
DEEP REINFORCEMENT LEARNING

HOMEWORK 7 | IMITATION RL, OFFLINE RL

Professor Mohammad Hossein Rohban

Sharif University of Technology
Spring 2026



Contents

1	Conservative Offline RL	1
2	Dagger	3
2.1	Theoretical Analysis	3
2.2	MiniGrid Implementation and Empirical Analysis	4
3	Generative Adversarial Imitation Learning	7
3.1	Theoretical Analysis	7
3.2	Implementation and Empirical Analysis	9
4	Offline DDPG with Behavior Cloning	12

Important Notes

- **Grading Policy:** This homework is considered complete once you reach a total of **100 points**. Up to *5 additional bonus points* may be earned; however, any score above 100 will still be recorded as 100. Bonus points do *not* transfer or overflow to other assignments.
- **Homework Deadline:** The assignment is due on the posted deadline. Please make sure to submit your work on time.
- **Delay Usage Policy:** You may use *at most 5 days* of your allowed late-day budget for this homework. Late days will be automatically applied if you submit after the deadline.
- **Cheating and Collaboration Policy:** You must write all solutions in your own words and produce all code independently. Discussing general ideas is allowed, but sharing written solutions, code, or using someone else's work (including from online sources, previous semesters, or AI tools without proper authorization) is strictly prohibited. Violations will result in academic misconduct procedures.

Advice

Please avoid asking LLMs for the answer. Sure, doing so might give you an immediate reward of 5 points, but in the long run, it won't help you become the next Rockstar of DRL. Remember the key lesson of this course: chasing immediate rewards is rarely the best strategy. Instead, struggle with the problems, read the papers, and truly understand the algorithms. The satisfaction of solving a challenging problem yourself is far greater than any grade. Good luck! [1]

Grading Breakdown

Task / Criteria	Points
Conservative Offline RL	18 pts
CQL Pessimism	6 pts
Conservative Duality	6 pts
Coverage Guarantees	6 pts
Dagger	38 pts
Compounding Error	5 pts
No-Regret DAgger	4 pts
Budgeted DAgger	4 pts
BFS Expert Oracle and Demonstration Dataset	5 pts
Behavior Cloning and Closed-Loop Evaluation	5 pts
DAgger and Budgeted Critical-State Queries	5 pts
Behavior Cloning, Distribution Shift, and DAgger	5 pts
Critical-State DAgger and Expert-Label Efficiency	5 pts
Generative Adversarial Imitation Learning	44 pts
Occupancy Duality	6 pts
Discriminator Geometry	5 pts
Adversarial Policy Updates	5 pts
PPO Backbone and Expert Demonstrations	6 pts
Discriminator and Stable Surrogate Rewards	6 pts
End-to-End GAIL and Empirical Diagnostics	6 pts
From Occupancy Matching to GAIL	5 pts
Reward Bias and Adversarial Training Dynamics	5 pts
Offline DDPG with Behavior Cloning	25 pts
Target Networks and Critic Learning	6 pts
Adaptive Behavior-Regularized Actor	5 pts
Controlled Offline-RL Ablation	5 pts
Extrapolation Error and Behavior Regularization	5 pts
Ablations, Multi-Seed Evidence, and Alpha Sensitivity	4 pts
Bonus	+5 pts
Clarity and Quality of Report and Codes	+5 pts
Total	125+5 pts

1 Conservative Offline RL

Work in an exact tabular discounted MDP with behavior policy π_β and $d^\beta(s) > 0$. For target policy π , let

$$(\mathcal{B}^\pi Q)(s, a) = r(s, a) + \gamma \mathbb{E}_{s', a' \sim P, \pi} [Q(s', a')].$$

Algorithm. Offline RL must improve a policy from a fixed dataset, so erroneous values for out-of-distribution actions cannot be corrected by new interaction. Conservative Q-learning (CQL) modifies temporal-difference learning by lowering the values of actions that the critic prefers but the dataset rarely contains. A common critic objective is

$$L_Q = \frac{1}{2} \mathbb{E}_{(s, a, r, s') \sim D} [Q(s, a) - y]^2 + \alpha \mathbb{E}_{s \sim D} \left[\log \sum_a e^{Q(s, a)} - \mathbb{E}_{a \sim \pi_\beta(\cdot | s)} Q(s, a) \right],$$

with the target

$$y = r + \gamma \mathbb{E}_{a' \sim \pi(\cdot | s')} \bar{Q}(s', a').$$

The first term is ordinary Bellman regression. The second creates a soft adversarial distribution concentrated on high-value actions, then pushes those values below the values of dataset actions. In the tabular limit, the update has an explicit downward correction determined by the density ratio between the adversarial action law and π_β .

Training alternates conservative critic updates, policy improvement against the conservative critic, and slow target-network updates. The coefficient α may be fixed or learned as a dual variable enforcing a desired conservative gap. The guarantee is relative to coverage: if the learned policy places mass on state-action pairs absent from the dataset, their values are not statistically identifiable without additional assumptions.

Question 1: CQL Pessimism

[6 Points]

Consider

$$\arg \min_Q \frac{1}{2} \mathbb{E}_{d^\beta \pi_\beta} [(Q - \mathcal{B}^\pi \hat{Q}^k)^2] + \alpha \mathbb{E}_{d^\beta} \left[\sum_a (\mu(a | s) - \pi_\beta(a | s)) Q(s, a) \right].$$

For parts 2-3, assume $\pi(\cdot | s) \ll \pi_\beta(\cdot | s)$ at every state; otherwise the corresponding density ratio and χ^2 divergence are infinite, and the coordinate calculations are restricted to the behavior-policy support.

1. Derive the exact coordinatewise minimizer and its Hessian. State necessary and sufficient support conditions for uniqueness.
2. Set $\mu = \pi$ and show that the policy-averaged correction equals $-\alpha \chi^2(\pi(\cdot | s) || \pi_\beta(\cdot | s))$. Derive the fixed point as a resolvent applied to this divergence penalty.
3. Prove state-value pessimism and obtain a sharp sup-norm lower-bias bound. Construct a counterexample showing that policy-averaged pessimism does not imply pointwise $\hat{Q}(s, a) \leq Q^\pi(s, a)$.

Question 2: Conservative Duality

[6 Points]

1. For $\tau > 0$, solve

$$\max_{\mu \in \Delta(\mathcal{A})} \{ \langle \mu, Q \rangle + \tau H(\mu) \}$$

- and the reference-policy variant with $-\tau \mathbf{D}_{\text{KL}}(\mu \parallel \rho)$. For the latter, assume $\rho(a) > 0$ for every action, or equivalently restrict μ to $\text{supp}(\rho)$. Derive both optimizers and optimal values from Fenchel conjugacy.
2. Formulate CQL as the Lagrangian relaxation of a constraint requiring a minimum value gap between behavior and adversarial action distributions. Derive KKT conditions and interpret α as a dual variable.

Question 3: Coverage Guarantees**[6 Points]**

1. For an initial distribution μ_0 , define the normalized discounted state occupancy

$$d^\pi(s) = (1 - \gamma) \sum_{t \geq 0} \gamma^t \Pr_{\mu_0, \pi}(s_t = s),$$

let d^β denote the same occupancy for the behavior policy, define the dataset distribution $\nu(s, a) = d^\beta(s) \pi_\beta(a \mid s)$, and use

$$\|g\|_{2, \nu} = (\mathbb{E}_{(s, a) \sim \nu} [g(s, a)^2])^{1/2}.$$

Define

$$C_\pi = \left\| \frac{d^\pi(s) \pi(a \mid s)}{\nu(s, a)} \right\|_\infty.$$

where the ratio is interpreted ν -almost surely. Assume $d^\pi \pi \ll \nu$ and $C_\pi < \infty$. If an estimate $\hat{Q}$ satisfies

$$\|\mathcal{B}^\pi \hat{Q} - \hat{Q}\|_{2, \nu} \leq \epsilon,$$

define

$$J(\pi) = \mathbb{E}_{s_0 \sim \mu_0, a_0 \sim \pi(\cdot \mid s_0)} [Q^\pi(s_0, a_0)], \quad \hat{J}(\pi) = \mathbb{E}_{s_0 \sim \mu_0, a_0 \sim \pi(\cdot \mid s_0)} [\hat{Q}(s_0, a_0)].$$

Prove

$$|J(\pi) - \hat{J}(\pi)| \leq \frac{\sqrt{C_\pi}}{1 - \gamma} \epsilon.$$

2. Construct two MDPs that induce identical offline datasets but assign different values to an unsupported action. Use the construction to prove an information-theoretic impossibility result without coverage.
3. Let Π_{cov} be a covered comparator class and $\Pi_{\text{alg}} \subseteq \Pi_{\text{cov}}$ the policy class searched by the algorithm. Let

$$\pi_{\text{cov}}^* \in \arg \max_{\pi \in \Pi_{\text{cov}}} J(\pi), \quad \varepsilon_{\text{app}} = J(\pi_{\text{cov}}^*) - \sup_{\pi \in \Pi_{\text{alg}}} J(\pi).$$

Suppose a pessimistic score $\underline{J}$ satisfies, uniformly over Π_{alg} ,

$$0 \leq J(\pi) - \underline{J}(\pi) \leq \varepsilon_{\text{est}},$$

and the returned policy $\hat{\pi}$ satisfies

$$\underline{J}(\hat{\pi}) \geq \sup_{\pi \in \Pi_{\text{alg}}} \underline{J}(\pi) - \varepsilon_{\text{opt}}.$$

Prove the suboptimality bound

$$J(\pi_{\text{cov}}^*) - J(\hat{\pi}) \leq \varepsilon_{\text{app}} + \varepsilon_{\text{est}} + \varepsilon_{\text{opt}}.$$

2 DAgger

2.1 Theoretical Analysis

Consider a finite-horizon sequential decision problem with horizon T , a deterministic expert policy π_E , and learner state law d_π^t , where d_π^t denotes the distribution of the state immediately before the action at time t when the complete trajectory is generated by π . The learner's self-induced imitation error is

$$M(\pi) = \sum_{t=1}^T \mathbb{E}_{s \sim d_\pi^t, a \sim \pi(\cdot|s)} [\mathbf{1}\{a \neq \pi_E(s)\}].$$

Thus, $M(\pi)$ is the expected total number of disagreements with the expert along trajectories induced by the learner.

Unless otherwise stated, the expert-distribution classification error ε is the time-averaged error

$$\varepsilon = \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{s \sim d_{\pi_E}^t, a \sim \pi(\cdot|s)} [\mathbf{1}\{a \neq \pi_E(s)\}].$$

The per-time error is the corresponding expectation at a particular time t . In the coupling argument, the expert and learner trajectories use the same initial state and transition randomness and therefore remain identical until their first action disagreement.

DAGger starts from an expert dataset D_1 . At iteration i , the current policy π_i is rolled out, the expert labels the visited states, the new state-action pairs are added to the aggregate dataset, and the supervised learner is updated to obtain π_{i+1} . Repeated visits are retained as separate examples unless deduplication is explicitly introduced.

For the no-regret analysis, use the following order of events: π_i is selected using information available before iteration i , π_i is rolled out, the resulting state distribution defines ℓ_i , and the loss is then revealed to the learner. Consequently, the assumed regret guarantee must apply to the realized adaptively generated loss sequence. A uniformly random iterate is selected independently after all iterations have been completed.

For budgeted DAGger, let each learner-visited candidate retain its rollout and time-step index. Given the candidate states, a state s is queried with probability $q_i(s)$, where $0 < q_i(s) \leq 1$. Query indicators are conditionally independent unless another sampling design is explicitly stated. Conditional variance refers to variance over these query indicators while holding the candidate states and any sampled action or gradient contributions fixed. The “fully labeled candidate-batch loss” is the empirical loss obtained by labeling every candidate with the same time and rollout averaging as in ℓ_i ; its expectation over rollout sampling equals ℓ_i .

Question 1: Compounding Error

[5 Points]

1. Starting only from an expert-distribution classification error bound ε , prove the sharp worst-case bound

$$M(\pi) \leq \min\{T, T^2\varepsilon\}$$

by coupling expert and learner trajectories up to their first disagreement.

2. Construct a family of finite-horizon MDPs attaining $\Omega(\min\{T, T^2\varepsilon\})$ up to universal constants. The construction must specify dynamics, expert, learner, and per-time error under the expert distribution.

Question 2: No-Regret DAgger**[4 Points]**

At iteration i , define

$$\ell_i(\pi) = \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{s \sim d_{\pi_i}^t, a \sim \pi(\cdot|s)} [\mathbf{1}\{a \neq \pi_E(s)\}].$$

1. From an average-regret guarantee, prove a self-induced loss bound for a uniformly random iterate and show the existence of a deterministic iterate with the same guarantee. Make explicit why the losses are chosen before or after the online learner's decision.
2. Explain why the theorem controls an average or best iterate but not the final iterate. Give a sufficient set of assumptions for a last-iterate guarantee, such as a contraction of the policy-update map, or strong convexity together with a convergent stable update and summable loss variation.

Question 3: Budgeted DAgger**[4 Points]**

1. Formalize selective expert querying by a state-dependent inclusion probability $q_i(s)$. Construct an importance-weighted empirical loss that is conditionally unbiased for the fully labeled candidate-batch loss, show that its expectation over rollout sampling equals ℓ_i , and derive its conditional variance.
2. Under a fixed expected label budget, solve for the variance-minimizing query distribution in terms of the per-state gradient norm or loss. Explain why policy entropy is only a proxy for this optimum.
3. Construct a confidently wrong policy for which entropy sampling never queries the states responsible for failure. Compare entropy with action margin, ensemble disagreement, novelty, and estimated cost-to-go as acquisition functions.

2.2 MiniGrid Implementation and Empirical Analysis

The implementation uses MiniGrid-DoorKey-16x16-v0. The agent must find and pick up a key, unlock and open a door, and navigate to the goal. The learner receives only the symbolic image observation, whose channels encode object type, color, and object state, and selects among the useful low-level actions turn left, turn right, move forward, pick up, and toggle. Actions outside this set must not be selected by the learned policy.

The privileged oracle has access to the complete environment state. For a fixed episode map containing the walls and the key, door, and goal locations, its dynamic planning state is

$$z = (x, y, d, k, o, \ell),$$

where (x, y) is the agent position, d is its orientation, k indicates whether it carries the key, and o and ℓ indicate whether the door is open and locked. The tuple z must always be interpreted together with the fixed map for that episode. Breadth-first search operates on this complete symbolic planning state and returns a shortest valid sequence of low-level actions. Whenever a label is requested, the oracle replans from the current environment state and returns the first action of the shortest plan.

Each demonstration sample must align the observation immediately before an action with the oracle action selected from that state. The stored reward and termination information correspond to the transition produced by that action. Where the environment API distinguishes termination from truncation, both flags must be retained. Success means that the agent reaches the goal before the episode terminates or is truncated, episode return is the sum of rewards, and episode length is the number of executed environment actions.

Behavior cloning minimizes

$$L_{\text{BC}}(\theta) = -\mathbb{E}_{(s,a) \sim D_E} [\log \pi_{\theta}(a | s)].$$

Full DAGger queries the oracle for every learner-visited state occurrence, including repeated occurrences. Critical-state DAGger first deduplicates the candidate pool and then queries only the k candidates with the largest policy entropy

$$H(\pi_{\theta}(\cdot | s)) = -\sum_a \pi_{\theta}(a | s) \log \pi_{\theta}(a | s).$$

Entropy must be computed from the policy distribution after restricting and renormalizing it over the useful DoorKey actions. The implementation must state whether deduplication uses exact image observations, complete symbolic states, or their combination, and whether it is performed only within the current iteration or also against previously labeled states. Ties in the top- k ranking must be resolved using a documented deterministic rule or seeded random rule.

One expert label is one oracle action requested for one retained training example. Cumulative label counts include the initial behavior-cloning dataset and all later DAGger queries; the initial and additional counts must also be reported separately. Matched evaluation budgets mean that every compared policy is evaluated with the same number of episodes, the same seeds or seed distribution, and the same episode horizon. Success rate, return, and episode length are averaged over all evaluation episodes. Any curve, heatmap, GIF, or table must identify whether it uses the current iterate or the best checkpoint so far, and the criterion used to choose a best checkpoint must remain fixed across both DAGger variants.

Implementation 1: BFS Expert Oracle and Demonstration Dataset

[5 Points]

Implement the privileged DoorKey planner and use it to construct the expert dataset.

1. Extract walls, key, door, and goal locations and encode the current symbolic state using position, orientation, key possession, and door status.
2. Implement symbolic transitions for turning, moving, picking up the key, and toggling the door. Invalid actions must leave the symbolic state unchanged.
3. Run BFS over symbolic states, retain parent and action pointers, and reconstruct a shortest plan when the goal is reached. Replan from the current environment state to obtain the oracle's next action.
4. Generate demonstrations across controlled seeds and store image observations, expert actions, positions, rewards, termination flags, episode identifiers, returns, lengths, and success indicators.

Verify that the oracle solves every smoke-test instance before generating the full dataset.

Implementation 2: Behavior Cloning and Closed-Loop Evaluation

[5 Points]

Implement the supervised imitation baseline.

1. Convert the symbolic image observation to channel-first floating-point tensors and normalize each semantic channel appropriately.
2. Train a compact convolutional policy with cross-entropy, a reproducible train/validation split, gradient clipping, and restoration of the best validation checkpoint. Mask policy selection to the useful DoorKey actions.
3. Evaluate the deterministic policy in complete environment rollouts. Report validation accuracy, success rate, average return, and average episode length; automatically locate and visualize at least one failure trajectory and plot a state-visitation heatmap.

Keep supervised validation metrics separate from closed-loop task metrics.

Implementation 3: DAGger and Budgeted Critical-State Queries**[5 Points]**

Starting from the BC model, implement two dataset-aggregation methods.

1. For standard DAGger, roll out the learner, query the BFS oracle at every visited state, append the new state-action pairs, retrain on the full aggregate, and retain the best evaluated policy across iterations.
2. For critical-state DAGger, collect learner-visited candidates together with policy entropy, remove duplicate symbolic/observational states, and request labels only for the top- k highest-entropy candidates per iteration.
3. Track cumulative expert labels, success rate, return, and episode length for both methods. Produce a success-versus-label curve, final visitation heatmaps, representative GIF rollouts, and a compact comparison table including BC.

Use matched evaluation budgets and clearly report whether the plotted score is the current iterate or the best checkpoint so far.

Question 4: Behavior Cloning, Distribution Shift, and DAGger**[5 Points]**

Analyze the relationship between supervised prediction and sequential control.

1. Why can a policy achieve high validation accuracy on expert data yet fail frequently in closed-loop DoorKey rollouts? Explain how an early classification mistake changes the future state distribution and causes compounding error.
2. Explain how DAGger changes the training distribution. Why are expert labels on learner-visited states more useful for recovery behavior than simply collecting more trajectories from the expert policy?
3. Interpret the BC failure GIF and the BC/DAGger visitation heatmaps. Identify evidence of off-expert drift, repeated local behavior, or newly learned recovery paths.
4. Discuss why an oracle with privileged full-state access can label a partially observed learner input, and identify a potential ambiguity if identical learner observations require different expert actions in different hidden states.

Ground the discussion in measured success rates and episode lengths.

Question 5: Critical-State DAGger and Expert-Label Efficiency**[5 Points]**

Compare full DAGger with entropy-budgeted querying.

1. Explain why policy entropy is a plausible query score and when it can fail, for example, when a confidently wrong policy assigns low entropy to an unfamiliar state.
2. Using the success-versus-label curve, compare the marginal improvement obtained per additional expert label. Does critical-state DAGger match full DAGger at a smaller label budget, or does selective querying omit important coverage?
3. Explain the role of duplicate removal. What are the trade-offs between deduplicating exact observations, symbolic states, or broader regions of the grid?
4. Propose one alternative acquisition rule, such as ensemble disagreement, action-margin sampling, novelty, or expected recovery value. State what extra model or computation it would require and what failure of entropy sampling it is intended to address.

Support the conclusion with cumulative label counts, final success rates, and the variability visible across evaluation episodes or seeds.

3 Generative Adversarial Imitation Learning

3.1 Theoretical Analysis

Assume a finite discounted MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \mu_0, \gamma)$ with $0 \leq \gamma < 1$, initial-state distribution μ_0 , and stationary stochastic policies. Define the normalized discounted state-action occupancy measure by

$$\rho_\pi(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr_\pi(s_t = s, a_t = a),$$

and let $\rho_E = \rho_{\pi^E}$ denote the expert occupancy. Under this normalization, $\sum_{s,a} \rho_\pi(s, a) = 1$, so expectations with respect to ρ_π are expectations under a probability distribution. Define the discounted return without normalization as

$$J_r(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right],$$

and use

$$\langle \rho, r \rangle = \sum_{s,a} \rho(s, a) r(s, a)$$

for the occupancy-reward pairing.

The occupancy polytope is the set of nonnegative vectors satisfying the normalized Bellman flow equations. When a question refers to a boundary policy, its tangent and normal cones are to be computed at the corresponding occupancy ρ_π in this polytope, using the standard convex-analytic cones in the ambient space $\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$. A recovered stationary policy must also be defined at states whose state occupancy is zero. For the integral-probability-metric results, the Lipschitz class uses a stated ground metric on $\mathcal{S} \times \mathcal{A}$, while the RKHS class uses a stated positive-semidefinite kernel. For the duality argument, take the reward space to be $\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ unless a different topological vector space is explicitly introduced.

GAIL compares expert and policy occupancies using a discriminator

$$D_\omega(s, a) = \sigma(x_\omega(s, a)),$$

where x_ω is the raw discriminator logit. The primary label convention assigns label 1 to expert samples and label 0 to policy samples, with equal class priors:

$$\max_{\omega} \mathbb{E}_{\rho_E} [\log D_\omega(s, a)] + \mathbb{E}_{\rho_\pi} [\log(1 - D_\omega(s, a))].$$

The reverse convention swaps the labels and therefore replaces D by $1 - D$. The pointwise discriminator and Jensen-Shannon calculations use the idealized objective above, without label smoothing or gradient penalties, and use the standard convention $0 \log 0 = 0$. Where both occupancies have zero density, the discriminator value does not affect the objective.

The policy cannot differentiate through sampled environment transitions. Instead, the discriminator logit is converted into one of the following per-step rewards:

$$r_{\text{gail}}(s, a) = \text{softplus}(x_\omega(s, a)) = -\log(1 - D_\omega(s, a)),$$

$$r_{\log D}(s, a) = -\text{softplus}(-x_\omega(s, a)) = \log D_\omega(s, a), \quad r_{\text{logit}}(s, a) = x_\omega(s, a).$$

Here, $x_\omega > 0$ means that the discriminator considers the pair more expert-like. In the episode-duration analysis, rewards are accumulated only until termination and the absorbing state has continuation value zero.

During each policy update, the discriminator parameters and resulting reward function are held fixed. Policy gradients are then estimated from on-policy rollouts, typically using a learned value baseline, GAE, and PPO.

In the two-timescale interpretation, discriminator updates form the faster process and policy updates form the slower process. With an optimal discriminator and normalized occupancies, the ideal adversarial objective is

$$\min_{\pi} 2\text{JS}(\rho_{\pi} \parallel \rho_E) - \lambda H(\pi),$$

up to the additive constant $-\log 4$, where the normalized discounted causal entropy is

$$H(\pi) = \mathbb{E}_{(s,a) \sim \rho_{\pi}} [-\log \pi(a \mid s)].$$

Question 1: Occupancy Duality

[6 Points]

1. Derive the normalized Bellman flow constraints and prove the converse recovery of a stationary policy from any feasible occupancy. Determine the tangent and normal cones of the occupancy polytope at a boundary policy.
2. Prove

$$J_r(\pi) - J_r(\pi_E) = \frac{1}{1-\gamma} \langle \rho_{\pi} - \rho_E, r \rangle$$

and derive tight integral-probability-metric bounds for reward classes given by the ℓ_{∞} ball, an RKHS unit ball, and a Lipschitz ball.

3. Let $\mathcal{D}$ be the normalized occupancy polytope and write $H(\rho)$ for causal entropy in occupancy form. Starting from the regularized apprenticeship-learning saddle problem

$$\sup_{r \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}} \left\{ -\psi(r) - \langle \rho_E, r \rangle + \inf_{\rho \in \mathcal{D}} [\langle \rho, r \rangle - \lambda H(\rho)] \right\},$$

use Fenchel-Rockafellar or an equivalent minimax duality theorem to derive

$$\inf_{\rho \in \mathcal{D}} \{ -\lambda H(\rho) + \psi^*(\rho - \rho_E) \}.$$

State domains, closure assumptions, and a qualification ensuring zero duality gap, and explain the specialization when $\lambda = 0$.

Question 2: Discriminator Geometry

[5 Points]

1. Solve the discriminator optimization pointwise under both label conventions. Substitute the optimum and derive the Jensen-Shannon divergence, including all zero-density boundary cases.
2. For discriminator logit x , compare $\text{softplus}(x)$, $-\text{softplus}(-x)$, and x as policy rewards. Derive their asymptotic gradients and prove how fixed-sign per-step rewards bias episode duration when the absorbing-state value is fixed at zero.

Question 3: Adversarial Policy Updates

[5 Points]

1. Starting from a fixed discriminator reward, derive the policy-gradient theorem in occupancy form. Show that a state-only baseline treated with stop-gradient preserves the expected policy gradient, derive the variance-minimizing state-dependent baseline, and state conditions under which a learned value function approximates it and reduces variance.
2. Derive GAE as an exponentially weighted sum of temporal-difference residuals. Quantify its bias when the critic is imperfect and its variance as a function of λ under a stated covariance model.

3. Model GAIL as a two-timescale stochastic approximation. State step-size and regularity conditions under which the discriminator tracks a local optimum, and explain why aggressively optimizing the discriminator can violate the assumptions needed by the policy update.

3.2 Implementation and Empirical Analysis

The implementation trains an expert and a GAIL policy on CartPole. The CartPole observation is a continuous state vector and its action is discrete. For these experiments, integer actions are converted to one-hot vectors before being concatenated with the state. The discriminator interface must also accept an action that has already been supplied as a continuous encoding.

First train an actor-critic expert using the true environment reward. Expert demonstrations are pre-action state-action pairs collected from deterministic greedy rollouts of the trained expert. Any temporal or random subsampling rule must be reproducible and reported, together with both the number of complete expert episodes and the number of retained state-action pairs.

A fixed-length rollout may contain complete episodes and an unfinished final episode. Store both the observation before each action and the next observation produced by that action. Let b_t be the value-bootstrap mask and c_t the GAE-continuation mask:

$$b_t = \begin{cases} 0, & \text{if the transition is a true termination,} \\ 1, & \text{if it is a time-limit truncation or nonterminal transition,} \end{cases}$$

and

$$c_t = \begin{cases} 0, & \text{if an episode terminates or is truncated after time } t, \\ 1, & \text{otherwise.} \end{cases}$$

The temporal-difference residual and backward GAE recursion are therefore

$$\delta_t = r_t + \gamma b_t V(s_{t+1}) - V(s_t), \quad \hat{A}_t = \delta_t + \gamma \lambda c_t \hat{A}_{t+1}.$$

At an unfinished rollout tail, bootstrap from the final observation when it is nonterminal, but end the recursion at the end of the stored buffer. At a time-limit truncation, also bootstrap from the final observation while preventing the recursion from crossing into the reset episode.

Each GAIL iteration uses the following fixed order. Collect a fresh on-policy rollout and retain its old action log-probabilities and value predictions. Update the discriminator using equally weighted expert and policy minibatches. One-sided label smoothing means that the expert target is $1 - \alpha$ for a stated $\alpha \in [0, \frac{1}{2})$, while the policy target remains 0. Classification accuracies are computed against the hard expert/policy classes using a threshold of $D_\omega(s, a) = 0.5$.

For the one-centered gradient penalty, pair expert and policy inputs and interpolate their concatenated state-action representations using the same independently sampled coefficient for the state and action components of each pair. The gradient norm is taken with respect to the complete interpolated input. For one-hot actions, these interpolated action vectors are used only to evaluate the penalty and are never submitted to the environment.

After the discriminator update, freeze its parameters, compute one of the three surrogate rewards defined in the theoretical subsection for every stored policy transition, and recompute GAE and value targets from those rewards. The surrogate rewards are treated as fixed scalar data during PPO; gradients must not pass from

PPO into the discriminator. Environment rewards are used to train the initial expert and to evaluate GAIL, but they must not contribute to GAIL advantages, value targets, or policy updates.

The theoretical occupancy is normalized and discounted. In the finite-rollout implementation, discriminator minibatches and empirical occupancy plots use the normalized empirical distribution over retained transitions, with each retained transition receiving equal weight. This is an undiscounted, finite-rollout empirical analogue rather than the exact discounted occupancy objective above. Expert and policy data must use the same sampling convention. For the continuous CartPole state, the state-occupancy visualization should use common bins and shared axes for an explicitly identified two-dimensional projection, such as cart position versus pole angle, and normalize each policy's plotted visitation counts to sum to one.

Periodic deterministic evaluation must use the same episode count, seeds or seed distribution, and maximum episode length across all experiments. Select the best checkpoint using one fixed criterion, such as mean deterministic environment return, and distinguish the final training iterate from the restored best checkpoint in reported results. In the reward comparison, hold the expert data, environment-interaction budget, rollout length, PPO schedule, and evaluation protocol fixed. In the demonstration study, vary only the number of expert demonstration episodes while reporting the resulting number of retained expert pairs.

Implementation 4: PPO Backbone and Expert Demonstrations

[6 Points]

Implement the on-policy components used by both expert training and GAIL.

1. Write deterministic evaluation and fixed-length rollout collection. Record observations, actions, environment rewards, log-probabilities, values, episode boundaries, and truncation bootstrap values.
2. Implement the backward GAE recursion and return both advantages and value targets. Correctly distinguish true termination, time-limit truncation, and an unfinished rollout tail.
3. Implement the minibatched PPO update with whole-batch advantage normalization, ratio clipping, value regression, entropy regularization, and global gradient-norm clipping. Train an expert on the true reward and collect subsampled greedy expert demonstrations.

Your implementation should pass the supplied rollout, GAE, and performance checks before it is used by GAIL.

Implementation 5: Discriminator and Stable Surrogate Rewards

[6 Points]

Implement the adversarial reward model.

1. Build a discriminator that consumes a state concatenated with either an integer action converted to one-hot form or an already continuous action encoding, and returns a raw logit rather than a probability.
2. Add the one-centered gradient penalty on interpolations between expert and policy samples. Interpolate the state and action with the same per-sample coefficient and retain the higher-order graph required to optimize the penalty.
3. Train with binary cross-entropy on logits, one-sided expert-label smoothing, and a small number of discriminator steps. Implement all three reward variants directly from the logit using stable operations; do not compute logarithms of materialized sigmoid probabilities.

Return discriminator loss and expert/policy classification accuracies for diagnostics.

Implementation 6: End-to-End GAIL and Empirical Diagnostics**[6 Points]**

Assemble the complete alternating training loop. The policy update must use only the discriminator-derived reward; environment rewards may be logged but must never enter GAE or PPO. Track environment return, discriminator loss and accuracies, mean surrogate reward, policy entropy, and periodic greedy evaluations. Save and restore the best evaluated actor-critic parameters.

Then run two compact studies: compare the three surrogate rewards and vary the number of expert demonstration episodes. Produce learning curves and final scores, plus a state-occupancy plot comparing the expert, learned policy, and a random policy. Use shared axes and controlled seeds so the comparisons are meaningful.

Question 4: From Occupancy Matching to GAIL**[5 Points]**

For fixed occupancies ρ_{π^E} and ρ_{π} :

1. derive the pointwise optimal discriminator $D^*(s, a)$ and show that substituting it into the discriminator objective yields $2 \text{JS}(\rho_{\pi^E} \parallel \rho_{\pi}) - \log 4$;
2. explain what divergence the policy minimizes at the saddle point and why matching the state-action occupancy addresses behavioral cloning's compounding-error problem;
3. explain why GAIL cannot backpropagate the discriminator gradient through a sampled environment transition, what policy-gradient estimator replaces that path, and how GAE, PPO clipping, advantage normalization, and the value baseline control the resulting variance;
4. identify the code hyperparameter corresponding to the causal-entropy coefficient and predict the effect of setting it to zero.

Question 5: Reward Bias and Adversarial Training Dynamics**[5 Points]**

Use the recorded diagnostics and reward-ablation curves to analyze the following.

1. Why does an always-positive reward favor long episodes, while an always-negative reward favors early termination when the absorbing state's value is defined as zero? Explain why these biases help or hurt on CartPole and give an environment in which the preference would reverse.
2. Why can a nearly perfect discriminator destroy the policy's learning signal when the policy is far from the expert? Relate your answer to saturated logits, the shape of softplus, gradient penalties, label smoothing, and the meaning of discriminator accuracies near 1 and near 0.5.
3. Compare expert-demonstration sample complexity with environment-interaction complexity. Explain why GAIL may learn from very few expert pairs yet still be impractical when additional environment interaction is costly.

Support claims with specific results from your runs rather than qualitative impressions alone.

4 Offline DDPG with Behavior Cloning

In this question we study offline reinforcement learning on Pendulum-v1. A noisy heuristic behavior policy first creates a fixed dataset

$$\mathcal{D} = \{(s_i, a_i, r_i, s'_i, d_i)\}_{i=1}^N.$$

After collection, training uses no new environment transitions. The learner is DDPG: a deterministic actor $\pi_\theta(s)$ proposes a continuous action and a critic $Q_\phi(s, a)$ estimates its discounted return. Slowly moving target networks define the bootstrapped critic target

$$y = r + \gamma(1 - d)Q_{\bar{\phi}}(s', \pi_{\bar{\theta}}(s')), \quad L_Q = \mathbb{E}_{\mathcal{D}}[(Q_\phi(s, a) - y)^2].$$

The target is computed without gradients. A true terminal transition removes the bootstrap term, while a time-limit truncation does not. After each update, target parameters track live parameters by Polyak averaging,

$$\bar{\vartheta} \leftarrow (1 - \tau)\bar{\vartheta} + \tau\vartheta.$$

Naive DDPG updates the actor by minimizing $-\mathbb{E}[Q_\phi(s, \pi_\theta(s))]$. In a static dataset this can exploit critic errors by selecting out-of-distribution actions whose predicted values are spuriously high. DDPG+BC counters this extrapolation error with a behavior-cloning penalty:

$$L_\pi = -\lambda \mathbb{E}_{s \sim \mathcal{D}}[Q_\phi(s, \pi_\theta(s))] + \mathbb{E}_{(s,a) \sim \mathcal{D}}[\|\pi_\theta(s) - a\|_2^2], \quad \lambda = \frac{\alpha}{\mathbb{E}[|Q_\phi(s, \pi_\theta(s))|] + \varepsilon}.$$

The detached scale factor makes the relative trade-off less sensitive to the critic's numerical scale. States are optionally normalized with fixed dataset statistics. Three controlled configurations isolate the effects of normalization and BC: naive DDPG without normalization, naive DDPG with normalization, and DDPG+BC with normalization. The comparison is repeated across seeds with matched minibatch sequences, followed by a sweep over α .

Implementation 7: Target Networks and Critic Learning

[6 Points]

Implement the core DDPG stabilization machinery.

1. Implement in-place, gradient-free Polyak averaging for each target/live parameter pair and verify the limiting behavior at $\tau = 0$ and $\tau = 1$.
2. Compute the one-step TD target with the target actor and target critic under `no_grad`. Preserve shape $(B, 1)$ and mask only true terminal transitions.
3. Train the critic by mean-squared regression to the detached target and update both target networks after each actor-critic step.

Implementation 8: Adaptive Behavior-Regularized Actor

[5 Points]

Implement one actor-loss function supporting both naive DDPG and DDPG+BC. When behavior regularization is disabled, the loss must depend only on the critic's value of the actor's actions. When it is enabled, add the mean-squared distance between predicted and logged actions and adapt the coefficient of the value term using a detached batch statistic of $|Q(s, \pi(s))|$. State the chosen statistic and justify why it reduces sensitivity to a positive multiplicative rescaling of the critic and is exactly invariant to such a rescaling when $\varepsilon = 0$ and the batch statistic is nonzero. Ensure gradients update the live actor but do not update the critic through the actor optimization step.

Implementation 9: Controlled Offline-RL Ablation**[5 Points]**

Fit one fixed state normalizer from the offline dataset and run the following configurations with the same model class:

1. naive DDPG without normalization;
2. naive DDPG with normalization;
3. DDPG+BC with normalization.

For each seed, reset both network initialization and replay-sampling generators so all configurations receive the same minibatch sequence. Report multi-seed learning curves and final mean $\pm$ standard deviation. Finally, sweep at least three logarithmically separated values of α while holding the dataset, seed, normalizer, and training budget fixed.

Question 1: Extrapolation Error and Behavior Regularization**[5 Points]**

Explain the feedback loop that produces extrapolation error in offline DDPG: the actor proposes an out-of-distribution action, the critic overestimates it, and the actor is then optimized to exploit that error without receiving a corrective environment sample. Address the following points.

1. Why are the same actor-critic updates less dangerous in online RL?
2. How does the BC term encourage the actor to remain near logged actions, why does this usually improve data support, and why does it not guarantee support for a multimodal behavior policy? What performance ceiling can behavior regularization impose?
3. Why must the TD target use slowly moving target networks and be detached from autograd?
4. Why should time-limit truncations retain the bootstrap term whereas true terminations should not?

Question 2: Ablations, Multi-Seed Evidence, and Alpha Sensitivity**[4 Points]**

Use your tables and plots to make an evidence-based argument.

1. Compare naive DDPG with and without state normalization. Does normalization materially change mean performance or stability?
2. Compare normalized naive DDPG with normalized DDPG+BC. Does BC produce an additional effect consistent with the extrapolation-error account?
3. Could a single seed have led to a different conclusion? Explain why offline actor-critic methods can be especially seed-sensitive.
4. Interpret the α sweep. Relate $\alpha \rightarrow 0$ to behavior cloning and $\alpha \rightarrow \infty$ to critic maximization under the stated loss convention, and explain what the best observed value says about how much this dataset's actions should be trusted relative to the learned critic.

Include the relevant final means, standard deviations, and sweep values in your answer.

References

- [1] Based on INF8250AE - Introduction to Reinforcement Learning. Fall 2025.
- [2] Ng, A. Y., and Russell, S. (2000). Algorithms for Inverse Reinforcement Learning. ICML.
- [3] Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. (2008). Maximum Entropy Inverse Reinforcement Learning. AAAI.
- [4] Ross, S., Gordon, G., and Bagnell, D. (2011). A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. AISTATS.
- [5] Kumar, A., Zhou, A., Tucker, G., and Levine, S. (2020). Conservative Q-Learning for Offline Reinforcement Learning. NeurIPS.
- [6] Ho, J., and Ermon, S. (2016). Generative Adversarial Imitation Learning. NeurIPS.