A Dual-SPMA Framework for Convex MDPs

Shervin Khamooshian **Ahmed Magd** **Pegah Aryadoost** **Danielle Nguyen**
Simon Fraser University
{ska309, ams80, paa40, tdn8}@sfu.ca

Abstract

This project empirically studies a unified framework for solving Convex Markov Decision Processes (CMDPs) by combining two key ideas: a Fenchel Duality reformulation and the Softmax Policy Mirror Ascent (SPMA) algorithm. CMDPs generalize standard Reinforcement Learning (RL) to handle objectives with convex constraints and multi-objective trade-offs, but directly solving them is computationally hard due to their high-dimensional constrained space. Using Fenchel duality, we reformulate the CMDP into a min–max problem that alternates between a convex optimization over dual variables and a policy optimization step solvable via SPMA. The project implements this dual–SPMA framework, evaluates its convergence and efficiency, and provides empirical insights into the practicality of dual formulations for CMDPs when combined with efficient policy optimization methods.

1 Motivation

While Reinforcement Learning (RL) has demonstrated remarkable success across many domains, the standard formulation remains insufficient for capturing complex real-world objectives. Scenarios involving constraints, multi-objective trade-offs, or imitation learning often require more expressive formulations than conventional RL. The Convex Markov Decision Process (CMDP) framework extends RL to handle such cases by introducing convex objectives and constraints.

Solving CMDPs directly is difficult since it involves optimization over a high-dimensional constrained space of stationary distributions. To address this, prior work has employed Fenchel Duality to reformulate the primal CMDP minimization into a min-max optimization problem that alternates between two simpler subproblems: (i) a *dual update*, maximizing a convex function with respect to the dual variables y, and (ii) a *policy update*, solving a standard RL task with a y-shaped reward using established policy optimization methods.

To solve the policy subproblem, we adopt the Softmax Policy Mirror Ascent (SPMA) algorithm Asad et al. [2024], a principled approach with provable linear convergence for convex, smooth objectives.

While Fenchel Duality and SPMA exist, their integration in a unified empirical framework is underexplored. The empirical validation of this theoretically-promising dual decomposition remains uninvestigated, a gap this project addresses.

Our project thus aims to empirically implement and analyze this combined framework, developing a solver that is both theoretically grounded and computationally efficient.

2 Related Work

RL formulations beyond linear rewards. Typical MDP objectives can be written as maximizing a linear reward expectation over the occupancy measure Zahavy et al. [2021]:

$$\max_{d_\pi} \langle r(s,a), d_\pi(s,a) \rangle .$$

However, many practical settings are not captured by a simple inner product over state–action pairs. For example, entropy-regularized methods such as SAC Haarnoja et al. [2018] introduce diversity via an entropy-regularized objective (e.g., $f(d) = -\langle r, d\rangle - \alpha H(\pi_d)$), and constrained policy optimization Achiam et al. [2017] imposes inequality constraints (e.g., $\langle c_i, d_\pi\rangle \leq B_i$ for one or more cost functions c_i).

CMDP Reformulation via Fenchel Duality. A broad class of these objectives can be reformulated as *convex minimization problems* over the occupancy measure, $f(d_\pi)$, yielding the general *Convex MDP (CMDP)* formulation Zahavy et al. [2021]:

$$\min_{d_\pi \in \mathcal{K}} f(d_\pi).$$

The standard RL problem, $\max_{d_\pi} \langle r, d_\pi\rangle$, is just one simple instance of this, where $f(d_\pi) = \langle -r, d_\pi\rangle$. To solve the *general* problem, we use **Fenchel duality** to rewrite $f(d_\pi)$ in its conjugate form: $f(d_\pi) = \max_y\{\langle y, d_\pi\rangle - f^*(y)\}$, where y is the dual variable and f^* is the Fenchel conjugate. This transforms the original minimization into the **minimax saddle-point problem** Miryoosefi et al. [2019]:

$$\min_{d_\pi \in \mathcal{K}} \max_{y} \langle y, d_\pi\rangle - f^*(y).$$

Such saddle-point problems motivate descent–ascent procedures alternating between the minimization and maximization subproblems Lin et al. [2019], Ying et al. [2024].

Optimization roles: dual vs. policy player. For the *dual* (maximization) updates, convex first-order methods apply; when strong convexity holds, accelerated rates (e.g., Nesterov) are available Nesterov [1983]. In practice, full gradients can be costly, so SGD/Adam variants are common Kingma and Ba [2017]. For the *policy* (minimization) side, each fixed dual iterate induces a shaped-reward RL problem solvable with standard policy optimization.

Policy-gradient geometry and SPMA. Geometry-aware updates refine policy gradients via trust regions or mirror maps. TRPO enforces a KL trust region with monotonic improvement Schulman et al. [2015]; PPO relaxes this with clipping but can be hyperparameter sensitive Schulman et al. [2017], Lascu et al. [2025]. MDPO casts updates as mirror descent Tomar et al. [2020], providing a foundation for later geometry-aware methods. Building on this, SPMA operates in *logit* space with a log-sum-exp mirror map, achieving $O(\log(1/\epsilon))$ convergence and strong empirical performance, and reduces policy updates to convex softmax classification under linear function approximation Asad et al. [2024].

Summary. Prior work (i) shows many RL objectives can be expressed as convex CMDPs via Fenchel duality; (ii) motivates alternating descent–ascent between dual and policy players; and (iii) develops geometry-aware policy optimization where SPMA offers both theoretical and practical advantages. Our study integrates these strands empirically: a dual update for the convex side and SPMA for the policy side within a unified CMDP solver.

3 Problem Formulation

MDP and occupancies. We consider a discounted infinite-horizon MDP $(\mathcal{S}, \mathcal{A}, P, \rho, \gamma)$ with $\gamma \in (0, 1)$, start-state distribution ρ, and transition kernel $P(\cdot|s, a)$. For a stationary policy π, its discounted state–action occupancy measure is

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

The feasible occupancies form a convex polytope $\mathcal{D}$ defined by standard flow constraints (full form moved for space to Appendix B). Any $d \in \mathcal{D}$ induces a policy via $\pi(a|s) = \frac{d(s,a)}{\sum_{a'} d(s,a')}$ when the denominator is nonzero.

Convex CMDP objective. Our goal is to minimize a convex function $f : \mathcal{D} \to \mathbb{R} \cup \{+\infty\}$ of the occupancy:

$$\min_{\pi} \; f(d_\pi) \qquad \Longleftrightarrow \qquad \min_{d \in \mathcal{D}} \; f(d). \tag{1}$$

Examples include constrained safety and entropy-regularized RL (details in Appendix A).

Fenchel duality and saddle formulation. Let f^* denote the Fenchel conjugate of f, $f^*(y) = \sup_{x \in \mathcal{D}} \langle y, x \rangle - f(x)$. By the Fenchel–Moreau representation,

$$f(d_\pi) = \max_{y \in \mathcal{Y}} \langle y, d_\pi \rangle - f^*(y),$$

for a suitable dual domain $\mathcal{Y}$. Thus (1) is equivalent to

$$\min_{\pi} \max_{y \in \mathcal{Y}} \mathcal{L}(\pi, y) := \langle y, d_\pi \rangle - f^*(y). \tag{2}$$

This is advantageous because for fixed y we obtain a standard RL problem, and for fixed π the y-update is a convex optimization. For fixed y, the term $\langle y, d_\pi \rangle$ corresponds to an RL problem with shaped reward ($r_y(s,a) = -y(s,a)$ in tabular form; or $r_y(s,a) = \phi(s,a)^\top y$ with linear features).

Regularity assumptions. We adopt standard conditions (communicating MDP; f proper/closed/convex; compact $\mathcal{D}$) ensuring existence of a saddle point and well-posedness; see Appendix B for statements and implications.

First-order structure. When f^* is differentiable (else use subgradients), the dual gradient is $\nabla_y \mathcal{L}(\pi, y) = d_\pi - \nabla f^*(y)$. For a fixed y, the policy player's objective is to solve $\min_{\pi_\theta} \langle y, d_{\pi_\theta} \rangle$. This is equivalent to maximizing the standard RL objective $J_y(\pi_\theta) := \langle -y, d_{\pi_\theta} \rangle$ using the shaped reward $r_y = -y$. The gradients for this saddle-point problem are thus:

$$\nabla_y \mathcal{L}(\pi, y) = d_\pi - \nabla f^*(y), \qquad \nabla_\theta J_y(\pi_\theta) = \mathbb{E}_{\pi_\theta}\big[\nabla_\theta \log \pi_\theta(a|s)\, Q^{\pi_\theta}_{r_y}(s,a)\big].$$

Our solution framework applies mirror ascent to both players in this min-max game. For the dual (maximization) step, we apply a general convex first-order mirror ascent method over $\mathcal{Y}$. For the policy (primal) step, we maximize $J_y(\pi_\theta)$ by employing Softmax Policy Mirror Ascent (SPMA), which is a specific mirror ascent algorithm using the log-sum-exp mirror map. Estimators and implementation details are in Appendix C.

4 Plan

We follow a four-phase plan (see Appendix, Fig. 1) with milestones on Nov. 10, Dec. 2, and Dec. 15. Tasks run in parallel with clear handovers to keep theory, algorithms, and experiments aligned.

Phase 1: Parallel Foundations (Oct. 22–Nov. 10). **A** reviews literature on convex MDPs, constrained policy gradients, and related optimization; **B** defines/implements the environment and sanity-checks it with a standard RL baseline; **C** implements the SPMA-based main algorithm under the CMDP formulation; **D** builds comparative CMDP baselines (e.g., constrained PG). **Milestone (Nov. 10):** review, environment, and both algorithmic branches completed and validated.

Phase 2: Integration (Nov. 10–Nov. 17). **A** prepares slides; **B+C** integrate the environment with the main algorithm and run end-to-end tests; **B+D** repeat integration with comparative methods to check consistency.

Phase 3: Ablation & Presentation (Nov. 17–Dec. 2). **C** runs ablations over RL/optimization choices; **D** varies CMDP objectives; **A+B** finalize slides and incorporate results as C/D complete analyses. **Milestone (Dec. 2):** ablations finished and included in the presentation.

Phase 4: Reporting (Dec. 2–Dec. 15). After the Dec. 2 presentation, all members draft, refine, and finalize the report, integrating motivation, methods, and empirical analysis into a cohesive document. **Milestone (Dec. 15):** final report submitted.

Work Division & Coordination. Leads: **A** (theory/writing), **B** (environments/integration), **C** (algorithms), **D** (comparative methods). Weekly syncs maintain visibility; results are cross-validated for correctness and reproducibility.

Expected Outcomes. (i) Primary and comparative CMDP implementations; (ii) verified environment and integrations; (iii) systematic ablations across algorithms/objectives; (iv) a complete presentation and technical report summarizing theoretical and empirical insights.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 22–31. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/achiam17a.html`.

Reza Asad, Reza Babanezhad Harikandeh, Issam H. Laradji, Nicolas Le Roux, and Sharan Vaswani. Fast convergence of softmax policy mirror ascent for bandits & tabular MDPs. In *OPT 2024: Optimization for Machine Learning*, 2024. URL `https://openreview.net/forum?id=f5OjNMXIik`.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018. URL `http://arxiv.org/abs/1801.01290`.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL `https://arxiv.org/abs/1412.6980`.

Razvan-Andrei Lascu, David Šiška, and Łukasz Szpruch. Ppo in the fisher-rao geometry, 2025. URL `https://arxiv.org/abs/2506.03757`.

Tianyi Lin, Chi Jin, and Michael I. Jordan. On gradient descent ascent for nonconvex-concave minimax problems. *CoRR*, abs/1906.00331, 2019. URL `http://arxiv.org/abs/1906.00331`.

Sobhan Miryoosefi, Kianté Brantley, Hal Daumé III, Miroslav Dudík, and Robert E. Schapire. Reinforcement learning with convex constraints. *CoRR*, abs/1906.09323, 2019. URL `http://arxiv.org/abs/1906.09323`.

Y. Nesterov. A method for solving the convex programming problem with convergence rate o(1/k2), 1983. URL `https://cir.nii.ac.jp/crid/1370862715914709505`.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015. URL `http://arxiv.org/abs/1502.05477`.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL `http://arxiv.org/abs/1707.06347`.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *CoRR*, abs/2005.09814, 2020. URL `https://arxiv.org/abs/2005.09814`.

Donghao Ying, Mengzi Amy Guo, Hyunin Lee, Yuhao Ding, Javad Lavaei, and Zuo-Jun Max Shen. Policy-based primal-dual methods for concave cmdp with variance reduction, 2024. URL `https://arxiv.org/abs/2205.10715`.

Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins, and Satinder Singh. Reward is enough for convex mdps. *Advances in Neural Information Processing Systems*, 34:25746–25759, 2021.

Appendix

A Convex Objective Examples

A.1 Constrained Safety

$f(d) = -\langle r, d\rangle + \lambda \max\{0, \langle c, d\rangle - \tau\}$, where c encodes costs (e.g., collision risk) and τ is a safety threshold. This formulation allows reward maximization while ensuring safety constraints are satisfied.

A.2 Entropy-Regularized RL

$f(d) = -\langle r, d\rangle - \alpha \cdot H(\pi_d)$, where $H(\pi_d) = -\sum_{s,a} d(s,a) \log \frac{d(s,a)}{\sum_{a'} d(s,a')}$ is the entropy of the policy induced by d, and $\alpha > 0$ controls exploration. This maximum-entropy framework encourages diverse behavior while maximizing reward and has been successfully applied in modern deep RL methods such as soft actor-critic.

B Regularity Assumptions and Their Implications

This appendix explains the regularity assumptions from Section 3 and their role in ensuring well-posedness and convergence of our convex MDP framework.

B.1 Overview of Assumptions

We assume:

1. The MDP is communicating.
2. The objective function f is proper, closed, and convex on $\mathcal{D}$
3. For discounted problems with $\gamma \in (0, 1)$, the feasible set $\mathcal{D}$ is nonempty and compact.

B.2 Communicating MDP

Definition. An MDP is communicating if every state can reach every other state under some policy: for all $(s, s') \in \mathcal{S} \times \mathcal{S}$, there exists policy π and finite T with $\Pr_\pi(s_T = s' | s_0 = s) > 0$.

Necessity and Consequences. *With* communicating assumption we obtain well-defined occupancy measures throughout the state space, meaningful policy gradients everywhere, and no dead-end states that trap policies; *without* it, consider a dead-end state s_{trap} with $P(s_{\text{trap}} | s_{\text{trap}}, a) = 1$ for all a where once entered, all occupancy mass concentrates there ($\lim_{t\to\infty} d_\pi(s_{\text{trap}}, \cdot) \to 1$), causing ill-defined policy gradients for other states, SPMA failures due to zero probabilities, and unbounded dual variables.

B.3 Properties of Objective Function f

B.3.1 Proper

Definition. Function f is proper if $f(d) > -\infty$ for all $d \in \mathcal{D}$ and $f(d) < +\infty$ for at least one d.

Necessity and Consequences. *With* properness the minimization problem is bounded below, at least one feasible solution exists, and the Fenchel conjugate $f^*(y) = \sup_{d\in\mathcal{D}} \langle y, d\rangle - f(d)$ is well-defined; *without* it, if $f(d) = -\infty$ for some d, optimization is unbounded below, if $f(d) = +\infty$ for all d, the problem is infeasible, and in either case the Fenchel conjugate becomes undefined.

B.3.2 Closed

Definition. Function f is closed (lower semicontinuous) if its epigraph $\text{epi}(f) = \{(d, \alpha) : f(d) \leq \alpha\}$ is closed, or equivalently, $f(d) \leq \liminf_{n\to\infty} f(d_n)$ for any sequence $\{d_n\} \to d$.

Necessity and Consequences. *With* closedness minimizers exist (infimum is attained), limit points of converging sequences remain feasible, and the Fenchel-Moreau duality $f = f^{**}$ holds, ensuring exact duality with zero gap; *without* it, the infimum may not be attained (optimal solution doesn't exist), converging sequences may have strictly better values than their limits, and Fenchel duality may have a gap, making our saddle formulation inexact.

B.3.3 Convex

Definition. Function f is convex if $f(\lambda d_1 + (1-\lambda)d_2) \leq \lambda f(d_1) + (1-\lambda)f(d_2)$ for all $d_1, d_2 \in \mathcal{D}$ and $\lambda \in [0, 1]$.

Necessity and Consequences. *With* convexity we obtain exact Fenchel duality (zero gap, so solving the dual solves the primal), any local minimum is global, subgradients exist everywhere, and the dual problem over y is convex with convergence guarantees; *without* it, a duality gap emerges ($\min_\pi \max_y \mathcal{L} \neq \max_y \min_\pi \mathcal{L}$) so our algorithm optimizes the wrong objective, multiple local minima trap optimization, SPMA's convergence guarantees fail, and saddle points may not exist.

Verification for Our Objectives. Our examples are convex:

- **Constrained safety**: $f(d) = -\langle r, d\rangle + \lambda \max\{0, \langle c, d\rangle - \tau\}$ is a sum of linear (convex) and composition of max (convex) with linear (convex), hence convex.
- **Entropy-regularized**: $f(d) = -\langle r, d\rangle - \alpha H(\pi_d)$ is a sum of linear and negative entropy (convex), hence convex.

B.4 Nonemptiness and Compactness of $\mathcal{D}$

Nonemptiness. For any MDP with $\gamma \in (0, 1)$, nonemptiness is automatic: any policy π induces a valid occupancy $d_\pi \in \mathcal{D}$. Without nonemptiness, the problem is vacuous.

Compactness. For discounted MDPs, $\mathcal{D}$ is compact: it is *bounded* because $\sum_{s,a} d_\pi(s, a) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t = 1$, so d_π lies in the unit simplex. It is *closed* because the flow constraints are linear equations, making $\mathcal{D}$ the intersection of the positive orthant (closed) with an affine subspace (closed).

Necessity and Consequences. *With* compactness saddle points exist (Minimax Theorem), minimizers exist (Extreme Value Theorem), iterates $\{(\pi_k, y_k)\}$ remain bounded with convergent subsequences, and dual variables stay bounded; *without* it, there is no saddle point guarantee, the infimum may not be attained, iterates may diverge to infinity, and convergence proofs break down.

B.5 Additional Conditions for Fast Convergence

Strong Convexity of f^*. If f^* is μ-strongly convex ($f^*(\lambda y_1 + (1-\lambda)y_2) \leq \lambda f^*(y_1) + (1-\lambda)f^*(y_2) - \frac{\mu}{2}\lambda(1-\lambda)\|y_1 - y_2\|^2$), then gradient methods achieve *linear* convergence $O((1-\mu/L)^k)$ instead of sublinear $O(1/k)$, with better conditioning and stable updates; if f^* lacks strong convexity, add regularization $f_\epsilon(d) = f(d) + \frac{\epsilon}{2}\|d\|^2$ to make f_ϵ^* strongly convex.

B.6 Summary

Together, these assumptions ensure: (1) the problem is well-posed with a unique solution, (2) Fenchel duality is exact with zero gap, (3) our alternating optimization converges to the global optimum, and (4) convergence is fast (linear rate under strong convexity). Without them, the algorithm may fail to converge, converge to suboptimal solutions, or optimize an incorrect objective.

C Practical Estimation of Occupancy Measures

While the problem formulation defines the occupancy measure $d_\pi(s, a)$ mathematically, implementing our algorithm requires *practical methods* to estimate d_π from simulation data. This appendix details

Assumption	With	Without
Communicating MDP	Well-defined occupancy; meaningful gradients	Degenerate distributions; trapped states
f proper	Bounded problem; feasible solution exists	Unbounded or infeasible
f closed	Minimizer exists; exact duality	Infimum not attained; duality gap
f convex	Exact duality; global optimality; convergence	Duality gap; local minima; wrong objective
$\mathcal{D}$ compact	Saddle point exists; bounded iterates	No saddle point; diverging iterates
f^* strongly convex	Linear convergence; stable updates	Slow sublinear convergence

Table 1: Summary of regularity assumptions and their implications.

two complementary approaches: one for tabular settings (small, discrete state-action spaces) and one for function approximation (large or continuous spaces).

C.1 Why We Need to Estimate d_π

The dual gradient from Section 3 requires the occupancy measure:

$$\nabla_y \mathcal{L}(\pi, y) = d_\pi - \nabla f^*(y).$$

While d_π has a formal definition as an infinite sum, we must estimate it from finite simulation data. The estimation method depends on the problem scale.

C.2 Tabular Case: Direct Occupancy Estimation

Setting. When the state-action space is small (e.g., fewer than 10,000 pairs) and discrete, we can explicitly track visits to each (s, a) pair.

Estimator. A Monte Carlo estimate of the discounted occupancy measure is obtained by running N trajectories (rollouts) and averaging the discounted visits:

$$\hat{d}_\pi(s, a) = \frac{1-\gamma}{N} \sum_{i=1}^{N} \sum_{t=0}^{T_i} \gamma^t \mathbb{I}[s_t^{(i)} = s, a_t^{(i)} = a],$$

where T_i is the length of the i-th trajectory, and $\mathbb{I}[\cdot]$ is the indicator function.

Procedure.

1. Run policy π in the environment for N full trajectories (rollouts).
2. For each trajectory i, and for each step t, add $(1-\gamma)\gamma^t$ to a running total for the visited state-action pair $(s_t^{(i)}, a_t^{(i)})$.
3. Divide the final total for each (s, a) pair by N to get the estimate.

Example. Consider running $N = 100$ rollouts with $\gamma = 0.9$.

- (A, right) is visited at $t = 0$ in 10 rollouts and $t = 1$ in 5 rollouts.
- Its estimated occupancy is: $\frac{1-0.9}{100} \left[(10 \times 0.9^0) + (5 \times 0.9^1)\right]$
- $\hat{d}_\pi(A, \text{right}) = 0.001 \times [10 + 4.5] = 0.0145$.

This is mathematically distinct from the simple count $\frac{15}{T}$.

Theoretical Justification. This is an unbiased Monte Carlo estimator of the true discounted occupancy measure:

$$\mathbb{E}[\hat{d}_\pi(s, a)] = d_\pi(s, a).$$

By the Law of Large Numbers, as the number of rollouts $N \to \infty$, $\hat{d}_\pi(s, a) \to d_\pi(s, a)$.

Use in Algorithm. In the outer loop of our algorithm, after running SPMA to update the policy π, we:

1. Estimate $\hat{d}_\pi$ using the procedure above.
2. Compute the dual gradient: $\nabla_y \mathcal{L} = \hat{d}_\pi - \nabla f^*(y)$
3. Update $y \leftarrow y + \eta \cdot \nabla_y \mathcal{L}$ (or a mirror ascent step).

C.3 Function Approximation: Indirect Estimation via Returns

Setting. When the state-action space is large (e.g., millions of pairs) or continuous (e.g., robot control with visual inputs), we cannot enumerate all (s, a) pairs or maintain explicit tables.

Key Insight. We do not need the full distribution $d_\pi(s, a)$ for all (s, a). The dual gradient only requires the inner product $\langle y, d_\pi \rangle$, which can be estimated indirectly.

Mathematical Equivalence. The inner product $\langle y, d_\pi \rangle$ equals the *discounted sum* of expected shaped rewards, scaled by $(1-\gamma)$. (Note: As established in Sec. 3, the policy player's reward is $r_y = -y$.)

$$\begin{aligned}
\langle y, d_\pi \rangle &= \sum_{s,a} y(s,a) \cdot d_\pi(s,a) \\
&= \sum_{s,a} y(s,a) \cdot \left[(1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr_\pi(s_t = s, a_t = a) \right] \\
&= (1-\gamma)\, \mathbb{E}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t y(s_t, a_t) \right] \\
&= (1-\gamma)\, \mathbb{E}_{\tau\sim\pi} \left[\text{Return with reward } y\right],
\end{aligned}$$

where the return $G_y^{(i)} = \sum_{t=0}^{T_i} \gamma^t y(s_t^{(i)}, a_t^{(i)})$ for trajectory i.

Estimator. We estimate $\langle y, d_\pi \rangle$ by averaging the returns over N rollouts and applying the $(1-\gamma)$ factor:

$$\langle y, d_\pi \rangle \approx (1-\gamma) \cdot \frac{1}{N} \sum_{i=1}^{N} G_y^{(i)},$$

where $G_y^{(i)} = \sum_{t=0}^{T_i} \gamma^t y(s_t^{(i)}, a_t^{(i)})$ is the discounted return for the i-th trajectory under shaped reward y.

Procedure.

1. Generate N trajectories by running policy π: $\tau^{(i)} = (s_0, a_0, s_1, a_1, \ldots)$
2. For each trajectory i, compute the discounted return using y as the reward:

$$G_y^{(i)} = \sum_{t=0}^{T_i} \gamma^t y(s_t^{(i)}, a_t^{(i)})$$

   where $y(s, a) = \phi(s, a)^\top w$ (linear features) or from the dual variable table.
3. Average the returns and multiply by $(1-\gamma)$: $\langle y, \hat{d_\pi} \rangle = (1-\gamma) \cdot \frac{1}{N} \sum_{i=1}^{N} G_y^{(i)}$

Example. Suppose y is parameterized. For trajectory 1 with $\gamma = 0.9$:

$$\begin{aligned}
t = 0: &\quad (s_0, a_0), \quad y(s_0, a_0) = 2.5 \\
t = 1: &\quad (s_1, a_1), \quad y(s_1, a_1) = 1.3 \\
t = 2: &\quad (s_2, a_2), \quad y(s_2, a_2) = -0.5
\end{aligned}$$

The discounted return (with $\gamma = 0.9$):

$$G_y^{(1)} = 1.0(2.5) + 0.9(1.3) + 0.81(-0.5) = 2.5 + 1.17 - 0.405 = 3.265$$

After collecting 100 trajectories with an average return of $\bar{G}_y \approx 3.05$, the estimate is:

$$\langle y, \hat{d_\pi} \rangle \approx (1 - 0.9) \times 3.05 = 0.1 \times 3.05 = 0.305$$

Parameterization. In function approximation, y is typically parameterized by w:

$$y(s,a) = \phi(s,a)^\top w \quad \text{or} \quad y(s,a) = \text{NeuralNet}_w(s,a).$$

The inner product $\langle y, d_\pi \rangle$ becomes $\langle w, \mathbb{E}_{d_\pi}[\phi(s,a)] \rangle$. The gradient of $\mathcal{L}$ w.r.t w is:

$$\nabla_w \mathcal{L}(\pi, w) = \mathbb{E}_{d_\pi}[\phi(s,a)] - \nabla f^*(w),$$

where $\mathbb{E}_{d_\pi}[\phi(s,a)]$ is the discounted feature expectation, estimated as:

$$\mathbb{E}_{d_\pi}[\phi(s,a)] \approx (1-\gamma) \cdot \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T_i} \gamma^t \phi(s_t^{(i)}, a_t^{(i)}).$$

Use in Algorithm. In the outer loop:

1. Collect N trajectories using policy π.
2. Estimate feature expectations $\mathbb{E}_{d_\pi}[\phi(s,a)]$ using the discounted sum above.
3. Compute dual gradient: $\nabla_w \mathcal{L} = \mathbb{E}_{d_\pi}[\phi(s,a)] - \nabla f^*(w)$.
4. Update $w \leftarrow w + \eta \cdot \nabla_w \mathcal{L}$ (or a mirror ascent step).

C.4 Comparison and When to Use Each Method

Aspect	**Tabular**	**Function Approximation**
State-action space	Small, discrete (< 10,000 pairs)	Large or continuous (> 10,000 or ∞)
What is estimated	Full $d_\pi(s,a)$ for all (s,a)	Only inner product $\langle y, d_\pi \rangle$
Estimation method	Discounted visits: $\frac{1-\gamma}{N} \sum_i \sum_t \gamma^t \mathbb{I}$	Discounted returns: $(1-\gamma)\frac{1}{N} \sum_i G_y^{(i)}$
Dual variable y	Table with one entry per (s,a)	Function: $y(s,a) = \phi(s,a)^\top w$
Memory	$O(\|\mathcal{S}\| \cdot \|\mathcal{A}\|)$	O(# parameters)
Examples	GridWorld, simple games	Robotics, Atari with images

Table 2: Comparison of tabular and function approximation estimation methods.

D Timeline (Supplemental)

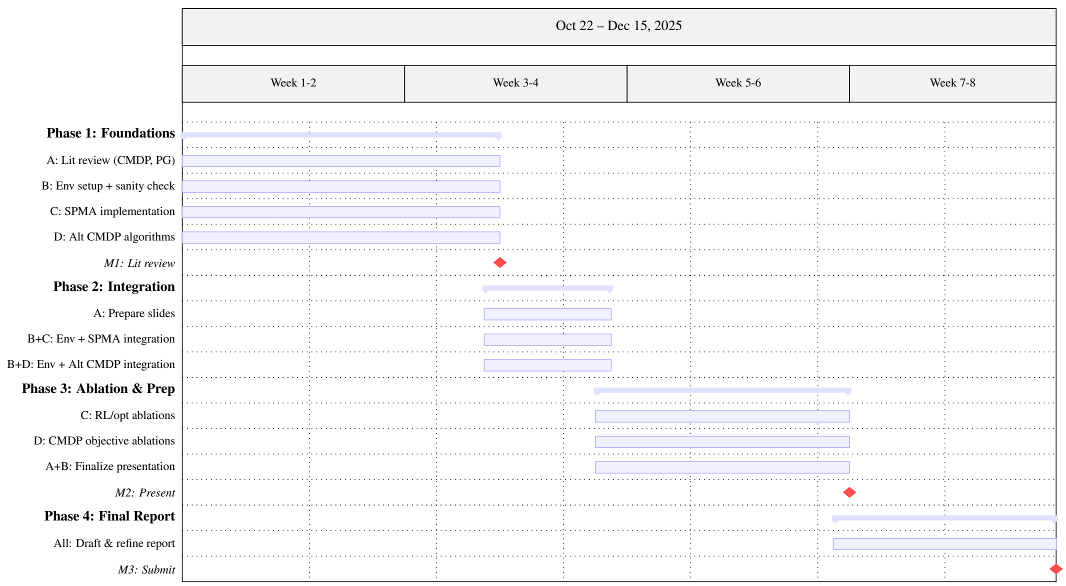

Figure 1: Project timeline with four phases and team coordination (A, B, C, D).