CONJUNCTIVE AMPLIFICATION OF REWARD-SCALE MISCALIBRATION IN COOPERATIVE MULTI-EVENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Dense reward shaping is widely used in cooperative MARL, but in some multi-event tasks it can cause catastrophic performance loss. Reward-scale sensitivity is well known in deep RL; what remains unclear is why an ordinary scale mismatch becomes catastrophic only for particular task structures. We show that the missing link is *conjunctive amplification*. Miscalibrated rewards moderately reduce the acquisition rates of individual necessary events. When success requires all of those events, the local losses compound: all-events success closely tracks the product of per-event acquisition rates (r=0.998). In a controlled Partial Relay environment, increasing dependence on the multi-event chain progressively narrows the empirically safe range across the tested cooperative reward weights, from all six values at ρ=0 to one at ρ=1 (225 runs; interaction p=2.7×10^{-16}). Running reward calibration substantially mitigates this structure-amplified performance loss, with gains increasing from +0.001 to +0.687 as ρ rises, and it generalizes across architectures and benchmarks, including a fourfold increase in Overcooked deliveries ($p\leq$0.004). We further show why static tuning is unreliable: the effective reward scale changes 3.3× during acquisition, so a good fixed scale can be identified only in hindsight. Finally, we introduce Event Order Consistency (EOC) as a low-cost behavioral descriptor of relay-family structure that can support calibration assessment before full training. These results support calibrating cooperative shaping when multi-event task structure can amplify small learning losses.

1 INTRODUCTION

Dense reward shaping is a standard response to sparse feedback in cooperative multi-agent reinforcement learning (MARL). Practitioners add intermediate rewards for progress, helping, and team success so that useful behavior can be learned before the sparse task reward is reached (Ng et al., 1999; Devlin & Kudenko, 2012; Hu et al., 2020). Yet the same shaping can sometimes reduce final performance rather than improve it. Several explanations are plausible: the shaping may reward the wrong behavior, shared rewards may distort incentives, or reward magnitude may alter optimization (Wang et al., 2022; Zhao et al., 2022; Henderson et al., 2018; Engstrom et al., 2020).

Reward-scale sensitivity is already known in deep RL. We ask *when task structure turns that familiar sensitivity into catastrophic failure*. This framing separates reward utility from reward scale and tests how multi-event structure changes the consequences of miscalibration.

Our central finding is a two-stage mechanism. First, reward miscalibration moderately degrades the acquisition of individual necessary events. Matched-magnitude controls with broken meanings and even scaled self-reward produce similar damage, showing that scale alone can generate this local degradation. Second, tasks that require several events to all succeed amplify those local losses multiplicatively. We call this *conjunctive amplification*. In the direct switch control, all-events success tracks the product of marginal event-acquisition rates with r=0.998. Thus, an ordinary scale error can look mild at each event while producing a sharp collapse at the task level.

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.

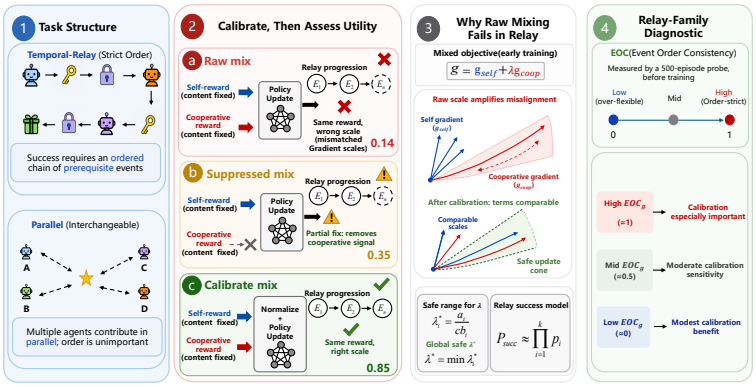


Figure 1: Overview. (1) Relay tasks compose success across prerequisite events. (2) Hidden Gift compares the same reward components under raw mixing, suppression, and calibration (team success, 5 seeds): 0.14, 0.35, and 0.85, respectively. (3) During early acquisition, the two reward streams are weakly aligned and exposed to uncalibrated scale; cooperative-stream-only normalization reaches 0.37, while calibrating both streams reaches 0.87. (4) A 500-episode EOC probe supports the relay-family gate; the graded EOC_g summarizes calibration sensitivity within that family.

Relay tasks provide a clean setting in which to expose this mechanism. In Hidden Gift, for example, one agent must obtain and use a shared key before another agent can reuse it. Across three relay tasks, adding the standard cooperative shaping mixture to a minimal self-interest reward reduces final success by 32–61%. This observation raises the central question of this paper: does the failure arise from the content of cooperative shaping, or from the interaction between reward scale and multi-event task structure?

We organize the paper around four linked questions. First, is the observed failure caused primarily by reward *content* or by reward *scale*? Second, which task property converts local scale damage into task-level collapse? Third, if scale is the problem, why does running calibration work when removal, annealing, or static retuning do not? Fourth, can calibration criticality be anticipated before paying the full training cost?

To answer these questions, we use Partial Relay, which varies the probability ρ that an episode requires the relay chain, while independently varying the cooperative weight λ. This controlled matrix connects task structure to scale sensitivity. We then use order-free controls to distinguish temporal order from the broader conjunctive requirement, analyze gradient dynamics to locate the vulnerable acquisition phase, and test a short behavioral probe for prediction.

Our contributions follow this causal sequence:

We show that scale miscalibration is sufficient to produce the observed shaping failure. Damage grows with reward weight and exposure, persists under semantically broken controls, and is distributed across the reward mixture. Together, these controls establish scale miscalibration as a sufficient source of catastrophic degradation even when harmful shaping semantics are absent.

We identify conjunctive amplification as the task-structural mechanism that makes the failure catastrophic. In Partial Relay, the safe range of λ narrows monotonically from six tested values to one as ρ rises. A separate order-free control rules out temporal order as the unique cause. A switch task then isolates the key structure: per-event degradation is similar under additive and conjunctive rewards, but all-events success follows $\prod_l p_l$ (r=0.998).

We show that running reward calibration substantially mitigates the amplification and explain why static fixes are unreliable. Normalization substantially recovers performance, with gain increasing from +0.001 to +0.687 as ρ rises. The effective reward scale changes 3.3× during acquisition, so static rescaling and fixed optimizer compensation cannot reliably match it. Gradient measurements show that the vulnerable period occurs while the event chain is still being learned.

We introduce EOC as a behavioral diagnostic within relay families. A 500-episode EOC probe supports the frozen calibration rule, while the graded EOC_g tracks calibration gain (r=0.933, p=0.020) and safe-λ window width (r=−0.998, p=1.4×10^{-4}). Conjunctive amplification pro-

vides the causal mechanism; EOC provides the relay-family gate, and EOC_g provides the graded sensitivity descriptor.

The same calibration contrast generalizes beyond the controlled grid world. It replicates without parameter sharing and with a MAPPO-style centralized critic ($p<10^{-4}$ each), and normalized shaped rewards substantially improve standard Overcooked layouts ($0.42\rightarrow1.78$ and $0.04\rightarrow0.15$, $p\leq0.004$).

2 Related Work

Reward Shaping: Content, Incentives, and Failure Modes Reward shaping can fail because the signal encodes the wrong objective. Classical examples include shaping-induced loops (Randløv & Alstrøm, 1998); potential-based shaping instead provides policy-invariance guarantees in single-agent settings (Ng et al., 1999), with equilibrium-preservation extensions under specific multi-agent conditions (Devlin & Kudenko, 2012). Reward misspecification is now treated as a broader failure class (Amodei et al., 2016; Skalse et al., 2022). In MARL, individually shaped rewards can pull agents away from the team objective (Wang et al., 2022), shared rewards can invite free-riding (Zhao et al., 2022), and recent work studies learned or dynamically updated shaping (Akella, 2025; Keddouri et al., 2026). Difference rewards (Wolpert & Tumer, 2001), learned individual rewards (Du et al., 2019), and social-influence bonuses (Jaques et al., 2019) likewise change what the shaping signal says. We hold the shaping construction fixed and test whether scale alone can produce catastrophic failure and which task structures amplify it.

Reward Scale and Calibration in Deep RL Reward magnitude can change learning even when the underlying objective is unchanged. Pop-Art normalizes value targets across magnitudes (van Hasselt et al., 2016); reward normalization and clipping are consequential PPO implementation choices (Engstrom et al., 2020; Andrychowicz et al., 2021); related methods improve scale robustness or balance heterogeneous reward distributions (Sullivan et al., 2023; Cho et al., 2025); and scale sensitivity contributes to reproducibility problems (Henderson et al., 2018). These results establish reward scale as a general optimization issue. Our contribution is to connect this general scale sensitivity to task structure: when task success depends on the joint completion of multiple necessary events, scale-induced losses at the individual-event level are amplified by conjunctive structure into task-level collapse. We further explain when reward calibration becomes critical and why calibration that adapts over training is more reliable than fixed compensation in these settings.

Cooperative MARL and Optimization Conflicts Value factorization (VDN, QMIX, QTRAN) (Sunehag et al., 2018; Rashid et al., 2018; Son et al., 2019), counterfactual baselines (Foerster et al., 2018), role-learning methods (Wang et al., 2020; 2021), and PPO variants (Yu et al., 2022; Kuba et al., 2022) address different sources of coordination and credit-assignment difficulty; independent learning remains a strong baseline (de Witt et al., 2020; Papoudakis et al., 2021; Gronauer & Diepold, 2022). Gradient-surgery methods such as PCGrad target conflicts between objectives at the optimizer level (Yu et al., 2020; Liu et al., 2021). We keep the learning algorithm and reward composition largely fixed and systematically vary reward scale and task structure, thereby isolating the interaction between reward calibration and multi-event structure. Together, these literatures motivate a controlled design that separates reward content, reward scale, and task structure.

3 Method

3.1 Reward Components and Calibration

We consider Dec-POMDPs (Oliehoek & Amato, 2016) with component rewards $r_t = \sum_k w_k r_t^{(k)}$. We separate the components into *self* terms, which reward an agent's own progress such as reaching, picking up, or unlocking, and *cooperative* terms, which reward team success, helping, or a share of a teammate's reward. A single multiplier λ scales all cooperative terms. Thus, $\lambda{=}0$ is self-only and $\lambda{=}1$ is the standard full mixture.

Our main calibration is the running transform $r \mapsto 0.1\,(r - \hat{\mu})/\hat{\sigma}$, applied to the summed scalar reward after a 100-step warm-up. We separately test RMS division and clipping as scale-control variants (Engstrom et al., 2020; Andrychowicz et al., 2021).

3.2 Task Families and Structural Controls

Three relay tasks provide the motivating failures. Hidden Gift (HG) uses two agents and one shared key: agent i must unlock and drop the key before agent j can reuse it. In Sequential Plate, agent A must hold a pressure plate so agent B can reach the goal. In MPE Relay, A must press a button before B's target opens. We also use MPE Spread (Lowe et al., 2017; Terry et al., 2021) and SMACv2 `terran_5v5`/`zerg_5v5` (Samvelyan et al., 2019; Ellis et al., 2023) via HARL (Zhong et al., 2024) as order-free references.

The causal workhorse is *Partial Relay*. Each episode requires the relay chain with probability ρ and is independently solvable otherwise. Varying ρ therefore changes how often success depends on the multi-event chain while leaving the reward construction fixed. Section 4.2 then uses separate order-free controls to determine whether temporal order itself is causal or whether the broader conjunctive requirement is responsible.

Standard-benchmark validation uses Overcooked (Carroll et al., 2019) (`cramped_room`, `coordination_ring`, `forced_coordination`) with its official implementation, layouts, and shaped-reward events. In Overcooked, scoring a delivery requires the chain onion$\times 3$ $\rightarrow$ cook $\rightarrow$ dish $\rightarrow$ serve.

3.3 Event Order Consistency

For relay-family tasks, we also record Event Order Consistency (EOC), the fraction of successful episodes whose logged events follow a canonical order:

$$\text{EOC} \;=\; \frac{\#\{\text{successful episodes whose events follow the canonical order}\}}{\#\{\text{successful episodes}\}}. \tag{1}$$

EOC uses rollout trajectories rather than reward values and serves as a relay-family behavioral descriptor. Section 4.5 evaluates its practical role. For order-free tasks such as MPE Spread, EOC is undefined. A graded variant used for construct validation is reported in Appendix D.

3.4 Training and Evaluation Protocol

Unless stated otherwise, we use IPPO (de Witt et al., 2020): independent PPO (Schulman et al., 2017) learners with GAE (Schulman et al., 2016) and shared parameters (Gupta et al., 2017), hidden size 64, entropy coefficient 0.15, 3000 episodes, 5 seeds, and 200-episode final evaluations. Within each experiment, compared reward conditions use identical learner hyperparameters and differ only in the stated reward treatment or structural intervention. Per-batch advantage normalization is on except in its ablation. Overcooked uses 8000 episodes and horizon 200; SMACv2 uses 5M steps and 3 seeds. All $\pm$ values for performance summaries are sample standard deviations across seeds; regression coefficients are reported with standard errors. Tests are two-sided Welch t-tests or stated regressions on per-seed values. The primary interaction and strict-relay calibration gain have $p<10^{-6}$; secondary mechanism tests report exact values, with Holm correction inside stated comparison families. Evaluation always uses the task-level metric reported in the corresponding table or figure, while event-level quantities are logged only for the dedicated mechanism analyses.

4 Results

4.1 Scale-Induced Shaping Failure

We begin by separating two explanations that are easy to conflate. A *content failure* means cooperative shaping points the policy toward the wrong behavior; a *calibration failure* means the signal may be useful in content but harmful at its realized scale. On the three relay tasks, the full raw mixture reduces final success by 32–61% ($p \leq 0.003$; Table 1), whereas the already calibrated SMACv2 pipeline benefits from dense shaping.

Table 1: Minimal versus full reward mixtures. Raw shaping hurts the relay tasks and MPE Spread; SMACv2 uses its calibrated standard pipeline.

Task	EOC	min	full
Hidden Gift (raw)	1.0	0.354 ± 0.082	0.138 ± 0.031
Sequential Plate (raw)	1.0	0.529 ± 0.036	0.359 ± 0.041
MPE Relay (raw)	1.0	0.957 ± 0.021	0.562 ± 0.053
MPE Spread (raw, return)	n/a[a]	-21.1 ± 0.3	-25.3 ± 2.7[b]
SMACv2 terran (std. pipeline, win)	n/a[a]	0.001 ± 0.002	$\mathbf{0.273 \pm 0.007}$
SMACv2 zerg (std. pipeline, win)	n/a[a]	0.058 ± 0.008	$\mathbf{0.263 \pm 0.056}$

[a]Order-free: success requires no prerequisite event chain, so EOC is undefined.
[b]On Spread the raw mixture also underperforms; Section 4.3 explains why.

The failure follows dose and exposure. Increasing cooperative weight strengthens the drop (MPE Relay $r{=}-0.96$, $p{=}6.6{\times}10^{-7}$; HG $r{=}-0.83$, $p{=}5.4{\times}10^{-6}$), and increasing exposure has the same effect (HG $r{=}-0.81$, $p{=}1.7{\times}10^{-5}$). Moving the same exposure window to early, middle, or late training has little effect (ANOVA $p{=}0.65$), indicating that cumulative dose matters more than one isolated window. At 10,000 episodes ($3.3\times$ the standard budget), minimal reward reaches 0.970 while the full mixture remains at 0.454 ($p{=}4.5{\times}10^{-7}$); withdrawing cooperative terms mid-training only raises 0.138 to 0.223 ($p{=}0.003$).

Matched-magnitude controls show that specifically harmful cooperative semantics are unnecessary. Scaling self-reward by $2.5\times$ reduces success by 47% ($0.354 \rightarrow 0.188$), misassigned cooperative rewards by 57% ($\rightarrow 0.151$), and random noise matches the full mixture (0.134 vs. 0.138, $p{=}0.81$). The team-success term is the most damaging individual component (0.164), while the other four yield 0.27–0.34 and the full stack is worst (0.138). Thus, scale perturbation is sufficient for severe performance degradation even without specifically harmful cooperative semantics. These experiments establish the source of the failure at the task level; the next section directly measures the event-level losses and asks which task structure amplifies them into collapse.

4.2 Conjunctive Amplification

We now hold reward construction fixed and manipulate task structure. In Partial Relay, $\rho \in \{0, 0.25, 0.5, 0.75, 1\}$ is crossed with raw $\lambda \in \{0, 0.03, 0.1, 0.3, 1, 3\}$ and normalized $\lambda \in \{0, 1, 3\}$. For the positive raw weights ($\lambda{>}0$; $n{=}125$ runs), regressing success on ρ, $\log_{10}\lambda$, and their product gives $\beta = -0.150 \pm 0.016$ ($p{=}2.7{\times}10^{-16}$). At $\rho{=}0$, success is flat across λ; at $\rho{=}1$, $\lambda : 0.03 \rightarrow 3$ reduces success from 0.392 to 0.100. The number of tested λ values less than 0.05 below each row's best narrows monotonically as $6, 5, 3, 2, 1$ (Spearman -1.0, permutation $p{=}0.017$). Task structure therefore determines how precisely reward scale must be controlled.

This interaction does not identify the structural variable by itself: increasing ρ changes temporal dependence, difficulty, and the number of events that must jointly succeed, while the $\rho{=}0$ row also sits at a ceiling. We therefore use a matched two-key control that removes the cross-agent prerequisite while preserving reward construction and approximately matches the $\lambda{=}0$ baseline difficulty to the hard relay (0.250 vs. 0.342). Despite probe EOC$=0$, increasing raw λ from 0 to 1 reduces success from 0.250 to 0.072 ($p{=}0.001$; $\lambda{=}3$ reaches 0.012), and normalization recovers 0.477. An easier order-free grid also collapses ($0.610 \rightarrow 0.073$), whereas an easier ordered grid is hurt less ($0.550 \rightarrow 0.366$) and is recovered by normalization (0.859). The structural explanation therefore shifts from strict temporal order to the broader requirement that several events jointly succeed.

The switch task isolates this requirement directly. Two agents light $L \in \{2, 3, 4\}$ switches; the conjunctive variant pays only when all switches are lit, whereas the additive variant pays proportionally per switch. Raw cooperative weight produces similar per-event degradation in both variants (smallest difference test $p{=}0.077$), but conjunctive all-switch success follows the product of individual marginal acquisition rates (per-run $r{=}0.998$; per-configuration $r{=}0.999$, bootstrap 95% CI $[0.998, 0.999]$, MAE 0.012). Per-event losses increase only from 10% at $L{=}2$ to 29% at $L{=}4$, while all-events success collapses by 17–80%. Normalization rescues both variants ($p \leq 0.011$), confirming that the local loss remains scale-sensitive even when final success is composed differently. Overcooked shows the same signature: intermediate events degrade moderately (onion -19%, pot-

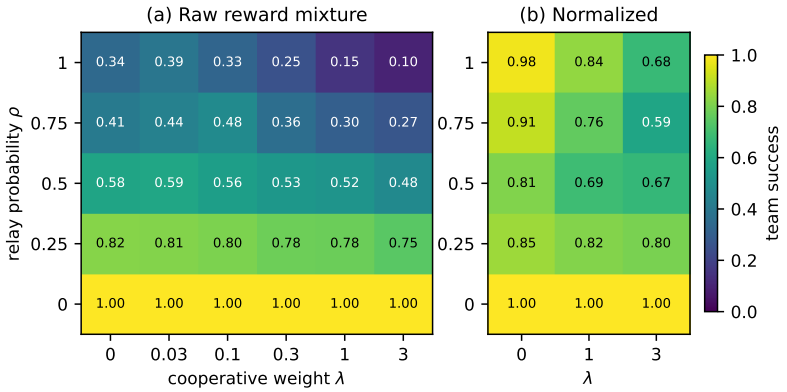


Figure 2: Structure–scale interaction in Partial Relay (5 seeds per cell, 225 runs). The raw safe range across the tested λ values narrows as relay dependence ρ increases, while normalization substantially widens it.

ting -42%, dish -2% n.s., soup -39%), but scored deliveries fall by 76%; the final two conditional stages are $0.75/0.57$ raw versus $0.83/0.98$ normalized. These results identify *conjunctive amplification*: modest event-level losses multiply into task-level collapse.

4.3 Running Reward Calibration

Running calibration acts on the local scale problem before conjunctive composition magnifies it. Across $\rho = 0, 0.25, 0.5, 0.75, 1$, the gain from normalizing the full mixture is $+0.001, +0.040, +0.171, +0.456, +0.687$ (all $p<0.005$ for $\rho \geq 0.25$), peaking at the strict relay ($p=4.8\times10^{-7}$; Figure 3). The original relay tasks show the same pattern (Table 2): HG improves from 0.138 to 0.845, Sequential Plate from 0.359 to 0.933, and MPE Relay from 0.562 to 0.993. Fixed rescaling shows the same dose response ($\times0.03 \rightarrow 0.777$, $\times0.1 \rightarrow 0.649$, $\times0.3 \rightarrow 0.272$). At the strict-relay corner, RMS-only running scaling reaches 0.686 from raw 0.152, centering raises it to 0.839, and clipping reaches 0.891. Different forms of scale control therefore produce the same qualitative rescue.

Calibration also separates scale-induced difficulty from the behavioral chain itself. Raw minimal success falls from 1.00 to 0.34 as ρ rises, whereas normalized minimal reward falls only to 0.98; on HG, normalizing minimal reward alone raises 0.354 to 0.973. Yet task structure is not erased: after normalization, adding cooperative terms costs nothing at $\rho=0$ but 0.12–0.15 at $\rho \geq 0.5$ ($p=0.007, 0.086, 0.009$ across the three higher-dependence rows). Tripling λ costs 0.164 at $\rho=1$ ($p=0.003$) and 0 at $\rho=0$, and the normalized interaction remains significant ($\beta=-0.404$, $p=0.025$). Calibration widens the safe region substantially while high-dependence tasks retain residual sensitivity.

Suppression is an incomplete substitute. At $\rho=1$, annealing λ to zero reaches 0.223 ± 0.031, worse than complete removal 0.342 ($p=2.1\times10^{-4}$) and far below running calibration 0.839 ($p=3.7\times10^{-6}$); withdrawing cooperative terms mid-training likewise reaches only 0.223. Moreover, cooperative content can be useful: on Sequential Plate, normalized full reward reaches 0.933 versus 0.536 for normalized minimal reward. In HG, where self reward already covers the chain, normalized minimal reward reaches 0.973 versus 0.845 for normalized full reward. Calibration therefore controls whether a signal is learnable at its scale, while reward design determines whether that signal is useful.

The effect extends beyond the controlled relay family. On Overcooked `cramped_room`, sparse reward never delivers, raw shaped reward yields 0.42 ± 0.09 deliveries, and normalized shaping reaches 1.78 ± 0.32 ($4.2\times$, $p=4.1\times10^{-4}$); on `coordination_ring`, the corresponding values are 0.00, 0.04 ± 0.04, and 0.15 ± 0.05 ($p=0.004$). At $2.5\times$ the budget, normalization still leads: 1.70 vs. 2.83 on `cramped_room` ($p=0.002$) and 0.15 vs. 1.32 on `coordination_ring` ($p=0.029$). Raw training can improve with more data, but it learns the event chain substantially more slowly.

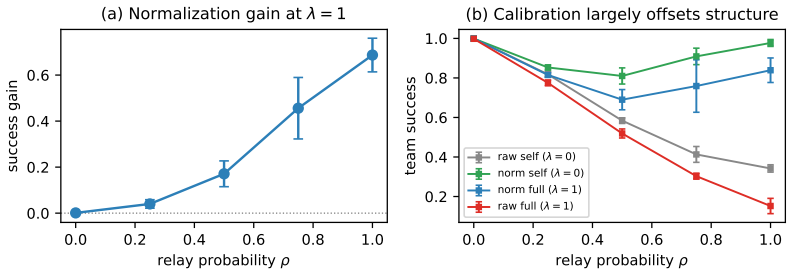


Figure 3: Calibration across relay dependence. Normalization gain increases with ρ, while normalized conditions remain substantially more robust as relay dependence rises.

Table 2: Final success under raw and normalized minimal/full rewards on the three relay tasks (5 seeds).

	min raw	**full raw**	**full norm.**	**min norm.**
Hidden Gift	0.354 ± 0.082	0.138 ± 0.031	0.845 ± 0.082	$\mathbf{0.973 \pm 0.012}$
Sequential Plate	0.529 ± 0.036	0.359 ± 0.041	$\mathbf{0.933 \pm 0.018}$	0.536 ± 0.042
MPE Relay	0.957 ± 0.021	0.562 ± 0.053	0.993 ± 0.008	$\mathbf{0.996 \pm 0.004}$

The result also survives architectural changes: non-shared IPPO improves from 0.252 to 0.827, a MAPPO-style centralized critic from 0.503 to 0.924 ($p<10^{-4}$ each), and a three-agent chain from 0.591 to 1.000. SMACv2 provides a complementary reference because its standard pipeline already includes the environment's reward scaling and HARL value normalization, consistent with the absence of the raw-collapse pattern in Table 1. These results establish calibration as a robust intervention across task families and architectures and motivate asking why a *running* calibration is especially effective.

4.4 Acquisition Dynamics and Scale Drift

The structural experiments explain how local damage becomes catastrophic; we now locate where that local damage enters learning. At eight checkpoints, we compute the policy gradient that each reward stream would produce separately. At ρ=1, self and cooperative gradients are less aligned over the first two checkpoints than at ρ=0 ($+0.45$ vs. $+0.61$, p=0.020). This is precisely when behaviors such as dropping the key or waiting have not yet been connected by the policy to eventual success. Under normalization, the policy acquires the chain and the cosine rises from $+0.30$ to $+0.82$; pooled checkpoint measurements show alignment and success rising together (r=0.52, descriptive because checkpoints repeat within seeds). Under raw rewards the corresponding correlation is r=0.04. Absolute gradient size alone is insufficient: the cooperative-to-self ratio is smaller in the fragile ρ=1 case than at ρ=0 (2.1 vs. 4.0, p=0.004). The vulnerable regime is therefore scale acting on weakly aligned acquisition updates. This localization is consistent with the exposure experiment in Section 4.1: that control moved a fixed exposure dose across training, whereas the gradient analysis measures the endogenous phase in which the reward streams are naturally least aligned. Moving the imposed window does not remove cumulative damage, but the acquisition phase remains where poorly aligned updates arise.

Per-stream interventions support this interpretation. At ρ=1, λ=1, cooperative-only normalization reaches 0.369 ± 0.020, above raw 0.152 but close to deletion 0.342; self-only normalization reaches 0.124, while normalizing both streams separately reaches 0.869 ± 0.032, close to total normalization 0.839 (p=0.42). Even self-only reward improves from 0.342 raw to 0.977 normalized. Overcooked shows the same pattern: either stream alone gives 0.74 ± 0.31 or 0.77 ± 0.21 deliveries, both streams give 1.67, and total normalization 1.78 (p=0.63 between the two full-calibration variants; both outperform either partial intervention at $p \leq 0.004$). Direction correction and within-batch rescaling are insufficient. PCGrad reaches 0.224 ± 0.027, versus deletion 0.342 and calibration 0.839; removing PPO advantage normalization changes raw success only to 0.254. Reward normalization without advantage normalization reaches 0.298 (0.340 at unit variance), compared with 0.839 when

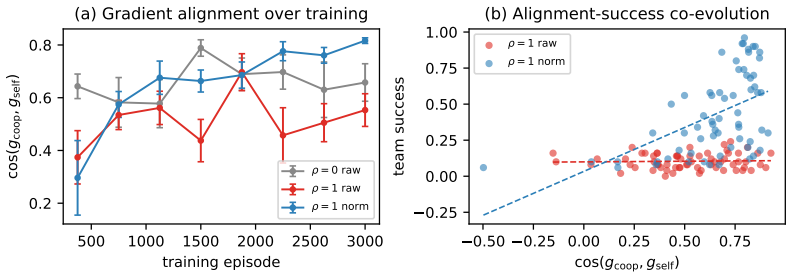


Figure 4: Gradient dynamics during acquisition (8 seeds). At ρ=1, self/cooperative gradients are weakly aligned early; under normalization, alignment rises as the chain is acquired.

reward and advantage normalization are combined. These controls separate reward-scale control from gradient surgery and within-batch estimator rescaling.

Static optimizer retuning also fails to reach the calibrated regime. Sweeping learning rate from 3×10^{-4} to 10^{-5}, entropy coefficient from 0.15 to 4.5, and the value coefficient leaves every raw setting at or below 0.257 ± 0.044; the jointly compensated settings fall to 0.03–0.05, far below normalization. The stronger explanation is scale drift. In strict-relay runs that acquire the chain, raw reward scale grows from 0.85 to 2.83 (3.3×), while calibrated scale remains bounded (Appendix Figure 5). Static rescales matched to early, middle, and late phases reach 0.579, 0.757, and 0.783, compared with 0.871 for running calibration; the late match is statistically close (p=0.061) but can be selected only after observing the trajectory. Freezing the normalizer after 20%, 50%, or 80% of training yields 0.747, 0.811, and 0.847, compared with 0.871 when it keeps adapting. Performance improves monotonically with how long the normalizer tracks the changing distribution (Holm-adjusted vs. running: p=$3\times10^{-4}, 0.081, 0.36$). Target scale also matters: 0.03 reaches 0.909, 0.1 reaches 0.839, and unit variance 0.208. Thus, a good fixed scale can exist, but running calibration adapts without phase-specific hindsight.

For intuition, let event l have reliability p_l, event-improving direction u_l, and effective stream scales c_s, c_c:

$$J \approx \prod_{l \le L} p_l, \qquad \dot{p}_l \propto \langle g, u_l \rangle, \qquad g = c_s g_{\text{self}} + \lambda c_c g_{\text{coop}}, \qquad \lambda^* = \min_{l \le L} \frac{c_s}{c_c} \frac{a_l}{b_l}. \tag{2}$$

When a stream opposes an unlearned event ($\langle g_{\text{self}}, u_l \rangle = a_l > 0$ while $\langle g_{\text{coop}}, u_l \rangle = -b_l \le 0$; a non-opposing stream imposes no constraint), event-wise directional constraints can narrow the global safe range of λ; the product then converts a local acquisition failure into task-level collapse. Equation 2 provides a conceptual link between event-wise constraints and conjunctive success. The scale-control, drift, and freezing interventions above provide the empirical evidence for running calibration.

4.5 Relay-Family Diagnosis with EOC

We next evaluate EOC as a behavioral descriptor of ordered relay structure before full training. We train a minimal-reward probe for 500 episodes and compute EOC from successful rollouts. Across 140 calibrated runs, final EOC remains 1.0 while success ranges from 0.18 to 1.00, confirming that EOC describes structure rather than performance.

We then freeze one rule on the two development tasks, HG and MPE Relay: probe EOC>0.5 triggers calibrated full reward; otherwise the raw mixture is used. The rule is fixed before any held-out evaluation. Applied unchanged to Sequential Plate, three Partial Relay levels, and Spread, all 20 relay-family runs trigger calibration while all five Spread probes are assigned to the low-criticality group. Sequential Plate nearly matches the normalized oracle (0.932 vs. 0.933); full results, including probe cost, appear in Appendix D. Spread still gains modestly from normalization ($-25.3 \rightarrow -19.3$, p=0.007), and the probe separates low-criticality from collapse-prone relay settings. Reward utility is evaluated separately through the normalized full-versus-minimal comparison: Sequential Plate benefits strongly from cooperative content after calibration (0.933 full vs.

0.536 minimal), whereas HG prefers the normalized minimal reward (0.973 vs. 0.845). Across the policy-gradient benchmarks tested here, the EOC rule separates the relay-family settings from the order-free Spread reference, while the graded EOC_g captures variation within the relay family. The appendix reports its correlations with engineered ρ as construct-validity checks.

5 Discussion and Limitations

5.1 Exploration Boundary

Our mechanism assumes that the learner encounters the behavioral chain at least occasionally. On Overcooked `forced_coordination`, sparse, raw shaped, and normalized shaped rewards all fail, with only one delivery in 1,500 evaluation episodes. QMIX (Rashid et al., 2018) likewise solves $\rho{=}0$ perfectly but reaches 0.00 on the strict relay under every reward treatment. When exploration never exposes the necessary events, there is no informative gradient for calibration to preserve.

5.2 Learner-Dependent Calibration

Our main normalization target is chosen for PPO-style policy-gradient learning. Applying that target directly to QMIX destabilizes it, while retargeting the scale to QMIX's own operating range restores performance; the reverse mismatch also hurts PPO (Appendix C). Value-based learners can address scale through different machinery, including PopArt-style value-target normalization (van Hasselt et al., 2016). These results support learner-dependent calibration targets rather than a single numeric target across algorithms.

5.3 Scope of EOC

The evidence for conjunctive amplification spans ordered and order-free controls, while EOC is evaluated as a behavioral diagnostic inside relay families and related tasks. EOC requires successful minimal-reward rollouts and event logs, and multiple valid canonical orders can reduce the measured consistency. A more general predictor of conjunctive dependence remains an open direction.

Finally, running normalization provides a simple and widely available calibration intervention. Our evidence spans two- and three-agent chains, two-key and switch controls, Overcooked, SMACv2, and Spread; broader algorithm families, task structures, and calibration methods remain open for evaluation.

6 Conclusion

This paper isolates a scale-driven failure mode of dense cooperative shaping and follows it from local optimization to final task success. Reward miscalibration moderately degrades the acquisition of necessary events. When task success is conjunctive across those events, the losses multiply, and a familiar scale sensitivity becomes catastrophic (J tracks $\prod_l p_l$, $r{=}0.998$). Running calibration prevents most of this amplification, separating collapse from success in the strict relay ($0.152 \rightarrow 0.839$) and in matched order-free controls ($0.07 \rightarrow 0.48$). Its advantage over static fixes follows from the learning process itself: the effective reward scale changes as the event chain is acquired. EOC provides a secondary relay-family descriptor of ordered structure. The practical implication is to separate reward utility from reward scale: retain useful cooperative information and calibrate the scale at which it is learned. This makes dense cooperative shaping more reliable in multi-event tasks by reducing the risk that small scale errors are amplified into task-level collapse.

Reproducibility Statement

All experiments use fixed seeds (5 unless stated), identical hyperparameters across compared conditions within each experiment, and fixed final-evaluation protocols (200 episodes unless stated); environments, reward components, evaluation sizes, and statistical procedures are specified in Sections 3–4 and Appendix A. Code, per-run result files, data-figure generators, and final schematic artwork will be released.

AI USE STATEMENT

Generative AI tools were used to assist with literature search, provide feedback on experimental design, support result interpretation, assist with code implementation and debugging, and support manuscript drafting, translation, revision, and grammar editing. The research hypotheses and core conceptual framing were developed and refined by the authors. All AI-assisted outputs incorporated into the work were reviewed and verified by the authors, who take responsibility for the final content.

REFERENCES

Aditya Akella. On the fundamental limitations of decentralized learnable reward shaping in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2511.00034*, 2025.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? A large-scale study. In *International Conference on Learning Representations*, 2021.

Micah Carroll, Rohin Shah, Mark K Ho, Thomas L Griffiths, Sanjit A Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-AI coordination. In *Advances in Neural Information Processing Systems*, volume 32, pp. 5175–5186, 2019.

Myungsik Cho, Jongeui Park, Jeonghye Kim, and Youngchul Sung. ARS: Adaptive reward scaling for multi-task reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 10583–10601, 2025.

Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip H S Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the StarCraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

Sam Devlin and Daniel Kudenko. Dynamic potential-based reward shaping. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 433–440, 2012.

Yali Du, Lei Han, Meng Fang, Tianhong Dai, Ji Liu, and Dacheng Tao. LIIR: Learning individual intrinsic reward in multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, pp. 4403–4414, 2019.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N Foerster, and Shimon Whiteson. SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 37567–37593, 2023.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep RL: A case study on PPO and TRPO. In *International Conference on Learning Representations*, 2020.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2974–2982, 2018.

Sven Gronauer and Klaus Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2):895–943, 2022.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *Autonomous Agents and Multiagent Systems (AAMAS) Workshops*, pp. 66–83, 2017.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, volume 32, pp. 3207–3214, 2018.

Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. Learning to utilize shaping rewards: A new approach of reward shaping. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 15931–15941, 2020.

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. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 3040–3049, 2019.

Faid Keddouri, Sohaib Houhou, Aissa Boulmerka, and Nadir Farhi. Regime-conditional stabilisation of LLM-augmented cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2607.04470*, 2026.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2022.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18878–18890, 2021.

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*, volume 30, pp. 6379–6390, 2017.

Andrew Y Ng, Daishi Harada, and Stuart J Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 278–287, 1999.

Frans A Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Springer, 2016.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2021.

Jette Randløv and Preben Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 463–471, 1998.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 4295–4304, 2018.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G J Rudner, Chia-Man Hung, Philip H S Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft multi-agent challenge. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 2186–2188, 2019.

John Schulman, Philipp Moritz, Sergey Levine, Michael I Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Joar Skalse, Nikolaus H R Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9460–9471, 2022.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 5887–5896, 2019.

Ryan Sullivan, Akarsh Kumar, Shengyi Huang, John P Dickerson, and Joseph Suarez. Reward scale robustness for proximal policy optimization via DreamerV3 tricks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 1352–1362, 2023.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 2085–2087, 2018.

Justin K Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, Niall L Williams, Yashas Lokesh, and Praveen Ravi. PettingZoo: Gym for multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15032–15043, 2021.

Hado van Hasselt, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. Learning values across many orders of magnitude. In *Advances in Neural Information Processing Systems*, volume 29, pp. 4287–4295, 2016.

Li Wang, Yupeng Zhang, Yujing Hu, Weixun Wang, Chongjie Zhang, Yang Gao, Jianye Hao, Tangjie Lv, and Changjie Fan. Individual reward assisted multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 23417–23432, 2022.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. ROMA: Multi-agent reinforcement learning with emergent roles. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 9876–9886, 2020.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. RODE: Learning roles to decompose multi-agent tasks. In *International Conference on Learning Representations*, 2021.

David H Wolpert and Kagan Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(2–3):265–279, 2001.

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. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 35, pp. 24611–24624, 2022.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020.

Minglu Zhao, Ning Tang, Annya L Dahmani, Yixin Zhu, Federico Rossano, and Tao Gao. Sharing rewards undermines coordinated hunting. *Journal of Computational Biology*, 29(9):1022–1030, 2022.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

A Experimental Configuration and Controls

A.1 Training Configuration

All reward conditions within each experiment use the same hyperparameters. Table 3 lists the learner configurations and training budgets.

Table 3: Hyperparameters. IPPO settings apply to all grid-world experiments (Hidden Gift, Sequential Plate, MPE Relay, Partial Relay, three-agent HG); the Overcooked learner is the same algorithm with a larger network for the richer observation.

	IPPO (grid worlds)	**IPPO (Overcooked)**	**QMIX**
learning rate	3×10^{-4}	3×10^{-4}	5×10^{-4}
hidden size	64 (128 for 3-agent)	256	64 (mixing 32)
entropy coefficient	0.15	0.05	—
PPO clip / epochs	0.2 / 4	0.2 / 4	—
γ / GAE λ	0.99 / 0.95	0.99 / 0.95	0.99 / —
rollout buffer	512 steps	1024 steps	replay 50,000
advantage normalization	on (except ablation)	on	—
exploration	stochastic policy	stochastic policy	ϵ: 1$\rightarrow$0.05 over 40k steps
episodes	3000 (5000 for 3-agent)	8000, horizon 200	3000
final evaluation	200 episodes	100 episodes	200 episodes

A.2 Reward Components

In the HG family, the self set is {own-door unlock, key pickup, progress-to-key, progress-to-door, step cost}, all at weight 1. The cooperative set is {team success, explicit key drop, teammate uses key, opportunity creation, share of teammate's reward}, all at weight λ. Normalization is the running transform defined in Section 3, applied to the summed scalar reward.

A.3 Learner Variants

SMACv2 experiments use HARL (Zhong et al., 2024) with the supplied configuration (5M steps, 3 seeds), including value normalization; the environment's `reward_scale` is active, as discussed in Section 4.3. The MAPPO-style variant keeps every IPPO hyperparameter and changes the critic to a centralized value network over the concatenated observations of all agents plus a one-hot agent id. The PCGrad baseline keeps every IPPO hyperparameter, adds one value head per reward stream with per-stream GAE and advantage normalization, and applies standard pairwise gradient projection between the two policy losses; value losses and entropy are not projected. SMACv2 win rates are the mean of each seed's last ten evaluations under the standard shaped mixture (damage+kill+win). Duplicate logs from interrupted launches are deduplicated by selecting the run with the largest recorded training step and then the most evaluations.

A.4 Switch-Task Control

The switch task isolates task composition while keeping the reward streams fixed. Two agents act on a 7×7 grid with $L \in \{2, 3, 4\}$ switches at fixed corners, a 50-step horizon, and actions stay, four moves, and toggle with range 1.5. Both variants share the same self stream (step cost -0.01, progress-to-nearest-unlit slope 0.3, own toggle $+1.0$) and cooperative stream at weight λ (partner toggle $+3.0$, mirrored partner-progress slope 0.3). The conjunctive version pays 10.0 to both agents when all switches are lit, whereas the additive version pays $10.0/L$ to both per toggle. The learner and budget are the IPPO grid-world configuration above, with 5 seeds per cell over $\{\text{conj}, \text{add}\} \times L \times \lambda \in \{0, 0.5, 1\}$ raw plus normalized λ=1 (120 runs). Evaluation logs each switch's final state, and the product is computed from the individual marginal rates $\prod_j \hat{P}(\text{switch}_j \text{ lit})$. The additional event logging leaves the outcomes of all 120 runs unchanged.

A.5 Static Compensation Controls

We evaluate whether fixed PPO hyperparameters can compensate for the raw reward scale. At the strict-relay corner (ρ=1, λ=1, raw), the exact matrix training loop is rerun with learning rate $\in \{3\times10^{-4}, 10^{-4}, 3\times10^{-5}, 10^{-5}\}$, entropy coefficient $\in \{0.15, 0.45, 1.5, 4.5\}$, and value coefficient $\in \{0.5, 0.05\}$, one axis at a time plus the jointly scale-compensated pair (lr 10^{-5}, entropy 4.5); 9 configurations, 5 seeds each. The default configuration yields 0.152 ± 0.043, matching the strict-relay result used in the main analysis.

A.6 Overcooked Event Decomposition

We measure the event-level acquisition pattern in the standard Overcooked `cramped_room` benchmark. The recipe of Section 4.3 is rerun with the final 100-episode evaluation logging the environment's statistics for onion pickup, onion potting, dish pickup, and soup pickup. The environment's "soup delivery" event log includes unscored deliveries, so scored deliveries are counted from the sparse reward, consistent with the main Overcooked evaluation. Raw shaped and normalized shaped use 5 seeds each, and the delivery values match Section 4.3 (0.42 ± 0.09 vs. 1.78 ± 0.32). The episode-level funnel uses nested stages (≥ 3 onion pickups; and ≥ 3 pottings; and ≥ 1 dish; and ≥ 1 soup; and ≥ 1 scored delivery), each conditional on all previous stages, computed from per-episode logs of an exact rerun (500 episodes per condition).

A.7 Scale Drift and Calibration Timing

We measure how the effective reward scale changes during strict-relay learning. The exact strict-relay corner loop is rerun with per-update logging of batch reward mean and standard deviation before and after the active transform, and of parameter-step norm. Seeds and RNG streams are unchanged, and the runs yield the same corner values (0.152 ± 0.043 raw, 0.839 ± 0.069 normalized). Phase scales are read from the normalized runs' pre-transform trajectory: $\sigma_{\text{early}}=0.85$ for the first 20% of updates, $\sigma_{\text{mid}}=1.64$ for the middle 20%, and $\sigma_{\text{late}}=2.83$ for the last 20%. Static-rescale runs multiply the raw reward by $0.1/\sigma$ with no mean subtraction or running statistics, with 5 seeds each. Freeze runs use the standard running normalizer but stop updating its mean and variance after 20%, 50%, or 80% of training, keeping the frozen transform thereafter. Freeze and running conditions use 10 seeds; static rescales use 5. The freeze family is compared against running with Holm correction over its three comparisons. The clipping control replaces the running transform with the pre-specified $r \mapsto \text{clip}(r, -1, 1)$ and uses 10 seeds.

B Learning-Curve Evidence

Figure 6 shows when the main raw and calibrated conditions separate during training.

C Full Structure–Scale Matrix and Boundary Controls

Table 4 reports every cell underlying the structure–scale matrix in Figure 2.

Table 4: Team success (mean ± std, 5 seeds) for every cell of the $\rho \times \lambda$ matrix underlying Figure 2.

	raw						normalized		
ρ	λ=0	0.03	0.1	0.3	1	3	λ=0	1	3
0.00	1.00±.00	1.00±.00	1.00±.00	1.00±.00	1.00±.00	1.00±.00	1.00±.00	1.00±.00	1.00±.00
0.25	.82±.02	.81±.01	.80±.01	.78±.01	.78±.02	.75±.03	.85±.01	.82±.01	.80±.02
0.50	.58±.02	.59±.04	.56±.02	.53±.02	.52±.03	.48±.01	.81±.05	.69±.06	.67±.05
0.75	.41±.04	.44±.03	.48±.05	.36±.02	.30±.02	.27±.03	.91±.05	.76±.15	.59±.06
1.00	.34±.02	.39±.03	.33±.02	.25±.03	.15±.04	.10±.03	.98±.02	.84±.07	.68±.05

C.1 Switch-Task Effect Sizes

The conjunctive−additive difference in relative per-event retention at λ=1 (raw) is $+0.001$, $+0.060$, and $+0.026$ at L=2, 3, 4, with Welch 95% confidence intervals $[-0.075, 0.078]$, $[-0.009, 0.128]$, and $[-0.053, 0.105]$ (p=0.965, 0.077, 0.466). The largest upper bound (0.128) remains small relative to the 17–80% all-events collapse, consistent with conjunctive amplification of local acquisition losses.

C.2 Value-Based Calibration Boundary

Calibration targets depend on the learning algorithm. PPO-targeted normalization (target scale 0.1) destabilizes QMIX at ρ=0 ($1.00 \rightarrow 0.445$), while retargeting to QMIX's unit scale recovers 0.948.

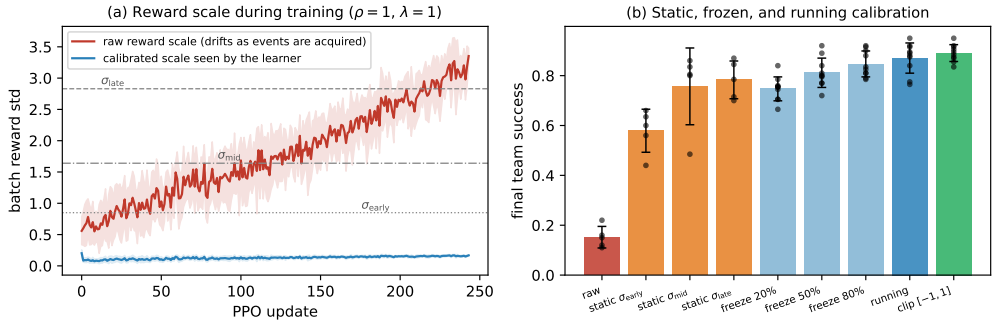


Figure 5: Reward-scale drift and calibration timing. (a) At the strict-relay corner (ρ=1, λ=1), the underlying raw batch-reward standard deviation (red) grows 3.3× across 5 normalized runs, while the calibrated scale seen by the learner (blue) remains within $[0.06, 0.20]$. Dashed lines mark the early/mid/late phase scales used for static rescaling. (b) Final success for phase-matched static rescales (orange), normalizers frozen during training (light blue), and the pre-specified clipping control (green); dots are seeds. Performance increases with later scale matching (static early vs. running p=5×10^{-4}; freeze-at-20% Holm p=3×10^{-4}), while static late and freeze-at-80% approach running calibration.

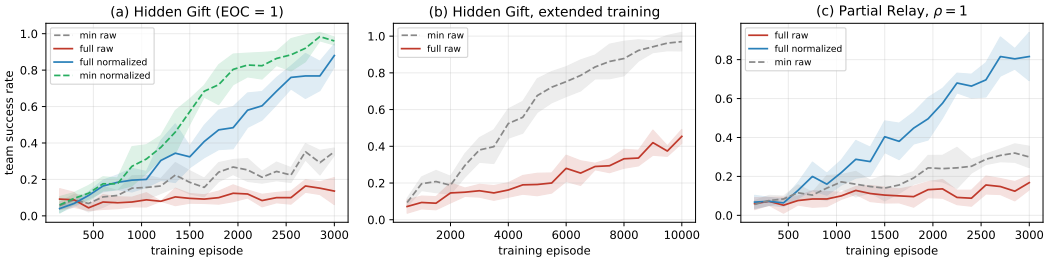


Figure 6: Learning curves for the main comparisons (mean ± std across 5 seeds). (a) Hidden Gift: both normalized conditions overtake both raw conditions before the end of the standard budget. (b) Extended Hidden Gift training: the raw full mixture remains below the raw minimal reward at 3.3× the standard budget. (c) Partial Relay at ρ=1: the normalized full mixture (blue) separates from the raw full mixture (red) early in training.

Conversely, unit-scale normalization reduces PPO performance at ρ=1 to 0.208 (Section 4.4). These results show that the effective calibration range depends on the learner.

D Held-Out EOC Evaluation

A graded index EOC_g multiplies EOC by the observed fraction of episodes that require the relay chain. Across Partial Relay levels it yields $0.00, 0.27, 0.61, 0.83, 1.00$. Because the second factor directly estimates relay exposure, its increase with engineered ρ provides a construct-validity check that EOC_g reflects the intended variation in relay dependence. Within this family, EOC_g covaries with calibration gain (r=0.933, p=0.020) and raw safe-window width (r=−0.998, p=1.4×10^{-4}).

Table 5 gives the full per-task results for the frozen relay-family heuristic in Section 4.5. The rule is fixed on the two development tasks and applied unchanged to the held-out rows; probe-training cost is included in the probe-rule column.

Table 5: Frozen probe rule (5 seeds). "Oracle" is the normalized full mixture; probe-training cost is included.

Task	full raw	min raw	probe rule	oracle
Development: HG	0.138 ± 0.031	0.354 ± 0.082	0.631 ± 0.076	0.845 ± 0.082
Development: MPE Relay	0.562 ± 0.053	0.957 ± 0.021	0.974 ± 0.016	0.993 ± 0.008
Sequential Plate	0.359 ± 0.041	0.529 ± 0.036	$\mathbf{0.932 \pm 0.036}$	0.933 ± 0.018
Partial Relay ρ=0.25	0.775 ± 0.016	0.818 ± 0.015	0.803 ± 0.010	0.815 ± 0.013
Partial Relay ρ=0.50	0.519 ± 0.026	0.584 ± 0.016	$\mathbf{0.614 \pm 0.045}$	0.690 ± 0.057
Partial Relay ρ=0.75	0.303 ± 0.017	0.413 ± 0.040	$\mathbf{0.578 \pm 0.036}$	0.759 ± 0.148
MPE Spread (return)[a]	-25.3 ± 2.7	-21.1 ± 0.3	-21.2 ± 0.8	-19.3 ± 0.3

[a]Return metric (higher is better); the probe assigns Spread to the low-criticality group, so the rule trains the raw mixture after the probe phase.

Sequential Plate reaches 0.932, compared with 0.933 for the normalized oracle. At ρ=0.25, where the raw safe window is wide, the tested treatments remain close. On Spread, the rule selects the raw mixture and obtains -21.2, close to the min-raw baseline of -21.1; the normalized oracle reaches -19.3.