Project Milestone — Literature Review: A Dual–SPMA Framework for Convex MDPs

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

Goal — why Convex MDPs?

Many reinforcement learning (RL) goals can be expressed more generally than "maximize a stationary reward." *Convex MDPs (CMDPs)* write the objective as a convex function of the discounted occupancy measure d_π, strictly generalizing standard RL (linear reward is a special case). This view covers imitation/occupancy matching, constrained/safe RL, and exploration while still allowing us to reuse standard RL subroutines once the problem is put into the right saddle-point form. We adopt this vantage and keep implementation specifics (dual update, occupancy estimators, FA projection) outside this milestone.

Paper 1: *Reward is Enough for Convex MDPs* (NeurIPS 2021)

What it shows. Zahavy et al. reformulate convex-in-occupancy objectives using *Fenchel duality*. If the CMDP objective is $\min_{d \in \mathcal{K}} f(d)$, then

$$\min_{d \in \mathcal{K}} f(d) \;=\; \min_{d \in \mathcal{K}} \max_{y \in \Lambda} \langle y, d \rangle - f^{(y),}$$

a convex–concave saddle between a *policy player* (choosing d) and a *cost player* (choosing y). For fixed y, the policy subproblem reduces to *standard RL with a shaped reward* $r_y(s,a) = -y(s,a)$ (or $r_y = \phi(s,a)^\top y$ with features). A meta-algorithm alternates a convex update for y with a low-regret RL learner for the policy, yielding $O(1/\sqrt{K})$ optimization error for averaged occupancies.

Why it matters for us. This paper provides the *outer-loop template*: update the dual y (convex ascent), and solve the policy step by running any strong RL algorithm on the shaped reward r_y. It justifies our claim that a modern policy optimizer (next section) can be *plugged in* as the policy player.

Paper 2: *Fast Convergence of Softmax Policy Mirror Ascent* (OPT 2024 / arXiv 2025)

What it shows. *Softmax Policy Mirror Ascent (SPMA)* performs mirror ascent in *logit space* with the log-sum-exp mirror map. In tabular MDPs the per-state update

$$\pi_{t+1}(a|s) = \pi_t(a|s)\big(1 + \eta\, A^{\pi_t}(s,a)\big)$$

avoids explicit normalization and attains *linear convergence* for sufficiently small constant step size (e.g., $\eta \le 1-\gamma$); with function approximation, SPMA projects via convex softmax-classification and converges linearly to a neighbourhood of the optimum. Empirically it competes with PPO/TRPO/MDPO.

Why it matters for us. In the saddle from Paper 1, the policy step is "just RL with r_y." *SPMA* is our chosen *policy player*: it is fast in tabular settings and has a principled FA extension via convex surrogates, making it a strong best response inside the CMDP saddle.

Paper 3: *Natural Policy Gradient Primal–Dual for CMDPs* (NeurIPS 2020)

What it shows. Ding et al. study a policy-based primal–dual algorithm for discounted CMDPs: *natural policy gradient* ascent for the policy and *projected subgradient* updates for the multiplier. With softmax policies they prove *dimension-free* $O(1/\sqrt{T})$ rates for the averaged optimality gap and constraint violation; with general function approximation, similar rates hold up to approximation error, and sample-based variants admit finite-sample guarantees.

Why it matters for us. NPG–PD addresses closely related CMDP goals using a different geometry (probability-space KL vs. SPMA's logit space) and provides a *principled baseline and comparator* for our evaluation (gap, violation, sample efficiency).

How the three papers connect (and to our project)

Synthesis. Zahavy et al. give the *CMDP-as-saddle* reduction (fix $y \Rightarrow$ standard RL with shaped reward r_y; update y via convex ascent). *SPMA* supplies a modern *policy player* with linear rates and a clean FA story. *NPG–PD* offers a policy-based *primal–dual* method with dimension-free sublinear guarantees and thus a natural *baseline*. Our milestone focuses on these literature links; implementation specifics (e.g., mirror-ascent update on y, occupancy estimators) can be placed in an appendix.