bayesian RL can estimate epistemic uncertainty: $p(\theta \mid \mathscr{D})$
  - Can help improve exploration (cf. Thompson sampling)
  - Can improve learning in bounded agents (uncertain Q $\Rightarrow$ winner's curse)



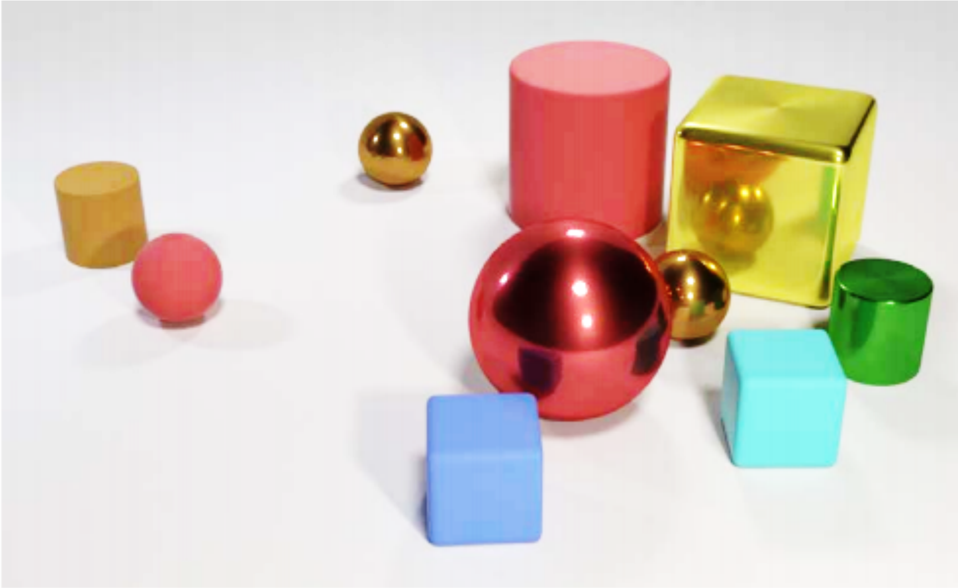
Optimization ⇔ RL



- Special considerations of optimization → RL:
  - Covariate shift
  - Temporal-Difference ⇒ non-stationary loss landscape
  - Saddle points in multi-agent RL
- RL → optimization: iterative optimization is a dynamical process
  - Gradient descent = maximize “reward” of descending loss landscape
  - Optimal control concepts (e.g. Langevin dynamics) key in analysis

Neuro-symbolic RL

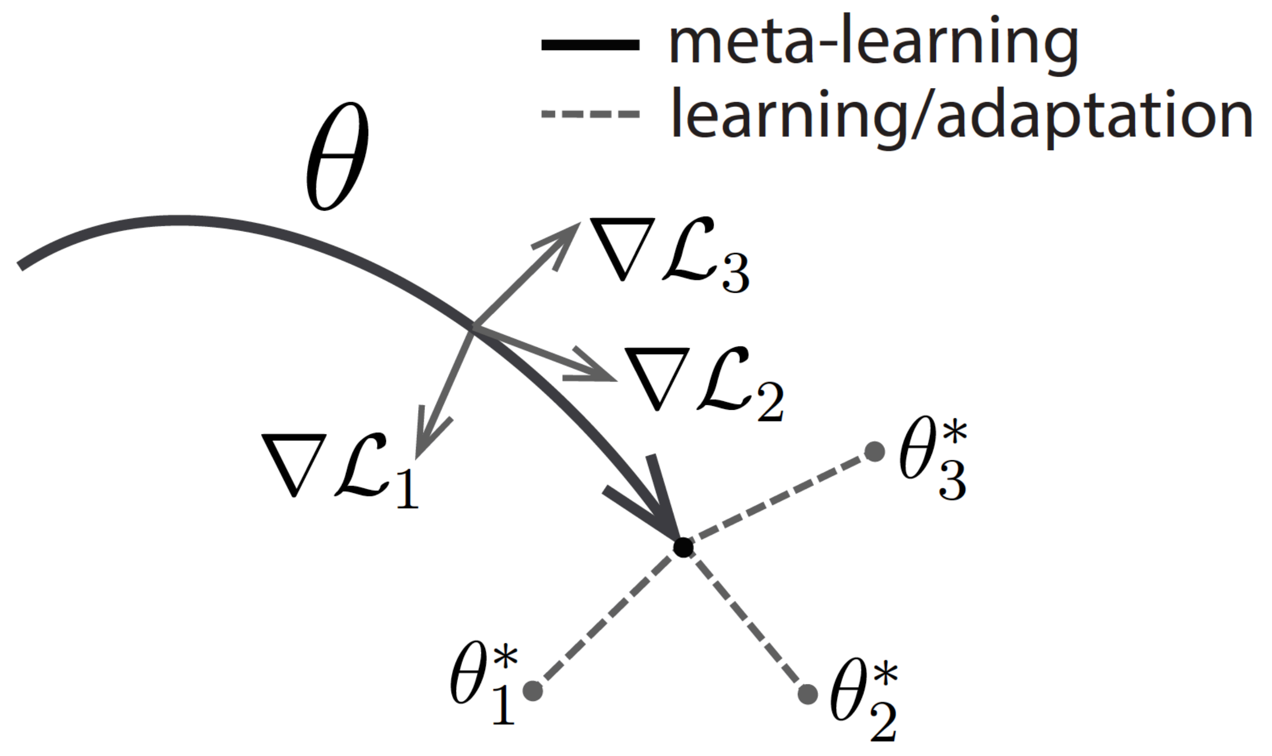
- Is there any benefit to discrete components in gradient-based methods?
  - E.g. modularity



- Structured control = discrete memory components
  - Can help sample efficiency, generalization, transfer, interpretability, ...
- How to learn under given structure?
- How to discover optimal structure?

Meta-learning ⇔ RL

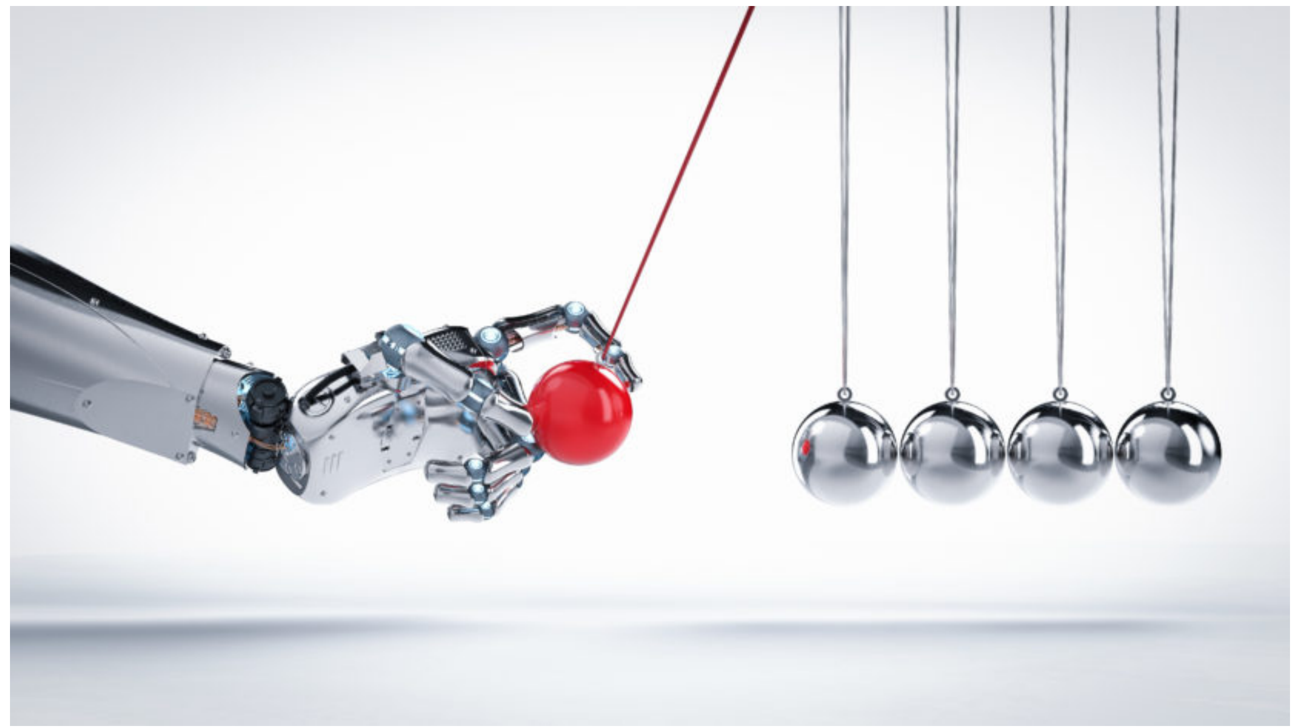
- Multi-task learning = transfer / share learning products between tasks
  - E.g. features, models, policies, skills
- Meta-learning = transfer / share learning of learner components
  - Network architecture = Neural Architecture Search (NAS)
  - Optimizer hyperparameters
  - Parameter initializations (MAML)

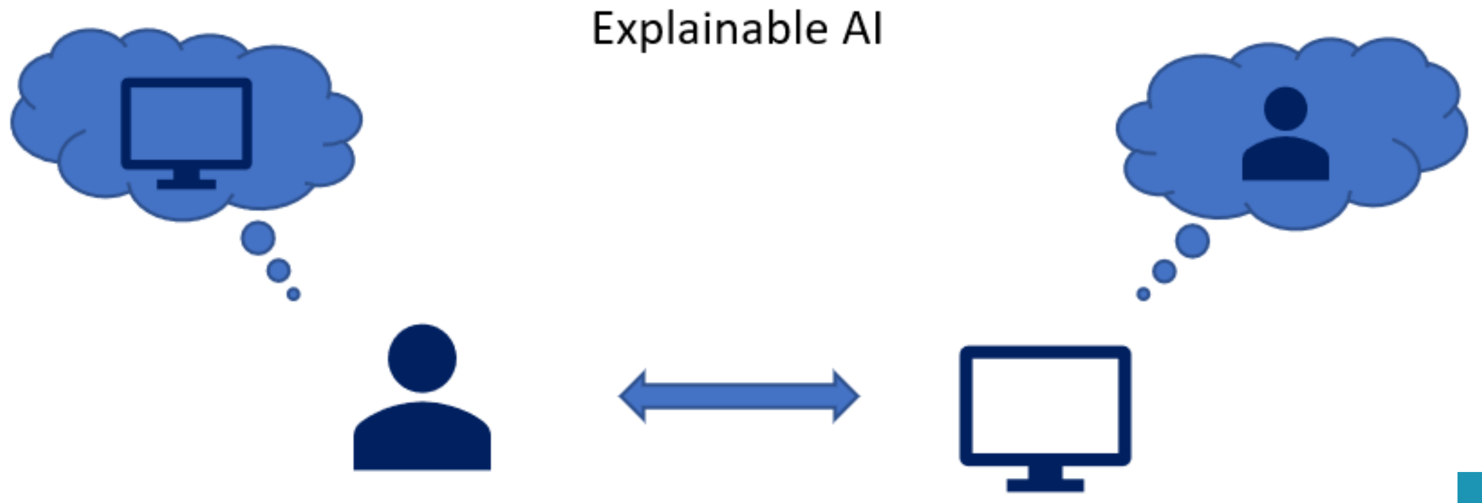


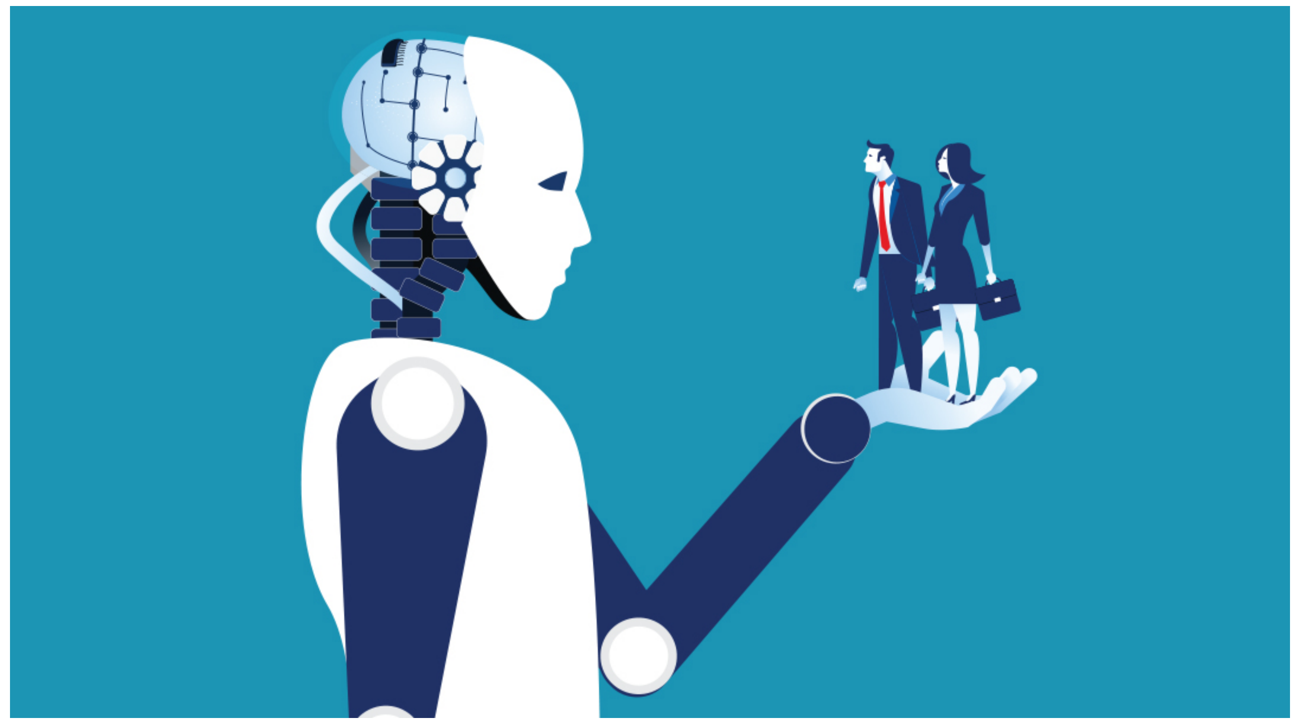
- Learning to perform sequence of tasks = sequential decision making
  - E.g. can use RNNs

Trends and open questions in ML

- Bayesian Deep Learning
- Optimization theory
- Neuro-symbolic AI
- Meta-learning / learning to learn
- Lifelong learning
- Causality
- Interpretability, explainability
- AI ethics: fairness, debiasing, alignment

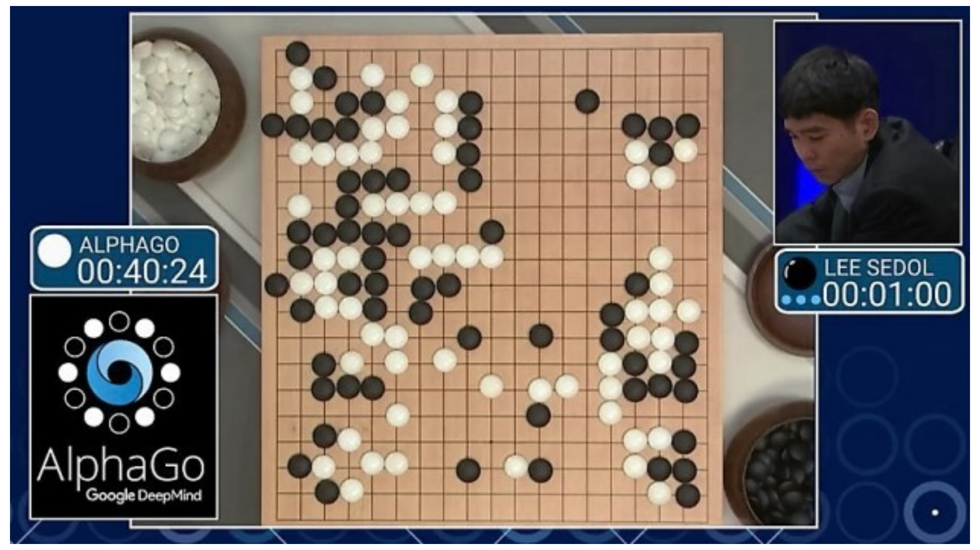


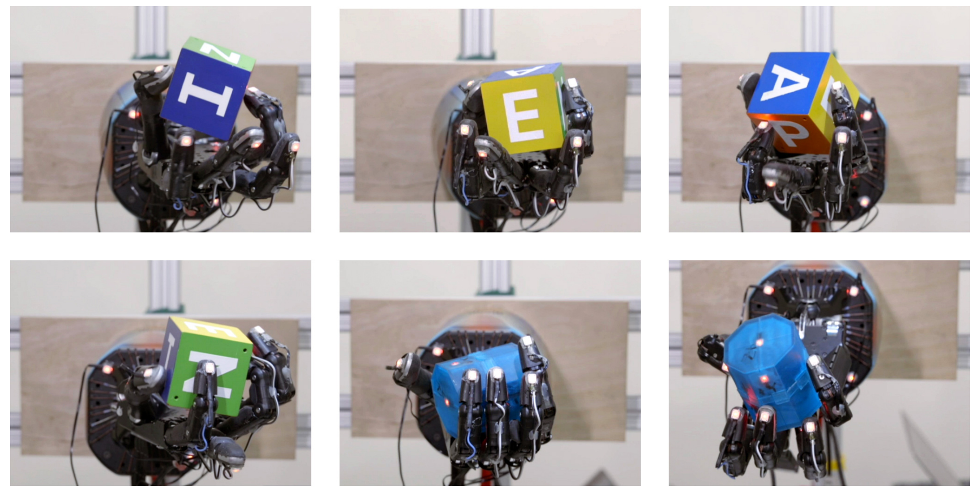




Reproducibility crisis

- Reinforcement learning has seen immense success
  - But remains largely irreproducible, hard to deploy
- Many algorithms are very sensitive to hyperparameters
- Very sensitive to parameter initialization
  - Need to evaluate over many runs, prone to cherry-picking
- Small implementation details may have unexpected effects
- How to go beyond this pre-paradigmatic phase?
  - Better RL theory
  - Build practical RL (and ML) as experimental field

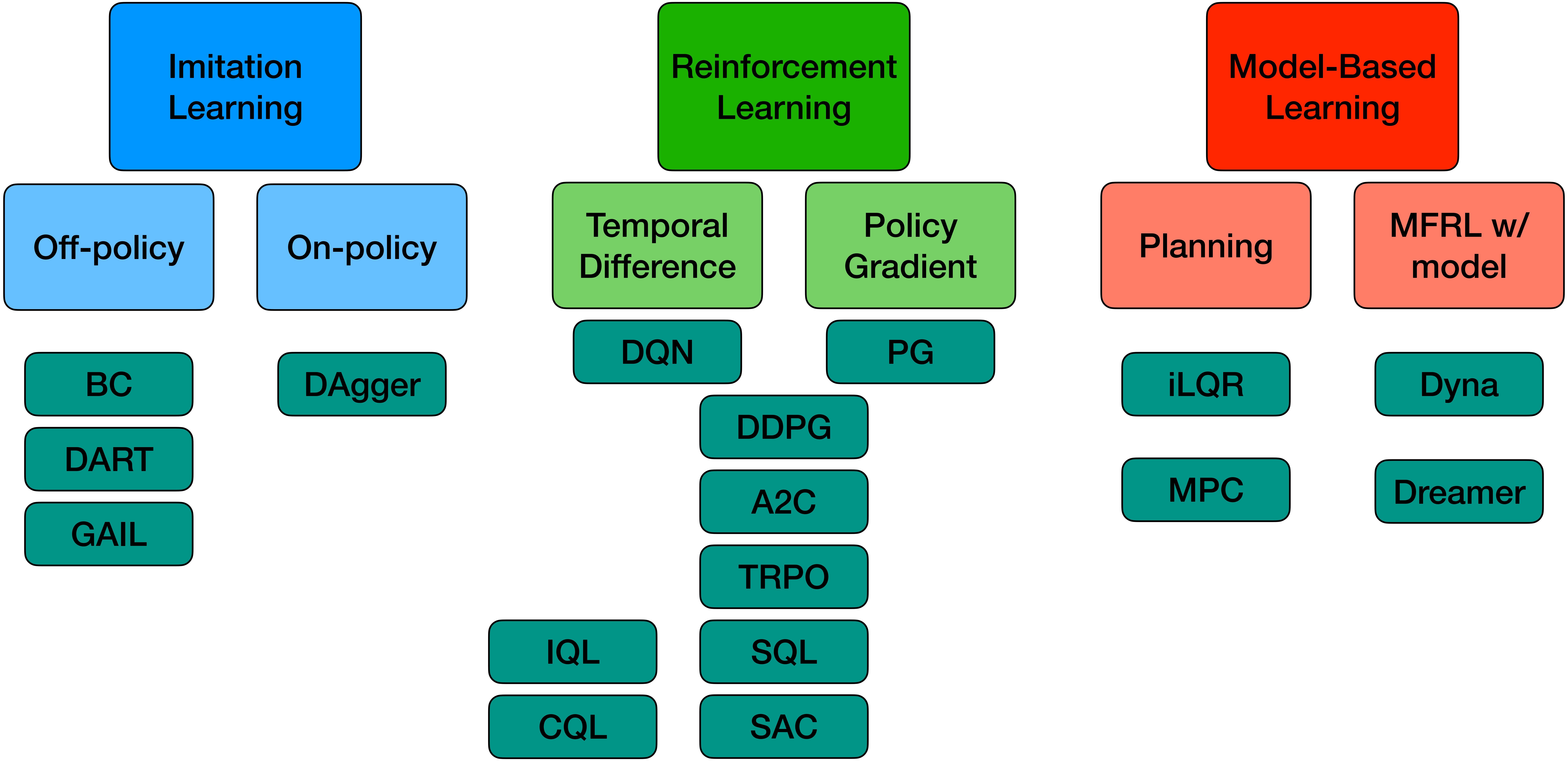




Other open questions

- Imitation learning / inverse RL
  - How to discover structure / memory features in teacher demonstrations?
- Bounded RL
  - How much “bounded” should the agent be?
  - How to anneal this coefficient?
- Structured control
  - Which structures are useful for (multi-task) control?
  - Which structures can we discover?
- Multi-task learning
  - How to discover which tasks are related / unrelated?

Questions?


Imitation Learning
Off-policy
On-policy
BC
DART
GAIL
DAgger
Reinforcement Learning
Temporal Difference
Policy Gradient
DQN
PG
DDPG
A2C
TRPO
IQL
SQL
CQL
SAC
Model-Based Learning
Planning
MFRL w/ model
iLQR
MPC
Dyna
Dreamer