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

1 Project Topic

Goal. We study a unified way to solve *Convex MDPs (CMDPs)* by combining a Fenchel-dual saddle formulation with a geometry-aware policy optimizer, *Softmax Policy Mirror Ascent (SPMA)*. CMDPs minimize a convex function of discounted occupancies and are equivalent to the saddle $\min_\pi \max_y \langle y, d_\pi \rangle - f^*(y)$. Fixing y turns the policy step into standard RL with shaped reward $r_y(s,a) = -y(s,a)$ (or $-\,\phi(s,a)^\top y$ under features). We alternate a mirror-ascent step on y with an SPMA policy step and return discounted occupancy (or feature-expectation) estimates for the next dual update (Fig. 1).

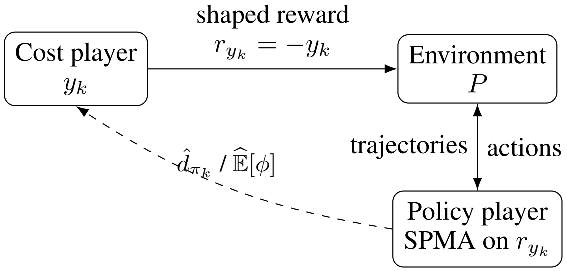

Figure 1: **Dual–SPMA loop.** Dual ascent chooses y_k, which induces a shaped reward for the SPMA policy step; discounted occupancies feed the next dual update.

2 Literature Review: Paper Summaries

We review three foundational papers that inform our Dual–SPMA framework: (1) the Fenchel duality approach for convex MDPs, (2) the SPMA policy optimizer with fast convergence, and (3) primal-dual methods for constrained MDPs.

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

Core idea. Many RL goals can be posed as $\min_{d\in\mathcal{K}} f(d)$ for convex f over the occupancy polytope $\mathcal{K}$. Using Fenchel conjugacy, $\min_{d\in\mathcal{K}} f(d) = \min_{d\in\mathcal{K}} \max_{\lambda\in\Lambda} \lambda\cdot d - f^*(\lambda)$, so for fixed λ the policy subproblem is vanilla RL with shaped reward $r_\lambda = -\lambda$.

Figure 2 illustrates this as a two-player game where the agent sees non-stationary rewards from the cost player. A meta-algorithm (Algorithm 1) alternates a *cost player* (FTL/OMD in λ, a convex ascent step) with a *policy player* (best response or low-regret RL, which reduces to "just RL" under the shaped reward), yielding $O(1/\sqrt{K})$ optimization error for averaged iterates under standard OCO assumptions. The paper shows best-response is ideal but often intractable in deep RL, so low-regret learners (e.g., UCRL2, MDPO) suffice; the guarantees hold for averaged occupancies $\bar{d}_\pi^K$ rather than single iterates. It unifies apprenticeship learning, CMDPs and pure exploration (Table 1). [Zahavy et al., 2021]

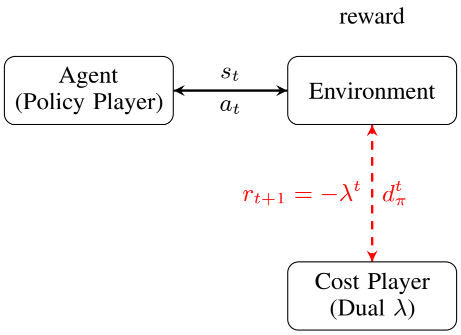

Figure 2: Convex MDP as a two-player game (adapted from Zahavy et al. [2021], Fig. 1). The cost player provides non-stationary shaped rewards $r_t = -\lambda^t$ to the agent, observing the resulting occupancy measures d_π^t. From the agent's perspective, this reduces to standard RL with time-varying rewards.

Algorithm 1: Meta-algorithm for Convex MDPs [Zahavy et al., 2021]

```
Input: Convex-concave payoff $\mathcal{L} : \mathcal{K} \times \Lambda \to \mathbb{R}$, algorithms $\mathrm{Alg}_\lambda$, $\mathrm{Alg}_\pi$, $K \in \mathbb{N}$
for $k = 1, \dots, K$ do
    $\lambda^k \leftarrow \mathrm{Alg}_\lambda(d_\pi^1, \dots, d_\pi^{k-1}; \mathcal{L})$ ;                // Cost player update
    $d_\pi^k \leftarrow \mathrm{Alg}_\pi(-\lambda^k)$ ;                // Policy: solve RL with $r = -\lambda^k$
Output: $\bar{d}_\pi^K = \frac{1}{K}\sum_{k=1}^K d_\pi^k$, $\bar{\lambda}^K = \frac{1}{K}\sum_{k=1}^K \lambda^k$
```

Relevance. This work justifies the saddle and the shaped-reward reduction we implement and informs our outer-loop design (dual MA + policy best response).

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

Core idea. *SPMA* performs mirror ascent in *logit* space using 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 achieves *linear convergence* for sufficiently small constant step-size, improving over softmax PG. For large problems, SPMA projects onto function classes via convex *softmax classification* subproblems and proves linear convergence to a neighbourhood under FA; empirically it competes with PPO/TRPO/MDPO. [Asad et al., 2024]

Relevance. We need a strong policy "best response" in Zahavy's saddle; SPMA provides the geometry and rates, and its FA projection matches our shaped-reward reduction.

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

Core idea. A policy-based primal–dual method: *natural policy gradient* (NPG) ascent for the policy and projected subgradient updates for the multiplier. Despite nonconcavity/nonconvexity under softmax parameterization, it proves *dimension-free* $O(1/\sqrt{T})$ bounds on averaged optimality gap and constraint violation; with FA, rates hold up to an approximation neighbourhood; sample-based variants have finite-sample guarantees. [Ding et al., 2020]

Relevance. NPG–PD is our principled CMDP baseline for both guarantees and practice; we compare Dual–SPMA against it in convergence/violation/sample-efficiency.

Table 1: Instantiations of the convex MDP framework (adapted from Zahavy et al. [2021], Table 1). Different choices of objective f and player algorithms recover well-known RL problems.

Application	**Objective** $f(d_\pi)$	**Cost Player**	**Policy Player**
Standard RL	$-\lambda \cdot d_\pi$ (linear)	FTL	RL
Apprenticeship Learning	$\|d_\pi - d_E\|_2^2$	FTL	Best Response
Pure Exploration	$d_\pi \cdot \log(d_\pi)$ (entropy)	FTL	Best Response
AL with ℓ_∞	$\|d_\pi - d_E\|_\infty$	OMD	Best Response
Constrained MDPs	$\lambda_1 \cdot d_\pi$ s.t. $\lambda_2 \cdot d_\pi \leq c$	OMD	RL
GAIL / State Matching	$\mathrm{KL}(d_\pi \| d_E)$	FTL	RL

3 How the Papers Relate to Each Other and to our project

The three papers provide complementary perspectives on our Dual–SPMA framework (Fig. 3):

Formulation. Zahavy et al. provide the *outer-loop template*: the Fenchel-dual saddle formulation that reduces constrained/structured RL to alternating between a cost player (dual ascent) and a policy player (shaped-reward RL). This formulation is key because fixing the dual variable y yields standard RL with shaped reward $r_y = -y$.

Policy player. SPMA (Asad et al.) supplies the *fast policy player* for the RL step in Zahavy's framework. It performs mirror ascent in logit space, achieving linear convergence rates in tabular settings and practical function approximation via convex softmax classification. This makes SPMA an ideal candidate for the policy subproblem.

Baseline and dual updates. NPG–PD (Ding et al.) offers a *policy-based CMDP baseline* with dimension-free sublinear guarantees. While we use SPMA (not NPG) as our policy player, Ding et al. inform how we handle constraint multipliers via projected subgradient updates and serve as our principled comparison baseline.

Table 2 summarizes how these three approaches differ in their formulation, policy optimization method, and convergence guarantees.

Table 2: Three perspectives that our project unifies or compares against.

Work	Objective / Saddle	Policy Player	Guarantees / Notes
Zahavy et al. (2021) [Zahavy et al., 2021]	$\min_d f(d)$; Fenchel dual $\min_d \max_\lambda \lambda \cdot d - f^*(\lambda)$	Best response / low-regret RL under $r_\lambda = -\lambda$	$O(1/\sqrt{K})$ via OCO; unifies AL, CMDPs, exploration
Asad et al. (2025) [Asad et al., 2024]	RL inner step (fixed y)	SPMA: $\pi_{t+1} = \pi_t(1 + \eta A)$; FA via convex projection	Linear (tabular); neighbourhood (FA); strong empirical results
Ding et al. (2020) [Ding et al., 2020]	Lagrangian CMDP $\max_\pi \min_{\lambda \geq 0} V_r^\pi + \lambda(V_g^\pi - b)$	NPG for π, projected subgradient for λ	Dimension-free $O(1/\sqrt{T})$ gap & violation (avg.)

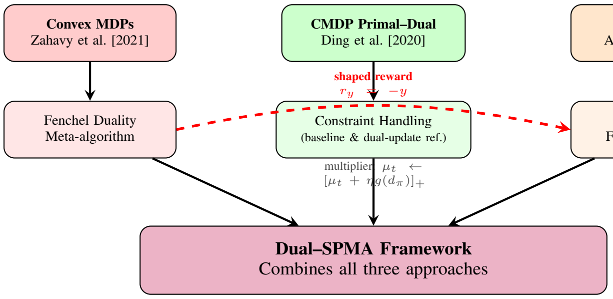

Figure 3: Overview of how the three papers inform our Dual–SPMA framework. The key insight is that Zahavy's shaped reward reduction ($r_y = -y$) enables us to use SPMA as the policy player, while Ding et al. inform our constraint handling and serve as a baseline.

4 What we will implement and measure

Method. Dual–SPMA: $y_{k+1} \leftarrow \mathrm{MA}\Big(y_k,\ \hat{d}_{\pi_k} - \nabla f^*(y_k)\Big)$; policy step: run SPMA for K_{in} epochs on r_{y_k}; return $\hat{d}_{\pi_k}$ (or $\widehat{\mathbb{E}}[\phi]$).

Metrics. (i) Saddle value $L(\pi, y)$ (when f^* known); (ii) constraint value/violation; (iii) policy return under r_y; (iv) convergence of $\|\hat{d}_\pi\|_1$ (tabular) or $\|\widehat{\mathbb{E}}[\phi]\|$ (FA); (v) wall-clock/sample efficiency. Baselines include NPG–PD.

References

R. Asad, R. B. Harikandeh, I. H. Laradji, N. L. Roux, and S. Vaswani. Fast convergence of softmax policy mirror ascent for bandits & tabular MDPs. 2024. URL `https://openreview.net/forum?id=f5OjNMXIik`.

D. Ding, K. Zhang, T. Başar, and M. R. Jovanović. Natural policy gradient primal-dual method for constrained markov decision processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

T. Zahavy, B. O'Donoghue, G. Desjardins, and S. Singh. Reward is enough for convex mdps. volume 34, pages 25746–25759, 2021.

A Discounted Occupancy Measures and Feature Expectations

This appendix provides formal definitions and estimation procedures for the discounted occupancy measure $\hat{d}_\pi$ and feature expectations $\widehat{\mathbb{E}}[\phi]$ that appear in our Dual–SPMA framework (Fig. 1).

A.1 Discounted Occupancy Measure (Tabular Setting)

Definition. For a policy π in an MDP with discount factor $\gamma \in [0, 1)$ and initial state distribution ρ, the **discounted occupancy measure** is defined as:

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

This quantity represents the discounted frequency with which policy π visits state-action pair (s, a). The normalization factor $(1 - \gamma)$ ensures that $\sum_{s,a} d_\pi(s, a) = 1$, making d_π a valid probability distribution over $\mathcal{S} \times \mathcal{A}$.

Properties. The occupancy measure uniquely characterizes a policy and satisfies:

- **Linearity of rewards:** The expected return can be expressed as $V_r^\pi(\rho) = \sum_{s,a} d_\pi(s, a)\, r(s, a) = \langle d_\pi, r\rangle$.
- **Polytope structure:** The set of all valid occupancy measures forms a convex polytope $K \subseteq \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, enabling convex optimization over policies.

Estimation. In practice, we estimate d_π from sampled trajectories. Given N trajectories of horizon H, the empirical estimate is:

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

where $\mathbb{I}[\cdot]$ is the indicator function and the normalization factor $(1-\gamma)/(1-\gamma^H)$ accounts for the finite horizon.

A.2 Feature Expectations (Function Approximation Setting)

Motivation. In large or continuous state-action spaces (e.g., robotics, Atari), tracking the full occupancy measure $d_\pi(s, a)$ becomes intractable due to the curse of dimensionality. Instead, we represent the occupancy through a finite-dimensional **feature map** $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, where $d \ll |\mathcal{S}| \times |\mathcal{A}|$.

Definition. The **feature expectation** under policy π is:

$$\mathbb{E}_{d_\pi}[\phi] = \sum_{s,a} d_\pi(s, a)\, \phi(s, a) \in \mathbb{R}^d.$$

This is a d-dimensional vector that summarizes the policy's occupancy distribution in feature space.

Shaped Reward under Features. In the Fenchel dual formulation with features, the dual variable becomes $y \in \mathbb{R}^d$ (instead of $y \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$), and the shaped reward is:

$$r_y(s, a) = -\phi(s, a)^\top y.$$

The saddle-point problem becomes:

$$\min_{\pi} \max_{y \in \mathbb{R}^d} \langle y, \mathbb{E}_{d_\pi}[\phi]\rangle - f^*(y).$$

Estimation. Given N sampled trajectories, we estimate the feature expectation as:

$$\widehat{\mathbb{E}}[\phi] = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{H-1} \gamma^t \, \phi(s_t^{(i)}, a_t^{(i)}) \cdot \frac{(1-\gamma)}{1-\gamma^H}.$$

This reduces the communication between the policy player and cost player from $|\mathcal{S}| \times |\mathcal{A}|$ dimensions to just d dimensions, making the algorithm scalable.

A.3 Role in the Dual–SPMA Algorithm

In our framework (Algorithm **??**, Fig. 1), at each iteration k:

1. The **cost player** provides a dual variable y_k that defines the shaped reward:

$$r_{y_k}(s,a) = \begin{cases} -y_k(s,a) & \text{(tabular)} \\ -\phi(s,a)^\top y_k & \text{(function approximation)} \end{cases}$$

2. The **policy player** runs SPMA with reward r_{y_k} to obtain policy π_k.
3. We estimate the occupancy from sampled trajectories:

$$\text{Estimate } = \begin{cases} \hat{d}_{\pi_k}(s,a) & \text{(tabular)} \\ \widehat{\mathbb{E}}_{d_{\pi_k}}[\phi] & \text{(function approximation)} \end{cases}$$

4. This estimate is fed back to the cost player for the next dual update:

$$y_{k+1} \leftarrow \text{MirrorAscent}\big(y_k, \hat{d}_{\pi_k} - \nabla f^*(y_k)\big).$$

Example: Apprenticeship Learning. In imitation learning, we have an expert policy with occupancy d_E (or feature expectation $\mathbb{E}_{d_E}[\phi]$). The objective is to minimize $f(d_\pi) = \|d_\pi - d_E\|_2^2$. The dual variable y_k adjusts to penalize deviations from the expert's behavior, and the policy player learns by maximizing the shaped reward r_{y_k}.

Computational Complexity.

- **Tabular:** $\mathcal{O}(|\mathcal{S}| \times |\mathcal{A}|)$ storage and communication per iteration.
- **Function approximation:** $\mathcal{O}(d)$ storage and communication, enabling scalability to large domains.

A.4 Numerical Example

Consider a simple 2-state, 2-action MDP with $\gamma = 0.9$ and uniform initial distribution. Suppose we run policy π and observe:

- State-action (s_1, a_1) visited at times $t = 0, 2, 4$ in one trajectory
- State-action (s_1, a_2) visited at time $t = 1$ in one trajectory

The empirical occupancy estimate (for a single trajectory with $H = 5$) would be:

$$\hat{d}_\pi(s_1, a_1) \approx (1-\gamma) \cdot \frac{\gamma^0 + \gamma^2 + \gamma^4}{1-\gamma^5} = 0.1 \cdot \frac{1 + 0.81 + 0.6561}{0.40951} \approx 0.601$$

In the feature approximation case with $\phi(s_1, a_1) = [1,0]^\top$ and $\phi(s_1, a_2) = [0,1]^\top$, we would compute:

$$\widehat{\mathbb{E}}[\phi] \approx (1-\gamma) \cdot \frac{[1,0] \cdot \gamma^0 + [0,1] \cdot \gamma^1 + [1,0] \cdot \gamma^2 + [1,0] \cdot \gamma^4}{1-\gamma^5}.$$

Appendix A: What Figure 1 Shows and Where It Comes From

A.1 Notation and objects in the diagram

- $\pi(a|s)$: stochastic policy; $\gamma \in (0,1)$: discount.
- **Discounted occupancy** of (s,a) under π: $d_\pi(s,a) = (1-\gamma) \sum_{t \geq 0} \gamma^t \, \Pr_\pi(s_t = s, a_t = a)$.
- $f : \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|} \to \mathbb{R}$: convex objective on occupancies (task-specific).

- f^*: Fenchel conjugate of f; $\langle u, v\rangle = \sum_{s,a} u(s,a)v(s,a)$.
- **Dual/cost variable** y (same dimension as costs or feature weights).
- **Environment** P: transition kernel; emits trajectories when run with a policy and reward.
- r_y: **shaped reward** induced by the current y (defined below).
- $\hat{d}_{\pi_k}$: an estimate of d_{π_k} (tabular counts) or $\widehat{\mathbb{E}}_\pi[\phi]$ with features $\phi(s,a)$.

A.2 From convex MDP to the saddle (why the cost player exists)

Given the convex CMDP objective

$$\min_{d\in\mathcal{K}} f(d) \quad \text{over occupancies } \mathcal{K},$$

Fenchel–Moreau gives the **convex–concave saddle**:

$$\min_{\pi}\max_{y}\ L(\pi, y) \quad \text{with} \quad L(\pi, y) = \langle y, d_\pi\rangle - f^*(y). \tag{1}$$

For a fixed y, the policy subproblem is *standard RL* with the shaped reward

$$r_y(s,a) = -\,y(s,a) \quad \text{(tabular)} \qquad \text{or} \quad r_y(s,a) = -\,\phi(s,a)^\top y \quad \text{(features)}. \tag{2}$$

This explains the top arrow in Fig. 1 ("shaped reward $r_{y_k} = -y_k$").

A.3 Dual/cost update (why the dashed feedback arrow exists)

Optimizing (1) in y requires a gradient (or follow-the-leader) step using the current occupancy:

$$\textbf{OMD/gradient ascent:} \qquad y_{k+1} = y_k + \eta_2\big(\underbrace{\hat{d}_{\pi_k}}_{\text{from the policy box}} - \nabla f^*(y_k)\big), \tag{3}$$

$$\textbf{FTL/meta:} \qquad \overline{d}_k = \tfrac{1}{k+1}\sum_{t=0}^{k} \hat{d}_{\pi_t}, \qquad y_{k+1} = \nabla f(\overline{d}_k). \tag{4}$$

Either choice justifies the dashed arrow "$\hat{d}_{\pi_k}/\widehat{\mathbb{E}}[\phi]$" feeding the cost player.

A.4 Policy player (SPMA) on the shaped reward

Given y_k, construct r_{y_k} via (2) and evaluate advantages $A^{\pi_k}_{r_{y_k}}(s,a) = Q^{\pi_k}_{r_{y_k}}(s,a) - V^{\pi_k}_{r_{y_k}}(s)$.

Tabular SPMA update. For stepsize $\eta_1 \le 1-\gamma$,

$$\pi_{k+1}(a|s) = \pi_k(a|s)\,\big(1 + \eta_1\, A^{\pi_k}_{r_{y_k}}(s,a)\big) \quad \text{for all } (s,a), \tag{5}$$

which needs no explicit normalization and yields linear convergence in tabular MDPs (for bounded rewards).

Function approximation (FA). Form the *tabular target* $\tilde{\pi}_{k+1}$ via (5) and *project* onto the policy class $\{\pi_\theta\}$ by the convex fit

$$\theta_{k+1} = \arg\min_\theta \sum_s \hat{d}_{\pi_k}(s)\,\mathrm{KL}\big(\tilde{\pi}_{k+1}(\cdot|s)\,\|\,\pi_\theta(\cdot|s)\big), \tag{6}$$

giving linear convergence to a neighbourhood determined by approximation/estimation error.

A.5 Estimating $\hat{d}_\pi$ (what the dashed arrow carries)

- **Tabular:** count discounted visits along rollouts: $\hat{d}_\pi(s,a) = (1-\gamma)\sum_{t=0}^{T}\gamma^t\mathbf{1}\{s_t = s, a_t = a\}$ (average over episodes).
- **Features:** accumulate feature expectations: $\widehat{\mathbb{E}}_\pi[\phi] = (1-\gamma)\sum_{t=0}^{T}\gamma^t\,\phi(s_t, a_t)$.

Either statistic is sufficient for (3)–(4).

A.6 Optional: explicit constraints (CMDPs)

If constraints $g_i(d_\pi) \le 0$ are present, augment (1) with multipliers $\mu_i \ge 0$:

$$L(\pi, y, \mu) = \langle y, d_\pi\rangle - f^*(y) \;+\; \sum_i \mu_i\, g_i(d_\pi).$$

Update multipliers by projected ascent $\mu_i^{k+1} = \big[\mu_i^k + \eta_2\, g_i(\hat{d}_{\pi_k})\big]_+$, and fold linear (or linearized) constraint costs into the shaped reward $r_{y_k} \leftarrow -\big(y_k + \sum_i \mu_i^{k+1} c_i\big)$ with $c_i = \nabla g_i(d_{\pi_k})$ if needed.

A.7 Practical settings and stability

- **Reward scaling:** clip/scale r_{y_k} to $[-1, 1]$ (or $[0, 1]$) to keep (5) valid.
- **Stepsizes:** tabular SPMA stable with $\eta_1 \leq 1 - \gamma$; tune η_2 for the dual.
- **Averaging:** report averaged policies/occupancies $\bar{\pi} = \frac{1}{K} \sum_{k=0}^{K-1} \pi_k$ as in the meta-algorithm.

A.8 One-iteration pseudocode (for Fig. 1)

1. **Input:** current dual y_k, policy π_k.
2. **Shaped reward:** build r_{y_k} via (2).
3. **Policy step (SPMA):** run (5) (tabular) or (6) (FA) using $A^{\pi_k}_{r_{y_k}}$ to obtain π_{k+1}.
4. **Estimate occupancy:** roll out π_{k+1} in P; compute $\hat{d}_{\pi_{k+1}}$ or $\widehat{\mathbb{E}}_{\pi_{k+1}}[\phi]$.
5. **Dual step:** update y_{k+1} using (3) or (4) (and μ if constraints).

A.9 Symbol quick-reference (for readers)

y	dual/cost variable (cost player)
r_y	shaped reward induced by y
P	environment dynamics (emits trajectories under π)
π	policy (policy player)
$\hat{d}_\pi$, $\widehat{\mathbb{E}}[\phi]$	discounted occupancy / feature expectations sent to the cost player