When Is Diversity Rewarded in Cooperative Multi-Agent Learning?

Michael Amir* **Matteo Bettini*** **Amanda Prorok**
Department of Computer Science and Technology
University of Cambridge
{ma2151,mb2389,asp45}@cl.cam.ac.uk

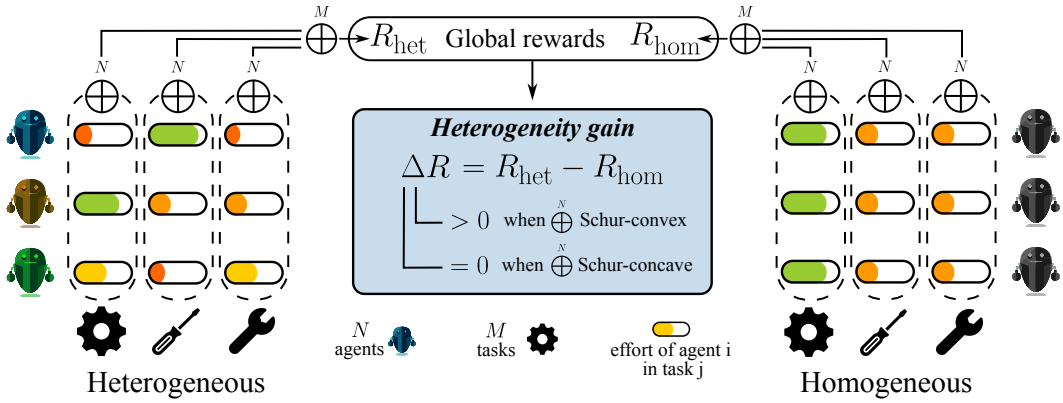


Figure 1: We study and categorize what reward structures lead to the need for behavioral heterogeneity in multi-agent multi-task environments.

Abstract

The success of teams in robotics, nature, and society often depends on the division of labor among diverse specialists; however, a principled explanation for *when* such diversity surpasses a homogeneous team is still missing. Focusing on multi-agent task allocation problems, our goal is to study this question from the perspective of reward design: what kinds of objectives are best suited for heterogeneous teams? We first consider an instantaneous, non-spatial setting where the global reward is built by two generalized aggregation operators: an *inner* operator that maps the N agents' effort allocations on individual tasks to a task score, and an *outer* operator that merges the M task scores into the global team reward. We prove that the curvature of these operators determines whether heterogeneity can increase reward, and that for broad reward families this collapses to a simple convexity test. Next, we ask what incentivizes heterogeneity to *emerge* when embodied, time-extended agents must *learn* an effort allocation policy. To study heterogeneity in such settings, we use multi-agent reinforcement learning (MARL) as our computational paradigm, and introduce *Heterogeneous Environment Design (HED)*, a gradient-based algorithm that optimizes the parameter space of underspecified MARL environments to find scenarios where heterogeneity is advantageous. Experiments in matrix games and an embodied Multi-Goal-Capture environment show that, despite the difference in settings, HED rediscovers the reward regimes predicted by our theory to maximize the advantage of heterogeneity, both validating HED and connecting our theoretical insights to reward design in MARL. Together, these results help us understand when behavioral diversity delivers a measurable benefit.

*Equal contribution, listed alphabetically.

Preprint. Under review.

1 Introduction

Collective systems—from robot fleets to insect colonies—tend to adopt one of two fundamental structures: a uniform shared blueprint or a set of distinct, specialized roles. In multi-agent learning, this dichotomy is reflected in the design choice between behavioral homogeneity (where all agents share one policy) and heterogeneity (where each agent learns its own) [1, 2, 3]. Although diversity unlocks role specialization and asymmetric information use, it also introduces extra coordination cost, representation overhead, and learning complexity [4]. This trade-off leads us to ask: under what conditions will heterogeneous agents actually outperform the best homogeneous baseline?

A natural setting to study this question in is *multi-agent task allocation*, where N agents allocate effort across M concurrent tasks. A homogeneous team is forced to adopt a single allocation vector (e.g., every agent spends 0.75 of its effort on task A and 0.25 on task B), whereas a heterogeneous team allows agents to specialize differently. This allocation setting underlies multi-agent problems such as coverage and load-balancing [5, 6]. In this work, we relate it to heterogeneity in cooperative navigation (Sec. 5), Colonel Blotto games, and level-based foraging environments (App. D) [7, 8, 9, 10].

Theoretical Insights. We first study a pure, non-spatial and instantaneous variant of multi-agent task allocation: each agent commits its effort allocation once, and the team is rewarded immediately (Sec. 2). We start from the observation that team reward in many effort–allocation problems can be expressed as a *generalized double aggregation*, $R(A) = \bigoplus_{j=1}^{M} \bigoplus_{i=1}^{N} r_{ij}$, where $A = (r_{ij})$ is the $N \times M$ matrix of agent effort allocations, the inner operator $\bigoplus$ aggregates over the N agents' efforts in each task and the outer operator $\bigoplus$ aggregates over resulting M task scores into a scalar global reward. Choosing both operators as sums recovers the $\sum_j \sum_i r_{ij}$ reward common in RL, whereas alternatives such as MAX, MIN, power means, or soft-max encode very different effort–reward relationships. Assuming such a reward structure, we compare the optimal heterogeneous reward, R_{het}, with the best reward attainable under a homogeneous allocation, R_{hom}, and define their difference as the *heterogeneity gain* $\Delta R = R_{\text{het}} - R_{\text{hom}}$ (Fig. 1). Our main insight is that ΔR is determined by the *curvature* of the two aggregators: specifically, whether they are *Schur-convex* or *Schur-concave*. These criteria immediately enable us to characterize the heterogeneity gain of broad families of reward functions (Table 3); for instance, the soft-max operator switches from Schur-concave to Schur-convex as its temperature increases. We also find exact expressions for ΔR in several important cases. These results help explain, for example, why a reward structure that involves a $\min$ aggregator (usually used to enforce that only one agent should pursue a goal) will require behavioral diversity from the agents [11]. We relate our findings to multi-agent reinforcement learning (MARL), where environments may be embodied and time-extended, by setting $R(A_t)$ as the stepwise reward over an allocation sequence $(A_t)_{t=1,\dots,T}$.

Algorithmic Search. To study heterogeneity in MARL settings not covered by our theoretical analysis, we develop *Heterogeneous Environment Design* (HED), a gradient-based co-design algorithm that learns environment parameters θ in under-specified tasks via backpropagation, leveraging differentiable simulation to find parameterizations that either maximize or minimize ΔR. In our experiments θ parameterizes the reward structure, i.e., the inner- and outer-aggregators, but any differentiable feature of the environment (e.g., obstacle placements) can be optimized. Maximizing the heterogeneity gain through HED allows us to discover previously unknown types of tasks where heterogeneity is essential; minimizing the gain steers an underspecified task toward regimes where homogeneous policies are sufficient, which may be desirable, e.g., when parameter sharing is preferred for learning efficiency (Sec. 4).

Experiments. We evaluate the validity of our theoretical insights, and HED, in simulation. To this end, we design single-shot and long-horizon reinforcement learning environments whose reward structure instantiates the kinds of aggregation operators we studied. First, in a continuous and a discrete matrix game, we test reward structures based on all nine possible combinations of $\{\min, \text{mean}, \max\}$, and find that the heterogeneity gains that result from the agents' learned policies match our theoretical predictions: concave outer aggregators and convex inner aggregators benefit heterogeneous teams. Next, we test the same aggregators in an embodied, partially observable MULTI-GOAL-CAPTURE environment, and find that our theory also transfers to this long-horizon MARL setting. Importantly, we also show that *reward structures that maximize heterogeneity are meaningful and practically useful*. Finally, we find that the empirical heterogeneity gain disappears as the richness of agents' observations is increased, recovering the finding that rich observations allow agents with identical policy networks to be behaviorally heterogeneous [1, 12].

We then turn to HED. Across two parameterizable families of aggregators (Softmax and Power-Sum), we show that, despite running on spatial and long-horizon environments, HED rediscovers the reward regimes predicted by our curvature theory to maximize the heterogeneity gain, both validating HED and confirming the connection between our theoretical insights and MARL reward design (Sec. 5).

1.1 Related Works

Behavioral Diversity in MARL. Behavioral heterogeneity—where capability-identical agents learn distinct policies—can markedly improve exploration, robustness, and reward [1]. Yet heterogeneity reduces parameter sharing and thus sample-efficiency, so a core practical question is *when* its benefits outweigh that cost. Existing MARL methods typically adopt one of two poles: endowing each agent with its own network, or enforcing parameter sharing so that all agents follow a single policy [13, 14, 15, 16, 17]. A substantial body of work explores the efficiency–diversity trade-off [18, 19] by interpolating between these extremes—e.g. injecting agent IDs into the observation [20, 13], masking different subsets of shared weights [21], sharing only selected layers [4], pruning a shared network into agent-specific sub-graphs [22], or producing per-agent parameters with a hypernetwork [23]. Further, several methods for promoting behavioral diversity in MARL have been proposed, such as: conditioning agents' policies on a latent representation [24], decomposing and clustering action spaces [25], dynamically grouping agents to share parameters [26], applying structural constraints to the agents' policies [11], or by additional tasks and intrinsic rewards that maximize diversity [4, 27, 28, 29, 30, 31, 32, 33]. While these studies demonstrate *how* to obtain diversity, they all presume tasks where heterogeneity is already advantageous; what is still missing—and what we supply here—is a principled characterization of *which* reward structures create that incentive in the first place.

Task Allocation. Classic resource–allocation settings, in which a team must divide finite effort among simultaneous objectives, are a central proving ground for cooperative MARL. In robotics, potential-field and market-based learning are the dominant tools for coverage, exploration, and load-balancing tasks [5, 6]. Game-theoretic analysis and, recently, MARL, play the same role in discrete counterparts such as Colonel-Blotto contests, where players decide how to spread forces over several "battlefields" [7, 8]. Embodied benchmarks like level-based foraging are heavily studied in MARL, and expose the tension between uniform and specialized effort allocations [9]. The survey of [34] highlights how cooperative performance is governed by the shape of the shared reward and the equilibria it induces. Our contribution sharpens this perspective: we prove that the *curvature* of nested aggregation operators characterizes when heterogeneous allocations dominate homogeneous ones, and introduce algorithmic tools for further exploring settings where diversity is needed.

Environment Co-design. Co-design is a paradigm where agent policies *and* their mission or environment are simultaneously optimized [35]. Our HED algorithm is related to PAIRED [36], a method which automatically designs environments in a curriculum such that an *antagonist* agent succeeds while the *protagonist* agent fails. This makes it so that resulting environments are challenging enough without being unsolvable. Similarly, HED designs environments that are advantageous to heterogeneous teams, while disadvantaging homogeneous teams. The key differences are: (1) the environment designer uses direct regret gradient backpropagation via a differentiable simulator instead of RL; this enables higher efficiency by directly leveraging all the environment gradient data available during collection while preventing RL-related issues identified in subsequent works [37, 38] such as exploration inefficiency and the need for a reward signal; and (2), the protagonist and antagonist are independent multi-agent teams instead of single agents.

2 Formal Setting: Multi-Agent Task Allocation

Consider a set of N agents and M tasks. Each agent $i \in \{1, \dots, N\}$ allocates effort among the tasks according to the budget constraints: $r_{i1}, r_{i2}, \dots, r_{iM} \geq 0$ with $\sum_{j=1}^{M} r_{ij} \leq 1$. We can consider both continuous allocations (r_{ij} can be any real number) and discrete allocations (r_{ij} restricted to some finite set of options), with most results in this work focusing on the continuous case. We collect all agents' allocations into an $N \times M$ matrix: $A = [r_{ij}]$[2].

[2]All results in this work can be extended to the case where $r_{i1}, r_{i2}, \dots, r_{iM} \geq B_{min}$ and $\sum_{j=1}^{M} r_{ij} \leq B_{max}$ for some arbitrary $B_{min}, B_{max} \in \mathbb{R}$.

For each task j let the j-th column of the effort matrix be $a_j = [r_{1j}, \ldots, r_{Nj}]^\top$. A *task-level aggregator* $T_j : \mathbb{R}^N \to \mathbb{R}$ maps these efforts to a *task score*, and an *outer aggregator* $U : \mathbb{R}^M \to \mathbb{R}$ combines the M scores into the team reward, $R(A) = U\big(T_1(a_1), \ldots, T_M(a_M)\big)$. Both T_j and U are *generalised aggregators*: symmetric and coordinate-wise non-decreasing, mirroring the familiar properties of $\sum$. When every task shares the same inner aggregator we simply drop the subscript and write T. To highlight the analogy with a double sum we also write $R(A) = \bigoplus_{j=1}^{M} \bigoplus_{i=1}^{N} r_{ij}$, where (in abuse of notation) the outer symbol $\bigoplus$ denotes U and the inner symbol $\bigoplus$ denotes T_j.

Homogeneous vs. Heterogeneous Strategies. A *homogeneous strategy* is one where all agents have the same allocation (i.e., devote the same amount of effort to a given task j): $r_{ij} = c_j \forall\, i, j$. In this case, the allocation matrix A consists of identical rows. We define $R_{\text{hom}} = \max_{(c_1,\ldots,c_M)\in\Delta_{\leq}^{M-1}} R\Big(A\Big)$ where $\Delta_{\leq}^{M-1} = \{(c_1, \ldots, c_M) \mid c_j \geq 0,\ \sum_j c_j \leq 1\}$ is the closed unit simplex. A *heterogeneous strategy* allows each agent i to choose any $(r_{i1}, \ldots, r_{iM}) \in \Delta_{\leq}^{M-1}$ independently. Then $R_{\text{het}} = \max_{A\in(\Delta_{\leq}^{M-1})^N} R\Big(A\Big)$. We define the *heterogeneity gain* as: $\Delta R = R_{\text{het}} - R_{\text{hom}}$. This quantity measures how much greater the overall reward can be when agents are allowed to specialize differently across tasks, compared to when they must behave identically. Characterizing when $\Delta R > 0$ is our main focus in this work.

MARL extension. In MARL, the value r_{ij} represents the 'local rewards' agents receive based on task performance, and the aggregate reward $R(A)$ can represent: (i) the payoff of a one-shot effort-allocation game, (ii) the return or sparse terminal reward of an episode, or (iii) the stepwise reward, giving the discounted return $\sum_{t=0}^{T} \gamma^t R\big(A_t\big)$ for a sequence $\big(A_t\big)_{t=1,\ldots T}$ of allocations[3]. A positive gain $\Delta R > 0$ implies heterogeneous agents can outperform homogeneous ones *if* the heterogeneous agents can always attain the optimal allocation; if not (as in our Multi-Goal-Capture experiment), this is *evidence of* an advantage to heterogeneity, but not a formal guarantee.

Examples. App. H contains concrete examples of generalized aggregators. In Sec. 5, we study two MARL environments whose reward structure conforms to our setting: a one-shot task allocation game and an embodied Multi-Goal-Capture environment, which can be seen as an extension of various cooperative navigation environments (see [10]). Finally, in App. D, we show how the heterogeneity gain of two well-known environments from the literature–Team Colonel Blotto games [8] and level-based foraging [9]–can be studied using the theory we develop in this work.

3 Analysis

Focusing on continuous allocations, we ask what properties of aggregators guarantee $\Delta R > 0$. We draw on the concept of Schur-convexity. Schur-convex functions can be understood as generalizing symmetric, convex aggregators: every convex and symmetric function is Schur-convex, but a Schur-convex function is not necessarily convex [39, 40]. Proofs for all results are available in App. F.

Since both the outer aggregator U and the task-level aggregators T_j are non-decreasing, an optimal effort allocation will always have each agents' efforts summing to 1. Hence, from here on, we **assume** without loss of generality that $\sum_{j=1}^{M} r_{ij} = 1$. We call such allocations **admissible**.

Definition 3.1 (Majorization)**.** *Let $x = (x_1, \ldots, x_N)$ and $y = (y_1, \ldots, y_N)$ be two vectors in $\mathbb{R}^N$ such that $\sum_{i=1}^{N} x_{(i)} = \sum_{i=1}^{N} y_{(i)}$. Let $x_{(1)} \geq x_{(2)} \geq \cdots \geq x_{(N)}$ and $y_{(1)} \geq y_{(2)} \geq \cdots \geq y_{(N)}$ be the components of x and y sorted in descending order. We say that x* majorizes *y (written $x \succ y$) if $\sum_{i=1}^{k} x_{(i)} \geq \sum_{i=1}^{k} y_{(i)}$ for $k = 1, 2, \ldots, N-1, N$.*

Definition 3.2 (Schur-Convex Function)**.** *A symmetric function $f : \mathbb{R}^N \to \mathbb{R}$ is* Schur-convex *if for any two vectors $x, y \in \mathbb{R}^N$ with $x \succ y$, we have $f(x) \geq f(y)$. If the inequality is strict whenever x and y are not permutations of each other, then f is said to be* strictly Schur-convex*. Similarly, f is* Schur-concave *if $f(x) \leq f(y)$ whenever $x \succ y$.*

Intuitively, $x \succ y$ means one can obtain y from x by repeatedly moving mass from larger to smaller coordinates, thereby making the vector more uniform. Schur-convexity is then a statement on a

[3]We can generalize (iii) by considering a sequence $(R_t)_{t=1\ldots,T}$ of allocation returns that varies per time step.

function's *curvature*: f is *Schur-convex* if it increases with inequality, or is *Schur-concave* if it increases with uniformity. We show here a connection between Schur-convexity (concavity) and ΔR.

Call an allocation matrix A *trivial* if there exists a task $j^\star$ such that every agent allocates its entire budget to that task, i.e. $r_{ij^\star} = B_{\max}$ and $r_{ij} = 0\ \forall i,\ \forall j \neq j^\star$; otherwise A is *non-trivial*. Then:

Theorem 3.1 (Positive Heterogeneity Gain via Schur-convex Inner Aggregators)**.** *Let $N, M \geq 2$, and assume that (i) each* task-level aggregator T_j *is strictly Schur-convex and (ii) the* outer aggregator U *is coordinate-wise strictly increasing. Then either all admissible optimal homogeneous allocations are trivial, or $\Delta R > 0$.*

If the task-level aggregator is instead Schur-concave, we can show there is no heterogeneity gain:

Theorem 3.2 (No Heterogeneity Gain via Schur-concave Inner Aggregators)**.** *Let $N, M \geq 2$. If each task-level aggregator T_j is* Schur-concave *then $\Delta R = 0$.*

We see that Schur-convexity of the inner aggregator produces $\Delta R > 0$, whereas Schur-concavity implies $\Delta R = 0$. Analyzing the outer aggregator U is trickier, because it acts on task-score vectors $\big(T_1(a_1), \ldots, T_M(a_M)\big)$ whose sum $\sum_{i=1}^{M} T_i(a_i)$ may vary, so majorization is not directly applicable. However, we can extend our analysis to U if our inner aggregators are *normalized* to keep the sum constant: $\sum_{i=1}^{M} T_i(a_i) = C$ for any admissible allocation. Assuming this, we can invoke majorization again, and the relationship between convexity and ΔR reverses: if the outer aggregator U is Schur-convex, the heterogeneity gain vanishes. Let us prove this.

Theorem 3.3 (No Heterogeneity Gain for Schur-Convex U with Constant-Sum Task Scores)**.** *Let $N, M \geq 2$. Suppose that for any admissible allocation A, (i) every task score is non-negative, and obeys $T_i(0, \ldots, 0) = 0$, and (ii) the sum of task score is always $\sum_{j=1}^{M} T_j\big(a_j\big) = C$. If U is* strictly Schur-convex *function, then $\Delta R = 0$.*

Sum-Form Aggregators. In App. E, we show that the above results reduce to a simple convexity test for *sum-form aggregators*: a broad class of aggregators that describes most reward structures we consider in this work. This makes testing whether $\Delta R > 0$ a simple computation in many cases.

Parameterizable Families of Aggregators. A core topic of this work is *reward design*: how can we craft team objectives that either advantage or disadvantage behavioral diversity? To do this, it is helpful to first identify an appropriate search space. Our theoretical analysis enables us to narrow down this search space, and focus on aggregators whose *curvature* can be parametrized. Many *family of aggregator functions* $\{\, f_t(\cdot)\}_{t\in\mathbb{R}}$ can be parametrized by a scalar t which controls whether the aggregation is *Schur-convex* or *Schur-concave*, and how strongly it penalizes (or favors) inequalities among the components. For example, the *softmax aggregator* $\sum_{i=1}^{N} \frac{\exp\big(t\cdot r_{ij}\big)}{\sum_{\ell=1}^{N}\exp\big(t\cdot r_{\ell j}\big)}$ is parametrized by its temperature, t, transitioning from being strictly Schur-concave when $t < 0$ to strictly Schur-convex when $t > 0$. We can define a space of reward functions by selecting both the task scores and outer aggregator to be softmax functions: let $T_j(A) = \sum_{i=1}^{N} \frac{\exp\big(t\cdot r_{ij}\big)}{\sum_{\ell=1}^{N}\exp\big(t\cdot r_{\ell j}\big)}\ r_{ij}$, and let $U\big(T_1(a_1), \ldots T_M(a_m)\big) = \sum_{j=1}^{M} \frac{\exp\big(\tau\cdot T_j(A)\big)}{\sum_{\ell=1}^{M}\exp\big(\tau\cdot T_\ell(A)\big)}\ T_j(A)$, where $t, \tau \in \mathbb{R}$ parametrize the inner and outer aggregators, respectively. The heterogeneity gain is then dependent on t and τ. As a case study, we derive lower bounds on ΔR when $N = M$ in Thm. 3.4. We *conjecture* the lower bound of Thm. 3.4 is actually the exact value of $\Delta R(t, \tau; N)$, but we were unable to prove this. Fig. 2 (righthand side) plots the heterogeneity gains when $N = M = 2$.

Theorem 3.4 (Exact softmax heterogeneity gain for $N = M$)**.** *Assume $N = M \geq 2$, and let $\sigma(t, N) := \frac{e^t}{e^t+N-1}$. The heterogeneity gain for softmax aggregators* ***(i)*** *equals $\Delta R(t, \tau; N) = 0$ when $t \leq 0$;* ***(ii)*** *is bounded below by $\sigma(t, N) - \dfrac{1}{N}$ when $t > 0, \tau \leq 0$; and* ***(iii)*** *is bounded below by $\max\big\{\sigma(t, N) - \sigma(\tau, N), 0\big\}$ when $t > 0, \tau \geq 0$.*

Tab. 3 contains more examples of aggregation operators parameterized by t. These families provide a search space for potential reward functions, allowing us to sweep smoothly from $\Delta R = 0$ to $\Delta R > 0$ reward regimes. As $t \to \pm\infty$, most such aggregators converge to either $\min$ or $\max$, and often reduce to the arithmetic mean for certain parameter choices, motivating us to ask what the

Discrete and continuous heterogeneity gains

	$T = \min$	$T = \text{mean}$	$T = \max$
		Outer $U = \min$	
ΔR_{F}	0	0	$(M-1)/M$
ΔR_{D}	0	$\lfloor N/M \rfloor / N$	$\mathbb{1}_{\{N \geq M\}}$
		Outer $U = \textit{mean}$	
ΔR_{F}	0	0	$(M-1)/M$
ΔR_{D}	0	0	$(\min\{M,N\}-1)/M$
		Outer $U = \max$	
ΔR_{F}	0	0	0
ΔR_{D}	0	0	0

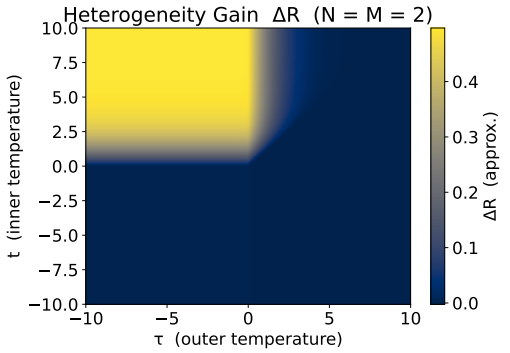


Figure 2: **Left:** Discrete (ΔR_{D}) and continuous-allocation (ΔR_{F}) heterogeneity gains for all $U, T \in \{\min, \text{mean}, \max\}$. The indicator $\mathbb{1}_{\{N \geq M\}}$ equals 1 if $N \geq M$ and 0 otherwise. **Right:** We plot the parametrized heterogeneity gains $\Delta R(t, \tau; N)$ when U and T are soft-max aggregators.

heterogeneity gain is when the outer and inner aggregator belong to the set $\{\min, \text{mean}, \max\}$. These aggregators are of special interest, since "$\min$" can be seen as a "maximally" Schur-concave function, "$\max$" can be seen as a "maximally" Schur-convex function, and "mean" is both Schur-convex and Schur-concave. Hence, it is worth asking what the heterogeneity gain is when the outer and inner aggregator belong to the set $\{\min, \text{mean}, \max\}$. We derive these gains in two cases: continuous allocations where $r_{ij} \in [0, 1]$, and discrete effort allocations where $r_{ij} \in \{0, 1\}$. The results are summarized in Fig. 2, lefthand side (formal derivation available in App. F).

4 Heterogeneous Environment Design (HED)

In complex scenarios where theory might be less applicable, we study heterogeneity through algorithmic search. We consider the setting of a Parametrized Dec-POMDP (PDec-POMDP, defined in App. K). A PDec-POMDP represents a Dec-POMDP [41], where the observations, transitions, or reward are conditioned on parameters θ. Hence, the return obtained by the agents, $G^{\theta}(\pi)$, can be differentiated with respect to θ: $\nabla_{\theta} G^{\theta}(\pi) = \frac{\partial}{\partial_{\theta}} G^{\theta}(\pi)$. In particular, computing this gradient in a differentiable simulator allows us to back-propagate through time and optimize θ via gradient ascent[4].

Algorithm 1 Heterogeneous Environment Design (HED).

input Environment parameters θ, environment learning rate α, heterogeneous agent policy π_{het}, homogeneous agent policy π_{hom}

for i in iterations **do**
  Batch$^{\theta}_{\text{het}}$= Rollout(θ,π_{het}) {rollout het policies in environment θ}
  Batch$^{\theta}_{\text{hom}}$= Rollout(θ,π_{hom}) {rollout hom policies in environment θ}
  HetGain$^{\theta}$ = ComputeGain(Batch$^{\theta}_{\text{het}}$,Batch$^{\theta}_{\text{hom}}$)
  if train_env (i) **then**
    $\theta \leftarrow \theta + \alpha \nabla_{\theta} \text{HetGain}^{\theta}$ {train environment via backpropagation}
  if train_agents(i) **then**
    $\pi_{\text{het}} \leftarrow$ MarlTrain(π_{het},Batch$^{\theta}_{\text{het}}$) {train het policies via MARL}
    $\pi_{\text{hom}} \leftarrow$ MarlTrain(π_{hom},Batch$^{\theta}_{\text{hom}}$) {train hom policies via MARL}

output final environment configuration θ, policies $\pi_{\text{het}}, \pi_{\text{hom}}$

Heterogeneous Environment Design (HED). We now consider the problem of learning the environment parameters θ to maximize the *empirical* heterogeneity gain. The empirical heterogeneity gain is defined as the difference in performance between heterogeneous and homogeneous teams in a given PDec-POMDP parametrization. Heterogeneous agents each learn different parameters, whereas homogeneous agents learn the same policy. We denote their policies as π_{het} and π_{hom}. Then, we can simply write the gain as: $\text{HetGain}^{\theta}(\pi_{\text{het}}, \pi_{\text{hom}}) = G^{\theta}(\pi_{\text{het}}) - G^{\theta}(\pi_{\text{hom}})$, repre-

[4]Although the same approach can train policies [42, 43], HED instead optimizes environment parameters and policies separately, using standard zeroth-order policy-gradient methods, to avoid being trapped in local minima.

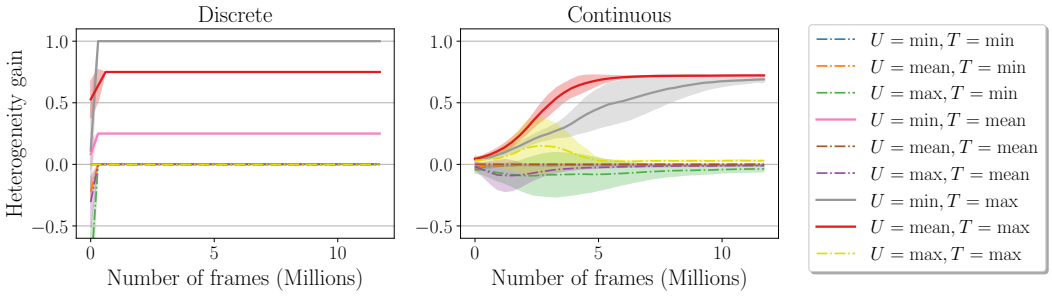


Figure 3: Heterogeneity gain for the discrete and continuous matrix games with $N = M = 4$ over training iterations. We report mean and standard deviation after 12M frames over 9 random seeds. The final results match the theoretical predictions in the Table of Fig. 2.

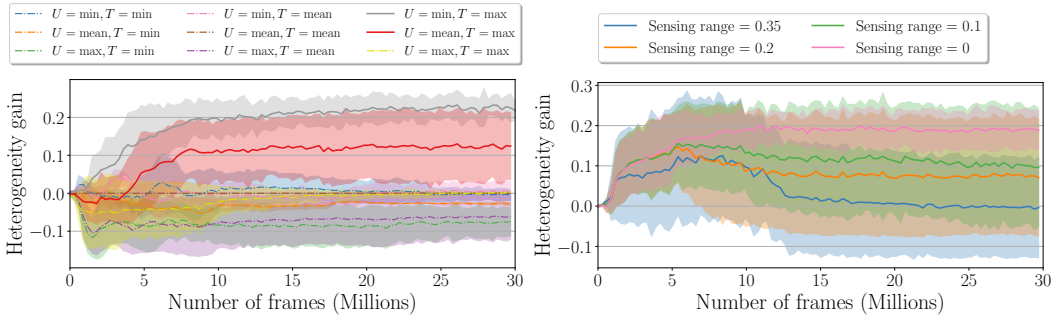
senting the return of heterogeneous agents minus that of homogeneous agents on environment parametrization θ. HED, shown in Alg. 1, learns θ by performing gradient ascent to maximize the gain: $\theta \leftarrow \theta + \alpha \nabla_\theta \text{HetGain}^\theta(\pi_{\text{het}}, \pi_{\text{hom}})$. The environment and the agents are trained in an iterative process. At every training iteration, HED collects roll-out batches in the current environment θ for both heterogeneous and homogeneous teams, computing the heterogeneity gain on the collected data. Then, it updates θ to maximize the heterogeneity gain. Finally, to train the agents, it uses MARL, with any on-policy algorithm (e.g., MAPPO [44]). The functions `train_env` and `train_agents` determine when to train each of the components in HED. In particular, we consider two possible training regimes: (1) *alternated*: where HED performs cycles of x agent training iterations followed by y environment training iterations and (2) *concurrent*: where agents train at every iteration and the environment is updated every x iterations. Note that by performing descent instead of ascent, HED can also be used to *minimize* the heterogeneity gain, which may be useful in homogeneous systems.

5 Experiments

To empirically ground our theoretical analysis, we conduct a three-stage experimental study in cooperative MARL. We first analyze a one-step, observation-free matrix game in which each agent allocates effort r_{ij} over M tasks, and consider reward structures defined by aggregator pairs $U, T \in \{\min, \text{mean}, \max\}$. We find that the agents' learned policies recover the exact heterogeneity gains derived in the theory (Fig. 2). Next, we transfer the same reward structures into an embodied, time-extended MULTI-GOAL-CAPTURE environment. We show that our curvature theory continues to be predictive in this setting as well, and discuss interesting learning dynamics emerging from the agents' embodiment. We also expose a trade-off between the heterogeneity gain and agents' observations, showing that when homogeneous agents can access richer observations, they can learn incrementally heterogeneous behaviors by leveraging behavioral typing and thus minimizing the gain: this highlights the difference between *neural* and *behavioral* heterogeneity [1].

Finally, to study HED, we parametrize the reward structure of MULTI-GOAL-CAPTURE using either parametrized Softmax or Power-Sum aggregators (App. H), and run HED to learn parameterizations that maximize the heterogeneity gain. HED learns the theoretically optimal aggregator instantiations: one the one hand, this validates HED's effectiveness at discovering heterogeneous missions, and on the other hand, showcases the generalizability of our theoretical insights to embodied environments. Implementation details and visualizations are available in App. C and App. J, respectively.

(i) Task Allocation. We consider a one-step observationless matrix game where N agents need to choose between M tasks. Their actions are effort allocations r_{ij} with $r_{ij} \geq 0, \sum_j r_{ij} = 1$, composing matrix A. With aggregators taken from the set $U, T \in \{\min, \text{mean}, \max\}$, our goal is to empirically confirm the heterogeneity gains derived in the theory *in a learning context*. Each time the game is played, all agents obtain the global reward $R(A)$ computed through the double aggregator. We consider two setups: (1) *Continuous* ($r_{ij} \in \mathbb{R}_{0 \leqslant x \leqslant 1}$): agents can distribute their efforts across tasks, (2) *Discrete* ($r_{ij} \in \{0, 1\}$): agents choose only one task. We train with $N = M = 4$ for 12 million steps. The evolution of the heterogeneity gains over training is shown in Fig. 6. The final results match *exactly* the theoretical predictions in the Fig. 2 table and our curvature theory: *concave* outer and *convex* inner aggregators favor heterogeneity. Additional results and details are in App. I.



(a) Gain for different aggregators.

(b) Gain w.r.t. observability when $U = \min, T = \max$.

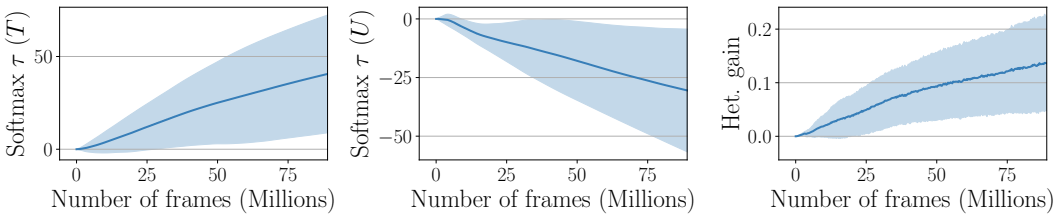
Figure 4: Heterogeneity gain for MULTI-GOAL-CAPTURE throughout training. We report mean and standard deviation for 30 million training frames over 9 (left), 4 (right) random seeds.

(ii) Multi-Goal-Capture. Next, we investigate a time-extended, embodied scenario called MULTI-GOAL-CAPTURE, based on multi-goal navigation missions [10]. Our goal is to show that the results obtained in the matrix game still hold in this more complex setting and that *aggregators that maximize the heterogeneity gain encode a practically useful global objective*.

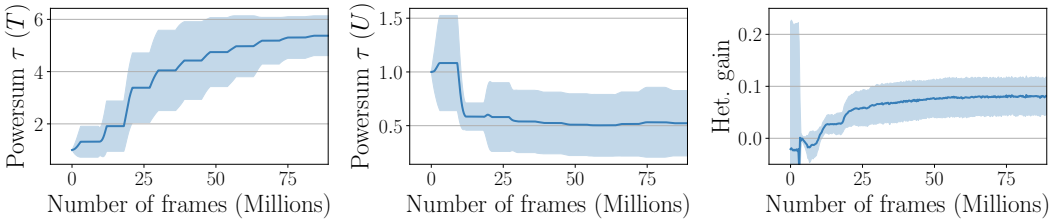
In MULTI-GOAL-CAPTURE, agents need to navigate to goals (visualization availabile in App. J). Each agent observes the relative position to the goals, and agent actions are continuous 2D forces that determine their direction of motion. The entries r_{ij}^t of matrix A^t at time t represent the local reward of agent i towards goal j, computed as $r_{ij}^t = \left(1 - d_{ij}^t / \sum_{j=1}^M d_{ij}^t\right) / (M-1)$, where d_{ij}^t is the distance between agent i and goal j. This makes it so that $\sum_{j=1}^M r_{ij}^t = 1$ and $r_{ij}^t \geqslant 0$. At each step, the agents receive the global reward $R(A^t)$, again with aggregators $U, T \in \{\min, \text{mean}, \max\}$. Different aggregator choices will yield different global objectives. For example, $U = \max, T = \max$ implies "at least one agent should go to at least one goal"; $U = \max, T = \min$ implies "all agents should go to the same goal", and so on. $U = \min, T = \max$, a concave-convex setting shown by our theory to favor heterogeneity, implies "each agent should go to a different goal and all goals should be covered" which is a natural goal for this scenario. This is because $T = \max$ encodes a task that needs just one agent to be completed (e.g., find an object), while $U = \min$ encodes that all tasks should be attended (i.e., agents need to diversify their choices).

After 30M environment frames (Fig. 4a) the empirical heterogeneity gains differ, numerically, from those of the static matrix-game because agents now realize their allocations r_{ij} through time-extended motion. *Nonetheless, our curvature theory still predicts when there is a heterogeneity gain* (Fig. 2): it is positive *only* for the concave–convex pairs $U = \min$, $T = \max$ and $U = \text{mean}$, $T = \max$. The heterogeneity gain is smaller in the latter case because learning dynamics matter: with $U = \min$, $T = \max$ the best homogeneous policy is unique–every agent must steer to the midpoint between the two goals–so homogeneous learners seldom find it, leaving room for heterogeneous policies to excel (see App. J). By contrast, $U = \text{mean}$, $T = \max$ admits a continuum of good homogeneous policies, which homogeneous teams execute more easily. For $U = \max$, $T = \min$ and $U = \max$, $T = \text{mean}$ the empirical heterogeneity gap is actually *negative*: this is because the reward peaks only when *all* agents choose the *same* goal, a coordination that heterogeneous teams learn more slowly, so they lag within the fixed training budget. Crucially, while additional training will eventually result in $\Delta R = 0$ in the latter case, the positive heterogeneity gains we report are *theoretically irreducible*, pinpointing scenarios where behavioral diversity is needed.

Observability-Heterogeneity Trade-Off: We now crystallize the relationship between environment observability and empirical heterogeneity gains. It is well known that *neurally* homogeneous agents (i.e., sharing the same neural network) can emulate diverse behavior by conditioning on the input context (behavioral typing) [12]. This can be achieved by naively appending the agent index to its observation [13] or by providing relevant observations that allows the agents to infer their role [1]. Behavioral typing is impossible in matrix games, as these games are observationless. However, it is possible in more complex games, such as our MULTI-GOAL-CAPTURE scenario. We augment agents in the positive gain scenario ($U = \min, T = \max$) with a range sensor, providing proximity readings for other agents within a radius. In Fig. 4b, we show that the heterogeneity gain decreases as the agent visibility increases (higher sensing radius). This is because, with a higher range, homogeneous



(a) HED with Softmax aggregators ($\tau \in \mathbb{R}$). $\tau \geq 0$ is Schur-convex; $\tau \leq 0$ is Schur-concave.



(b) HED with Power-Sum aggregators ($\tau \in [0.3, 6]$). $\tau \geq 1$ is Schur-convex; $\tau \leq 1$ is Schur-concave.

Figure 5: HED results in MULTI-GOAL-CAPTURE. The two leftmost columns report the evolution of aggregator parameters through training, while the rightmost column shows the obtained heterogeneity gain. This result empirically demonstrates that HED rediscovers the reward structure predicted by our theory to maximize the gain, *making the inner aggregator convex, and the outer aggregator concave*. We report mean and standard deviation for 90M training frames over 13 random seeds.

agents can sense each other and coordinate to pursue different goals. This result highlights the tight interdependence between the heterogeneity gain and agents' observations.

(iii) Heterogeneous Reward Design. We apply HED to MULTI-GOAL-CAPTURE, and ask whether it finds the same aggregator parameterizations predicted by our theory to maximize the heterogeneity gain. We turn the environment into a PDec-POMDP by parameterizing the reward as $R^\theta(A^t) = \bigoplus_{j=1}^{M}{}^{\theta} \bigoplus_{i=1}^{N}{}^{\theta} r_{ij}^t$, with parametrized inner and outer aggregators $U^\theta = \bigoplus^\theta, T^\theta = \bigoplus^\theta$. Our goal is to learn the parameters $\theta = (\tau_1, \tau_2)$, parametrizing T and U respectively, that maximize the heterogeneity gain. We consider two parametrized aggregators from Tab. 3: Softmax and Power-Sum. *Softmax*: We parameterize both U^θ and T^θ using Softmax. We initialize $\tau_1 = \tau_2 = 0$, so U and T are initially *mean*, and run HED. In Fig. 5a, we show that, to maximize the heterogeneity gain, HED learns to maximize τ_1, making the inner aggregator T Schur-convex, while minimizing τ_2, making the outer aggregator U Schur-concave. The result is exactly the reward function predicted by our theory to maximize the gain. *Power-Sum*: We parametrize both aggregators with Power-Sum. We initialize both functions to $\tau_1 = \tau_2 = 1$, representing *sum*, and run HED. We constrain $\tau_{1,2} \in [0.3, 6]$ to stabilize learning. In Fig. 5b, we show that HED learns to maximize τ_1, making T Schur-convex, while minimizing τ_2, making U Schur-concave; again rediscovering the optimal parametrization our theory predicts. Hence, these results simultaneously validate HED and our curvature theory.

6 Discussion

This work introduces tools for both *diagnosing* and *designing* reward functions that incentivize heterogeneity in cooperative MARL. In task allocation settings, our theory shows that the advantage of behavioral diversity is a predictable consequence of reward *curvature*: if the inner aggregator is Schur-convex, amplifying inequality, and the outer aggregator is Schur-concave, amplifying uniformity, heterogeneous policies are strictly superior; reversing the curvature removes the benefit. Complementing this analysis, and covering settings where our theory doesn't apply, the proposed HED algorithm automatically steers underspecified environments to either side of the diversity boundary, letting us encourage or suppress heterogeneity and providing a sandbox for studying its advantages. Together, these results help turn the choice of heterogeneity from an ad-hoc heuristic into a controllable design dimension, and help reconcile past mixed results on parameter sharing.

A key remaining open question concerns how the environment's *transition dynamics* interact with reward curvature to shape heterogeneity gains. It is also important to consider how one can extend our findings to *inseparable tasks*. We expand on these open questions, and other limitations, in App. L.

Acknowledgments and Disclosure of Funding

This work is supported by European Research Council (ERC) Project 949940 (gAIa) and ARL DCIST CRA W911NF-17-2-0181. We gratefully acknowledge their support.

References

[1] Matteo Bettini, Ajay Shankar, and Amanda Prorok. Heterogeneous multi-robot reinforcement learning. *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, page 1485–1494, 2023.

[2] Matteo Bettini, Ajay Shankar, and Amanda Prorok. System neural diversity: Measuring behavioral heterogeneity in multi-agent learning. *arXiv preprint arXiv:2305.02128*, 2023.

[3] Max Rudolph, Sonia Chernova, and Harish Ravichandar. Desperate times call for desperate measures: towards risk-adaptive task allocation. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2592–2597, 2021.

[4] Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3991–4002, 2021.

[5] Jayesh K. Gupta, Maxim Egorov, and Mykel J. Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2017.

[6] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor–critic for mixed cooperative–competitive environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[7] Brian Roberson. The colonel blotto game. *Economic Theory*, 29(1):1–24, 2006.

[8] Joseph Christian G Noel. Reinforcement learning agents in colonel blotto. *arXiv preprint arXiv:2204.02785*, 2022.

[9] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS)*, 2021.

[10] J Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.

[11] Matteo Bettini, Ryan Kortvelesy, and Amanda Prorok. Controlling behavioral diversity in multi-agent reinforcement learning. *Forty-first International Conference on Machine Learning*, 2024.

[12] Joel Z Leibo, Julien Perolat, Edward Hughes, Steven Wheelwright, Adam H Marblestone, Edgar Duéñez-Guzmán, Peter Sunehag, Iain Dunning, and Thore Graepel. Malthusian reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1099–1107, 2019.

[13] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16*, pages 66–83. Springer, 2017.

[14] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

[15] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. *Proceedings of the AAAI conference on artificial intelligence*, 32, 2018.

[16] Ryan Kortvelesy and Amanda Prorok. Qgnn: Value function factorisation with graph neural networks. *arXiv preprint arXiv:2205.13005*, 2022.

[17] Sainbayar Sukhbaatar, arthur szlam, and Rob Fergus. Learning multiagent communication with backpropagation. *Advances in Neural Information Processing Systems*, 29, 2016.

[18] Filippos Christianos, Georgios Papoudakis, Muhammad A Rahman, and Stefano V Albrecht. Scaling multi-agent reinforcement learning with selective parameter sharing. *International Conference on Machine Learning*, pages 1989–1998, 2021.

[19] Wei Fu, Chao Yu, Zelai Xu, Jiaqi Yang, and Yi Wu. Revisiting Some Common Practices in Cooperative Multi-Agent Reinforcement Learning. *International Conference on Machine Learning*, pages 6863–6877, 2022.

[20] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[21] Xinran Li, Ling Pan, and Jun Zhang. Kaleidoscope: Learnable masks for heterogeneous multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 37:22081–22106, 2024.

[22] Woojun Kim and Youngchul Sung. Parameter sharing with network pruning for scalable multi-agent deep reinforcement learning. *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1942–1950, 2023.

[23] Kale-ab Abebe Tessera, Arrasy Rahman, and Stefano V Albrecht. Hypermarl: Adaptive hypernetworks for multi-agent rl. *arXiv preprint arXiv:2412.04233*, 2024.

[24] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. ROMA: Multi-agent reinforcement learning with emergent roles. *Proceedings of the 37th International Conference on Machine Learning*, 119:9876–9886, 13–18 Jul 2020.

[25] T Wang, T Gupta, B Peng, A Mahajan, S Whiteson, and C Zhang. Rode: learning roles to decompose multi- agent tasks. *Proceedings of the International Conference on Learning Representations*, 2021.

[26] Mingyu Yang, Jian Zhao, Xunhan Hu, Wengang Zhou, Jiangcheng Zhu, and Houqiang Li. Ldsa: Learning dynamic subtask assignment in cooperative multi-agent reinforcement learning. *Advances in neural information processing systems*, 35:1698–1710, 2022.

[27] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. *International Conference on Machine Learning*, pages 3040–3049, 2019.

[28] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-Based Multi-Agent Exploration. *International Conference on Learning Representations*, 2019.

[29] Jiechuan Jiang and Zongqing Lu. The emergence of individuality. *Proceedings of the 38th International Conference on Machine Learning*, 139:4992–5001, 18–24 Jul 2021.

[30] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32, 2019.

[31] Shunyu Liu, Yihe Zhou, Jie Song, Tongya Zheng, Kaixuan Chen, Tongtian Zhu, Zunlei Feng, and Mingli Song. Contrastive identity-aware learning for multi-agent value decomposition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(10):11595–11603, 2023.

[32] Shunyu Liu, Jie Song, Yihe Zhou, Na Yu, Kaixuan Chen, Zunlei Feng, and Mingli Song. Interaction pattern disentangling for multi-agent reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[33] Tianxu Li, Kun Zhu, Juan Li, and Yang Zhang. Learning distinguishable trajectory representation with contrastive loss. *Advances in Neural Information Processing Systems*, 37:64454–64478, 2024.

[34] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

[35] Zhan Gao, Guang Yang, and Amanda Prorok. Co-optimization of environment and policies for decentralized multi-agent navigation. *arXiv preprint arXiv:2403.14583*, 2024.

[36] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

[37] Minqi Jiang, Michael D Dennis, Jack Parker-Holder, Jakob Nicolaus Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[38] Jack Parker-Holder, Minqi Jiang, Michael D Dennis, Mikayel Samvelyan, Jakob Nicolaus Foerster, Edward Grefenstette, and Tim Rocktäschel. That escalated quickly: Compounding complexity by editing levels at the frontier of agent capabilities. In *Deep RL Workshop NeurIPS 2021*, 2021.

[39] A Wayne Roberts and Dale E Varberg. *Convex Functions: Convex Functions*, volume 57. Academic Press, 1974.

[40] Josip E Peajcariaac and Yung Liang Tong. *Convex functions, partial orderings, and statistical applications*. Academic Press, 1992.

[41] Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

[42] Jie Xu, Miles Macklin, Viktor Makoviychuk, Yashraj Narang, Animesh Garg, Fabio Ramos, and Wojciech Matusik. Accelerated policy learning with parallel differentiable simulation. In *International Conference on Learning Representations*, 2022.

[43] Yunlong Song, Sang bae Kim, and Davide Scaramuzza. Learning quadruped locomotion using differentiable simulation. In *8th Annual Conference on Robot Learning*, 2024.

[44] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

[45] Omry Yadan. Hydra - a framework for elegantly configuring complex applications. Github, 2019.

[46] Matteo Bettini, Ryan Kortvelesy, Jan Blumenkamp, and Amanda Prorok. Vmas: A vectorized multi-agent simulator for collective robot learning. *Proceedings of the 16th International Symposium on Distributed Autonomous Robotic Systems*, 2022.

[47] Albert Bou, Matteo Bettini, Sebastian Dittert, Vikash Kumar, Shagun Sodhani, Xiaomeng Yang, Gianni De Fabritiis, and Vincent Moens. Torchrl: A data-driven decision-making library for pytorch. In *The Twelfth International Conference on Learning Representations*, 2023.

[48] Michael R Garey, David S Johnson, et al. A guide to the theory of np-completeness. *Computers and intractability*, pages 37–79, 1990.

[49] Aditya Modi, Nan Jiang, Satinder Singh, and Ambuj Tewari. Markov decision processes with continuous side information. In *Algorithmic learning theory*, pages 597–618. PMLR, 2018.

A Computational Resources Used

For the realization of this work, we have employed computational resources that have gone towards: experiment design, prototyping, and running final experiment results. Simulation and training are both run on GPUs, no CPU compute has been used. Results have been stored on the WANDB cloud service. We estimate:

- 300 compute hours on an NVIDIA GeForce RTX 2080 Ti GPU.
- 500 compute hours on an NVIDIA L40S GPU.

Simply verifying our results using the available code will take considerably less compute hours (around a day).

B Code and Data Availability

We attach the code in the supplementary materials. The code contains instructions on how to reproduce the experiments in the paper and dedicated YAML files containing the hyperparameters for each experiment presented. The YAML files are structured according to the HYDRA [45] framework which allows smooth reproduction as well as systematic and standardized configuration. We further attach all scripts to reproduce the plots in the paper from the experiment results.

C Implementation Details

For all experiments, we use the MAPPO MARL algorithm [44]. Environments are implemented in the multi-agent environment simulator VMAS [46], and trained using TorchRL [47]. Both the actor and critic are two-layer MLPs with 256 neurons per layer and Tanh activation. Further details, such as hyperparameter choices, are available in the attached code and YAML configuration files.

D Colonel Blotto & Level-Based Foraging

We describe how two well-known settings from the literature fit into our theoretical framework, and check what our theoretical results say about their heterogeneity gain.

D.1 Team Colonel Blotto (fixed adversary)

The *Colonel Blotto game* is a well-known allocation game studied in both game theory and MARL [7, 8]. It is used to model election strategies and other resource-based competitions. In the team variant with fixed adversary, N friendly colonels (agents) each distribute a (fixed and equal) budget of troops $r_{ij} \geq 0, \sum_{j=1}^{M} r_{ij} = 1$ across M battlefields $j \in \{1, \dots, M\}$ (our tasks). A fixed adversary selects a *stochastic* opposing allocation strategy, i.e. a distribution π_{adv} over vectors $a = (a_1, \dots, a_M)$ which is fixed throughout training and evaluation. Let $s_j = \sum_{i=1}^{N} r_{ij}$ denote the team force committed to battlefield j. Our agents win against the adversary if the troops they allocate to a given field surpass the troops allocated by the adversary. The expected value secured on battlefield j is therefore

$$T_j(a_j) \;=\; v_j\, \mathbb{E}_{a\sim\pi_{\text{adv}}}\big[\mathbb{1}[\, s_j > a_j \,]\big] \;=\; v_j \Pr_{a\sim\pi_{\text{adv}}}\big[s_j > a_j\big],$$

where $1[x > y]$ denotes the indicator function. This is a **thresholded-sum** that remains symmetric and coordinate-wise non-decreasing in every agent's contribution r_{ij}. Aggregating across battlefields with a value-weighted sum yields the team reward

$$R(A) \;=\; \sum_{j=1}^{M} T_j(a_j) \;=\; \underbrace{\sum_{j=1}^{M}}_{U}\; T_j\Big(\underbrace{\sum_{i=1}^{N}}_{\oplus}\, r_{ij}\Big),$$

so the game fits the double-aggregation structure $R(A) = \bigoplus_j \bigoplus_i r_{ij}$ assumed in our analysis.

Heterogeneity Gain: This is a continuous allocation game, and the inner aggregator T_j is an indicator function over the sum of troop allocations to battlefield j. This function is Schur-concave (and Schur-convex at the same time!). Hence, by Thm. 3.2, heterogeneous colonel teams, where each colonel has a distinct troop allocation strategy, have no advantage over homogeneous teams, where all colonels employ the same allocation strategy: $\Delta R = 0$. This makes sense, as it makes no difference whether two different colonels allocate $x/2$ troops to a battlefield, or one colonel allocates x troops to the battlefield.

Our analysis also tells us what happens when we change T_j: this provides insights for generalizations of the Colonel Blotto game. For example, maybe the troops of different colonels don't cooperate as well with each other, such that two colonels allocating $x/2$ troops to a battlefield results in a lower T_j-value than a single colonel allocating x troops. In this case, T_j becomes strictly Schur-convex, and Thm. 3.1 tells us that $\Delta R > 0$ as long as the optimal allocation is non-trivial. Hence, heterogeneous teams are advantaged.

D.2 Level-Based Foraging

The well-known *level-based foraging* (LBF) benchmark, based on the knapsack problem [48], is a deceptively challenging, embodied MARL environment, where N agents are placed on a grid with M food items, and are tasked with collecting them. Each item j has an integer level L_j that must be met or exceeded by the combined skills of the agents standing on that cell before it can be collected [9]. Let agent i's skill be e_i. At a given step the binary variable

$$r_{ij} \;\in\; \{0, e_i\}, \qquad \text{with } \sum_{j=1}^{M} r_{ij} \le e_i,$$

denotes whether i contributes its skill to item j. In our setting, we assume all agents are equally skilled, so $e_i = 1\ \forall i$. Collecting these variables thus yields an allocation matrix $A = [r_{ij}] \in \{0, e_i\}^{N \times M}$, which again matches our framework.

Inner aggregator. A food item is harvested if the summed skill on its cell reaches the threshold, so

$$T_j(a_j) \;=\; L_j \; \mathbb{1}\big[\textstyle\sum_{i=1}^{N} r_{ij} \;\ge\; L_j\big], \qquad a_j = (r_{1j}, \ldots, r_{Nj})^\top.$$

This **threshold–sum** is symmetric and monotone, depending only on the sum of its arguments and therefore simultaneously Schur-convex and Schur-concave.

Outer aggregator. The stepwise team reward is the sum of harvested item values,

$$R(A) \;=\; \sum_{j=1}^{M} T_j\Big(\sum_{i=1}^{N} r_{ij}\Big) \;=\; \underbrace{\sum_{j=1}^{M}}_{U}\; T_j(\underbrace{\sum_{i=1}^{N}}_{\oplus}\; r_{ij}) \;=\; \bigoplus_{j=1}^{M}\bigoplus_{i=1}^{N} r_{ij},$$

so LBF also conforms to the double-aggregation form $R(A) = \bigoplus_j \bigoplus_i r_{ij}$.

MARL Environment Reward. In the level-foraging environment, items that are picked up either disappear; replace themselves with different items; or replace themselves with the same item (possibly at a different cell). In all of these cases we can represent the cumulative reward as $\sum_{t=0}^{T} \gamma^t R_t\big(A_t\big)$ for some sequence $(R_t)_{t=1,\ldots T}$ of rewards adhering to the above reward structure.

Heterogeneity Gain: We analyze the heterogeneity gap of a specific stepwise reward R.

Because this is an embodied environment where each agent can either stand on an item ($r_{ij} = 1$) or not ($r_{ij} = 0$), effort allocations are *discrete*. Our continuous curvature test therefore does not apply directly, but the discrete analysis in Fig. 2 (left panel) does.

The table in Fig. 2 tells us something about the case where all items have level $L_j = 1$. In this case, since we assumed $e_i = 1$ for all agents, the inner aggregator reduces to

$$T_j(a_j) = \mathbb{1}\Big[\sum_{i=1}^{N} r_{ij} \geq 1\Big] = \max_i r_{ij},$$

while the outer aggregator is an unnormalized sum, which becomes the *mean* when divided by M. Hence $R(A) = \sum_j \max_i r_{ij}$, which, up to the constant $1/M$, is exactly the case $U =$ mean, $T =$ max of Fig. 2. That table shows

$$\frac{1}{M}\Delta R \;=\; \frac{\min\{M, N\} - 1}{M},$$

so the heterogeneity gap is *strictly positive* whenever the team could in principle cover more than one item ($\min\{M, N\} > 1$). Intuitively, a homogeneous team can only collect one item per step (all agents flock to the same cell), whereas heterogeneous agents may spread out and capture up to $\min\{M, N\}$ items simultaneously.

This analysis can be extended to the case where all items have the same level $L > 1$ and $L \mid N$ by grouping agents into $\tilde{N} := N/L$ *agent teams*, each bundle contributing exactly L units of skill. This yields

$$\frac{1}{ML}\Delta R \;=\; \frac{\min\{M, \tilde{N}\} - 1}{M}.$$

(We omit the formal analysis, which is not difficult). Thus, if the team can form at least two such bundles ($\tilde{N} > 1$), heterogeneity is again advantageous. If it cannot, then $\Delta R = 0$, and there is no advantage to heterogeneity.

When the levels $\{L_j\}$ differ, an exact closed form is harder, but in general we expect $\Delta R > 0$ whenever there is some combination of items that the heterogeneous team can collect, which in total is worth more than the largest single item that can be collected if all N agents stand on its cell.

In LBF, therefore, our theory suggests that behavioral diversity is often advantageous. Note that (unlike the Colonel Blotto game) since LBF is an embodied, time-extended MARL environment, this analysis does not *formally guarantee* an advantage to RL-based heterogeneous agent teams: rather, it identifies that there are effort allocation strategies that will give these teams an advantage over homogeneous teams. The agents must still *learn* and be able to execute these strategies to gain this advantage (e.g., they must learn how to move to attain the desired allocations).

E Sum-Form Aggregators

Many useful reward functions are *sum-form aggregators*:

Definition E.1 (Sum-Form Aggregator)**.** *A task-level aggregator $f : \mathbb{R}^N \to \mathbb{R}$ for task j is a **sum-form aggregator** if it can be written as: $f(x_j) = \sum_{i=1}^{N} g(x_j)$, where $g_j : \mathbb{R} \to \mathbb{R}$ is differentiable. We say f is (strictly)* convex or concave *if g is (strictly) convex or concave, respectively.*

Tab. 3 contains examples. When our aggregators have this form, Schur-convexity (concavity) is determined by whether g is convex (concave)–a simple computational test. This is because of the following *known* connection between sum-form aggregators and Schur-convexity/concavity:

Lemma E.1 (Schur Properties of Sum-Form Aggregators [40])**.** *Given sum-form task-level aggregator $f(x) = \sum_{i=1}^{N} g(x_i)$, the following holds:* **(i)** *if g is (strictly) convex, then f is (strictly) Schur-convex; and* **(ii)** *if g is (strictly) concave, then f is (strictly) Schur-concave.*

This lemma simplifies checking the conditions of our heterogeneity gain results. For example, the following corollary can be used to establish $\Delta R > 0$ for many of the aggregators in Tab. 3:

Corollary E.1 (Convex-Concave Positive Heterogeneity Gain)**.** *Let $N, M \geq 2$. Let $g : [0, 1] \to \mathbb{R}_{\geq 0}$ be a non-negative strictly convex function satisfying $g(0) = 0$, and let $h : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be a strictly concave, increasing function satisfying $h(0) = 0$. If each task-level aggregator is a strictly convex sum-form aggregator $T_j(a_j) = \sum_{i=1}^{N} g\big(r_{ij}\big)$, and the outer aggregator is a strictly concave sum-form aggregator $U(y) = \sum_{j=1}^{M} h(y_j)$, then $\Delta R > 0$.*

Proof of Corollary E.1. We will apply Theorem 3.1 by verifying its conditions:

First, by Lemma E.1, since g is strictly convex, the (identical) task-level aggregators $T_j(x) = \sum_{i=1}^{N} g(x_i)$ are strictly Schur-convex, satisfying condition (i) of Theorem 3.1.

Second, the outer aggregator $U(y_1, \ldots, y_M)$ is strictly increasing at every coordinate by definition, satisfying condition (ii).

Hence, the conditions of Thm. 3.1 apply. To establish $\Delta R > 0$, it remains to verify that the optimal allocation is non-trivial: it distributes effort across at least two tasks. In any admissible *homogeneous* solution, each of the N agents chooses the same effort-distribution $(c_1, \ldots, c_M)$ on tasks, with $\sum_j c_j = 1$. Then task j's reward is $T_j = N\, g(c_j)$, so $R(A) = \sum_{j=1}^{M} h\big(N\, g(c_j)\big)$. The trivial, *all-agent single-task allocation* uses $(c_j = 1, c_{k\neq j} = 0)$. Its reward is therefore $R_{\text{corner}} = h\big(N\, g(1)\big) + \sum_{k\neq j} h\big(N\, g(0)\big) = h\big(N\, g(1)\big)$ since $g(0) = 0$ and $h(0) = 0$.

Strict concavity of h implies that $h\big(N\, g(1)\big) < N \cdot h(g(1))$. Hence, agents can attain a better reward by allocating effort 1 to N different tasks rather than a single task. This shows that the best solution *must* use at least two nonzero c_j, completing the proof. □

F Formal Analysis

F.1 Proof of Thm. 3.1

Proof of Thm. 3.1. Let A_{hom} be an optimal homogeneous allocation (i.e., $R(A_{hom}) = R_{hom}$), whose ith row is the vector

$$c = (c_1, \ldots, c_M) \quad \text{with} \quad \sum_{j=1}^{M} c_j = 1.$$

Then each column j of A_{hom} is the uniform vector $u_j = (c_j, c_j, \ldots, c_j)^\top \in \mathbb{R}^N$. Hence the task-level reward is $T_j(u_j)$, and the overall reward is

$$R\big(A_{\text{hom}}\big) = U\big(T_1(u_1), \ldots, T_M(u_M)\big).$$

Because $\sum_j c_j = 1$, there is at least one task j with $c_j > 0$. We construct a heterogeneous allocation A_{het} such that each column x_j in A_{het} has the same sum as the corresponding column in A_{hom}.

The total effort allocated to a task j can be expressed as $\lfloor Nc_j \rfloor + f_k$, where $0 \leq f_j < 1$. First, we assign $\lfloor Nc_j \rfloor$ agents to allocate effort 1 to task j, for every task j. These agents are all distinct. This leaves us with $\sum_j f_j = N - \sum_j \lfloor Nc_j \rfloor$ agents that have not allocated any effort yet. Let i be the first of those agents. We have agent i allocate f_1 effort to task 1, f_2 effort to task 2, and so on, until we arrive at a task k such that $f_1 + \ldots + f_k = 1 + s$, for some $s > 0$. We have i allocate $f_k - s$ to this task k. Then, we move to agent $i + 1$, and allocate the remaining fractional efforts in the same manner (and in particular, allocating s effort to task k), until agent $i + 1$ overflows. Then we move to agent $i + 2$, and so on. This ensures that we have allocated N effort in total across the agents, and that every agent's effort allocation sums exactly to 1, so is feasible.

Let x_j be the jth column of A_{het}. We note the following fact: any non-uniform vector whose sum is Nc_j majorizes the uniform vector u_j. Hence, $T_j(x_j) \geq T_j(u_j)$, with equality only if $x_j = u_j$. This means that if $A_{het} \neq A_{hom}$, then

$$R\big(A_{\text{het}}\big) = U\big(T_1(x_1), \ldots, T_M(x_M)\big) > U\big(T_1(u_1), \ldots, T_M(u_M)\big) = R\big(A_{\text{hom}}\big).$$

We note that $A_{hom} = A_{het}$ only if A_{hom} is a trivial allocation, as A_{het} contains at least one agent allocating effort 1 to some task, and A_{hom}'s agents only allocate fractional efforts, if it is non-trivial. Otherwise, since $R(A_{hom}) = R_{hom}$, the above inequality implies $\Delta R = R_{\text{het}} - R_{\text{hom}} > 0$. This completes the proof. □

F.2 Proof of Thm. 3.2

Proof of Thm. 3.2. Let A be an arbitrary feasible allocation, and let A_{hom} be a *homogeneous* allocation with the same column sums. Concretely, for each column j, define

$$s_j \;=\; \sum_{i=1}^{N} r_{ij} \quad \text{and} \quad u_j \;=\; \big(\tfrac{s_j}{N}, \tfrac{s_j}{N}, \ldots, \tfrac{s_j}{N}\big)^{\top},$$

so u_j is the *uniform* distribution of total mass s_j across N agents. Then construct

$$A_{\text{hom}} \;=\; \begin{pmatrix} \frac{s_1}{N} & \cdots & \frac{s_M}{N} \\ \vdots & \ddots & \vdots \\ \frac{s_1}{N} & \cdots & \frac{s_M}{N} \end{pmatrix},$$

which is clearly *homogeneous* (each row is the same), and respects each column sum s_j. Since $\sum_j s_j = N$, each row sums to 1, hence the allocation is feasible. By Schur-concavity of T_j, for each column j we have

$$a_j \;\succ\; u_j \quad\Longrightarrow\quad T_j(a_j) \;\le\; T_j(u_j),$$

unless a_j is u_j. In other words, *any* deviation from the uniform vector with the same sum $\sum_{i=1}^{N} a_{ji} = s_j$ will not increase $T_j(a_j)$ under Schur-concavity. Hence for each column j of A, $T_j(a_j) \;\le\; T_j(u_j)$,. Since U is non-decreasing in each coordinate,

$$R\big(A\big) \;=\; U\big(T_1(a_1), \ldots, T_M(a_M)\big) \;\le\; U\big(T_1(u_1), \ldots, T_M(u_M)\big) \;=\; R\big(A_{\text{hom}}\big).$$

This implies $\Delta R \;=\; 0$. □

F.3 Proof of Thm. 3.3

Proof of Thm. 3.3. By hypothesis, the components of the task score vector

$$\mathbf{T}(\mathbf{A}) = \Big(T_1(a_1),\, T_2(a_2),\, \ldots,\, T_M(a_M)\Big)$$

always sum to C. By strict Schur-convexity, the maximum value of U over such vectors is attained precisely at an extreme point of the C-simplex, i.e. at some permutation of $(C, 0, \ldots, 0)$. Hence, we seek to find an allocation of efforts, A_{corner}, that causes $\mathbf{T}(\mathbf{A})$ to equal this vector.

Let each agent i invest *all* of its effort into task 1. This is the trivial allocation. Then the first column of A_{corner} is $(1, 1, \ldots, 1)^{\top}$, and all other columns a_j are zero. Since task scores sum to C, we get $T_1\big(a_1\big) = C, \quad T_j\big(a_j\big) = 0 \;\; \text{for } j \neq 1$. By assumption (2), we infer that the vector of task-level scores is indeed $(C, 0, \ldots, 0)$.

Notice that *each row of A_{corner} is the same* $(1, 0, \ldots, 0)$, making A_{corner} a *homogeneous* allocation. Hence, we attained the maximum possible reward $R(\mathbf{A})$ through a homogeneous allocation, implying $\Delta R = 0$. □

F.4 Proof of Thm. 3.4

Before proving the statement, let's write the expressions for homogeneous and heterogeneous optima. For each task j, we defined

$$T_j(A) = \sum_{i=1}^{N} \frac{\exp\big(t \cdot r_{ij}\big)}{\sum_{\ell=1}^{N} \exp\big(t \cdot r_{\ell j}\big)}\; r_{ij},$$

while defining the outer aggregator to be

$$U\big(T_1(a_1), \ldots T_M(a_m)\big) = \sum_{j=1}^{M} \frac{\exp\big(\tau \cdot T_j(A)\big)}{\sum_{\ell=1}^{M} \exp\big(\tau \cdot T_\ell(A)\big)}\; T_j(A),$$

where $t, \tau \in \mathbb{R}$ are temperature parameters. In the **homogeneous setting**, where all agents share the same allocation $c = (c_1, \ldots, c_M)$, we therefore have $T_j(A) = \sum_{i=1}^{N} \frac{\exp\big(t\, c_j\big)}{\sum_{\ell=1}^{N} \exp\big(t\, c_j\big)}\, c_j = c_j$. Thus,

$$R_{\text{hom}} = \max_{c \in \Delta^{M-1}} \quad \sum_{j=1}^{M} \frac{\exp\big(\tau\, c_j\big)}{\sum_{\ell=1}^{M} \exp\big(\tau\, c_\ell\big)}\, c_j$$

where Δ^{M-1} is the simplex of all admissible allocations.

In the general **heterogeneous setting**, each row $(r_{i1}, \ldots, r_{iM})$ can be different. Then

$$T_j(A) \;=\; \sum_{i=1}^{N} \frac{\exp\big(t\, r_{ij}\big)}{\sum_{\ell=1}^{N} \exp\big(t\, r_{\ell j}\big)}\, r_{ij},$$

and we choose $A \in (\Delta^{M-1})^N$ to maximize

$$R_{\text{het}} \;=\; \max_{A} \sum_{j=1}^{M} \frac{\exp\big(\tau\, T_j(A)\big)}{\sum_{k=1}^{M} \exp\big(\tau\, T_k(A)\big)}\, T_j(A).$$

Keeping these expressions in mind, we proceed with the proof of Thm. 3.4.

Reminder: assuming $N = M \geq 2$, we want to prove $\Delta R(t, \tau; N) = 0$ when $t \leq 0$, and

$$\boxed{\Delta R(t, \tau; N) \geq \begin{cases} \sigma(t, N) - \dfrac{1}{N}, & t > 0, \;\; \tau \leq 0, \\ \max\big\{\sigma(t, N) - \sigma(\tau, N),\, 0\big\}, & t > 0, \;\; \tau \geq 0. \end{cases}}$$

otherwise, where $\sigma(t, N) \;:=\; \frac{e^t}{e^t + N - 1}$.

Proof of Thm. 3.4. When $t \leq 0$, T_j is Schur–concave, so $\Delta R = 0$ by Thm. 3.2. We assume $t > 0$ for the rest of the proof.

Homogeneous optimum. If every row of A equals the same allocation $c \in \Delta^{N-1}$, then $T_j(A) = c_j$. U is Schur–concave for $\tau \leq 0$, and Schur–convex for $\tau \geq 0$, hence it is maximized by the uniform distribution in the former case, and by a 1-hot vector in the latter case, yielding:

$$R_{\text{hom}} \;=\; \max_{c \in \Delta^{N-1}} U(c) \;=\; \begin{cases} \dfrac{1}{N}, & \tau \leq 0, \\ \sigma(\tau, N), & \tau > 0. \end{cases} \tag{H}$$

Lower bound on R_{het}. The *trivial* allocation, where every agent works on the same task, produces $R_{\text{trivial}} = \sigma(\tau, N)$. The *spread* allocation, where agent i works exclusively on task i, makes each column "one-hot"; this gives $T_j = \sigma(t, N)$ for all j, and plugging this into U, we get $R_{\text{spread}} = \sigma(t, N)$. Consequently

$$R_{\text{het}} \;\geq\; \max\{\sigma(t, N),\, \sigma(\tau, N)\}. \tag{L}$$

Combining (H) and (L) gives the desired lower bound. □

G Deriving the $\{\min, \text{mean}, \max\}$ heterogeneity gains in the Fig. 2 table

We derive these heterogeneity gain case-by-case. Tab. 1 summarizes the derivation for continuous allocations ($r_{ij} \in [0, 1]$), and Tab. 2 does the same for discrete effort allocations ($r_{ij} \in \{0, 1\}$).

Table 1: All nine extreme cases of inner/outer aggregators belonging to the set $\{\min, \text{mean}, \max\}$. In each cell, we show the best possible outcome for Heterogeneous vs. Homogeneous allocations and the resulting ΔR.

	$T = \min$	$T = \text{mean}$	$T = \max$
$U = \min$	**Inner:** $T_j = \min_i r_{ij}$. **Best** R_{het}, R_{hom}**:** All must have $r_{ij} \geq x$ to push $\min_i r_{ij} = x$, so $x \leq 1/M$. $\implies T_j = 1/M$. **Outer:** $\min_j T_j = 1/M \implies R = 1/M$. **Gap:** 0.	**Inner:** $T_j = \frac{1}{N}\sum_i r_{ij}$ (avg over i). **Maximize** $\min_j T_j$**:** Both R_{het}, R_{hom} must make T_j all equal (for best min), so $T_j = 1/M$. **Outer:** $\min_j T_j = 1/M \implies R = 1/M$. **Gap:** 0.	**Inner:** $T_j = \max_i r_{ij}$. **Outer:** picks $\min_j T_j$. R_{het}: $\min_j T_j = 1 \implies R = 1$. R_{hom}: $\min_j T_j = 1/M \implies R = 1/M$. **Gap:** $1 - \frac{1}{M} = \frac{M-1}{M}$.
$U = \text{mean}$	**Inner:** $T_j = \min_i r_{ij} = 1/M$. **Outer:** simple avg $\frac{1}{M}\sum_j T_j$. Since $\sum_j T_j = M \cdot (1/M) = 1 \implies R = 1/M$. Both R_{het}, R_{hom} same $\implies \Delta R = 0$. **Gap:** 0.	**Inner:** $T_j = \frac{1}{N}\sum_i r_{ij}$. Then $\sum_j T_j = 1$. **Outer:** avg$= \frac{1}{M}\sum_j T_j$. Hence $R = \frac{1}{M} \cdot 1 = \frac{1}{M}$. Same for R_{het}, R_{hom}. **Gap:** 0.	**Inner:** $T_j = \max_i r_{ij}$. **Outer:** avg$= \frac{1}{M}\sum_j T_j$. R_{het}: sum $= M \implies R = 1$. R_{hom}: sum $= 1 \implies R = 1/M$. **Gap:** $1 - \frac{1}{M} = \frac{M-1}{M}$.
$U = \max$	**Inner:** $T_j = \min_i r_{ij}$ can be made 1 for one task. **Outer:** picks $\max_j T_j = 1 \implies R = 1$ Same for R_{het}, R_{hom}. **Gap:** 0.	**Inner:** $T_j =$ avg over i. **Outer:** picks $\max_j T_j$. Both R_{het}, R_{hom} can put all effort into one task to get $T_j = 1$, so $R = 1$. **Gap:** 0.	**Inner:** $T_j = \max_i r_{ij}$. **Outer:** picks $\max_j T_j$. Both R_{het}, R_{hom} can achieve $\max_j = 1 \implies R = 1$. **Gap:** 0.

Table 2: A "9 extreme cases" table for *discrete, one-task-per-agent* allocations.

	min	mean	max
min	**Inner:** $T_j \to \begin{cases} 1, & \text{if all agents pick } j, \\ 0, & \text{otherwise.} \end{cases}$ **Outer:** $\min_j T_j$. To get $R > 0$, must have $T_j > 0$ for *every* j (i.e. all agents pick *all* tasks, impossible). Hence $R_{\text{het}} = R_{\text{hom}} = 0$ typically, $\Delta R = 0$.	**Inner:** $T_j = \frac{\lvert\mathcal{I}_j\rvert}{N}$ **Outer:** $\min_j T_j$. $R_{het} = \lfloor N/M \rfloor / N$. $R_{hom} = 0$. $\Delta R = \lfloor N/M \rfloor / N$.	**Inner:** $T_j \to \begin{cases} 1, & \text{if at least 1 agent picks } j, \\ 0, & \text{if no agent picks } j. \end{cases}$ **Outer:** $\min_j T_j$. - *Heterogeneous* can choose s distinct tasks. If want $\min_j = 1$, must pick *all* M tasks. That requires $N \geq M$. Then $R = 1$. - *Homogeneous* covers only 1 task $\implies \min_j = 0$ for $M > 1 \implies R = 0$. $\Delta R = 1$ if $N \geq M$, else 0.
mean	**Inner:** $T_j = 1$ only if all pick j, else 0. Summation $\sum_j T_j$ is number of tasks chosen by *all* agents. Usually 0 or 1. **Outer:** Average across j. $R = \frac{1}{M}\sum_j T_j$. $\implies R = 1/M$, $\Delta R = 0$.	**Inner:** $T_j = \frac{\lvert\mathcal{I}_j\rvert}{N}$. **Outer:** Average across tasks: $R = \frac{1}{M}\sum_{j=1}^{M} \frac{\lvert\mathcal{I}_j\rvert}{N} = \frac{1}{M}$. No matter how agents are distributed, $\sum_{j=1}^{M} \lvert\mathcal{I}_j\rvert = N$. Hence $R_{\text{het}} = R_{\text{hom}} = \frac{1}{M}$, $\Delta R = 0$.	**Inner:** $T_j = 1$ if chosen by at least 1 agent, else 0. **Outer:** Average across j: $\frac{1}{M}\sum_j T_j$. This is $\frac{1}{M} \cdot \big(\#$ of tasks chosen$\big)$. - *Heterogeneous* can pick up to $\min(M, N)$ tasks, so $R = \frac{\min(M,N)}{M}$. - *Homogeneous* covers exactly 1 task $\implies R = 1/M$. $\Delta R = \frac{\min(M,N)-1}{M}$.
max	**Inner:** $T_j = 1$ only if all pick j, else 0. **Outer:** $\max_j T_j$. $\Delta R = 0$.	**Inner:** $T_j = \lvert\mathcal{I}_j\rvert / N$. **Outer** ($\tau \to +\infty$)**:** $\max_j T_j$. We can place *all* agents on one task, get $T_j = 1$. Then $R = 1$. Same for homogeneous or heterogeneous. $\Delta R = 0$.	**Inner:** $T_j = 1$ if at least 1 picks j, else 0. **Outer:** $\max_j T_j = 1$ if any agent picks j. Even a single task yields $R = 1$. So $R_{\text{hom}} = R_{\text{het}} = 1$, $\Delta R = 0$.

H Parametrized Families of Aggregators

The Table in this section illustrates several families of *generalized aggregators* that the analysis in this paper applies to. The scalar t parametrizes each family of aggregators, continuously shifting the aggregators from Schur-concave to Schur-convex.

Table 3: Illustrative families of parametric (and one nonparametric) aggregators $f_t(x)$. Changing the real parameter t can switch between Schur-convex and Schur-concave behaviors (on nonnegative inputs), or control how strongly the aggregator favors "peaked" vs. "uniform" distributions. As $t \to \pm\infty$ or $t \to 0$, many reduce to well-known extremes such as $\max$, $\min$, or the arithmetic mean.

Name	Definition	Schur Property & Limits
Power-Sum	$f_t(x) = \sum_{i=1}^{N} (x_i)^t, \quad x_i \geq 0, \quad t > 0$	• Strictly *Schur-convex* for $t > 1$. • Strictly *Schur-concave* for $0 < t < 1$. • At $t = 1$, it is linear (both Schur-convex and Schur-concave). • Undefined at $t \leq 0$ if any $x_i = 0$, though one can extend with limits.
Power-Mean	$M_t(x) = \left(\frac{1}{N} \sum_{i=1}^{N} (x_i)^t \right)^{1/t}, \quad x_i \geq 0,\ t \neq 0$	• Strictly *Schur-convex* for $t > 1$. • Strictly *Schur-concave* for $0 < t < 1$. • Reduces to arithmetic mean at $t = 1$. • As $t \to \infty$, converges to $\max_i x_i$; as $t \to -\infty$, converges to $\min_i x_i$.
Log-Sum-Exp (LSE)	$\mathrm{LSE}_t(x) = \frac{1}{t} \ln\left(\sum_{i=1}^{N} e^{t\, x_i} \right), \quad t \neq 0$	• Strictly *Schur-convex* for $t > 0$. • Strictly *Schur-concave* for $t < 0$. • As $t \to \infty$, approaches $\max_i x_i$; as $t \to -\infty$, approaches $\min_i x_i$.
Softmax Aggregator	$\mathrm{Softmax}_t(x) = \sum_{i=1}^{N} \frac{e^{t\, x_i}}{\sum_{j=1}^{N} e^{t\, x_j}} x_i, \quad t \in \mathbb{R}$	• Strictly *Schur-convex* for $t > 0$. • Strictly *Schur-concave* for $t < 0$. • As $t \to \infty$, converges to $\max_i x_i$; as $t \to -\infty$, converges to $\min_i x_i$. • At $t = 0$, each weight is $\frac{1}{N}$, so $\mathrm{Softmax}_0(x) = \frac{1}{N} \sum_i x_i$.

I Additional Results in the Multi-Agent Multi-Task Matrix Game

We report further details and results on the heterogeneity gains obtained in the multi-agent multi-task matrix game.

I.1 Game formulation

In Tab. 4 we provide an example of the pay-off matrix in this game for $N = M = 3$.

Table 4: Example of a Multi-Agent Multi-Task matrix game for $N = M = 3$. Agents choose their actions $A = (r_{ij})$ and receive the global reward $R(A) = \bigoplus_{j=1}^{M} \bigoplus_{i=1}^{N} r_{ij}$.

		Tasks		
		1	2	3
Agents	1	r_{11}	r_{12}	r_{13}
	2	r_{21}	r_{22}	r_{23}
	3	r_{31}	r_{32}	r_{33}

I.2 $N = M = 2$

We train with $N = 2, M = 2$ for 100 training iterations (each consisting of 60,000 frames). We report the results for the **continuous** case in Tab. 5 and for the **discrete** case in Tab. 6. The evolution of the heterogeneity gains over training is shown in Fig. 6.

Table 5: Heterogeneity gain $\Delta R \in \mathbb{R}_{0 \leqslant x \leqslant 1}$ of the **continuous** matrix game with $N = M = 2$. The results match the theoretical analysis in the Table of Fig. 2. We report mean and standard deviation after 6 million training frames over 9 different random seeds.

		T		
		Min	Mean	Max
U	Min	-0.002 ± 0.002	0.000 ± 0.003	0.504 ± 0.007
	Mean	-0.002 ± 0.002	0.000 ± 0.000	0.496 ± 0.001
	Max	-0.003 ± 0.002	-0.001 ± 0.001	0.003 ± 0.001

Table 6: Heterogeneity gain $\Delta R \in \mathbb{R}_{0 \leqslant x \leqslant 1}$ of the **discrete** matrix game with $N = M = 2$. The results match the theoretical analysis in Fig. 2. We report mean and standard deviation after 6 million training frames over 9 different random seeds.

		T		
		Min	Mean	Max
U	Min	0.0 ± 0.0	0.5 ± 0.0	1 ± 0.0
	Mean	0.0 ± 0.0	0.0 ± 0.0	0.5 ± 0.0
	Max	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

I.3 $N = M = 4$

In the case $N = M = 4$, the evolution of the heterogeneity gains during training is shown in Fig. 3. We further report the final obtained gains for the **continuous** case in Tab. 7 and for the **discrete** case in Tab. 8.

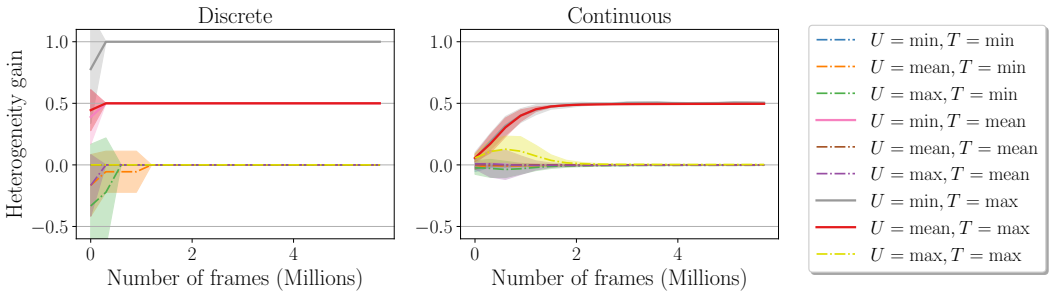


Figure 6: Heterogeneity gain for the discrete and continuous matrix games with $N = M = 2$ over training iterations. We report mean and standard deviation after 6 million training frames over 9 different random seeds. The final results match the theoretical predictions in Fig. 2.

Table 7: Heterogeneity gain $\Delta R \in \mathbb{R}_{0 \leqslant x \leqslant 1}$ of the **continuous** matrix game with $N = M = 4$. The results match the theoretical analysis in Fig. 2. We report mean and standard deviation after 12 million training frames over 9 different random seeds.

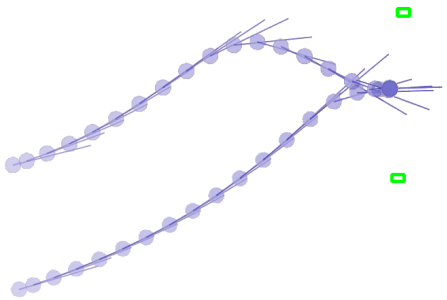
		T		
		Min	Mean	Max
U	Min	-0.003 ± 0.002	0.000 ± 0.001	0.690 ± 0.026
	Mean	-0.002 ± 0.000	0.000 ± 0.000	0.722 ± 0.002
	Max	-0.037 ± 0.023	-0.009 ± 0.005	0.029 ± 0.006

Table 8: Heterogeneity gain $\Delta R \in \mathbb{R}_{0 \leqslant x \leqslant 1}$ of the **discrete** matrix game with $N = M = 4$. The results match the theoretical analysis in the Table of Fig. 2. We report mean and standard deviation after 12 million training frames over 9 different random seeds.

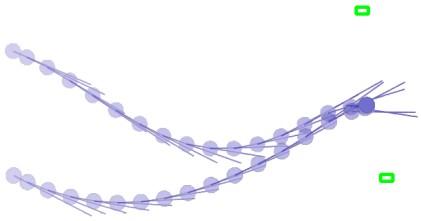
		T		
		Min	Mean	Max
U	Min	0.0 ± 0.0	0.25 ± 0.0	1.0 ± 0.0
	Mean	0.0 ± 0.0	0.0 ± 0.0	0.75 ± 0.0
	Max	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

J MULTI-GOAL-CAPTURE

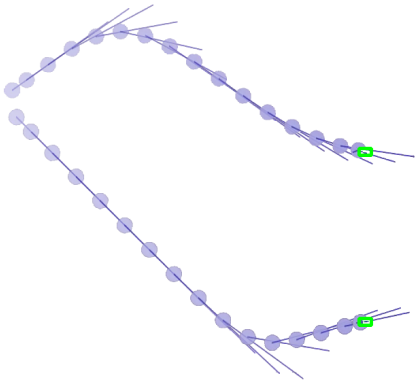
In Fig. 7 we juxtapose two representative $N = M = 2$ roll-outs of the MULTI-GOAL-CAPTURE environment for *homogeneous* teams (top row) and *heterogeneous* teams (bottom row) when $U = \min$, $T = \max$. Consistent with the discussion in Sec. 5, homogeneous agents steer to the geometric midpoint between the two goals, producing almost overlapping paths–this is suboptimal, as they cannot cover both goals. On the other hand, heterogeneous agents exaggerate their differences–taking sharply diverging trajectories and ensuring one goal each.



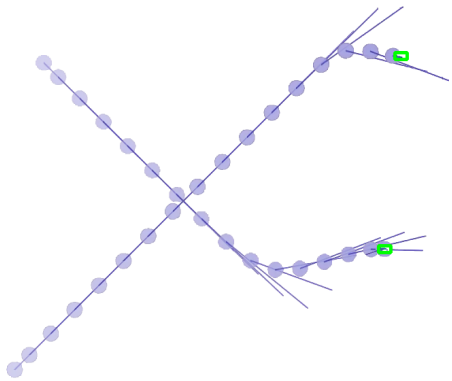
(a) Homogeneous run 1



(b) Homogeneous run 2



(c) Heterogeneous run 1



(d) Heterogeneous run 2

Figure 7: **Behaviour under the concave–convex aggregator $U = \min$, $T = \max$.** Each dot is an agent position; line segments indicate instantaneous velocity; green squares mark goal locations. Homogeneous policies collapse to a single "mid-point" route, while heterogeneous policies split and follow distinct paths to cover both goals. Note how the heterogeneous agents *exaggerate* the difference in their trajectories, rather than head directly to the goal: this is an outcome of the reward structure, which encourages maximal diversity.

K Parametrized Dec-POMDP

A Parametrized Decentralized Partially Observable Markov Decision Process (PDec-POMDP) is defined as a tuple

$$\left\langle \mathcal{N}, \mathcal{S}, \{\mathcal{O}_i\}_{i\in\mathcal{N}}, \{\sigma_i^\theta\}_{i\in\mathcal{N}}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \mathcal{R}^\theta, \mathcal{T}^\theta, \gamma, s_0^\theta \right\rangle_\theta,$$

where $\mathcal{N} = \{1, \dots, n\}$ denotes the set of agents, $\mathcal{S}$ is the state space, and, $\{\mathcal{O}_i\}_{i \in \mathcal{N}}$ and $\{\mathcal{A}_i\}_{i \in \mathcal{N}}$ are the observation and action spaces, with $\mathcal{O}_i \subseteq \mathcal{S},\ \forall i \in \mathcal{N}$. Further, $\left\{\sigma_i^\theta\right\}_{i \in \mathcal{N}}$ and $\mathcal{R}^\theta$ are the agent observation and reward functions, such that $\sigma_i^\theta : \mathcal{S} \mapsto \mathcal{O}_i$, and, $\mathcal{R}^\theta : \mathcal{S} \times \{\mathcal{A}_i\}_{i \in \mathcal{N}} \mapsto \mathbb{R}$. $\mathcal{T}^\theta$ is the stochastic state transition model, defined as $\mathcal{T}^\theta : \mathcal{S} \times \{\mathcal{A}_i\}_{i \in \mathcal{N}} \mapsto \Delta\mathcal{S}$, which outputs the probability $\mathcal{T}^\theta(s^t, \{a_i^t\}_{i \in \mathcal{N}}, s^{t+1})$ of transitioning to state $s^{t+1} \in \mathcal{S}$ given the current state $s^t \in \mathcal{S}$ and actions $\{a_i^t\}_{i \in \mathcal{N}}$, with $a_i^t \in \mathcal{A}_i$. γ is the discount factor. Finally, $s_0^\theta \in \mathcal{S}$ is a the initial environment state. A PDec-POMDP represents a set of traditional Dec-POMDPs [41], where the observation function, the transition function, the reward function, and the initial state are conditioned on parameters θ. This formalism is similar to the concepts of Underspecified POMDP [36] and contextual MDP [49].

Agents are equipped with (possibly stochastic) policies $\pi_i(a_i|o_i)$, which compute an action given a local observation. Their objective is to maximize the discounted return:

$$G^\theta(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t \mathcal{R}^\theta \left(s^t, a^t\right) \middle| s^{t+1} \sim \mathcal{T}^\theta(s^t, a^t), a_i^t \sim \pi_i(o_i^t), o_i^t = \sigma_i^\theta(s_t) \right],$$

where π, a are the vectors of all agents' policies and actions. $G^\theta(\pi)$ represents the expected sum of discounted rewards starting in state s_0^θ and following policy π in a PDec-POMDP parametrized by θ.

L Limitations and Open Questions

We list a number of limitations, open questions, and possible extensions.

L.1 Theoretical scope

- **Inseparable tasks.** This work considers task allocation problems where tasks are separable (i.e., the subtasks are independent), and that team reward can be factorized as an *inner* symmetric aggregator followed by an *outer* symmetric aggregator. Extending the theory to *inseparable* rewards, where performance on one task causally affects performance on another task (e.g., coverage with overlap penalty, or pairwise task constraints), is an open problem.
- **Beyond task-allocation RL domains.** While the benchmark domains we study (matrix game, multi-goal capture) and the additional settings covered in App. D (Colonel Blotto, level-based foraging) all fit naturally into the task-allocation template, many notable RL domains—team autonomous driving, multi-robot manipulation—might not be representable within a task allocation framework. Our heterogeneity analysis does not directly apply to these settings, and extending our results to them is important for getting a complete picture of the benefits of heterogeneity.

L.2 Algorithmic assumptions

- **Differentiable simulation.** HED requires $\nabla_\theta G^\theta(\boldsymbol{\pi})$, hence a simulator that is end-to-end differentiable and tractable to back-propagate through. Many realistic environments still rely on non-smooth physics or black-box generators, requiring us to modify HED for these settings. We note that there are good methods for learning environment parameters in non-differentiable settings, including the PAIRED algorithm [36] and the MARL-based environment design algorithm of [35], which can potentially be used to extend HED—but we leave such extensions to future work.

L.3 Open questions

i. **What is the connection between the transition function and heterogeneity?** Our analysis is reward-centric: the curvature criterion reasons only about the team reward. In a Dec-POMDP, however, heterogeneity can be beneficial purely because agents are constrained by *state transitions*. When do state transition dynamics benefit heterogeneity?

ii. **Learning dynamics vs. reward structure.** The theory predicts whether a given reward structure *enables* an advantage to heterogeneous teams, not whether a particular learning algorithm will learn in response to it. Our experiments suggest, empirically, that heterogeneous agents will, in practice, learn to exploit heterogeneous reward structures; but can a formal link be established between our reward structure insights and what reward the learning dynamics converge to in practice?

Tackling these challenges would sharpen our understanding of *when* and *how* diversity should be engineered in cooperative multi-agent learning.