WHO BEARS THE BURDEN? LEARNING RESPONSIBILITY FOR SHARED CONSTRAINTS IN MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

When multiple agents share a cost budget, a common Lagrange multiplier can enforce the aggregate constraint but does not determine how its penalty should be allocated across agents. Uniform penalties ignore heterogeneity in the rewards agents sacrifice, while agent-specific multipliers may still rely on the same aggregate cost signal. We introduce ***L**agrang**i**an **R**esponsibility **A**llocation* (LiRA), which learns each agent's share of a common multiplier by optimizing social welfare over a finite training horizon. The multiplier enforces the aggregate budget, while responsibility shares redistribute its influence without modifying the original rewards or constraints. For convex games under standard regularity conditions, varying these shares induces a smooth family of normalized generalized Nash equilibria in which active constraints remain at their budgets while welfare varies. To optimize responsibility before convergence, we derive a welfare gradient that accounts for both learning updates and the induced change in data distribution. Across CityLearn, MABIM, Harvest, and MetaDrive, spanning 3 to 400 agents, LiRA improves average social welfare by up to 29% over uniform and agent-specific multiplier baselines. Grid and driving costs remain within budget, inventory violations decrease, and Harvest makes more effective use of available budget.

1 INTRODUCTION

Consider two delivery robots approaching a narrow corridor. One carries an urgent package, while the other is traveling to collect a package. Either robot may yield to avoid a collision, but delaying the loaded robot incurs a larger loss. A shared collision-avoidance constraint excludes unsafe outcomes, but does not determine how the burden of satisfying the constraint should be allocated between the robots. This ambiguity arises more broadly in multi-agent systems subject to shared constraints, including power limits, resource budgets, and safety requirements, when agents incur heterogeneous costs for adapting their behavior.

A shared Lagrange multiplier specifies the strength of the penalty associated with a common constraint, but does not determine how that penalty should be apportioned across agents. Applying the same penalty to all agents leaves this allocation implicit. Introducing agent-specific multipliers does not resolve the issue when each multiplier is driven by the same aggregate constraint cost. We instead make the allocation of constraint responsibility explicit. Specifically, a responsibility share determines the fraction of the shared penalty applied to each agent's objective, while the common multiplier continues to enforce the aggregate constraint. The original rewards and constraint budget remain unchanged. Responsibility allocation therefore modifies which agents are incentivized to adapt without altering the underlying constrained problem.

This additional degree of freedom induces a nontrivial equilibrium-selection problem. Under convexity, monotonicity, and standard regularity conditions, varying the responsibility shares generates a smooth family of normalized generalized Nash equilibria. Along this family, each active constraint remains satisfied at its budget, while social welfare may vary (Theorem 1). Motivated by this observation, we introduce ***L**agrang**i**an **R**esponsibility **A**llocation* (LiRA), which optimizes responsibility shares to select among feasible equilibria according to social welfare.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

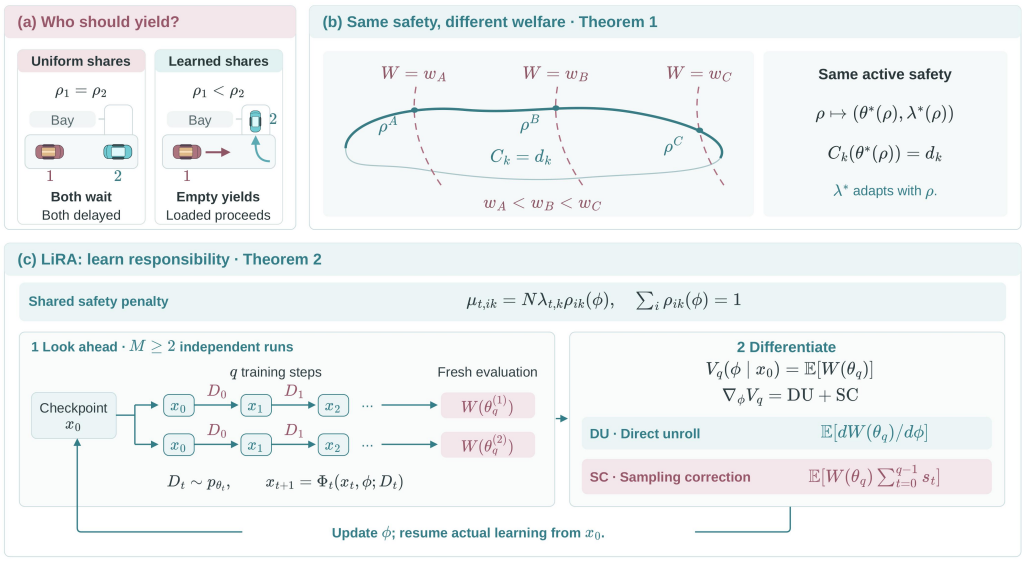

Figure 1: **Overview of LiRA. (a)** In the illustrated corridor scenario, uniform shares leave both robots waiting; learned shares let the empty robot yield so that the loaded robot proceeds. Both outcomes avoid collision but differ in welfare. **(b)** Under Theorem 1's conditions, varying responsibility ρ selects a smooth equilibrium family. For each active constraint k, cost C_k stays at its budget d_k, while welfare W can vary and the shared equilibrium multiplier λ^* adapts to ρ. **(c)** From checkpoint x_0, LiRA runs $M \geq 2$ independent lookaheads of q learner updates, each followed by fresh evaluation. It combines direct unrolling (DU) with sampling correction (SC) to estimate $\nabla_\phi W$, update ϕ, and restart from x_0.

LiRA does not require the underlying agents to converge to equilibrium before responsibility allocation can be optimized. Instead, it evaluates a candidate allocation through the welfare obtained after a finite lookahead of the learning dynamics. Changing the responsibility shares influences both the policy updates within this lookahead and the distribution of data collected for subsequent updates. We derive a gradient estimator that accounts for both effects by differentiating through the learning dynamics and correcting for the induced change in the sampling distribution (Theorem 2). Independent lookaheads initialized from a common training checkpoint allow this gradient to be estimated without modifying the primary learning trajectory. Figure 1 summarizes the resulting framework, from responsibility allocation in the motivating corridor example to its implementation within the learning procedure.

We test LiRA on four multi-agent tasks, including building energy management in CityLearn (Nweye et al., 2025) with 3 agents, inventory control in MABIM (Yang et al., 2023) with 400 agents, resource harvesting in Harvest (Beattie et al., 2020) with 7 agents, and autonomous driving in MetaDrive (Li et al., 2023) with 4 agents. LiRA has the highest mean welfare in all four tasks (Section 4.1). It keeps grid costs within budget in CityLearn, reduces safety violations in MetaDrive, and cuts order-rejection penalties in MABIM. A matched CityLearn episode (Section 4.2) traces the gain to individual buildings: the two that overcool receive a larger share of the grid-electricity penalty, cool less, and recover comfort while district electricity falls.

2 Related Work

Shared safety in multi-agent learning. Constrained reinforcement learning trades reward against explicit cost budgets (Altman, 1999). Single-agent methods enforce constraints via trust regions (Achiam et al., 2017; Yang et al., 2020b) or Lagrangian penalties (Tessler et al., 2019; Stooke et al., 2020). With multiple agents, one learner's update shifts the feasible set for the others. Constrained policy optimization and decentralized descent–ascent address these coupled dynamics (Gu et al.,

2021; Lu et al., 2021), and regret and violation bounds exist for certain two-player settings (Ding et al., 2023). All of these methods enforce the shared cost, but none asks which agents should absorb it. LiRA keeps the shared cost feedback and adds that distributional choice.

Equilibria under shared constraints. When agents share constraints, the feasible set of each depends on the others' choices. Generalized Nash equilibria formalize such interdependence (Facchinei & Kanzow, 2010). Rosen's normalized equilibria fix the ratios between agents' constraint multipliers (Rosen, 1965) and parameterized variational inequalities trace the resulting family (Nabetani et al., 2011; Dreves, 2018). Recent safe-learning work studies convergence to such equilibria under state-wise constraints (Li & Azizan, 2024). Our responsibility shares set the multiplier ratios directly. Welfare then selects among the equilibria they index.

Learning through the agents' response. Varying allocations can improve welfare at equilibrium, but agents are still learning when shares must be chosen. Incentive design modifies rewards in anticipation of agents' adaptive response (Yang et al., 2020a; 2022; Zheng et al., 2022). Bilevel formulations and differentiable equilibrium blocks offer complementary approaches (Thoma et al., 2024; Shen et al., 2024; Thoma et al., 2026). LiRA differs in that it allocates the penalty of an existing constraint rather than designing new rewards. Its gradient combines meta-gradient differentiation through learner updates (Xu et al., 2018; Franceschi et al., 2017; Grefenstette et al., 2019) with a sampling correction for the data collected by intermediate policies (Vuorio et al., 2022). Iterative and implicit hypergradients connect finite-window responses to equilibrium sensitivity (Grazzi et al., 2020).

3 METHOD

The Lagrangian penalty for a shared constraint has a scale, set by the multiplier, and a distribution across agents, usually left at $1/N$. LiRA adds a learnable responsibility vector that controls the distribution as the multiplier continues to track aggregate cost. We formalize this allocation, show that it selects among equilibria with different welfare at the same active costs, and derive a gradient for choosing shares from a finite training window.

3.1 RESPONSIBILITY ALLOCATION

Consider a Markov game with N agents and K shared constraints. Agent i uses policy $\pi_{\theta_i}(a_i \mid o_i)$, where a_i is the action and o_i is the local observation. The joint policy factorizes across agents and induces trajectory distribution p_θ, where $\theta := [\theta_i]_{i=1}^N$. On a trajectory τ of length H, let $r_{i,h}$ denote agent i's reward and $c_{k,h}$ the shared cost for constraint k at step h. With discount factor $\gamma \in [0, 1]$ and budget d_k, define

$$J_i(\theta) = \mathbb{E}_{\tau\sim p_\theta}\left[\sum_{h=0}^{H-1} \gamma^h r_{i,h}\right], \qquad C_k(\theta) = \mathbb{E}_{\tau\sim p_\theta}\left[\sum_{h=0}^{H-1} \gamma^h c_{k,h}\right] \le d_k. \tag{1}$$

We measure social welfare using the original rewards, $W(\theta) = \sum_i \alpha_i J_i(\theta)$, with fixed positive weights α_i.

LiRA introduces a responsibility allocation. Let $\rho_{ik}(\phi)$ be agent i's share of the penalty for constraint k, with $\rho_{ik} \ge 0$ and $\sum_i \rho_{ik} = 1$. At learner update t, the shared multiplier $\lambda_{t,k} \ge 0$ sets the overall penalty scale. Agent i receives coefficient $\mu_{t,ik} = N\lambda_{t,k}\rho_{ik}(\phi)$ and trains on

$$r^{\text{eff}}_{i,h;t} = r_{i,h} - \sum_{k=1}^{K} \mu_{t,ik} c_{k,h}. \tag{2}$$

Equal shares, $\rho_{ik} = 1/N$, recover the usual Lagrangian coefficient $\mu_{t,ik} = \lambda_{t,k}$. A larger ρ_{ik} makes constraint k matter more to agent i's update. It does not alter the reward, cost, or budget used to evaluate the resulting policies. We optimize unconstrained logits ϕ_{ik} and map them to these nonnegative shares with a softmax. A small positive floor can keep every agent exposed to the shared penalty. Appendix C.3 gives the exact map.

The multiplier retains the standard Lagrangian primal–dual update from constrained reinforcement learning (Altman, 1999; Tessler et al., 2019; Stooke et al., 2020). At a scheduled dual update, let $\widehat{C}_{t,k}$ be the estimated cost on the same scale as its budget d_k, let $\beta_\lambda > 0$ be the dual step size, and let $[z]_+ = \max\{z, 0\}$. The shared multiplier follows

$$\lambda_{t+1,k} = [\lambda_{t,k} + \beta_\lambda(\widehat{C}_{t,k} - d_k)]_+. \tag{3}$$

Each multiplier sees aggregate cost, regardless of who receives its penalty. The factor N in $\mu_{t,ik}$ ensures that equal shares recover the original coefficient, so varying ρ redistributes pressure rather than rescaling it uniformly. We next show that this redistribution can change welfare even when an active constraint stays at its budget.

3.2 Welfare varies along the cost surface

In Figure 1(a), either robot can yield. A feasible outcome need not be the best one for their combined rewards. The constraint says what is forbidden, not who should give way. Generalized Nash equilibria formalize such constrained best responses (Facchinei & Kanzow, 2010). Among them, a normalized equilibrium fixes the ratios of agents' constraint multipliers (Rosen, 1965). The responsibility shares set exactly those ratios, so changing ρ selects a different equilibrium under the same shared constraint.

To make this precise, consider a positive responsibility vector ρ shared by all constraints. Let $\mathcal{R}$ be an interior region of the simplex, $\theta^*(\rho)$ the selected normalized equilibrium, and $\lambda^*(\rho)$ its shared multipliers. Let $\mathcal{A}$ denote the active shared constraints. The regularity assumptions are collected in Appendix A.1.

Theorem 1 (Equilibrium selection at fixed active safety)**.** *Under assumptions (A1)–(A3) in Appendix A.1, each $\rho \in \mathcal{R}$ selects a unique normalized equilibrium, and $\theta^*(\rho)$ and $\lambda^*(\rho)$ vary smoothly with ρ. Every active shared cost remains at its prescribed budget:*

$$C_k(\theta^*(\rho)) = d_k \qquad (k \in \mathcal{A}). \tag{4}$$

If welfare has a nonzero derivative along the simplex at ρ, then nearby responsibility allocations attain different welfare at those same active costs.

Proof can be found in Appendix A.2.

The mechanism is visible in the agents' first-order conditions. A larger share raises the constraint's marginal price for one agent relative to another. When their marginal rewards for yielding differ, the joint policy shifts along the active cost surface. Total cost stays at the budget because the shared multiplier compensates, but the agents divide the adjustment differently (Figure 1(b)). The theorem makes this precise through the nonzero welfare derivative. It does not guarantee that every direction improves welfare, only that improvement is generically available.

Appendix A.3 extends the result locally to separate shares for each constraint. During training, however, the agents have not settled at an equilibrium, and we need to choose an allocation from the response they actually produce.

3.3 Choosing shares during training

Theorem 1 identifies the opportunity at equilibrium. To realize it during training, LiRA evaluates a proposed allocation by the welfare it produces after a finite number of updates, making the training process itself the object we differentiate.

Let x_t collect everything needed to resume training at update t, including the policies, critics, optimizer state, and shared multipliers. Let $\mathcal{D}_t$ be the batch collected at that update, and let Φ_t denote one differentiable lookahead step. Then

$$\mathcal{D}_t \sim p_{\theta_t}(\mathcal{D}), \qquad x_{t+1} = \Phi_t(x_t, \phi; \mathcal{D}_t). \tag{5}$$

The lookahead map may update the actors, critics, optimizer state, and shared multipliers according to the chosen probe. It needs only be differentiable along a sampled training window. Our experiments use proximal policy optimization (PPO) (Schulman et al., 2017).

Starting from checkpoint x_0, let q be the lookahead horizon. Holding the allocation fixed for these q updates gives the finite-training objective

$$\max_{\phi} V_q(\phi \mid x_0), \qquad V_q(\phi \mid x_0) = \mathbb{E}_{\mathcal{D}_{0:q-1}}[W(\theta_q)]. \tag{6}$$

The objective measures welfare after the next q updates under the proposed shares, using the agents' original rewards.

Changing a share affects an update on a fixed training batch. The new policy then collects a different batch for the following update. To account for this second path, define the training-batch score

$$s_t = \frac{\mathrm{d}}{\mathrm{d}\phi} \log p_{\theta_t}(\mathcal{D}_t), \tag{7}$$

where the total derivative follows the recorded learner updates for a fixed realized training history.

The distinction first appears after an update has changed the policy. With x_0 fixed, the initial batch is collected before the proposed allocation has changed any policy. A second batch can reflect the allocation through the first update. A two-update lookahead is therefore the shortest window that exposes both the direct update and the changed-data path. Longer windows follow more of the ensuing response but also require more sampled updates.

Theorem 2 (Finite-training responsibility gradient)**.** *Suppose x_0 is fixed and the learner and sampling laws satisfy the regularity conditions in Appendix B. Then*

$$\nabla_\phi V_q = \underbrace{\mathbb{E}\left[\frac{\mathrm{d}W(\theta_q)}{\mathrm{d}\phi}\right]}_{g^{\mathrm{DU}}} + \underbrace{\mathbb{E}\left[W(\theta_q)\sum_{t=0}^{q-1} s_t\right]}_{g^{\mathrm{SC}}}. \tag{8}$$

Proof can be found in Appendix B.

The direct-unrolling (DU) term differentiates the recorded updates at fixed data. Its complement, the sampling-correction (SC) term, accounts for the data collected by intermediate policies (Yang et al., 2020a; 2022; Vuorio et al., 2022). Policy scores let us estimate the two terms without a simulator derivative. Appendix B gives the derivation.

3.4 Estimating the LiRA update

The gradient in Eq. (8) can be estimated without committing any proposed update to the live learner. From its current checkpoint, LiRA runs $M \geq 2$ independent continuations, each for q updates at the same allocation. For continuation m, let $\widehat{W}_m$ be its welfare on fresh terminal rollouts, $\widehat{g}_m^{\mathrm{DU}}$ the direct-response estimate, and $s^{(m)} = \sum_{t=0}^{q-1} s_t^{(m)}$ the score of its training history.

A lookahead of q steps sets the response used to choose responsibility. After that, h live updates advance the learner before the next choice. Update schedules for the two phases can differ. Appendix D.1 specifies the CityLearn probe and live schedules.

For its sampling term, continuation m uses the other continuations' mean welfare as a baseline,

$$b_m = \frac{1}{M-1}\sum_{j \neq m} \widehat{W}_j. \tag{9}$$

This gives

$$\widehat{g} = \frac{1}{M}\sum_{m=1}^{M}\left[\widehat{g}_m^{\mathrm{DU}} + (\widehat{W}_m - b_m)s^{(m)}\right]. \tag{10}$$

Because b_m is independent of the score from continuation m, centering preserves the expected gradient (Appendix B.5). For an exact differentiable lookahead, Eq. (10) is unbiased. In CityLearn we use the stopped-batch approximation described in Appendix C.2, while holding the shared multiplier fixed over the short probe to isolate responsibility redistribution at the current dual scale.

Terminal welfare is estimated from the original rewards on fresh rollouts. Its direct gradient follows the final policy's score back through the recorded learner updates. The history score accumulates the policy scores of batches collected during the lookahead. Reusing a training batch for multiple PPO epochs does not create new sampled actions in that history. These distinctions keep the two terms in Eq. (10) tied to the same finite response.

Algorithm 1 LiRA: learning responsibility from finite continuations

Require: Checkpoint x_0, allocation logits ϕ_0, outer cycles L, lookahead q, independent continuations $M \geq 2$, live updates per cycle h, step size η_ρ.

for $\ell = 0, \ldots, L-1$ **do**
  for $m = 1, \ldots, M$ **do**
    Restore $x_0^{(m)} \leftarrow x_\ell$ and set $s^{(m)} \leftarrow 0$.
    for $t = 0, \ldots, q-1$ **do**
      Collect a fresh batch $\mathcal{D}_t^{(m)}$ with the policy in $x_t^{(m)}$.
      $s^{(m)} \leftarrow s^{(m)} + s_t^{(m)}$, the batch score in Eq. (7).
      $x_{t+1}^{(m)} \leftarrow \Phi_t(x_t^{(m)}, \phi_\ell; \mathcal{D}_t^{(m)})$.
    end for
    On fresh terminal rollouts, estimate $\widehat{W}_m$ and $\widehat{g}_m^{\mathrm{DU}}$.
  end for
  Compute b_m for each m and $\widehat{g}$ by Eq. (10).
  Ascend ϕ_ℓ along $\widehat{g}$ with step size η_ρ to obtain $\phi_{\ell+1}$.
  Restore x_ℓ; train the live learner for h fresh updates under $\rho(\phi_{\ell+1})$, obtaining $x_{\ell+1}$.
end for
return the final policy in x_L and $\rho(\phi_L)$.

The continuations are discarded after the allocation step. Only the fresh live updates produce the next checkpoint, including its optimizer and dual state. One outer cycle thus takes Mq lookahead learner updates and h live updates, plus terminal rollouts. Appendix C specifies the lookahead implementation.

4 Experiments

The theory above assumes convex games with smooth payoffs. We now test whether LiRA produces welfare gains when agents train with nonlinear function approximators for a finite time. Four tasks test welfare and cost across different scales and constraint types. CityLearn then lets us inspect individual agents. Throughout, we read costs alongside rewards to distinguish gains that spend unused budgets from gains at nearly the same shared cost.

4.1 Welfare and shared costs across tasks

The shared penalty makes different decisions in our four tasks. In CityLearn v2 (Nweye et al., 2025), three buildings control their energy devices for thermal comfort. They draw electricity from the same district, so their combined net demand faces one grid cap: reducing demand in one building can relieve a cost borne by all three. In MABIM (Yang et al., 2023), 400 product agents manage replenishment across an inventory network to maximize total profit, subject to shared order-rejection caps across two warehouse-store links. An allocation now distributes pressure across hundreds of decisions and two distinct constraints. Harvest on DeepMind Lab2D (Beattie et al., 2020) has seven agents collecting a common resource under one shared cost. MetaDrive (Li et al., 2023) has four driving agents and a shared limit on the rate of safety events. These tasks differ in scale and in what an agent gives up to reduce cost. In all four benchmarks, we set $\alpha_i = 1$ for every agent, so welfare is the sum of agents' original returns.

We compare LiRA with two baselines. Uniform responsibility sets $\rho_{ik} = 1/N$, and each agent receives an equal share of each shared multiplier. The per-agent Lagrangian baseline (PAL) instead learns a separate multiplier for each agent. LiRA keeps a shared multiplier and learns its responsibility shares. Within each task, methods use the same inner learner and matched training seeds

under a common evaluation protocol. Table 1 gives held-out means and sample standard deviations over three training seeds, or six for MetaDrive. Rewards and costs retain their native units: welfare is compared within each task, and each cost against its own budget. MABIM's rejection budgets are calibrated on validation data, with eight matched action seeds in testing. MetaDrive starts each method from the same actor checkpoint and uses 13 held-out episode seeds. Appendix D.1 gives further details of the protocol.

Table 1: **Welfare and shared costs across four tasks.** Held-out mean ± sample standard deviation over matched training seeds (three per task; six for MetaDrive). Costs and budgets use the units shown in each row. Bold marks the highest mean welfare.

Task (N, K)	Outcome	Budget	Uniform	PAL	LiRA
CityLearn $(3, 1)$	Welfare (10^3)	–	-16.87 ± 3.04	-16.87 ± 3.04	$\mathbf{-14.51} \pm 1.66$
	Grid excess (kWh)	28.76	28.44 ± 5.97	28.44 ± 5.97	28.73 ± 7.48
MABIM $(400, 2)$	Welfare (10^6)	–	-596.31 ± 5.00	-597.27 ± 3.76	$\mathbf{-591.36} \pm 4.48$
	Rejections C_0 (10^6)	20.35	21.51 ± 0.25	21.56 ± 0.23	21.30 ± 0.18
	Rejections C_1 (10^3)	32.69	27.71 ± 2.31	27.54 ± 2.21	28.68 ± 0.90
Harvest $(7, 1)$	Welfare	–	42.73 ± 12.22	44.73 ± 3.79	$\mathbf{50.27} \pm 2.53$
	Shared cost	68.00	33.24 ± 10.08	33.38 ± 5.18	36.82 ± 2.40
MetaDrive $(4, 1)$	Welfare	–	49.72 ± 22.20	60.94 ± 13.45	$\mathbf{64.26} \pm 19.98$
	Events / 10^3 steps	1.00	1.017 ± 0.343	1.337 ± 1.466	0.890 ± 0.407

LiRA leads in mean welfare on all tasks (Table 1). The welfare ranking is consistent across domains, but the cost response differs in each.

CityLearn. Uniform and PAL both return -16.87k in mean comfort reward. PAL starts with equal multipliers and receives identical district-cost feedback, so its multipliers stay equal. LiRA raises the welfare to -14.51k, and the mean grid exceedance moves from 28.44 to 28.73 kWh, against a budget of 28.76-kWh. The gain comes at nearly the same district cost. Section 4.2 asks what the buildings do differently to obtain it.

MABIM. With 400 agents and two constraints, LiRA reaches -591.36 million versus Uniform's -596.31 million. It also reduces the more strained rejection cost from 21.51 to 21.30 million. Both still exceed the budget of 20.35-million, so the welfare gain accompanies a cost improvement rather than a cost trade.

Harvest. The shared cost is well within budget: 33–37 versus 68. The welfare rises from 42.73 (Uniform) to 50.27 using the unused allowance.

MetaDrive. LiRA reaches 64.26 in welfare (Uniform: 49.72; PAL: 60.94) and records 0.890 safety events per thousand steps, below both the 1.00 limit and Uniform's 1.017. PAL improves welfare, but at 1.337 events. Learning shares of a common penalty outperforms individually learned penalties in both welfare and safety alike.

4.2 What the buildings do differently

Each CityLearn building earns a thermal-comfort reward. The shared cost accumulates the positive excess of district net electricity over a per-step grid cap across 719 steps, with a budget of 28.76-kWh. For an overcooled building, easing cooling serves both objectives at once. It saves electricity and improves comfort. For a building that is already too warm, the comfort effect reverses.

Figure 2 follows a matched 719-step evaluation under Uniform and LiRA. It folds the temperatures into hour-of-day means, while Table 2 reports the allocations and totals from those same episodes. The shaded 16–18h price peak gives context for the daily response. The cost of shared grid-cap is accumulated throughout the whole evaluation.

Under Uniform, Building 1 is below the comfort band for 314 steps and Building 2 for 232. Building 3 is above it for 318. The first two buildings are overcool, while the third tends to run warm.

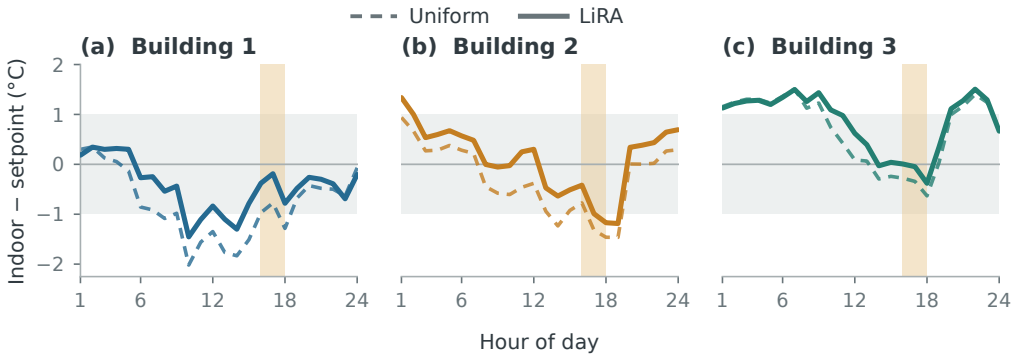

Figure 2: **Hourly temperature relative to setpoint in CityLearn.** Each point averages observations at that hour in the matched 719-step seed-1102 evaluation. Uniform is dashed; LiRA is solid. Gray marks the ± 1°C comfort band, and tan the 16–18h price peak. The episode totals appear in Table 2.

Table 2: **CityLearn responsibility and episode outcomes.** Seed-1102 evaluation over 719 steps. Uniform assigns $\rho = 1/3$ to each building; changes are LiRA minus Uniform. Reward is native building comfort reward.

	Learned ρ	Electricity change (kWh)	Reward change (10^3)
Building 1	0.395	−37.6	+4.43
Building 2	0.387	−32.7	+0.48
Building 3	0.218	−18.9	−0.03

LiRA moves Building 1's daytime temperatures toward the comfort band. Building 2, coldest late in the day, also warms: in 18h at the marked peak, its mean is 1.17°C below the setpoint instead of 1.46°C. Building 3 begins above the band, so warming it would offer little relief. The same district cost signal reaches buildings with different thermal needs.

The mean cooling action drops from 0.197 to 0.184 in Building 1 and from 0.312 to 0.287 in Building 2. Building 3 also cools less (0.180 to 0.169), but starts from a warmer state.

The learned shares put more responsibility on the overcooled buildings: $\rho = 0.395$ for Building 1 and 0.387 for Building 2, against 0.218 for Building 3 and $1/3$ each under Uniform.

Episode electricity use drops by 37.6 and 32.7 kWh for Buildings 1 and 2, and their cold steps drop to 269 and 164. However, the comfort gains differ: 4.43k versus 0.48k. Building 1 also has much higher baseline cold discomfort (586.1 versus 214.5 °C-steps), so the reduction in cold discomfort more than offsets the small increase in warm discomfort. Building 3 uses 18.9 kWh less, with almost no change in comfort reward (−0.03k).

Together, these building-level changes increase episode welfare by 4.88k as district cap exceedance falls from 25.42 to 23.78 kWh. (The three electricity reductions sum to 89.2 kWh. The smaller change in cap exceedance reflects that shared cost counts only electricity above the cap at each step.) The welfare improvement in this matched episode arises because buildings for which reduced cooling has a smaller impact on comfort and a greater benefit for grid feasibility receive more responsibility and adjust their behavior accordingly. Appendix D.2 provides the reward breakdown.

4.3 Sensitivity of the outer loop

The shares are learned from a finite training response, so their quality depends on the outer-loop design: how far ahead to look (q), how many independent lookaheads to average (M), and how fast the shares move (η_ρ). We vary each one at a time around the CityLearn setting used in Table 1, keeping the other two fixed, with three matched seeds and $h = 10$ live PPO updates per cycle.

Table 3: **CityLearn local search around the main setting.** Each row changes one parameter from $(q, M, \eta_\rho) = (2, 8, 0.05)$. Welfare and grid excess are mean $\pm$ sample standard deviation over three matched seeds. The grid-excess budget is 28.76 kWh.

Varied choice	Setting	Welfare (10^3)	Grid excess (kWh)
Main setting	$(2, 8, 0.05)$	$\mathbf{-14.51} \pm 1.66$	28.73 ± 7.48
Lookahead	$q = 3$	-18.13 ± 4.42	29.34 ± 4.67
	$q = 4$	-19.03 ± 8.39	31.52 ± 4.20
Replicates	$M = 4$	-16.02 ± 4.05	28.91 ± 4.02
	$M = 12$	-16.68 ± 3.87	29.12 ± 2.38
Step size	$\eta_\rho = 0.025$	-20.67 ± 7.14	32.97 ± 5.65
	$\eta_\rho = 0.10$	-15.78 ± 3.76	32.75 ± 12.50

The center $(q, M, \eta_\rho) = (2, 8, 0.05)$ gives the highest mean welfare among these local choices (-14.51k), with 28.73 kWh of grid excess against a 28.76-kWh budget. Looking three or four steps ahead instead of two yields -18.13k or -19.03k, without reducing grid excess. Four or twelve independent lookaheads also give lower mean welfare than eight, at similar mean costs.

The responsibility step matters most in this sweep. Leaving it at 0.025 gives -20.67k and 32.97 kWh; doubling it to 0.10 gives -15.78k and 32.75 kWh. Neither neighbor improves the center's welfare–cost outcome. This setting produces the district-level gain in Table 1 and the building responses in Section 4.2.

5 Conclusion

A shared cost limit constrains what agents may do collectively but leaves open who adjusts. LiRA makes this division learnable. At equilibrium, responsibility shares can change welfare even as active costs stay in their budgets. The finite-training gradient extends this to nonconvergent learners by estimating welfare's response to a proposed allocation through independent lookaheads. Across four tasks, the learned shares give the highest mean welfare among the compared methods. In CityLearn, the welfare gain traces to individual buildings: the two that overcool receive a larger penalty share, cool less, and recover comfort while district electricity falls.

6 Limitations

Theorem 1 requires own-concave payoffs, convex shared costs, and a strongly monotone weighted payoff field with a fixed active-constraint set. Deep policy networks violate these conditions, and our finite-training gradient (Theorem 2) does not guaranty welfare improvement in such settings by itself. Closing this gap, for instance, by establishing local improvement guarantees for nonconvex learners under shared constraints, is a substantial open problem. The responsibility shares are also constant across states; every state sees the same allocation. A state-dependent $\rho(s)$ could, for example, shift responsibility during peak-demand hours, but it turns the outer problem from optimizing a point on the simplex into learning a function, and the welfare gradient would need to track how the allocation itself reshapes the state distribution.

AI Use Statement

We used generative AI tools to assist with the code implementation of the method, the design and execution of experiments, the organization of experimental data, and manuscript polishing. The authors take responsibility for the final content.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31, 2017. URL https://proceedings.mlr.press/v70/achiam17a.html.

Eitan Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999. URL https://www-sop.inria.fr/members/Eitan.Altman/PAPERS/h.pdf.

Charles Beattie, Thomas Köppe, Edgar A. Duéñez-Guzmán, and Joel Z. Leibo. DeepMind Lab2D. *arXiv preprint arXiv:2011.07027*, 2020. URL https://arxiv.org/abs/2011.07027.

Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo Jovanovic. Provably efficient generalized lagrangian policy optimization for safe multi-agent reinforcement learning. In *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 315–332, 2023. URL https://proceedings.mlr.press/v211/ding23a.html.

Axel Dreves. How to select a solution in generalized Nash equilibrium problems. *Journal of Optimization Theory and Applications*, 178:973–997, 2018. URL https://doi.org/10.1007/s10957-018-1327-0.

Francisco Facchinei and Christian Kanzow. Generalized Nash equilibrium problems. *Annals of Operations Research*, 175:177–211, 2010. URL https://doi.org/10.1007/s10479-009-0653-x.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1165–1173, 2017. URL https://proceedings.mlr.press/v70/franceschi17a.html.

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3748–3758, 2020. URL https://proceedings.mlr.press/v119/grazzi20a.html.

Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. Generalized inner loop meta-learning. *arXiv preprint arXiv:1910.01727*, 2019. URL https://arxiv.org/abs/1910.01727.

Shangding Gu, Jakub Grudzien Kuba, Munning Wen, Ruiqing Chen, Ziyan Wang, Zheng Tian, Jun Wang, Alois Knoll, and Yaodong Yang. Multi-agent constrained policy optimisation. *arXiv preprint arXiv:2110.02793*, 2021. URL https://arxiv.org/abs/2110.02793.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. MetaDrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3461–3475, 2023. doi: 10.1109/TPAMI.2022.3190471. URL https://doi.org/10.1109/TPAMI.2022.3190471.

Zeyang Li and Navid Azizan. Safe multi-agent reinforcement learning with convergence to generalized Nash equilibrium. *arXiv preprint arXiv:2411.15036*, 2024. URL https://arxiv.org/abs/2411.15036.

Songtao Lu, Kaiqing Zhang, Tianyi Chen, Tamer Başar, and Lior Horesh. Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8767–8775, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17062.

Koichi Nabetani, Paul Tseng, and Masao Fukushima. Parametrized variational inequality approaches to generalized Nash equilibrium problems with shared constraints. *Computational Optimization and Applications*, 48:423–452, 2011. URL https://doi.org/10.1007/s10589-009-9256-3.

Kingsley Nweye, Kathryn Kaspar, Giacomo Buscemi, Tiago Fonseca, Giuseppe Pinto, Dipanjan Ghose, Satvik Duddukuru, Pavani Pratapa, Han Li, Javad Mohammadi, Luis Lino Ferreira, Tianzhen Hong, Mohamed Ouf, Alfonso Capozzoli, and Zoltan Nagy. CityLearn v2: Energy-flexible, resilient, occupant-centric, and carbon-aware management of grid-interactive communities. *Journal of Building Performance Simulation*, 18(1):17–38, 2025. doi: 10.1080/19401493.2024.2418813. URL https://doi.org/10.1080/19401493.2024.2418813.

J. B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, 33(3):520–534, 1965. URL https://www.jstor.org/stable/1911749.

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

Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and RLHF. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 44774–44799, 2024. URL https://proceedings.mlr.press/v235/shen24g.html.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID lagrangian methods. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9133–9143, 2020. URL https://proceedings.mlr.press/v119/stooke20a.html.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1805.11074.

Vinzenz Thoma, Barna Pasztor, Andreas Krause, Giorgia Ramponi, and Yifan Hu. Contextual bilevel reinforcement learning for incentive alignment. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.01575.

Vinzenz Thoma, Georgios Piliouras, and Luke Marris. Deep incentive design with differentiable equilibrium blocks. *arXiv preprint arXiv:2603.07705*, 2026. URL https://arxiv.org/abs/2603.07705.

Risto Vuorio, Jacob Beck, Shimon Whiteson, Jakob Foerster, and Gregory Farquhar. An investigation of the bias-variance tradeoff in meta-gradients. *arXiv preprint arXiv:2209.11303*, 2022. URL https://arxiv.org/abs/2209.11303.

Zhongwen Xu, Hado van Hasselt, and David Silver. Meta-gradient reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://arxiv.org/abs/1805.09801.

Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. Learning to incentivize other learning agents. In *Advances in Neural Information Processing Systems*, volume 33, 2020a. URL https://arxiv.org/abs/2006.06051.

Jiachen Yang, Ethan Wang, Rakshit Trivedi, Tuo Zhao, and Hongyuan Zha. Adaptive incentive design with multi-agent meta-gradient reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 1436–1445, 2022. URL https://ifaamas.org/Proceedings/aamas2022/pdfs/p1436.pdf.

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J. Ramadge. Projection-based constrained policy optimization. In *International Conference on Learning Representations*, 2020b. URL https://arxiv.org/abs/2010.03152.

Xianliang Yang, Zhihao Liu, Wei Jiang, Chuheng Zhang, Li Zhao, Lei Song, and Jiang Bian. A versatile multi-agent reinforcement learning benchmark for inventory management. *arXiv preprint arXiv:2306.07542*, 2023. URL https://arxiv.org/abs/2306.07542.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, David C. Parkes, and Richard Socher. The AI economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18):eabk2607, 2022. URL https://doi.org/10.1126/sciadv.abk2607.

A EQUILIBRIUM SELECTION

We write the normalized constrained game as a variational inequality, prove that a regular equilibrium branch moves smoothly with responsibility while its active shared costs remain fixed, and extend the argument to constraint-specific responsibility columns. Two exact games in Appendix A.4 exhibit welfare variation at fixed active cost and the benefit of separate constraint columns.

A.1 NORMALIZED GAME AND ASSUMPTIONS

Agent i chooses $\theta_i \in \Theta_i \subset \mathbb{R}^{n_i}$, where each Θ_i is nonempty, compact, convex, and full-dimensional. Let $\Theta = \prod_i \Theta_i$ and

$$\mathcal{K} = \{\theta \in \Theta : C_k(\theta) \le d_k,\ k = 1, \dots, K\}.$$

For this subsection, the same positive share ρ_i is used for every constraint. The effective multiplier in agent i's objective is $N\lambda_k\rho_i$. Dividing its stationarity condition by the positive factor $N\rho_i$ leaves a common shared multiplier λ_k across agents. This gives the weighted negative-payoff field

$$F_\rho(\theta) = \operatorname{col}_{i=1}^{N}\left[-\frac{1}{N\rho_i}\nabla_{\theta_i} J_i(\theta)\right]. \tag{11}$$

A normalized equilibrium is a solution of

$$\langle F_\rho(\theta^*), \theta - \theta^* \rangle \ge 0 \qquad \text{for every } \theta \in \mathcal{K}. \tag{12}$$

This is Rosen's positive-weight normalization for a shared-constraint game (Rosen, 1965).

We use the following conditions.

- **(A1)** Each J_i and C_k is C^2 on a neighborhood of Θ. For fixed θ_{-i}, J_i is concave in θ_i, and each C_k is convex in θ. There exists $\bar{\theta} \in \operatorname{int}\Theta$ with $C_k(\bar{\theta}) < d_k$ for every k.
- **(A2)** Responsibility ranges over a connected, relatively open set $\mathcal{R}$ in the simplex interior. For every $\rho \in \mathcal{R}$, the weighted field is strongly monotone: there exists $m_\rho > 0$ such that

$$\langle F_\rho(\theta) - F_\rho(\vartheta), \theta - \vartheta \rangle \ge m_\rho \|\theta - \vartheta\|^2 \qquad (\theta, \vartheta \in \Theta). \tag{13}$$

- **(A3)** On the region under consideration, the active shared constraints and active local bounds are fixed. Active local bounds can be written as owner-separable convex C^2 inequalities $h_{i\ell}(\theta_i) \le 0$. Their gradients together with those of the active shared constraints are linearly independent. Active multipliers are positive, and inactive inequalities remain strictly slack.

The same conditions can be imposed only at a reference point, in which case the conclusions below hold on a sufficiently small neighborhood.

A.2 PROOF OF THEOREM 1

Existence and uniqueness. The feasible set $\mathcal{K}$ is nonempty, compact, and convex, and F_ρ is continuous. For any $\eta > 0$, the projected map

$$\theta \mapsto \Pi_{\mathcal{K}}(\theta - \eta F_\rho(\theta))$$

maps $\mathcal{K}$ continuously into itself. Brouwer's theorem therefore gives a fixed point, and the projection optimality condition is exactly Eq. (12). If θ^* and ϑ^* are two solutions, adding their two variational inequalities yields

$$\langle F_\rho(\theta^*) - F_\rho(\vartheta^*), \theta^* - \vartheta^* \rangle \le 0.$$

Strong monotonicity in Eq. (13) forces $\theta^* = \vartheta^*$. Hence each ρ selects a unique normalized policy.

Smooth dependence on responsibility. Let $\mathcal{A}$ index the active shared constraints and $\mathcal{L}$ the active local bounds. Let G_C contain the active shared-cost gradients as rows and G_h the active local-bound gradients. With local multipliers $\nu_{\mathcal{L}}$ and $z = (\theta, \lambda_{\mathcal{A}}, \nu_{\mathcal{L}})$, define the active KKT residual

$$\mathcal{G}(z, \rho) = \begin{bmatrix} F_\rho(\theta) + G_C^\top \lambda_{\mathcal{A}} + G_h^\top \nu_{\mathcal{L}} \\ C_{\mathcal{A}}(\theta) - d_{\mathcal{A}} \\ h_{\mathcal{L}}(\theta) \end{bmatrix} = 0. \tag{14}$$

Because agent i's weighted stationarity block is scaled by $1/(N\rho_i)$, Eq. (14) is equivalent to its original stationarity condition with shared multiplier $N\lambda_k\rho_i$. Own-concavity and convex feasible best-response sets then make these KKT conditions sufficient for a constrained best response, so the selected policy is a generalized Nash equilibrium.

Let Q be the derivative in θ of the first row of Eq. (14), and let $G = \mathrm{col}(G_C, G_h)$. The Jacobian in the primal-dual variables is

$$\partial_z \mathcal{G} = \begin{bmatrix} Q & G^\top \\ G & 0 \end{bmatrix}, \qquad Q = \partial_\theta F_\rho + \sum_{k \in \mathcal{A}} \lambda_k \nabla^2 C_k + \sum_{\ell \in \mathcal{L}} \nu_\ell \nabla^2 h_\ell. \tag{15}$$

The symmetric part of Q is positive definite by strong monotonicity, convexity, and nonnegative multipliers. Suppose (u, v) lies in the kernel: $Qu + G^\top v = 0$ and $Gu = 0$. Then $u^\top Qu = 0$ forces $u = 0$, and linear independence of the active gradients gives $v = 0$. The Jacobian is nonsingular.

The implicit function theorem gives a locally unique C^1 branch $z^*(\rho)$. Positive active multipliers and strict slackness persist by continuity. Uniqueness of the normalized policy means overlapping local branches agree, giving a smooth selection on the full region.

Active costs and welfare. For every active shared constraint k,

$$\lambda_k^*(C_k(\theta^*) - d_k) = 0.$$

Assumption (A3) keeps $\lambda_k^* > 0$, so $C_k(\theta^*(\rho)) = d_k$ throughout the branch. If welfare has a nonzero intrinsic derivative with respect to responsibility, then moving a small distance in the improving direction changes welfare while all active costs remain at those same budgets. This proves the remaining claims of Theorem 1.

A.3 Constraint-specific responsibility columns

The main theorem uses one responsibility vector for all constraints. LiRA allows each shared constraint to have its own column. Around a regular primal-dual point, the same argument applies to these independent columns.

Let η be nonredundant local coordinates for the product of allocation simplexes, and write $\rho_{ik}(\eta)$ for the responsibility of agent i for constraint k. Define agent i's stationarity field

$$F_i(\theta, \lambda, \eta) = -\nabla_{\theta_i} J_i(\theta) + N \sum_k \lambda_k \rho_{ik}(\eta) \nabla_{\theta_i} C_k(\theta). \tag{16}$$

For active local multipliers ξ, the residual contains $F + G_h^\top \xi$, $C_{\mathcal{A}} - d_{\mathcal{A}}$, and $h_{\mathcal{L}}$. Let H_F be the derivative of its first block in θ, and let A_R be the derivative in $\lambda_{\mathcal{A}}$; its (i, k) block is $N\rho_{ik}\nabla_{\theta_i} C_k$. The corresponding Jacobian is

$$M = \begin{bmatrix} H_F & A_R & G_h^\top \\ G_C & 0 & 0 \\ G_h & 0 & 0 \end{bmatrix}. \tag{17}$$

Proposition 1 (Local independent-column selection)**.** *Suppose a reference primal-dual point satisfies this residual, primal and dual feasibility, and strict complementarity. Assume the smoothness and convexity conditions of (A1), smooth owner-local constraints, and nonsingularity of M. Then a locally unique C^1 primal-dual branch exists for nearby η. Its policies are generalized Nash equilibria, its active costs equal the same budgets, and nonzero welfare sensitivity gives different welfare at those same active costs.*

Proof. Apply the implicit function theorem to the full residual. Continuity preserves multiplier signs and inactive slackness, while own-concavity makes the resulting KKT points constrained best responses. Complementarity pins every active cost at its budget, and the welfare derivative gives the final claim. □

If equality of the entire cost vector is required when some constraints are inactive, let $Q_I(\eta) = C_I(\theta^*(\eta))$ collect those inactive costs. When $\partial_\eta Q_I$ has locally constant rank, its level sets are smooth. A tangent direction in $\ker(\partial_\eta Q_I)$ with nonzero welfare derivative therefore generates a local curve along which the full cost vector is fixed and welfare changes.

A.4 Exact examples

Two small games make the equilibrium selection result concrete. The first shows welfare varying along an active-cost level set. The second shows why different constraints may prefer different responsibility allocations.

A.4.1 Welfare variation at fixed active cost

Let $N \geq 2$ agents choose scalar actions $u_i \in \mathbb{R}$. Agent i prefers action b_i, and the shared budget is $d > 0$. Let $\alpha_i > 0$ be the fixed welfare weight for agent i, set $B = \sum_i b_i$ and $S = \sqrt{2d}$, and assume $B > S$. Define

$$J_i(u) = -\tfrac{1}{2}(u_i - b_i)^2, \qquad C(u) = \tfrac{1}{2}\Big(\sum_i u_i\Big)^2 \leq d. \tag{18}$$

Compact action intervals containing the response family below can be imposed without affecting the solution. For any simplex allocation, the positive-aggregate active equilibrium is

$$\lambda^* = \frac{B - S}{NS} > 0, \qquad u_i^*(\rho) = b_i - (B - S)\rho_i, \qquad C(u^*(\rho)) = d. \tag{19}$$

To see this, stationarity gives $b_i - u_i = N\lambda\rho_i \sum_j u_j$. Summing across agents and using the active constraint $\sum_i u_i = S$ gives Eq. (19). The weighted field is diagonal and strongly monotone, so this family satisfies the assumptions of Theorem 1.

Substituting the equilibrium into welfare gives

$$W(u^*(\rho)) = -\tfrac{1}{2}(B - S)^2 \sum_i \alpha_i \rho_i^2, \qquad \rho_i^* = \frac{\alpha_i^{-1}}{\sum_j \alpha_j^{-1}}. \tag{20}$$

The active cost is identical for every allocation, yet welfare is not. Unequal welfare weights make the uniform allocation strictly suboptimal; with equal weights, uniform is optimal but welfare still changes away from it.

A.4.2 Why separate constraint columns can matter

Now consider two agents with two-dimensional actions $u_1 = (u_{11}, u_{12})$ and $u_2 = (u_{21}, u_{22})$. Both prefer $b = (1, 1)$, and the shared constraints are $C_k(u) = u_{1k} + u_{2k} \leq 1$ for $k = 1, 2$. Let

$$J_1(u) = -\tfrac{1}{2}\,\|u_1 - b\|^2 + (1/4, -1/4)^\top u_2, \tag{21}$$

$$J_2(u) = -\tfrac{1}{2}\,\|u_2 - b\|^2 + (-1/4, 1/4)^\top u_1, \tag{22}$$

with welfare $W = J_1 + J_2$. Let p_k be agent 1's responsibility share for constraint k; agent 2 receives $1 - p_k$. The equilibrium is

$$\lambda_1^* = \lambda_2^* = \tfrac{1}{2}, \qquad u_1^* = (1 - p_1, 1 - p_2), \qquad u_2^* = (p_1, p_2). \tag{23}$$

The two stationarity equations in each coordinate first determine the shared multiplier and then the individual actions. The losses are strongly convex and the constraints are linear, so these points are global constrained best responses. Both shared costs equal one for every $(p_1, p_2) \in [0, 1]^2$.

At this equilibrium,

$$W(p_1, p_2) = -\tfrac{1}{2}\left[p_1^2 + (1-p_1)^2 + p_2^2 + (1-p_2)^2\right] + \tfrac{1}{2}(p_1 - p_2). \tag{24}$$

Its Hessian is $-2I_2$, so the unique optimum is $(p_1, p_2) = (3/4, 1/4)$ with $W = -3/8$. If the two responsibility columns are tied, $p_1 = p_2 = p$, then $W = -(p^2 + (1-p)^2) \le -1/2$, with equality at $p = 1/2$. Independent columns therefore improve the best tied welfare by $1/8$ without changing either shared cost. The two constraints prefer opposite allocations because the reward externalities have opposite signs across coordinates.

B Finite-training responsibility gradient

Theorem 2 differentiates welfare through a random sequence of learner updates. The proof conditions on the sampled training history, differentiates welfare through the recorded updates at fixed data, and then differentiates the probability of that history. These two derivatives yield the direct-unrolling and sampling-correction terms.

B.1 Training-history distribution

Let $\phi \in \mathcal{U} \subset \mathbb{R}^p$ be the allocation parameter and let x_0 be fixed during one outer update. Write $\mathbf{D} = (\mathcal{D}_0, \dots, \mathcal{D}_{q-1})$ for the complete sequence of training batches. Given a realized history, the learner states $x_t(\phi; \mathcal{D}_{0:t-1})$ are deterministic. For fresh batches, the joint sampling law is

$$p_\phi(\mathbf{D} \mid x_0) = \prod_{t=0}^{q-1} p_{\theta_t(\phi;\mathcal{D}_{0:t-1})}(\mathcal{D}_t). \tag{25}$$

With a persistent collector, the same argument uses the corresponding history-conditioned sampling factors.

To separate the two sources of dependence, let $\mathrm{D}_\phi^{\mathrm{fix}}$ denote the total derivative $\mathrm{d}/\mathrm{d}\phi$ in Eq. (8), taken while the realized raw trajectories are held fixed. Observations, actions, rewards, costs, terminal flags, and other exogenous samples are fixed; learner quantities derived from them are not. For a scalar, $\mathrm{D}_\phi^{\mathrm{fix}}$ is a column gradient. In particular,

$$\mathrm{D}_\phi^{\mathrm{fix}} W(\theta_q) = (\mathrm{D}_\phi^{\mathrm{fix}} \theta_q)^\top \nabla_\theta W(\theta_q), \qquad s_t = (\mathrm{D}_\phi^{\mathrm{fix}} \theta_t)^\top \nabla_{\theta_t} \log p_{\theta_t}(\mathcal{D}_t). \tag{26}$$

The proof assumes that the conditional sampling laws share parameter-independent support, the recorded learner maps and welfare are differentiable almost surely, and differentiation can be exchanged with integration. For piecewise-smooth learner updates, these conditions are understood on the realized differentiable branches.

B.2 Proof of Theorem 2

Using Eq. (25),

$$\nabla_\phi V_q = \nabla_\phi \int W(\theta_q(\phi; \mathbf{D}))\, p_\phi(\mathbf{D} \mid x_0)\, \mathrm{d}\mathbf{D} \tag{27}$$

$$= \int \left[\mathrm{D}_\phi^{\mathrm{fix}} W(\theta_q) + W(\theta_q) \mathrm{D}_\phi^{\mathrm{fix}} \log p_\phi(\mathbf{D} \mid x_0)\right] p_\phi(\mathbf{D} \mid x_0)\, \mathrm{d}\mathbf{D} \tag{28}$$

$$= \mathbb{E}\left[\mathrm{D}_\phi^{\mathrm{fix}} W(\theta_q) + W(\theta_q) \sum_{t=0}^{q-1} s_t\right]. \tag{29}$$

The first term differentiates welfare through the learner for a fixed training history. The second differentiates the probability of seeing that history. This proves Eq. (8).

B.3 POLICY-SCORE FORM

The sampling correction can be evaluated without differentiating the environment. Let P denote the transition law and p_0 the reset law. For a trajectory with states s_h, local observations $o_{i,h}$, and joint actions a_h,

$$p_\theta(\tau) = p_0(s_0) \prod_{h=0}^{H-1} \left[P(s_{h+1} \mid s_h, a_h) \prod_{i=1}^{N} \pi_{\theta_i}(a_{i,h} \mid o_{i,h}) \right]. \tag{30}$$

At a fixed realized trajectory, the environment factors do not depend on ϕ. The batch score therefore reduces to policy scores:

$$s_t = \sum_{\tau \in \mathcal{D}_t} \sum_{h \in \tau} \sum_{i=1}^{N} \mathrm{D}_\phi^{\mathrm{fix}} \log \pi_{\theta_{t,i}}(a_{i,h} \mid o_{i,h}). \tag{31}$$

Each sampled action appears once, even if the same batch is reused for multiple policy-optimization epochs.

B.4 TERMINAL WELFARE ESTIMATE

For a terminal rollout τ, let

$$R_W(\tau) = \sum_{h=0}^{H-1} \gamma^h \sum_i \alpha_i r_{i,h}$$

be its original social return. Given the training history, draw $E \geq 1$ fresh terminal rollouts $\tau_1, \ldots, \tau_E$ under π_{θ_q} and define

$$\widehat{W} = \frac{1}{E} \sum_{e=1}^{E} R_W(\tau_e), \qquad \widehat{g}^{\mathrm{DU}} = \frac{1}{E} \sum_{e=1}^{E} R_W(\tau_e) (\mathrm{D}_\phi^{\mathrm{fix}} \theta_q)^\top \nabla_\theta \log p_{\theta_q}(\tau_e). \tag{32}$$

Conditioned on the training history, these estimate $W(\theta_q)$ and $\mathrm{D}_\phi^{\mathrm{fix}} W(\theta_q)$, respectively. In a general-sum game, the social return weights every agent's policy score, preserving the effect of one agent's action on the others' rewards.

B.5 LEAVE-ONE-OUT CENTERING

For independent continuations $m = 1, \ldots, M$ from the same checkpoint, let $s^{(m)} = \sum_{t=0}^{q-1} s_t^{(m)}$ and let b_m be the baseline in Eq. (10). A score has conditional mean zero:

$$\mathbb{E}[s^{(m)} \mid x_0, \phi] = \int \nabla_\phi p_\phi(\mathbf{D} \mid x_0) \, \mathrm{d}\mathbf{D} = 0. \tag{33}$$

Because b_m is constructed only from the other independent continuations, $\mathbb{E}[b_m s^{(m)} \mid x_0, \phi] = 0$. Together with the conditional unbiasedness of the terminal estimates,

$$\mathbb{E}[\widehat{g} \mid x_0, \phi] = \nabla_\phi V_q(\phi \mid x_0). \tag{34}$$

Thus leave-one-out centering changes neither term in expectation. The continuation estimates need not be mutually independent after centering, because their baselines share the other continuations' returns; only the independence between a continuation's score and its own baseline is used.

C LOOKAHEAD IMPLEMENTATION

C.1 PPO AND CRITIC UPDATES

Let ϑ_i denote agent i's actor parameters within a PPO epoch and θ_i^{beh} the parameters used to collect the batch. Let $\widehat{A}_{i,h}^R$ and $\widehat{A}_{k,h}^C$ be the reward and shared-cost advantages. For a fixed responsibility

allocation, the effective advantage $\widehat{A}^{\text{eff}}_{i,h}$ and policy ratio $\zeta_{i,h}$ are

$$\widehat{A}^{\text{eff}}_{i,h} = \widehat{A}^{R}_{i,h} - \sum_{k} N\lambda_k \rho_{ik}(\phi)\widehat{A}^{C}_{k,h}, \qquad \zeta_{i,h}(\vartheta_i) = \frac{\pi_{\vartheta_i}(a_{i,h} \mid o_{i,h})}{\pi_{\theta^{\text{beh}}_i}(a_{i,h} \mid o_{i,h})}. \tag{35}$$

Let $\epsilon_{\text{PPO}} > 0$ be the clipping radius, $c_{\text{ent}} \geq 0$ the entropy coefficient, $\mathcal{H}$ policy entropy, and $\widehat{\mathbb{E}}$ a masked average over valid samples. The clipped loss is

$$\widehat{\mathcal{L}}_i(\vartheta_i) = -\widehat{\mathbb{E}}\Big[\min\Big\{\zeta_{i,h}(\vartheta_i)\widehat{A}^{\text{eff}}_{i,h}, \text{clip}(\zeta_{i,h}(\vartheta_i), 1-\epsilon_{\text{PPO}}, 1+\epsilon_{\text{PPO}})\widehat{A}^{\text{eff}}_{i,h}\Big\}\Big] - c_{\text{ent}}\widehat{\mathbb{E}}[\mathcal{H}(\pi_{\vartheta_i})]. \tag{36}$$

PPO clips the combined reward-and-cost advantage (Schulman et al., 2017). In CityLearn, the reward advantages are the per-building, per-step comfort rewards standardized within each batch; the cost advantages are the district cap excess standardized over that batch. The critics regress to the corresponding unnormalized per-step sum of building rewards and district cap excess.

C.2 What the lookahead differentiates

The CityLearn probe starts from copies of the live actors, critics, and Adam moments. Each fresh batch gives one functional PPO actor–critic update and one differentiable Adam step. The next batch is collected under the updated probe policy. The shared multiplier is held at its checkpoint value during this q-step probe to isolate responsibility redistribution at the current dual scale; it is updated in the live learner on the schedule specified in Appendix D.1.

Within a probe update, stored behavior log-probabilities, normalized advantages, and critic targets are held fixed. Differentiation follows the actor and critic updates and the carried Adam moments, while the training-history score accounts for actions collected under the changing probe policies. The PPO implementation therefore uses a stopped-batch approximation to the exact direct derivative in Theorem 2: its sampling correction does not restore derivatives omitted from the update itself.

C.3 Allocation and online updates

For logits ϕ_{ik} and a fixed floor $\varepsilon_\rho \in [0, 1/N)$,

$$\rho_{ik}(\phi) = \varepsilon_\rho + (1 - N\varepsilon_\rho)\frac{\exp \phi_{ik}}{\sum_j \exp \phi_{jk}}. \tag{37}$$

The floor keeps shares away from zero; setting it to zero recovers softmax. Adding a constant to a logit column leaves its allocation unchanged. The equilibrium proofs use nonredundant local simplex coordinates, whereas autodiff can operate directly on the logits. An ascent optimizer applies Eq. (10) to update ϕ.

Algorithm 1 uses a lookahead from the current checkpoint. Each of its M independent continuations takes q probe updates at fixed responsibility, collecting fresh training batches and terminal rollouts. After the outer step, the live learner takes h updates from that checkpoint under the new allocation, producing the full learner state x^+. Held-out evaluation measures its policy without changing that state. The next cycle uses $x_0 \leftarrow x^+$ and retains the updated ϕ, carrying forward the actors, critics, optimizer state, and shared multipliers from the live continuation.

C.4 Computation and evaluation

The terminal rollouts used in the meta-gradient train ϕ; separate reporting rollouts evaluate the resulting policy. Appendix D.1 records benchmark settings and evaluation conventions for Table 1. A responsibility-update cycle uses Mq probe batches and h live batches, in addition to terminal and held-out evaluation rollouts. The lookahead cost is part of the method's computation budget.

D EXPERIMENTAL PROTOCOLS AND REWARD COMPONENTS

D.1 BENCHMARK SETUP FOR SECTION 4.1

We use the same training seed identifiers across methods within each task: three seeds, except six for MetaDrive. We first summarize the held-out evaluations within each seed; Table 1 reports the mean and sample standard deviation across these seed-level values. Costs and welfare retain each benchmark's native units.

CityLearn. We use CityLearn v2 (Nweye et al., 2025) with the three buildings in `citylearn_challenge_2023_phase_1`. Welfare is the sum of native ComfortReward across buildings and steps. Shared cost accumulates only the positive excess of district net electricity over the per-step grid cap. The $d = 0.04$ experiments have a 719-step cap-exceedance budget of 28.76 kWh (719×0.04) and use seeds 1101–1103, 30 outer cycles, $h = 10$ live PPO updates per cycle, 12-step live rollouts, four PPO epochs, shared-dual initialization 2.0, and dual learning rate 0.05. For each CityLearn lookahead, every probe step collects a fresh 12-step batch and makes one functional PPO/Adam actor–critic update. The shared multiplier stays fixed during the probe. A separate 12-step terminal rollout supplies the mean per-step comfort reward for the allocation update; its action scores and the training-batch scores are both divided by the 12-step batch length. Before the outer optimizer, this scales Eq. (10)'s unnormalized score estimate by $1/12$. After the allocation update, the live learner instead takes $h = 10$ fresh batches with four PPO epochs each, then updates the shared multiplier once from their mean grid excess. That dual update compares mean per-step excess against 0.04 kWh. The held-out values in Table 1 are 719-step episode totals, rather than the probe's short-rollout objective. The LiRA reference point uses $(q, M, \eta_\rho) = (2, 8, 0.05)$ in both Table 1 and Table 3. The local search changes one coordinate at a time on the same three training seeds, keeping $h = 10$ live updates per outer cycle.

In this setting, PAL starts with three equal per-building multipliers. Their updates use the same district cost and budget, so they remain equal and produce the same result as Uniform.

MABIM. The inventory experiment has $N = 400$ controlled SKU agents and $K = 2$ shared rejection-cost coordinates: warehouse 0/store 2 and warehouse 1/store 1. For Table 1, we use model seeds 1101–1103, $q = 5$, $M = 6$ lookaheads, ten outer cycles (50 committed updates), and budgets calibrated on a validation period separate from the held-out test period. Held-out evaluation pairs the same eight action seeds across LiRA, Uniform, and PAL. Each training seed contributes the mean of these eight held-out rollouts to Table 1; the displayed spread is across training seeds.

Harvest. The Lab2D Harvest task (Beattie et al., 2020) uses seven agents on `10gen_terran` and one episode-level shared cost with budget 68. The three training seeds are 10411, 10442, and 10473. The LiRA setting uses 60 response updates, $q = 10$, $M = 4$, and responsibility step size 0.15, with an allocation update every five response updates. The shared dual starts at 0.3 and uses step size 0.05. Each training seed is evaluated on five held-out episodes with native stochastic resets.

MetaDrive. MetaDrive uses six matched training seeds (211–216), a shared dual learning rate $\lambda_{\mathrm{lr}} = 4.0$ across all three arms, and the active training-time cost budget $d = 0.001$ per actual environment step (shown in Table 1 as $d = 1.00$ events per 1000 actual held-out steps, the same threshold on that display scale). All three arms start from the same competent fixed actor checkpoint and are evaluated on the same 13 held-out episode seeds; welfare is the mean per-episode welfare and cost is the number of native shared-cost events per 1000 steps actually run. Within each training seed, we average welfare over the 13 episodes and divide the total event count by their total realized steps to obtain the cost rate. Table 1 then aggregates these six seed-level values.

D.2 CITYLEARN MECHANISM DETAILS

Figure 2 uses seed 1102 with $q = 2$, $M = 8$, and $\eta_\rho = 0.05$. Comfort states are classified against the effective $\pm 1^{\circ}$C ComfortReward band. Table 4 separates the cold, warm, and in-band contributions to the comfort changes in Table 2.

Table 4: CityLearn comfort components in the matched seed-1102 episode. Cold excess is measured under Uniform; reward changes are LiRA minus Uniform.

	Cold excess (°C·steps)	ΔR_{cold}	ΔR_{warm}	$\Delta R_{\mathrm{in-band}}$
Building 1	586.1	+4528.01	-102.80	+6.55
Building 2	214.5	+813.09	-340.49	+6.26
Building 3	70.2	+159.58	-205.55	+11.62

Building 1 begins with the greatest cold burden, so its reduction in cooling yields the largest comfort recovery. Across all three buildings, episode welfare increases by 4876.26 while district cap excess moves from 25.4156 to 23.7787 kWh.