K-SPARSE WASSERSTEIN DISTRIBUTIONALLY ROBUST VALUE FACTORIZATION FOR COOPERATIVE MARL

Anonymous authors
Paper under double-blind review

ABSTRACT

Robust value factorization for MARL under concurrent actuator faults is combinatorial: allowing up to K failures among N agents creates $\sum_{k=1}^{K}\binom{N}{k}$ fault patterns per Bellman backup. We introduce K-Sparse Wasserstein Distributionally Robust MARL (KW-DRO), which limits each fault realization to K agents and uses a Wasserstein radius ρ to control the expected fault count. For the standard single-hidden-layer QMIX mixer, we show that learner-side fault damage is monotone submodular under convex activation and nonnegative utility drops. KW-DRO therefore constructs candidates in $O(NK)$ time with a $(1-1/e)$ percardinality guarantee, exactly for VDN, and solves probability allocation by an exact lower-convex-hull computation. The solver agrees with reference linear programs, and greedy search matches exhaustive selection on all 1,536 tested QMIX states. On RWARE, nominal VDN exhibits late policy collapse in long training runs; KW-DRO at $\rho = 0.5$ keeps all $20/20$ runs alive, compared with $14/20$ under the nominal backup. Matched controls attribute this stabilization to moderate pessimism rather than Wasserstein geometry alone. Anonymous code available: `https://anonymous.4open.science/r/kw-dro`.

1 INTRODUCTION

Autonomous mobile robots in automated warehouses must coordinate in narrow shared corridors while executing decentralized policies. Value factorization methods such as VDN (Sunehag et al., 2018) and QMIX (Rashid et al., 2020b) make this coordination scalable by learning a centralized team value while preserving decentralized greedy action selection. However, in high-density fulfillment centers, a single stalled drive actuator can immobilize an entire corridor queue, creating a severe operational hazard. Protecting the fleet against such hardware stalls introduces a severe combinatorial challenge: if at most K of N robots may fail simultaneously, the adversary must evaluate $\sum_{k=1}^{K}\binom{N}{k}$ nonempty joint failure patterns at every Bellman backup (recovering $2^N - 1$ only when $K = N$). Figure 1 shows one such cascade. All four robots request `FORWARD`, but forcing the leader (red ring) to `noop` leaves the queue stationary; dashed rings mark the pre-step positions. Loaded robots cannot pass beneath standing shelves, so a stall can propagate along a one-tile corridor.

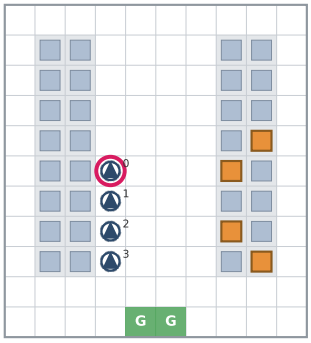

Figure 1: Example: one stalled actuator can immobilize a queue.

Wasserstein distributionally robust optimization (DRO) provides a natural framework to evaluate Bellman updates against perturbed transition laws (Blanchet & Murthy, 2019; Gao & Kleywegt, 2023; Derman & Mannor, 2020), but standard formulations fail to capture sparse concurrent failures. Under the natural Hamming metric with a nominal fault-free law, the Wasserstein transport cost satisfies $W_1(P, \widehat{P}) = \mathbb{E}_P[\|\boldsymbol{\xi}\|_1]$. Consequently, bounding the expected failure count is mathematically redundant with the Wasserstein radius and still permits probability distributions with positive mass on catastrophic fleet-wide failure, inducing extreme conservatism. A physically realistic multi-robot

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.

formulation must explicitly restrict the adversary's support to sparse realization-wise failure patterns, while keeping the inner optimization tractable within the deep RL training loop.

We introduce K**-Sparse Wasserstein Distributionally Robust MARL (KW-DRO)**. The ambiguity set restricts its support to patterns with at most K simultaneous actuator faults, while a Wasserstein radius ρ controls the probability mass assigned to them. KW-DRO replaces the nominal successor value with the worst-case expectation over this support-constrained set, without requiring a learned transition model. Our central result is that a standard convex monotone QMIX mixer makes this adversary tractable. For a single-hidden-layer mixer with nonnegative, state-conditioned weights independent of the utilities, a convex nondecreasing activation, and nonnegative utility drops, learner-side fault damage is monotone submodular. Greedy selection therefore obtains a $(1 - 1/e)$ per-cardinality guarantee using $O(NK)$ mixer evaluations rather than combinatorial enumeration (Nemhauser et al., 1978), and is exact for additive VDN. The remaining distributional optimization has a geometric solution: for any supplied candidates, the worst-case expectation is the minimum of their lower convex hull over feasible costs up to ρ. This yields an exact batched $O(M \log M)$ solver for heterogeneous transport costs and an optimal adversary supported on at most two atoms.

For VDN, the greedy chain recovers the full inner optimum. For QMIX, the guarantee compares greedy and full backups applied to the same value function; it does not compare their learned fixed points. The computed greedy backup depends on K and ρ only through $\min(K, \rho)$, so $K > 1$ is inactive when $\rho \leq 1$. Section 3 gives the core results, and Appendix A collects standard supporting facts, complete proofs, and the classical contraction result for the fixed full-support operator.

We evaluate the optimization and the learning behavior separately. Comparisons with 2,000 reference linear programs verify the inner solver across five regimes (Section 5.1). Section 5.2 reports a late VDN collapse that shorter training budgets miss and a two-sided radius regime between collapse and failure to learn. At the successful radius, $K > 1$ is inactive, so this experiment studies the effect of the robust backup rather than the concurrent-fault search itself. Our contributions in this work are:

- **A sparse fault ambiguity set.** We impose the concurrent-failure limit on the support of the Wasserstein ambiguity set, separating admissible fault patterns from the probability mass transported onto them (Section 3).
- **Tractable robust value factorization.** We prove that the stated QMIX mixer induces monotone submodular fault damage, giving $O(NK)$ greedy candidate construction with a $(1 - 1/e)$ per-cardinality guarantee and exact selection for VDN (Section 3.2).
- **A tractable robust backup and verified implementation.** We transfer the per-cardinality greedy guarantee to the robust backup and implement its probability-allocation LP by a lower convex hull; combined with exact VDN candidate selection, this gives an exact full inner solution for VDN (Sections 3 and 5.1).

2 Related Work

Value factorization for cooperative MARL. VDN uses additive per-agent utilities, while QMIX and its extensions preserve decentralized greedy action selection through structured joint-value functions (Sunehag et al., 2018; Rashid et al., 2020b; Son et al., 2019; Wang et al., 2021; Rashid et al., 2020a). Distributional variants factorize return distributions or risk-sensitive objectives (Sun et al., 2021; Qiu et al., 2021; Shen et al., 2023), and MAPPO instead trains decentralized policies with a centralized critic (Yu et al., 2022). KW-DRO uses the utilities and monotonicity of existing value-factorization architectures to optimize sparse actuator-fault patterns.

Distributionally robust reinforcement learning. Robust MDPs optimize against transition ambiguity, and Wasserstein DRO equips distribution shifts with a transport geometry that often admits tractable reformulations (Iyengar, 2005; Blanchet & Murthy, 2019; Mohajerin Esfahani & Kuhn, 2018; Gao & Kleywegt, 2023; Derman & Mannor, 2020; Yang, 2017). Robust MARL has additionally considered uncertainty in agents' states (He et al., 2023). Our ambiguity set instead governs a conditional fault channel: after the nominal successor is realized, an adversary redistributes probability over action-availability patterns containing at most K failed actuators.

Most directly related, DrIGM develops distributionally robust value factorization that aligns individual robust greedy actions with the robust joint greedy action under history–action rectangular transition-kernel ambiguity (Qu et al., 2026). KW-DRO instead retains the nominal environment kernel and

places a support-constrained Wasserstein set on sparse post-transition actuator faults. This change of uncertainty object turns the inner problem into selection among combinatorially many fault subsets followed by adversarial probability allocation; under the stated mixer conditions, greedy selection has a submodular approximation guarantee for QMIX and is exact for VDN, while the lower convex hull allocates probability exactly over the supplied candidates. Appendix D gives the detailed comparison of ambiguity sets, optimization roles, and guarantee scopes. A warehouse group-DRO formulation instead targets exogenous shifts in package-induction regimes and searches over empirical operating groups (Liu et al., 2025); its detailed contrast with actuator-level uncertainty is also deferred to Appendix D.

Robust subset selection and positioning. Selecting at most K failed components is a cardinality-constrained set-function problem, for which greedy maximization has a $(1 - 1/e)$ guarantee under monotone submodularity (Nemhauser et al., 1978), while general interdiction objectives need not be tractable (Topkis, 1978; Chestnut & Zenklusen, 2017). We identify this structure within standard monotone value factorization: fault damage is submodular for the stated QMIX mixer and modular for additive VDN, reducing candidate construction from combinatorial enumeration to $O(NK)$. KW-DRO connects decentralized value factorization, support-constrained Wasserstein robustness, and guaranteed sparse fault search.

3 Theory: Sparse Fault Selection from Convex Monotone Mixing

We isolate the part of the robust backup that is both combinatorial and specific to value factorization. A realization-wise support constraint defines admissible fault patterns; the stated QMIX mixer then makes their estimated damage submodular, permitting guaranteed greedy selection. Standard facts about product-set maximization, Wasserstein distance to a point mass, probability allocation by a lower convex hull, robust Bellman contraction, and IGM are collected in Appendix A.

3.1 K-Sparse Fault Ambiguity

Consider a fully cooperative multi-agent MDP with agents $\mathcal{N} = \{1, \dots, N\}$, joint action space $\mathcal{A} = \prod_i \mathcal{A}_i$, bounded shared reward, and discount $\gamma \in [0, 1)$. A fault vector $\boldsymbol{\xi} \in \{0, 1\}^N$ indicates which agents are forced to `noop`at the successor. Let $\mathcal{A}_i(1) = \{\texttt{noop}\}$, $\mathcal{A}_i(0) = \mathcal{A}_i$, and $\mathcal{A}(\boldsymbol{\xi}) = \prod_i \mathcal{A}_i(\xi_i)$. The corresponding continuation value is

$$V_Q^{\boldsymbol{\xi}}(s') \triangleq \max_{\boldsymbol{a}' \in \mathcal{A}(\boldsymbol{\xi})} Q(s', \boldsymbol{a}'). \tag{1}$$

The nominal law for this conditional fault channel is $\widehat{P} = \delta_{\mathbf{0}}$; stochasticity in the environment transition remains in the outer expectation over $s' \sim P_0(\cdot \mid s, \boldsymbol{a})$. We explicitly impose rectangularity at the $(s, \boldsymbol{a}, s')$ level: after each nominal successor is realized, the adversary may choose its conditional fault law independently at that tuple. Faults are transient action-availability restrictions at s' rather than perturbations of the nominal transition.

Definition 1 (K-Sparse Wasserstein Ambiguity Set)**.** *For radius $\rho \geq 0$ and concurrent-failure limit $K \in \{1, \dots, N\}$, let*

$$\Xi_K \triangleq \{\boldsymbol{\xi} \in \{0, 1\}^N : \|\boldsymbol{\xi}\|_0 \leq K\}, \quad M \triangleq |\Xi_K| = \sum_{k=0}^{K} \binom{N}{k}, \tag{2}$$

$$\mathcal{P}_{\rho,K}(\widehat{P}) \triangleq \left\{ P \in \Delta(\{0, 1\}^N) : \operatorname{supp}(P) \subseteq \Xi_K, W_1(P, \widehat{P}) \leq \rho \right\}, \tag{3}$$

where W_1 uses the Hamming ground metric.

Because the nominal fault law is a point mass, $W_1(P, \widehat{P}) = \mathbb{E}_P[\|\boldsymbol{\xi}\|_1]$ (Lemma 5). Thus K limits the number of simultaneous faults in every realization, whereas ρ limits their expected count under the adversarial law. The proof, the non-Dirac caveat, and the relation to stochastic RWARE transitions are in Appendix A.1.

For a monotone factorization $Q(s', \boldsymbol{a}') = f_{\text{mix}}(Q_1(\tau_1', a_1'), \dots, Q_N(\tau_N', a_N'); s')$, maximization over the product set $\mathcal{A}(\boldsymbol{\xi})$ reduces to one mixer evaluation (Lemma 6). Write $q_i = \max_a Q_i(\tau_i', a)$,

$\widetilde{q}_i = Q_i(\tau'_i, \texttt{noop})$, and let $\boldsymbol{q}^S$ replace q_i by $\widetilde{q}_i$ for $i \in S$. The learner-side damage caused by fault set S is

$$g(S) \triangleq f(\boldsymbol{q}) - f(\boldsymbol{q}^S). \tag{4}$$

This is damage to the represented successor value, not a claim about physical queue immobilization.

3.2 Core Result: Submodular Fault Damage

Monotonicity alone does not imply submodularity. The result requires the following convex one-hidden-layer mixer, which matches the implementation used here.

Lemma 1 (Submodular fault damage)**.** *Let*

$$f(\boldsymbol{q}) = \boldsymbol{w}_2^\top \sigma(W_1^\top \boldsymbol{q} + \boldsymbol{b}_1) + b_2 \tag{5}$$

have elementwise nonnegative W_1 and $\boldsymbol{w}_2$, weights and biases independent of $\boldsymbol{q}$ conditional on s', a convex nondecreasing activation σ, and drops $q_i - \widetilde{q}_i \geq 0$. Then g in Eq. (4) is normalized, monotone, and submodular. Let S_k be the prefix after k greedy additions. For every $k \in \{1, \ldots, K\}$,

$$g(S_k) \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \max_{|S| \leq k} g(S) \geq (1 - e^{-1}) \max_{|S| \leq k} g(S), \tag{6}$$

and constructing all K prefixes uses $O(NK)$ mixer evaluations. For additive VDN, g is modular and greedy selection is exact.

To see the diminishing-returns structure, take $S \subseteq T$ and $j \notin T$. The corresponding mixer inputs satisfy $\boldsymbol{q}^S \geq \boldsymbol{q}^T$, while faulting agent j lowers hidden-unit preactivation e by the same nonnegative amount $d_e = W_1^{je}(q_j - \widetilde{q}_j)$ in both cases. Convexity gives

$$\sigma(a_e) - \sigma(a_e - d_e) \geq \sigma(a'_e) - \sigma(a'_e - d_e) \qquad \text{for } a_e \geq a'_e, \tag{7}$$

and summing with nonnegative output weights yields $g(S \cup \{j\}) - g(S) \geq g(T \cup \{j\}) - g(T)$. Appendices A.6–A.8 give the complete proof and show how each assumption enters. In particular, a concave monotone activation reverses the diminishing-returns inequality, and double-Q evaluation need not preserve $q_i \geq \widetilde{q}_i$. The implemented QMIX mixer uses state-conditioned, utility-independent nonnegative weights and ELU with its default convex parameterization; the standard target maximization includes noopand therefore gives nonnegative drops.

3.3 Guarantee for the Robust Backup

For a supplied candidate set with values $\boldsymbol{v}$ and Hamming costs $\boldsymbol{c}$, define

$$\Lambda_\rho(\boldsymbol{v}, \boldsymbol{c}) \triangleq \min_{\boldsymbol{p} \in \Delta^{M-1}} \boldsymbol{p}^\top \boldsymbol{v} \quad \text{subject to} \quad \boldsymbol{p}^\top \boldsymbol{c} \leq \rho. \tag{8}$$

Each distribution $\boldsymbol{p}$ maps to the point $(\boldsymbol{p}^\top \boldsymbol{c}, \boldsymbol{p}^\top \boldsymbol{v})$ in the convex hull of $\{(c_m, v_m)\}_{m=1}^M$. The constraint retains points with cost coordinate at most ρ, and minimizing expected value selects the lowest feasible point on the hull. Because that point lies on a vertex or edge, an optimum uses at most two atoms. Theorem 7 formalizes this characterization and gives an $O(M \log M)$ construction. Section 4 uses its batched implementation for probability allocation.

Fix a realized successor s' and let $D_k^\star(s') = \max_{|S| \leq k} g_{s'}(S)$ and $D_k^{\mathrm{gr}}(s') = g_{s'}(S_k)$. For $\ell \in \{\star, \mathrm{gr}\}$, define

$$\mathcal{D}_{s'}^\ell(\rho) \triangleq \max_{\boldsymbol{p} \in \Delta^K} \sum_{k=0}^K p_k D_k^\ell(s') \quad \text{subject to} \quad \sum_{k=0}^K k p_k \leq \rho. \tag{9}$$

Aggregation by cardinality is lossless for $\ell = \star$, so $\mathcal{D}_{s'}^\star$ is the full-support inner damage over Ξ_K. For $\ell \in \{\star, \mathrm{gr}\}$, define the corresponding Bellman operator by

$$(\mathcal{T}_\ell Q)(s, \boldsymbol{a}) \triangleq R(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim P_0(\cdot|s,\boldsymbol{a})} \left[(1 - d(s')) \left(V_Q(s') - \mathcal{D}_{s'}^\ell(\rho)\right)\right], \tag{10}$$

and write $\mathcal{T}_{\mathrm{full}} \equiv \mathcal{T}_\star$ and $\mathcal{T}_{\mathrm{greedy}} \equiv \mathcal{T}_{\mathrm{gr}}$.

Corollary 2 (Greedy robust-backup guarantee)**.** *Under Lemma 1, for every realized successor s',*

$$\alpha \mathcal{D}^{\star}_{s'}(\rho) \le \mathcal{D}^{\mathrm{gr}}_{s'}(\rho) \le \mathcal{D}^{\star}_{s'}(\rho), \qquad \alpha \triangleq \min_{1\le k\le K} \left[1 - \left(1 - \frac{1}{k}\right)^{k}\right] \ge 1 - e^{-1}. \tag{11}$$

Consequently, for every $(s, \boldsymbol{a})$,

$$0 \le (\mathcal{T}_{\mathrm{greedy}} Q)(s, \boldsymbol{a}) - (\mathcal{T}_{\mathrm{full}} Q)(s, \boldsymbol{a}) \tag{12}$$

$$\le \gamma(1-\alpha)\, \mathbb{E}_{s'\sim P_0(\cdot|s,\boldsymbol{a})} \left[(1 - d(s'))\mathcal{D}^{\star}_{s'}(\rho)\right]. \tag{13}$$

For VDN, $\alpha = 1$ and the greedy and full backups coincide.

The comparison applies the two operators to the same Q. It is neither a fixed-point bound nor a convergence guarantee for learned function approximation. The full fixed-set robust operator is a classical γ-contraction (Theorem 8), but the practical greedy support depends on the current utilities, so that argument does not transfer to Algorithm 1. Appendix A.9 gives the aggregation and transfer proof.

3.4 When the Concurrency Budget Is Active

The greedy chain contains one candidate at every integer cost $k \in \{0, \dots, K\}$. Submodularity makes its marginal damages nonincreasing, hence its candidate values form a decreasing convex sequence.

Proposition 3 (Collapse along the greedy chain)**.** *Under Lemma 1, $\mathcal{T}_{\mathrm{greedy}}$ depends on (K, ρ) only through $\min(K, \rho)$. For VDN, greedy is optimal at every cardinality, so the same statement holds for $\mathcal{T}_{\mathrm{full}}$. In particular, $K > 1$ does not change either backup when $\rho \le 1$.*

Indeed, if $d_k = g(S_k) - g(S_{k-1})$, then submodularity gives $d_{k+1} \le d_k$ and monotonicity gives $d_k \ge 0$. Thus $v_k = v_0 - g(S_k)$ is a decreasing convex chain, so the lower-hull solution interpolates at cost $\min(K, \rho)$. This limitation is specific to the nested greedy chain: the exact cardinality-wise maximizers of a general monotone submodular function need not be nested, and the full-support operator need not collapse (Proposition 12 and its witness are in Appendix A.11). The successful runs at $\rho \le 1$ therefore exercise only the nominal and single-fault frontier. We evaluate the $K > 1$ combinatorial mechanism through exhaustive candidate tests, separately from the observed training-stability effect.

4 Method: K-Sparse Wasserstein DRO-MARL

KW-DRO changes one component of value-based training: the successor bootstrap. For each replay transition, it constructs a nested sequence of sparse fault candidates and then assigns adversarial probability mass to those candidates under radius ρ. The first stage costs $O(NK)$ mixer evaluations; the second is an exact lower-hull computation over $K + 1$ points.

4.1 Greedy Sparse Fault Candidates

At successor histories τ'_i, define the nominal and stalled utilities

$$q_i \triangleq \max_a Q_i(\tau'_i, a), \qquad \widetilde{q}_i \triangleq Q_i(\tau'_i, \texttt{noop}). \tag{14}$$

Starting from the nominal vector, Algorithm 1 repeatedly forces to `noop`the remaining agent with largest marginal reduction in the mixed value. It returns one atom at each fault cost $k \in \{0, \dots, K\}$. By Lemma 1, every prefix has the per-cardinality $(1 - 1/e)$ guarantee for QMIX. For VDN, $g(S) = \sum_{i\in S}(q_i - \widetilde{q}_i)$, so selecting agents in descending order of $q_i - \widetilde{q}_i$ makes every prefix exact and costs $O(N \log N)$ in total.

All remaining-agent trials in one greedy round are batched into one mixer call. Thus the construction requires K batched rounds rather than enumerating $\sum_{k\le K} \binom{N}{k}$ patterns. The guarantee requires $q_i \ge \widetilde{q}_i$, which the standard target backup provides because `noop`is included in the maximization. Double-Q target evaluation does not preserve this ordering; the implementation therefore rejects double-Q with $K > 1$ rather than running an unguaranteed greedy chain.

Algorithm 1 Greedy K-sparse fault candidates

Require: Nominal utilities $\boldsymbol{q}$, stalled utilities $\widetilde{\boldsymbol{q}}$, mixer $f(\cdot; s')$, limit K
$S \leftarrow \emptyset$; $v_1 \leftarrow f(\boldsymbol{q}; s')$; $c_1 \leftarrow 0$
for $k = 1$ **to** K **do**
  $j^\star \leftarrow \arg\max_{j \notin S} \left[f(\boldsymbol{q}^S; s') - f(\boldsymbol{q}^{S \cup \{j\}}; s') \right]$
  $S \leftarrow S \cup \{j^\star\}$; $v_{k+1} \leftarrow f(\boldsymbol{q}^S; s')$; $c_{k+1} \leftarrow k$
end for
return $\boldsymbol{v}, \boldsymbol{c}$

Algorithm 2 Lower-hull algorithm for $\Lambda_\rho(\boldsymbol{v}, \boldsymbol{c})$

Require: Values $\boldsymbol{v} \in \mathbb{R}^M$, costs $\boldsymbol{c} \in \mathbb{R}^M_{\geq 0}$ with $c_1 = 0$, radius ρ
Sort (c_m, v_m) by c_m ascending and v_m descending
$\mathcal{H} \leftarrow \textsc{LowerConvexHull}(\{(c_m, v_m)\}_{m=1}^M)$
$\bar{c} \leftarrow \min(\rho, \arg\min_{c \in \mathcal{H}} h(c))$
Find the hull edge $(c_l, v_l) \to (c_r, v_r)$ containing $\bar{c}$
$\theta \leftarrow (\bar{c} - c_l)/(c_r - c_l)$, or 0 if $c_l = c_r$
return $(1-\theta)v_l + \theta v_r$ and $(1-\theta)\boldsymbol{e}_l + \theta \boldsymbol{e}_r$

4.2 Exact Probability Allocation

Algorithm 2 applies Theorem 7 to the candidate values and integer transport costs returned above. Sorting equal-cost points by descending value ensures that the lowest point is retained at the maximal cost; Appendix A.3 gives the edge case and its verification.

The output is the exact worst-case expectation and an optimal distribution over the supplied candidates. For VDN these candidates contain the optimum at every cardinality, so the combined procedure solves the full K-sparse inner problem exactly. For QMIX, the hull stage is exact and the candidate stage carries the guarantee of Lemma 1.

4.3 Training Objective

For VDN or QMIX, let $\boldsymbol{v}_{\text{next}}(\theta^-)$ be the candidate values produced from the target utilities and target mixer, and let $r = \sum_i r_i$ be RWARE's team reward obtained by summing its per-agent reward list. Let $d_{t+1} = \mathbb{I}\{s' \text{ is terminal}\}$, excluding time-limit truncation; RWARE has no terminal state, so $d_{t+1} = 0$ here. KW-DRO replaces the nominal bootstrap by

$$y_{\text{DRO}} = r + \gamma(1 - d_{t+1})\, \Lambda_\rho\big(\boldsymbol{v}_{\text{next}}(\theta^-), \boldsymbol{c}\big)\,, \tag{15}$$

$$\mathcal{L}(\theta) = \mathbb{E}\Big[\big(Q_{\text{tot}}(s, \boldsymbol{a}; \theta) - y_{\text{DRO}}\big)^2\Big]\,. \tag{16}$$

The target is evaluated without gradients, leaving the optimizer and decentralized action rule unchanged. The joint sparse construction requires per-agent utilities and therefore applies directly to VDN and QMIX. Potential-based reward shaping is an optional environment-side aid for sparse rewards, not part of KW-DRO. Its definition, invariance conditions, and reporting convention are given in Appendix B.

5 Experiments

The experiments address two questions: whether the proposed optimization recovers the exact inner solution, and how the resulting backup affects long-horizon learning. Appendices C and F provide additional diagnostics and complete training settings.

5.1 Operator Verification

We compare Algorithm 2 with a general-purpose linear-program solver on $2{,}000$ random instances spanning equal, duplicated, and continuous transport costs (Appendix C.10). No absolute difference exceeds 4.5×10^{-7}, every returned distribution is feasible, and ratio-greedy is suboptimal in the

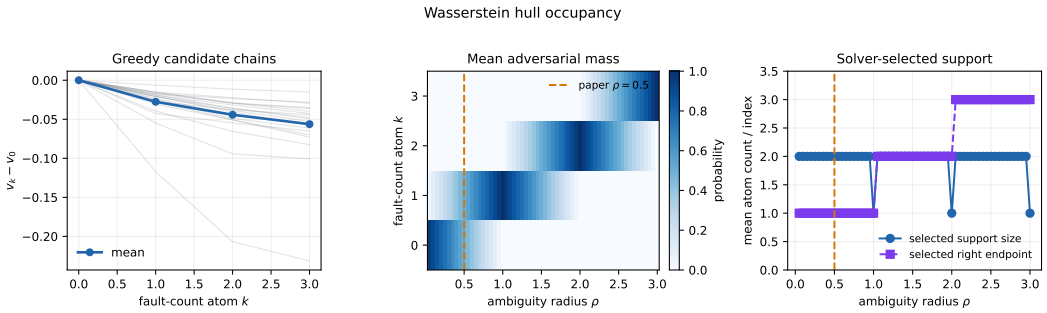

Figure 2: Wasserstein hull occupancy on a trained KW-DRO QMIX checkpoint. The plots use 512 successor states from the seed-1 greedy rollout on `small-8ag` with $K = 3$. Left: greedy candidate values relative to the nominal value; the blue line is the statewise mean and gray lines show a representative subset. Center: mean adversarial mass on each fault-count atom as ρ varies. Right: support size and largest active atom. The dashed line marks the training radius $\rho = 0.5$, where all sampled states use only the nominal and single-fault atoms. Integer radii can reduce the returned support to one atom, producing the isolated dips in support size. The plotted support is an extreme optimum; degenerate instances may admit other optimal distributions with the same robust value.

unequal-cost regimes. On trained utilities, the predicted $\min(K, \rho)$ structure also holds to machine precision.

Figure 2 complements the synthetic LP tests with candidate values produced by a trained policy. Although the implementation constructs the full $K = 3$ chain, the inner solution at $\rho = 0.5$ lies on its first hull segment and assigns mass only to $k = 0$ and $k = 1$. At larger radii, atoms representing two and three simultaneous faults become active. This single-checkpoint view characterizes the operating regime; the multi-seed greedy–exhaustive comparisons below evaluate candidate selection more broadly.

Greedy and exhaustive QMIX search agree on the tested states. We evaluate three trained QMIX checkpoints on `small-8ag`, each using 512 successor states sampled from its own rollout. At $N = 8$ and $K = 3$, exhaustive selection evaluates all 93 admissible fault sets, whereas the greedy chain uses 24 mixer evaluations. Greedy selects the same set as exhaustive search at every cardinality on all 1536 successors, giving approximation ratio 1.0 and zero robust-target gap for every $\rho \in \{0.5, 1, 2, 3\}$.

The same agreement holds on 512 independent environment resets and on a randomly initialized mixer of the same architecture. At the native actuator-fault scale, pairwise non-modularity is below 0.5% of mean singleton damage on every trained seed. These instances are therefore close to the additive VDN case, and greedy search recovers the exhaustive solution throughout the tested sample. Appendix C.11 reports the full comparison and an amplified-drop sensitivity analysis.

KW-DRO with QMIX. We also train the architecture covered by the submodularity result. Across 30 QMIX runs at 10^7 steps, both the nominal and KW-DRO backups keep every policy alive: 10/10 runs per arm on `small-8ag` and 5/5 per arm on `tiny-4ag`. Sustained delivery rates are 98.7% versus 99.6% on `small-8ag` and 92.6% versus 99.2% on `tiny-4ag`. Deliveries per 1000 steps favor KW-DRO in both environments, but the paired comparisons are unresolved ($p = 0.084$ and $p = 0.313$). These runs establish that the greedy chain and hull solver can be used throughout QMIX training; the saturated survival endpoint does not show an additional performance benefit. Appendix C.12 reports the full results, and Appendix C.2 gives the target-update calibration used to obtain competent QMIX policies.

5.2 Long-Horizon Training Behavior

Late collapse under nominal VDN. At 10^7 steps, correctly configured value-based methods learn RWARE (Appendix C.2), but nominal VDN later falls to persistent zero delivery on 4 of 10 seeded `small-8ag` runs.



Figure 3: Late VDN collapse on `small-8ag` over 10^7 steps, using the same seeded cohort as Table 1. All ten runs per arm are shown, ordered by seed with no trajectory selection. Faint curves are raw evaluation delivery rates and bold curves are trailing-five-evaluation means. Four nominal runs enter a terminal zero-delivery block; all ten $K = 3, \rho = 0.5$ runs remain alive.

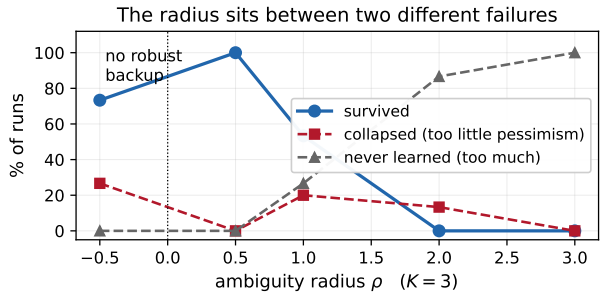

Figure 4: Learning outcomes across the ambiguity radius for VDN, with five runs on each of three environments per radius. Small radii permit late policy collapse, while large radii prevent learning; $\rho = 0.5$ is the only tested value at which all 15 runs remain alive. The seeded comparisons at $\rho \in \{0, 0.5, 1\}$ appear in Table 1.

The collapse begins after roughly three-quarters of training and forms a terminal zero-delivery block (Figure 3). Across both seeded environments, all six non-alive nominal runs had previously learned the task. Appendix C.5 reports the same pattern in three earlier cohorts.

A two-sided radius window. The radius separates two qualitatively different failure modes. Under the nominal backup ($\rho = 0$), $6/20$ seeded runs first learn and then collapse. At $\rho = 0.5$, all $20/20$ runs remain alive. At $\rho = 1$, survival falls to $9/20$, but none of the remaining runs collapses after learning: all 11 fail to form a working policy. Moderate pessimism therefore prevents late collapse, while a larger radius changes the failure mode from instability to failure to learn.

Figure 4 shows a stability window rather than a monotone robustness–performance trade-off: no run remains alive beyond $\rho = 2$. At $\rho = 0.5$, Proposition 3 makes the implemented backup identical for every $K \geq 1$; this training result therefore isolates the effective radius but does not distinguish concurrency budgets.

Comparison with generic pessimism. On 512 replay successors, the robust target lies an average of 0.36% below the nominal bootstrap at $\rho = 0.5$ and 0.72% below it at $\rho = 1$. For each radius, we compare a scalar reduction matched to this measured checkpoint-level mean, along with double Q-learning as a standard correction for optimistic targets. Each arm uses 10 seeded runs per environment and nominal evaluation.

At $\rho = 0.5$, KW-DRO and double Q both reach $20/20$ survival, compared with $14/20$ under the nominal backup. The matched 0.36% constant reaches $18/20$, and its direct comparison with KW-DRO is unresolved at this sample size ($p = 0.50$). Moderate pessimism stabilizes the observed collapse, but these comparisons do not identify Wasserstein geometry as the cause.

Table 1: Seeded VDN controls after 10^7 steps, 10 seeds per environment. *Survival* follows the endpoint in Appendix F; p is the exact environment-stratified comparison with nominal VDN. *Deliveries* are measured at the final checkpoint over surviving runs. Matched constants reproduce each robust arm's checkpoint-level mean bootstrap reduction but not its state-dependent placement. **Bold** marks $20/20$ survival and, among arms with at least eight survivors, the smallest dispersion.

	survival				small-8ag		tiny-4ag	
Training backup	s-8ag	t-4ag	total	p	mean	σ	mean	σ
nominal	$6/10$	$8/10$	$14/20$	—	9.85	11.48	10.90	6.85
KW-DRO, $\rho = 0.5$	$10/10$	$10/10$	$\mathbf{20/20}$	0.020	6.53	**4.38**	11.33	**6.41**
constant, 0.36%	$9/10$	$9/10$	$18/20$	0.235	3.74	5.00	13.77	11.06
KW-DRO, $\rho = 1$	$6/10$	$3/10$	$9/20$	0.200	10.40	6.89	4.50	1.97
constant, 0.72%	$10/10$	$10/10$	$\mathbf{20/20}$	0.020	11.22	9.70	14.21	10.27
double Q	$10/10$	$10/10$	$\mathbf{20/20}$	0.020	9.82	10.28	14.12	6.47

At $\rho = 1$, KW-DRO and its checkpoint-matched 0.72% constant achieve $9/20$ and $20/20$ survival, respectively ($p = 1.5 \times 10^{-4}$). All 11 unsuccessful KW-DRO runs fail before learning rather than collapsing afterward. The state-dependent robust backup and the scalar control thus produce different training regimes at the larger radius. Among high-survival arms, KW-DRO at $\rho = 0.5$ also has the smallest observed survivor-conditioned dispersion in both environments. This comparison is descriptive; the primary endpoint remains survival. Appendices C and E report mechanism diagnostics, actuator-training controls, reference baselines, and the evaluation audit.

Actuator-only deployment. Deployment evaluation isolates actuator faults because observation masking changes throughput independently of the fault intervention. With the mask removed, KW-DRO and its matched constant yield 9.2 and 8.0 deliveries per 1000 steps at $\rho = 0.5$ ($p = 0.37$), while the corresponding values at $\rho = 1$ are 3.9 and 13.3 ($p = 0.002$). These tests resolve no deployment difference at the working radius and reproduce the unfavorable higher-radius regime observed during training. Appendix C.9 gives the fault protocol, the masking analysis, and complete results.

6 Conclusion

KW-DRO makes sparse-fault backups tractable for cooperative value factorization. Its support constraint limits each fault realization to at most K agents, while the Wasserstein radius ρ controls the expected fault count. Under the stated single-hidden-layer convex monotone QMIX mixer, learner-side fault damage is submodular, yielding an $O(NK)$ greedy search with a $(1 - 1/e)$ per-cardinality guarantee. Candidate selection is exact for VDN, and probability allocation over any supplied candidates reduces to an exact lower-convex-hull computation.

The experiments verify these computational components and identify a separate learning phenomenon. The hull solver agrees with reference linear programs, and greedy and exhaustive fault selection coincide on the tested QMIX checkpoints. In long-horizon RWARE training, nominal VDN sometimes learns and then collapses to persistent zero delivery. A moderate robust backup prevents this collapse in the seeded cohort, although double Q and the matched scalar control show that the stabilization is not specific to Wasserstein geometry. Moreover, the successful radius activates only the nominal and single-fault frontier, leaving the effect of $K > 1$ during learning unresolved. The present contribution is therefore a scalable sparse-fault backup together with a bounded case study of its training behavior. A natural next step is to estimate the ambiguity geometry from operational fault data and evaluate the method in regimes where concurrent-fault atoms are active.

AI Disclosure

In this work, we used generative AI tools for brainstorming, polishing writing, and code completion. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019.

Stephen R. Chestnut and Rico Zenklusen. Hardness and approximation for network flow interdiction. *Networks*, 69(4):378–387, 2017. doi: 10.1002/net.21739.

Filippos Christianos, Lukas Schäfer, and Stefano Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. *Advances in neural information processing systems*, 33: 10707–10717, 2020.

Esther Derman and Shie Mannor. Distributional robustness and regularization in reinforcement learning. *arXiv preprint arXiv:2003.02894*, 2020.

Sam Devlin and Daniel Kudenko. Theoretical considerations of potential-based reward shaping for multi-agent systems. In *Tenth international conference on autonomous agents and multi-agent systems*, pp. 225–232. ACM, 2011.

Rui Gao and Anton Kleywegt. Distributionally robust stochastic optimization with wasserstein distance. *Mathematics of Operations Research*, 48(2):603–655, 2023.

Sihong He, Songyang Han, Sanbao Su, Shuo Han, Shaofeng Zou, and Fei Miao. Robust multi-agent reinforcement learning with state uncertainty. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

Guangyi Liu, Suzan Iloglu, Michael Caldara, Joseph W Durham, and Michael M. Zavlanos. Distributionally robust multi-agent reinforcement learning for dynamic chute mapping. In *Forty-second International Conference on Machine Learning*, 2025.

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1–2):115–166, 2018.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Wei Qiu, Xinrun Wang, Runsheng Yu, Rundong Wang, Xu He, Bo An, Svetlana Obraztsova, and Zinovi Rabinovich. Rmix: Learning risk-sensitive policies for cooperative reinforcement learning agents. *Advances in Neural Information Processing Systems*, 34:23049–23062, 2021.

Chengrui Ray Qu, Christopher Yeh, Kishan Panaganti, Eric Mazumdar, and Adam Wierman. Distributionally robust cooperative multi-agent reinforcement learning with value factorization. In *International Conference on Learning Representations*, volume 2026, pp. 9420–9446, 2026.

Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10199–10210, 2020a.

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, 2020b.

Siqi Shen, Chennan Ma, Chao Li, Weiquan Liu, Yongquan Fu, Songzhu Mei, Xinwang Liu, and Cheng Wang. Riskq: risk-sensitive multi-agent reinforcement learning value factorization. *Advances in Neural Information Processing Systems*, 36:34791–34825, 2023.

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 *International conference on machine learning*, pp. 5887–5896. PMLR, 2019.

Wei-Fang Sun, Cheng-Kuang Lee, and Chun-Yi Lee. Dfac framework: Factorizing the value function via quantile mixture for multi-agent distributional q-learning. In *International conference on machine learning*, pp. 9945–9954. PMLR, 2021.

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 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pp. 2085–2087, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

Donald M. Topkis. Minimizing a submodular function on a lattice. *Operations Research*, 26(2): 305–321, 1978. doi: 10.1287/opre.26.2.305.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. {QPLEX}: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021.

Insoon Yang. A convex optimization approach to distributionally robust Markov decision processes with Wasserstein distance. *IEEE Control Systems Letters*, 1(1):164–169, 2017. doi: 10.1109/LCSYS.2017.2711553.

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.

A PROOFS

This appendix gives proofs and supporting details for the results in Section 3. Throughout, $\mathcal{N} = \{1, \dots, N\}$ indexes agents and $\Delta^{M-1} = \{\boldsymbol{p} \in \mathbb{R}^M : \boldsymbol{p} \geq 0, \sum_m p_m = 1\}$ denotes the probability simplex on M atoms.

A.1 SETUP RECALLED

Fix a state-action pair $(s, \boldsymbol{a})$ and write s' for its successor. Each robot has action set $\mathcal{A}_i = \{\texttt{noop}, \texttt{forward}, \texttt{left}, \texttt{right}, \texttt{toggle-load}\}$. A fault pattern is a binary vector $\boldsymbol{\xi} \in \{0,1\}^N$ with $\xi_i = 1$ meaning "agent i's actuator has stalled, so its action at s' is forced to `noop`". The pattern does not change s'; it restricts the joint action space available there to the product set $\mathcal{A}(\boldsymbol{\xi})$, giving the fault-constrained continuation value $V^{\boldsymbol{\xi}}(s') = \max_{\boldsymbol{a}' \in \mathcal{A}(\boldsymbol{\xi})} Q(s', \boldsymbol{a}')$ of (1).

The candidate set admitted by budget K is $\Xi_K = \{\boldsymbol{\xi} : \|\boldsymbol{\xi}\|_0 \leq K\}$, enumerated as $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_M$ with $\boldsymbol{\xi}_1 = \mathbf{0}$. We abbreviate

$$v_m \triangleq V^{\boldsymbol{\xi}_m}(s'), \qquad c_m \triangleq \|\boldsymbol{\xi}_m\|_1 = \text{ number of stalled robots in pattern } m, \qquad c_1 = 0,$$

so that $v_1 = V(s')$ is the ordinary greedy value. The nominal law is the Dirac $\widehat{P} = \delta_{\boldsymbol{\xi}_1} = \delta_{\mathbf{0}}$ on $\{0,1\}^N$.

Two facts about $V^{\boldsymbol{\xi}}$ are used repeatedly. Both are immediate, and we record them so that later proofs can cite them rather than re-derive them.

Lemma 4 ($V^{\boldsymbol{\xi}}$ is 1-Lipschitz in Q)**.** *For any two bounded action-value functions $Q, \widetilde{Q}$ and any $\boldsymbol{\xi}$,* $\left|V_Q^{\boldsymbol{\xi}}(s') - V_{\widetilde{Q}}^{\boldsymbol{\xi}}(s')\right| \leq \|Q - \widetilde{Q}\|_\infty$.

Proof. A supremum over a fixed non-empty set is 1-Lipschitz in the function being maximized: for every $\boldsymbol{a}' \in \mathcal{A}(\boldsymbol{\xi})$ we have $Q(s', \boldsymbol{a}') \le \widetilde{Q}(s', \boldsymbol{a}') + \|Q - \widetilde{Q}\|_\infty$, and taking the maximum of both sides over $\mathcal{A}(\boldsymbol{\xi})$ gives $V_Q^{\boldsymbol{\xi}} \le V_{\widetilde{Q}}^{\boldsymbol{\xi}} + \|Q - \widetilde{Q}\|_\infty$. Exchanging the roles of Q and $\widetilde{Q}$ gives the reverse inequality, and the two together give the absolute value. The set $\mathcal{A}(\boldsymbol{\xi})$ is non-empty because `noop`is always available. □

Lemma 5 (The Wasserstein constraint is linear here)**.** *With Hamming ground metric* $d(\boldsymbol{\xi}, \boldsymbol{\xi}') = \|\boldsymbol{\xi} - \boldsymbol{\xi}'\|_1$ *and* $\widehat{P} = \delta_{\mathbf{0}}$*, every* $P = \sum_m p_m \delta_{\boldsymbol{\xi}_m}$ *satisfies* $W_1(P, \widehat{P}) = \sum_m p_m c_m$*.*

Proof. $W_1(P, \widehat{P}) = \inf_\Pi \int d \, d\Pi$ over couplings Π with marginals P and $\widehat{P}$. Because the second marginal is a point mass at $\mathbf{0}$, there is exactly one coupling: it must send all of the mass p_m sitting at $\boldsymbol{\xi}_m$ to $\mathbf{0}$. There is no infimum to take. Its cost is $\sum_m p_m \, d(\boldsymbol{\xi}_m, \mathbf{0}) = \sum_m p_m \|\boldsymbol{\xi}_m\|_1 = \sum_m p_m c_m$. □

Lemma 5 is also the formal content of the remark in Section 3 that an expectation-form budget is redundant: $\mathbb{E}_P[\|\boldsymbol{\xi}\|_1] = \sum_m p_m c_m$ is literally the same quantity as $W_1(P, \widehat{P})$, so imposing both $W_1 \le \rho$ and $\mathbb{E}_P[\|\boldsymbol{\xi}\|_1] \le K$ yields the single constraint $\sum_m p_m c_m \le \min(\rho, K)$, and the budget contributes nothing that the radius does not already say.

The Dirac nominal is a hypothesis, not a convenience. Every use we make of W_1 passes through Lemma 5, which replaces the Wasserstein constraint by the linear surrogate $\sum_m p_m c_m \le \rho$. That replacement is exact only because $\widehat{P}$ is a point mass, which forces a unique coupling. For a non-Dirac nominal, the surrogate is not merely loose; it fails to be a bound in either direction. Take $\mathcal{X} = \{0, 1\}$ with $d(0, 1) = 1$, nominal $\nu = \frac{1}{2}\delta_0 + \frac{1}{2}\delta_1$, and $\rho = \frac{1}{4}$. The measure $P = \delta_0$ has $W_1(P, \nu) = \frac{1}{2} > \rho$, so it lies outside the true ball, yet its surrogate cost $\sum_m p_m d(x_m, 0) = 0$ satisfies the constraint. Conversely $P = \nu$ has $W_1 = 0$ and lies inside the true ball, yet its surrogate cost is $\frac{1}{2} > \rho$ and it is rejected. Neither set contains the other.

This matters for the present environment. RWARE resamples the requested-shelf set whenever a delivery completes, so the successor of a state-action pair is not deterministic and the true nominal environment kernel is not a Dirac. We therefore read $\widehat{P} = \delta_{\mathbf{0}}$ as a statement about the *fault* channel only: conditional on the realized shelf resampling, no actuator fails under the nominal. Shelf resampling is an exogenous source of stochasticity that the ambiguity set does not model and the adversary does not control. Our robustness claims are claims about the conditional fault channel, and none of them extend to distribution shift in the order stream.

The training ambiguity set and runtime fault wrapper are distinct. The robust backup conditions on a sampled nominal successor s' and optimizes an arbitrary law $P \in \mathcal{P}_{\rho,K}$ over next-step action-availability patterns. The environment wrapper instead selects failed agents before executing the current joint command and overwrites their physical actions with `noop`. Its random mode samples agents independently and then caps the realized count at K; its targeted mode applies a heuristic ranking. The wrapper neither receives ρ nor samples the optimizer returned by the inner problem. Runtime fault evaluations are therefore out-of-distribution stress tests under a separate transient fault channel, not samples from, or empirical validation of, the Wasserstein ambiguity set. In the registered environments communication is disabled, so forcing an agent to `noop`coincides with restricting its full action to the singleton {`noop`}; with message bits enabled, the wrapper would preserve communication while disabling only the physical actuator.

A.2 Fault-Constrained Maximization over a Product Action Set

Lemma 6 (Fault-constrained values decompose)**.** *Let* $Q(s', \boldsymbol{a}') = f_{\mathrm{mix}}(Q_1(\tau_1', a_1'), \ldots, Q_N(\tau_N', a_N'); s')$ *with* f_{mix} *nondecreasing in every utility. Define* $q_i^{\boldsymbol{\xi}} = Q_i(\tau_i', \texttt{noop})$ *when* $\xi_i = 1$ *and* $q_i^{\boldsymbol{\xi}} = \max_{a_i'} Q_i(\tau_i', a_i')$ *otherwise. Then*

$$V^{\boldsymbol{\xi}}(s') = f_{\mathrm{mix}}(\boldsymbol{q}^{\boldsymbol{\xi}}; s'). \tag{17}$$

Because $\mathcal{A}(\boldsymbol{\xi}) = \prod_i \mathcal{A}_i(\xi_i)$ is a product set, each utility can be maximized independently. For any $\boldsymbol{a}' \in \mathcal{A}(\boldsymbol{\xi})$ and each i,

$$Q_i(\tau_i', a_i') \;\le\; \max_{b \in \mathcal{A}_i(\xi_i)} Q_i(\tau_i', b) \;=\; q_i^{\boldsymbol{\xi}},$$

where the equality is the definition of $q_i^{\boldsymbol{\xi}}$: for $\xi_i = 1$ the set $\mathcal{A}_i(1)$ is the singleton $\{\texttt{noop}\}$ so its maximum is $Q_i(\tau_i', \texttt{noop})$, and for $\xi_i = 0$ it is all of $\mathcal{A}_i$. Applying f_{mix}, which is nondecreasing in every argument, to the vector inequality gives

$$Q(s', \boldsymbol{a}') = f_{\mathrm{mix}}\big(Q_1(s', a_1'), \ldots, Q_N(s', a_N'); s'\big) \;\le\; f_{\mathrm{mix}}(\boldsymbol{q}^{\boldsymbol{\xi}}; s').$$

Taking the maximum of the left-hand side over $\boldsymbol{a}' \in \mathcal{A}(\boldsymbol{\xi})$ yields $V^{\boldsymbol{\xi}}(s') \le f_{\mathrm{mix}}(\boldsymbol{q}^{\boldsymbol{\xi}}; s')$.

For the reverse, let $a_i^\dagger \in \arg\max_{b \in \mathcal{A}_i(\xi_i)} Q_i(\tau_i', b)$ for each i, which exists because each $\mathcal{A}_i(\xi_i)$ is finite and non-empty ($\texttt{noop}$is always available). Because $\mathcal{A}(\boldsymbol{\xi})$ is a product set, the assembled $\boldsymbol{a}^\dagger = (a_1^\dagger, \ldots, a_N^\dagger)$ is itself a member of $\mathcal{A}(\boldsymbol{\xi})$ (which is the step that fails for a non-product constraint), and by construction $Q(s', \boldsymbol{a}^\dagger) = f_{\mathrm{mix}}(\boldsymbol{q}^{\boldsymbol{\xi}}; s')$. Hence $V^{\boldsymbol{\xi}}(s') \ge f_{\mathrm{mix}}(\boldsymbol{q}^{\boldsymbol{\xi}}; s')$, and the two inequalities give equality. □

Role of monotonicity. The proof uses only the sign of $\partial f_{\mathrm{mix}}/\partial Q_i$, not its magnitude or convexity. If f_{mix} decreases in coordinate i, raising Q_i can lower the total, and the per-agent maximizer need not maximize the mixed value. In a numerical validation over all 2^N fault sets, the identity holds to machine precision for randomly initialized monotone mixers and fails after sign-flipping one coordinate's contribution.

A.3 Lower-Hull Solution of the Probability-Allocation LP

Theorem 7 (Exact hull characterization)**.** *Let h be the lower convex hull of the points $\{(c_m, v_m)\}_{m=1}^M$, let $c^\star$ be the leftmost minimizer of h, and let $c_{\max} = \max_m c_m$. Then*

$$\Lambda_\rho(\boldsymbol{v}, \boldsymbol{c}) = \min_{c \in [0, \min(\rho, c_{\max})]} h(c) = h(\min(c^\star, \rho))\,. \tag{18}$$

An optimal distribution exists with support on at most the two endpoints of the hull edge containing $\min(c^\star, \rho)$, and the value and weights are computable in $O(M \log M)$ exact-arithmetic operations.

By Lemma 5, the inner problem is $\Lambda_\rho(\boldsymbol{v}, \boldsymbol{c})$ from (8).

Step 1: the reachable set is a planar convex hull. Define the linear map $\Psi : \mathbb{R}^M \to \mathbb{R}^2$ by

$$\Psi(\boldsymbol{p}) \;\triangleq\; \Big(\textstyle\sum_m p_m c_m,\; \sum_m p_m v_m\Big),$$

which sends a distribution to its (expected cost, expected value) pair. Let $\mathcal{C} \triangleq \mathrm{conv}\{(c_m, v_m) : m = 1, \ldots, M\} \subset \mathbb{R}^2$.

We claim $\Psi(\Delta^{M-1}) = \mathcal{C}$, i.e. the pairs reachable by some distribution are exactly the points of the hull.

($\subseteq$) For $\boldsymbol{p} \in \Delta^{M-1}$, $\Psi(\boldsymbol{p}) = \sum_m p_m (c_m, v_m)$ is by definition a convex combination of the points (c_m, v_m), hence lies in $\mathcal{C}$.

($\supseteq$) Conversely, any point of $\mathcal{C}$ is by definition of the convex hull a convex combination $\sum_m \lambda_m (c_m, v_m)$ with $\boldsymbol{\lambda} \in \Delta^{M-1}$, and then $\Psi(\boldsymbol{\lambda})$ is that point.

The objective defining $\Lambda_\rho(\boldsymbol{v}, \boldsymbol{c})$ is the second coordinate of $\Psi(\boldsymbol{p})$, and its constraint says the first coordinate is at most ρ. Thus

$$\Lambda_\rho(\boldsymbol{v}, \boldsymbol{c}) \;=\; \min\{\, v \;:\; (c, v) \in \mathcal{C},\; c \le \rho \,\}. \tag{19}$$

Thus the M-dimensional LP is equivalent to finding the lowest point of this convex polygon with cost coordinate at most ρ.

Step 2: the lowest point at each abscissa is the lower convex hull. Define, for c in the range $[\min_m c_m, \max_m c_m] = [0, \max_m c_m]$,

$$h(c) \triangleq \min\{\, v : (c, v) \in \mathcal{C} \,\}.$$

This minimum is attained because $\mathcal{C}$ is compact (the convex hull of finitely many points) and the vertical slice at c is a non-empty closed interval whenever c is in the stated range. The function h is convex: for c_a, c_b in range and $t \in [0,1]$, pick $(c_a, h(c_a))$ and $(c_b, h(c_b))$ in $\mathcal{C}$; their convex combination $\big(tc_a + (1-t)c_b,\ t\,h(c_a) + (1-t)h(c_b)\big)$ lies in $\mathcal{C}$ since $\mathcal{C}$ is convex, so by definition of h as a minimum, $h(tc_a + (1-t)c_b) \le t\,h(c_a) + (1-t)h(c_b)$. Moreover h is exactly the lower convex hull of the point set in the sense used in Theorem 7: it is piecewise linear with breakpoints at a subset of the (c_m, v_m), and it is the largest convex function lying weakly below all of them. Combining with (19),

$$\Lambda_\rho(\boldsymbol{v}, \boldsymbol{c}) \;=\; \min_{c \in [0, \min(\rho, c_{\max})]} h(c), \qquad c_{\max} \triangleq \max_m c_m.$$

Step 3: locating the constrained minimizer. Let $c^\star \in \arg\min_{c \in [0, c_{\max}]} h(c)$ (any minimizer; the set of minimizers is an interval, and we take its left endpoint for definiteness). Because h is convex, it is non-increasing on $[0, c^\star]$ and non-decreasing on $[c^\star, c_{\max}]$. Hence:

- If $\rho \ge c^\star$, the unconstrained minimizer is feasible and the constrained minimum is $h(c^\star)$.
- If $\rho < c^\star$, then h is non-increasing throughout $[0, \rho]$, so its minimum over that interval is at the right endpoint, $h(\rho)$.

Both cases are summarized by $\min_{c \in [0, \min(\rho, c_{\max})]} h(c) = h(\bar{c})$ with $\bar{c} \triangleq \min(c^\star, \rho)$, which is (18).

Step 4: an optimal distribution has at most two atoms. The point $(\bar{c}, h(\bar{c}))$ lies on the graph of h, which is a piecewise-linear curve whose vertices are among the (c_m, v_m). Therefore $(\bar{c}, h(\bar{c}))$ lies on some segment joining two of the original points, say (c_l, v_l) and (c_r, v_r) with $c_l \le \bar{c} \le c_r$, and can be written as

$$(\bar{c}, h(\bar{c})) = (1-\theta)(c_l, v_l) + \theta(c_r, v_r), \qquad \theta = \frac{\bar{c} - c_l}{c_r - c_l} \in [0,1]$$

(with $\theta \triangleq 0$ if $c_l = c_r$). Setting $\boldsymbol{p}^\star = (1-\theta)\boldsymbol{e}_l + \theta \boldsymbol{e}_r$ gives a feasible distribution with $\Psi(\boldsymbol{p}^\star) = (\bar{c}, h(\bar{c}))$, so it attains the optimum and has at most two non-zero entries. If $\bar{c} = c^\star$ is a vertex of h, then $\theta \in \{0, 1\}$ and $\boldsymbol{p}^\star$ is a single atom.

Step 5: complexity. Sorting the M points by abscissa costs $O(M \log M)$. Andrew's monotone chain then sweeps them once, maintaining a stack and popping whenever the last three points fail to turn in the required direction; each point is pushed and popped at most once, so the sweep is $O(M)$. Locating the edge containing $\bar{c}$ is a binary search over at most M hull vertices, $O(\log M)$, and the interpolation in Step 4 is $O(1)$. Total: $O(M \log M)$. □

Remark: why the ratio-greedy heuristic fails. It is tempting to treat (LP) as a fractional knapsack: sort candidates by "damage per unit transport cost" $r_m = (v_1 - v_m)/c_m$ and fill greedily until the budget ρ is exhausted. Fractional knapsack is exact for a *single* budget constraint, but (LP) has a second coupling constraint, $\sum_m p_m = 1$. Concretely take

$$\boldsymbol{v} = (10,\ 0,\ 1), \qquad \boldsymbol{c} = (0,\ 10,\ 1), \qquad \rho = 10.$$

The ratios are $r_2 = (10-0)/10 = 1$ and $r_3 = (10-1)/1 = 9$, so greedy processes candidate 3 first and assigns it $p_3 = \min(\rho/c_3,\ 1) = \min(10, 1) = 1$. This exhausts the probability mass, leaving nothing for candidate 2, and returns objective value 1. The optimum instead puts $p_2 = 1$, which is feasible ($1 \cdot 10 \le 10$) and attains 0. The hull construction gets this right: the points are $(0, 10), (1, 1), (10, 0)$, the lower hull passes through all three, its least ordinate is at $c^\star = 10 \le \rho$, and Step 3 returns $h(10) = 0$.

Remark: tie-breaking in Step 5. Candidates sharing an abscissa form a vertical stack of which only the lowest belongs to the lower hull. The monotone chain's popping test is degenerate for such a stack (three collinear vertical points yield a zero-area cross product) while the current point is pushed unconditionally, so whichever equal-cost point comes last in the sort order is the one retained. The consequence is confined to one place. At an interior abscissa the next point, at strictly larger cost, pops the spurious vertex, so the hull and the reported value are correct under either tie order. At the maximal abscissa nothing comes later: the spurious point survives as the right endpoint of the hull and is returned as soon as ρ reaches that cost, understating the adversary. Ordering ties by descending value puts the minimum last and closes this. Empirically, over 400 random instances with duplicated integer costs an ascending tie order disagrees with a reference LP on 40, worst absolute error 14.1, all of them at the maximal abscissa with $\rho \geq \max_m c_m$; the descending order disagrees on none.

A.4 Classical Contraction of the Full Fixed-Set Operator

Theorem 8 (Contraction of the full robust operator)**.** *The robust Bellman operator $\mathcal{T}_{\mathrm{DRO}}$ in Eq. (20), induced by a fixed nonempty rectangular ambiguity correspondence $\mathcal{P}_{\rho,K}$, is a γ-contraction in $\|\cdot\|_\infty$ and has a unique fixed point.*

Let $P_0(\cdot \mid s, \boldsymbol{a})$ denote the nominal transition kernel for the environment dynamics, and let the fixed ambiguity set $\mathcal{P}_{\rho,K}$ act on the conditional fault channel at the realized successor. Rectangularity is imposed at the $(s, \boldsymbol{a}, s')$ level: the adversary chooses a fault law independently for each realized s', which is the conditioning the implementation uses and the level at which Step 2 below is stated. Define

$$(\mathcal{T}_{\mathrm{DRO}}Q)(s,\boldsymbol{a}) \triangleq R(s,\boldsymbol{a}) + \gamma \mathbb{E}_{s'\sim P_0(\cdot|s,\boldsymbol{a})}\left[\inf_{P\in\mathcal{P}_{\rho,K}(\widehat{P})} \mathbb{E}_{\boldsymbol{\xi}\sim P}[V_Q^{\boldsymbol{\xi}}(s')]\right]. \tag{20}$$

Write $V_Q^{\boldsymbol{\xi}}(s') \triangleq \max_{\boldsymbol{a}'\in\mathcal{A}(\boldsymbol{\xi})} Q(s',\boldsymbol{a}')$ and let $\mathcal{P} \triangleq \mathcal{P}_{\rho,K}(\widehat{P})$, suppressing its dependence on $(s, \boldsymbol{a}, s')$ where applicable.

Step 1: every fault-constrained value is non-expansive. This is Lemma 4: $\left|V_Q^{\boldsymbol{\xi}}(s') - V_{Q'}^{\boldsymbol{\xi}}(s')\right| \leq \|Q - Q'\|_\infty$ for every $\boldsymbol{\xi}$ and every s', because a maximum over a fixed non-empty set is 1-Lipschitz in the function being maximized. The bound is uniform in $\boldsymbol{\xi}$, which is what the next step needs. Note that the restricted action set $\mathcal{A}(\boldsymbol{\xi})$ is identical for both Q and Q', because the adversary's choice cannot depend on the value function; hence, no extra term arises from the constraint.

Step 2: the worst-case expectation is non-expansive. The key point is that $\mathcal{P}$ is fixed independently of Q. For each s', let $F_{s'}(Q) \triangleq \inf_{P\in\mathcal{P}} \mathbb{E}_{\boldsymbol{\xi}\sim P}[V_Q^{\boldsymbol{\xi}}(s')]$. For any $\varepsilon > 0$ choose $P_\varepsilon \in \mathcal{P}$ with $\mathbb{E}_{P_\varepsilon}[V_{Q'}^{\boldsymbol{\xi}}(s')] \leq F_{s'}(Q') + \varepsilon$. Then

$$F_{s'}(Q) - F_{s'}(Q') \;\leq\; \mathbb{E}_{P_\varepsilon}[V_Q^{\boldsymbol{\xi}}(s')] - \mathbb{E}_{P_\varepsilon}[V_{Q'}^{\boldsymbol{\xi}}(s')] + \varepsilon \;\leq\; \|Q - Q'\|_\infty + \varepsilon,$$

using the definition of the infimum for the first inequality and Step 1 for the last. Letting $\varepsilon \downarrow 0$ and swapping Q, Q' gives $|F_{s'}(Q) - F_{s'}(Q')| \leq \|Q - Q'\|_\infty$ uniformly in s'.

Step 3: assemble. For every $(s, \boldsymbol{a})$, the reward term cancels, so

$$\left|(\mathcal{T}_{\mathrm{DRO}}Q)(s,\boldsymbol{a}) - (\mathcal{T}_{\mathrm{DRO}}Q')(s,\boldsymbol{a})\right| \leq \gamma \mathbb{E}_{s'\sim P_0}\left[|F_{s'}(Q) - F_{s'}(Q')|\right] \leq \gamma \|Q - Q'\|_\infty.$$

Taking the supremum over $(s, \boldsymbol{a})$ yields $\|\mathcal{T}_{\mathrm{DRO}}Q - \mathcal{T}_{\mathrm{DRO}}Q'\|_\infty \leq \gamma\|Q - Q'\|_\infty$. Since $\gamma < 1$ and the space of bounded action-value functions with $\|\cdot\|_\infty$ is complete, Banach's fixed-point theorem gives a unique fixed point $Q^\star_{\mathrm{rob}}$ and $\|Q^{(k)} - Q^\star_{\mathrm{rob}}\|_\infty \leq \gamma^k \|Q^{(0)} - Q^\star_{\mathrm{rob}}\|_\infty$. □

Scope. The argument requires only that $\mathcal{P}_{\rho,K}$ be a fixed, non-empty set of probability measures. It also applies to KL balls, total-variation balls, and other rectangular ambiguity sets, recovering the classical robust-MDP contraction of Iyengar (2005). Theorem 8 is therefore not specific to Definition 1.

A.5 Represented Iterates Preserve IGM

Proposition 9 (What monotonic mixing preserves)**.** *Every represented iterate remains in the monotonic value-function class, so decentralized per-agent* $\arg\max$ *remains consistent with the centralized greedy action. This architectural consistency does not imply that the unrestricted robust fixed point is representable by the mixer.*

Let $\mathcal{Q}_{\text{mono}}$ be the representable monotonic value-function class and let Π_{mono} denote projection into that class in the training loss. Write $\widehat{\mathcal{T}} = \Pi_{\text{mono}} \circ \mathcal{T}_{\text{DRO}}$.

(1) By construction $\widehat{\mathcal{T}} = \Pi_{\text{mono}} \circ \mathcal{T}_{\text{DRO}}$ outputs an element of $\mathcal{Q}_{\text{mono}}$ at every iteration, since Π_{mono} maps into $\mathcal{Q}_{\text{mono}}$ by definition. For any $Q_{tot} \in \mathcal{Q}_{\text{mono}}$ with mixing function f satisfying $\partial f / \partial Q_i \geq 0$, the Individual-Global-Max property holds: if $a_i^\dagger \in \arg\max_{a_i} Q_i(\tau_i, a_i)$ for each i, then for any joint $\boldsymbol{a}$ we have $Q_i(\tau_i, a_i^\dagger) \geq Q_i(\tau_i, a_i)$ componentwise, and a function non-decreasing in each argument preserves componentwise inequalities, so $f(\boldsymbol{Q}(\boldsymbol{a}^\dagger)) \geq f(\boldsymbol{Q}(\boldsymbol{a}))$. Hence $\boldsymbol{a}^\dagger \in \arg\max_{\boldsymbol{a}} Q_{tot}(s, \boldsymbol{a})$, and decentralized per-agent $\arg\max$ reproduces the centralized greedy action.

(2) Theorem 8 concerns $\mathcal{T}_{\text{DRO}}$ acting on *unrestricted* bounded action-value functions, and its fixed point $Q^\star_{\text{rob}}$ is characterized by $Q^\star_{\text{rob}} = \mathcal{T}_{\text{DRO}} Q^\star_{\text{rob}}$ with no reference to $\mathcal{Q}_{\text{mono}}$. The standard two-agent matrix-game separations show that monotonic mixing cannot represent every joint action-value function: taking the discount to zero reduces the fixed point to the reward function, and an arbitrary payoff matrix need not be monotonically factorizable. Thus $Q^\star_{\text{rob}}$ need not be representable, and no equality between a fixed point of the projected training operator, if one exists, and $Q^\star_{\text{rob}}$ follows. The proposition establishes consistency of decentralized execution at represented iterates, not optimality for the unrestricted robust objective. □

A.6 Proof of Lemma 1 (Submodularity)

We first isolate the one-dimensional fact that drives everything, then assemble the set-function statement.

A.6.1 A one-dimensional lemma on convex functions

Lemma 10 (Increments of a convex function slide upward)**.** *Let* $\sigma : \mathbb{R} \to \mathbb{R}$ *be convex, let* $d \geq 0$*, and let* $a' \leq a$*. Then*

$$\sigma(a) - \sigma(a - d) \;\geq\; \sigma(a') - \sigma(a' - d). \tag{21}$$

In words: the amount a convex function rises across a window of fixed width d *is non-decreasing as the window slides to the right.*

Proof. Write $h \triangleq a - a' \geq 0$. If $h = 0$ the two sides are equal, and if $d = 0$ both sides are 0; so assume $h + d > 0$. Rearranging, (21) is equivalent to

$$\sigma(a) + \sigma(a - h - d) \;\geq\; \sigma(a - h) + \sigma(a - d). \tag{22}$$

Set the two outer points $u \triangleq a$ and $w \triangleq a - h - d$, so that $w \leq a - h \leq u$ and $w \leq a - d \leq u$. Both middle points are convex combinations of u and w:

$$a - h = \lambda_1 u + (1 - \lambda_1) w \;\text{ with }\; \lambda_1 = \frac{d}{h + d}, \qquad a - d = \lambda_2 u + (1 - \lambda_2) w \;\text{ with }\; \lambda_2 = \frac{h}{h + d}.$$

(To verify the first: $\lambda_1 u + (1 - \lambda_1) w = u - (1 - \lambda_1)(u - w) = a - \frac{h}{h+d}(h + d) = a - h$, and similarly for the second.) Note $\lambda_1, \lambda_2 \in [0, 1]$ and, crucially, $\lambda_1 + \lambda_2 = 1$. Applying convexity to each,

$$\sigma(a - h) \leq \lambda_1 \sigma(u) + (1 - \lambda_1)\sigma(w), \qquad \sigma(a - d) \leq \lambda_2 \sigma(u) + (1 - \lambda_2)\sigma(w).$$

Adding these and using $\lambda_1 + \lambda_2 = 1$,

$$\sigma(a-h) + \sigma(a-d) \;\leq\; (\lambda_1 + \lambda_2)\sigma(u) + \big(2 - \lambda_1 - \lambda_2\big)\sigma(w) \;=\; \sigma(u) + \sigma(w) \;=\; \sigma(a) + \sigma(a - h - d),$$

which is (22). □

Lemma 10 requires no differentiability, so it covers ReLU and other non-smooth activations without a separate argument. For twice-differentiable σ it is the familiar statement $\sigma'' \geq 0$; we use the increment form because it is what the proof actually needs.

A.6.2 SET-UP AND NOTATION

Fix a next state s' and suppress it. Let

$$q_i \triangleq \max_a Q_i(\tau_i', a), \qquad \tilde{q}_i \triangleq Q_i(\tau_i', \texttt{noop}), \qquad \delta_i \triangleq q_i - \tilde{q}_i \geq 0,$$

the last inequality holding because $\tilde{q}_i$ is the value of one particular action while q_i is the maximum over all of them. For $S \subseteq \mathcal{N}$ let $\boldsymbol{q}^S$ denote the vector with $(\boldsymbol{q}^S)_i = \tilde{q}_i$ for $i \in S$ and q_i otherwise; note $\boldsymbol{q}^\emptyset = \boldsymbol{q}$ and that $S \subseteq T$ implies $\boldsymbol{q}^T \leq \boldsymbol{q}^S$ componentwise. The mixer is

$$f(\boldsymbol{q}) = \sum_{e=1}^{E} w_2^e \, \sigma\big(z_e(\boldsymbol{q})\big) + b_2, \qquad z_e(\boldsymbol{q}) \triangleq \sum_{i=1}^{N} W_1^{ie} q_i + b_1^e,$$

with $W_1^{ie} \geq 0$, $w_2^e \geq 0$, and σ non-decreasing and convex. Finally $g(S) \triangleq f(\boldsymbol{q}) - f(\boldsymbol{q}^S)$.

A.6.3 NORMALIZATION AND MONOTONICITY

$g(\emptyset) = f(\boldsymbol{q}) - f(\boldsymbol{q}) = 0.$

For monotonicity, let $S \subseteq T$. Then $\boldsymbol{q}^T \leq \boldsymbol{q}^S$ componentwise, so for each e, $z_e(\boldsymbol{q}^T) \leq z_e(\boldsymbol{q}^S)$ because z_e has non-negative coefficients W_1^{ie}. Since σ is non-decreasing, $\sigma(z_e(\boldsymbol{q}^T)) \leq \sigma(z_e(\boldsymbol{q}^S))$; multiplying by $w_2^e \geq 0$ and summing over e gives $f(\boldsymbol{q}^T) \leq f(\boldsymbol{q}^S)$, hence $g(T) \geq g(S)$. So g is non-decreasing, and in particular $g \geq 0$.

A.6.4 SUBMODULARITY

We use the diminishing-returns characterization: g is submodular if and only if for all $S \subseteq T \subseteq \mathcal{N}$ and all $j \notin T$,

$$g(S \cup \{j\}) - g(S) \;\geq\; g(T \cup \{j\}) - g(T). \tag{23}$$

Expanding g, the left side is $f(\boldsymbol{q}^S) - f(\boldsymbol{q}^{S\cup\{j\}})$ and the right side is $f(\boldsymbol{q}^T) - f(\boldsymbol{q}^{T\cup\{j\}})$ (the common term $f(\boldsymbol{q})$ cancels on each side). So (23) is equivalent to

$$f(\boldsymbol{x}) - f(\boldsymbol{x} - \delta_j \boldsymbol{e}_j) \;\geq\; f(\boldsymbol{y}) - f(\boldsymbol{y} - \delta_j \boldsymbol{e}_j), \qquad \boldsymbol{x} \triangleq \boldsymbol{q}^S, \;\; \boldsymbol{y} \triangleq \boldsymbol{q}^T, \tag{24}$$

where we used that $j \notin T \supseteq S$, so passing from S to $S \cup \{j\}$ (resp. T to $T \cup \{j\}$) lowers exactly coordinate j, from q_j to $\tilde{q}_j$, i.e. by δ_j. Note $\boldsymbol{x} \geq \boldsymbol{y}$ componentwise and $x_j = y_j = q_j$.

Now expand both sides of (24) through the mixer. Put

$$a_e \triangleq z_e(\boldsymbol{x}), \qquad a_e' \triangleq z_e(\boldsymbol{y}), \qquad d_e \triangleq W_1^{je} \, \delta_j.$$

Three observations:

1. $a_e \geq a_e'$ for every e, because $\boldsymbol{x} \geq \boldsymbol{y}$ and z_e has non-negative coefficients.
2. $d_e \geq 0$, because $W_1^{je} \geq 0$ and $\delta_j \geq 0$.
3. Lowering coordinate j by δ_j lowers the pre-activation z_e by exactly d_e: $z_e(\boldsymbol{x} - \delta_j \boldsymbol{e}_j) = a_e - d_e$, and likewise $z_e(\boldsymbol{y} - \delta_j \boldsymbol{e}_j) = a_e' - d_e$. The same d_e appears on both sides, which is what makes Lemma 10 applicable.

Therefore

$$f(\boldsymbol{x}) - f(\boldsymbol{x} - \delta_j \boldsymbol{e}_j) = \sum_e w_2^e \big[\sigma(a_e) - \sigma(a_e - d_e)\big],$$

$$f(\boldsymbol{y}) - f(\boldsymbol{y} - \delta_j \boldsymbol{e}_j) = \sum_e w_2^e \big[\sigma(a_e') - \sigma(a_e' - d_e)\big],$$

the constants b_2 canceling. By Lemma 10 applied for each e with $a = a_e$, $a' = a_e'$, $d = d_e$,

$$\sigma(a_e) - \sigma(a_e - d_e) \;\geq\; \sigma(a_e') - \sigma(a_e' - d_e).$$

Multiplying each of these by $w_2^e \geq 0$ preserves the inequality, and summing over e yields (24). Hence g is submodular.

A.6.5 THE GREEDY GUARANTEE

g is normalized, monotone non-decreasing and submodular. Apply the theorem of Nemhauser et al. (1978) separately to each prefix length $k \in \{1, \dots, K\}$. The first k steps of the chain are exactly the greedy algorithm for the constraint $|S| \le k$, hence

$$g(S_k) \;\ge\; \Big(1 - \big(1 - \tfrac{1}{k}\big)^k\Big) \max_{|S|\le k} g(S) \;\ge\; \big(1 - e^{-1}\big) \max_{|S|\le k} g(S).$$

The second inequality uses $(1 - 1/k)^k \le e^{-1}$. Constructing all prefixes through S_K evaluates at most N marginal gains in each of K rounds, for $O(NK)$ mixer evaluations. □

Why convexity rather than concavity. Submodularity requires the marginal damage of a fault to decrease after other coordinates have been lowered. Through the mixer, this means that $\sigma(a) - \sigma(a-d)$ increases with a, which follows from convexity. A concave activation reverses the inequality, makes g supermodular, and removes the greedy approximation guarantee. Standard QMIX uses ELU, which is convex, as are ReLU and softplus.

A.7 WHERE THE HYPOTHESES OF LEMMA 1 COME FROM, AND HOW THEY ARE LOST

Standard QMIX satisfies each hypothesis, but common variants need not. Greedy still returns a chain when a hypothesis fails, without retaining the $(1 - e^{-1})$ bound.

What g describes. The damage function g describes the learner's factored estimate of successor value. Physical damage can instead couple faults through spatial configuration: two faults in one corridor interact differently from faults in separate aisles. Lemma 1 makes the backup adversary tractable but does not impose submodularity on physical outcomes; Appendix C.9 evaluates those outcomes directly.

Non-negative mixing weights. The hypernetworks emit W_1 and $\boldsymbol{w}_2$ through an absolute value, so non-negativity holds by construction. It is the same constraint QMIX already imposes for IGM, which is why the sparse adversary costs nothing extra here.

Weights independent of $\boldsymbol{q}$ (hypothesis (ii)). The hypernetworks are conditioned on the state s, not on the utilities. Were the weights a function of $\boldsymbol{q}$, the two sides of the diminishing-returns inequality would no longer share a common increment and the proof of Lemma 10 would fail.

Convex activation. ReLU and softplus are convex; ELU is convex for $\alpha \le 1$, which is the default. At $\alpha = 2$ it is not, and the guarantee lapses (see *Why convexity and not concavity* above for the direction of the reversal).

Non-negative drops (hypothesis (i)). $\delta_i = q_i - \tilde{q}_i \ge 0$ is free under the standard backup, since $q_i = \max_a Q_i(\tau_i', a)$ maximizes over an action set containing `noop`. It is not free under double Q-learning, where $q_i = Q_i^{\text{targ}}\big(\arg\max_a Q_i^{\text{online}}\big)$ bears no such relation to $\tilde{q}_i$. Measured over 4800 instances, the double-Q backup produced a negative δ_j on 1651 of them (34.4%, worst -59.93), against 0 for the standard backup. Accordingly, we restrict double-Q experiments to $K = 1$; for $K > 1$, negative drops invalidate the greedy approximation guarantee.

A.8 VDN IS EXACTLY SOLVABLE

Corollary 11 (VDN is exactly solvable)**.** *For additive factorization ($\sigma = \mathrm{id}$, $W_1 = I$, $\boldsymbol{w}_2 = \mathbf{1}$) the damage $g(S) = \sum_{i\in S}(q_i - \tilde{q}_i)$ is modular, so greedy is exact and the worst K-sparse pattern is obtained by taking the K largest per-agent drops in $O(N \log N)$.*

For additive factorization, $\sigma = \mathrm{id}$, $E = N$, $W_1 = I$ and $\boldsymbol{w}_2 = \mathbf{1}$, so $f(\boldsymbol{q}) = \sum_i q_i + b_2$. Then

$$g(S) = f(\boldsymbol{q}) - f(\boldsymbol{q}^S) = \sum_i q_i - \Big(\sum_{i\notin S} q_i + \sum_{i\in S} \tilde{q}_i\Big) = \sum_{i\in S}(q_i - \tilde{q}_i) = \sum_{i\in S} \delta_i,$$

which is a modular (additive) set function. For a modular objective under a cardinality constraint, the marginal gain of adding i is δ_i regardless of the current set, so greedy simply selects the K largest δ_i,

and any set of the K largest δ_i is optimal: for any $|S| \le K$, $g(S) = \sum_{i \in S} \delta_i \le \sum_{i \in \text{top-}K} \delta_i$ since all $\delta_i \ge 0$. Selecting the K largest of N numbers costs $O(N \log N)$ by sorting (or $O(N)$ by selection). The identity $\sigma = \mathrm{id}$ is both convex and concave, consistent with g being both sub- and supermodular here. □

A.9 Proof of Corollary 2 (Greedy Robust-Backup Guarantee)

Aggregation is lossless. For any feasible P on Ξ_K let $q_k = P(\{S : |S| = k\})$. Then $\sum_k k q_k = \sum_S P(S)|S| \le \rho$, and by monotonicity $\sum_S P(S) g(S) = \sum_k q_k \mathbb{E}_P[g(S) \,|\, |S| = k] \le \sum_k q_k D_k^\star$, so $\mathcal{D}^\star \ge$ the full-support optimum. Conversely, given feasible q, place mass q_k on an argmax set of cardinality at most k; its cost is at most k, so the resulting P is feasible and attains $\sum_k q_k D_k^\star$. The two optima coincide.

Transfer. The two LPs share a feasible set and differ only in the objective coefficients, which satisfy $\alpha D_k^\star \le D_k^{\mathrm{gr}} \le D_k^\star$ pointwise by Lemma 1. A maximum over a fixed set is monotone in the coefficients and positively homogeneous, giving the sandwich.

Value form. Conditional on a realized successor s', the sampled target is $r + \gamma(1 - d(s'))(v_{\mathrm{nom}}(s') - \mathcal{D}^\ell_{s'})$. Subtracting the full and greedy forms converts the damage sandwich into the conditional target sandwich. Taking the outer expectation over $s' \sim P_0(\cdot \mid s, \boldsymbol{a})$ gives the operator bound in Corollary 2.

The comparison applies $\mathcal{T}_{\mathrm{greedy}}$ and $\mathcal{T}_{\mathrm{full}}$ to the same Q. It does not compare their fixed points, and the Q-dependent greedy chain does not inherit the contraction argument of Appendix A.4. The constant is $\alpha = \min_{1 \le k \le K} \left(1 - (1 - 1/k)^k\right)$, attained at $k = K$ since $1 - (1 - 1/k)^k$ decreases monotonically toward its limit $1 - e^{-1}$ as k grows. We verify the ratio $\mathcal{D}^{\mathrm{gr}}/\mathcal{D}^\star$ and the aggregation identity over 3,600 random monotone-submodular instances from two generators, with $n \in \{4, 6, 8\}$, $K \in \{1, 2, 3\}$, and $\rho \sim U(0, K)$. The realized ratio never falls below α; the worst case is 0.870, compared with $\alpha = 0.7037$ at $K = 3$, and the aggregation identity holds to 3.55×10^{-15}.

A.10 Proof of Proposition 3 (Collapse along the greedy chain)

Let $d_k \triangleq g(S_k) - g(S_{k-1})$ be the marginal damage at greedy step k. Greedy selection and submodularity imply $d_{k+1} \le d_k$, while monotonicity gives $d_k \ge 0$. Since $v_k = v_0 - g(S_k)$, the increments $v_k - v_{k-1} = -d_k$ are nondecreasing. Thus the points (k, v_k) form a decreasing convex chain, every point lies on its lower hull, and the unconstrained hull minimum is attained at $k = K$. Theorem 7 therefore evaluates the chain at $\min(K, \rho)$, proving part (i). For VDN, Corollary 11 shows that the chain contains a damage-maximizing set at every cardinality, so it is also the cardinality frontier of the full support; the same argument proves part (ii). □

A.11 A Monotone Submodular Damage Whose Budget Does Not Collapse

Proposition 12 (The full set retains a distinct budget)**.** *There exists a monotone submodular damage function on four agents for which* $\mathcal{T}_{\mathrm{full}}$ *is not a function of* $\min(K, \rho)$*. At* $\rho = 3/2$*, its full-set backups differ for* $K = 2$ *and* $K = 3$ *despite equal* $\min(K, \rho)$*.*

Proposition 12 asserts that $\mathcal{T}_{\mathrm{full}}$ need not be a function of $\min(K, \rho)$. The collapse in Proposition 3 needs the cardinality frontier $G_k \triangleq \max_{|S| \le k} g(S)$ to have non-increasing marginals. Greedy delivers this along its own chain by construction, but the frontier of the full set is assembled from the optimal set at each cardinality, and those need not be nested.

Here is an exact witness on $\mathcal{N} = \{0, 1, 2, 3\}$:

$$g(S) \triangleq \begin{cases} |S|/3, & 0 \notin S, \\ 1/3 + |S|/6, & 0 \in S. \end{cases} \tag{25}$$

This formula specifies all 16 set values as rational numbers. For any $j \ne 0$, the marginal gain is $1/3$ before agent 0 is selected and $1/6$ afterward. For agent 0, its marginal gain on a set S not containing it is $1/2 - |S|/6$. All marginals are nonnegative and can only decrease as the conditioning set grows, proving monotonicity and submodularity exactly.

Table 2: QMIX calibration over 10^7 steps and five seeds per cell. Values are alive runs and sustained nominal delivery rate.

	tiny-4ag		small-8ag	
Configuration	alive	delivery	alive	delivery
target interval 200	0/5	0.0	0/5	0.0
target interval 2000	5/5	97.8	5/5	97.4
fewer updates per step	5/5	71.2	5/5	87.6
both	5/5	95.0	5/5	99.2

The resulting cardinality frontier is

cardinality k	0	1	2	3	4
G_k	0	$\frac{1}{2}$	$\frac{2}{3}$	1	1
$G_k - G_{k-1}$	–	$\frac{1}{2}$	$\frac{1}{6}$	$\mathbf{\frac{1}{3}}$	0

with respective maximizers $\{0\}$, any pair, and $\{1, 2, 3\}$ at cardinalities one, two, and three. These maximizers need not be nested, and the frontier marginal increases from $1/6$ to $1/3$.

Taking $v_k = -G_k$ and reading the lower convex hull at $c = \min(K, \rho)$ as in Theorem 7, set $\rho = 3/2$. For $K = 2$, interpolation between $(1, -1/2)$ and $(2, -2/3)$ gives $-7/12$. For $K = 3$, the point $(2, -2/3)$ lies above the segment from $(1, -1/2)$ to $(3, -1)$, which gives $-5/8$ at the same radius. Thus equal $\min(K, \rho) = 3/2$ yields distinct full-set backups, proving Proposition 12.

B Potential-Based Reward Shaping

RWARE emits reward only when a requested shelf reaches a goal. Reward shaping is disabled in the primary results and is not part of KW-DRO. When enabled in the present implementation, we add the per-agent potential

$$\Phi_i(s) = \begin{cases} 1.0 - 0.05 \min_{g \in \mathcal{G}} \|p_i - g\|_1, & \text{if agent } i \text{ carries a shelf,} \\ 0.2 - 0.02 \min_{s_k \in \mathcal{S}_{\text{all}}} \|p_i - s_k\|_1, & \text{otherwise,} \end{cases} \tag{26}$$

and reward

$$R_i' = R_i + w\big(\gamma \Phi_i(s') - \Phi_i(s)\big), \qquad w > 0. \tag{27}$$

Scaling by w preserves the potential-based form, provided the shaping discount equals the learner discount; both are 0.99 in the reported configuration. The shelf set and agent poses are part of the environment state, so Φ_i is a state function. The relevant multi-agent result is preservation of Nash equilibria under per-agent potential-based shaping (Devlin & Kudenko, 2011), extending the single-agent policy-invariance construction of Ng et al. (1999). Experiments identify when shaping is enabled, and delivery counts always use the unshaped environment reward.

C Supplementary Experiments

C.1 Task and Observation Structure

Figure 5 illustrates why a zero score can represent either an untrained policy or a policy that completes several unrewarded stages but not delivery. Figure 6 records the observation mask used in the withdrawn adversarial evaluation discussed in Appendix E.

C.2 Benchmark Calibration

QMIX is sensitive to target-network update frequency on RWARE. At 10^7 steps, changing the target update interval from 200 to 2000 changes sustained delivery from zero to above 97% on both displayed environments.

This calibration establishes that the late-collapse analysis concerns policies that first learn the task, rather than a permanently flat baseline.

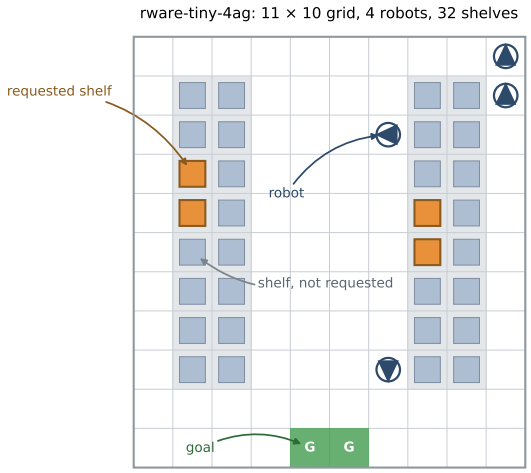

Figure 5: RWARE task anatomy. A delivery requires five stages: navigate to a requested shelf, load it, carry it to a goal, arrive for the only $+1$ reward, then return and unload it. A fixed-seed uniform-random rollout produces reward on just 5 of 160,000 agent-transitions (0.003%). Loaded robots cannot pass beneath standing shelves, making the one-tile corridors susceptible to the queue-level blockage illustrated in Figure 1.

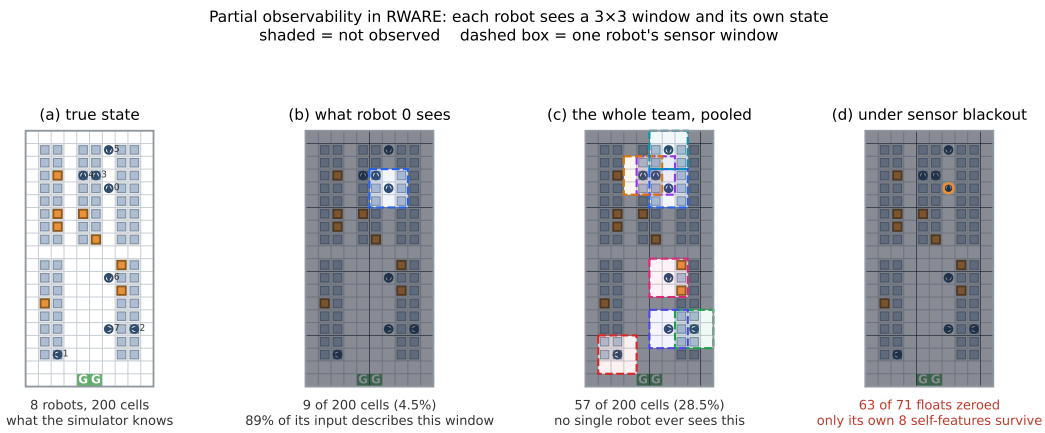

Figure 6: Partial observability on `small-8ag`. Each robot receives its own state and a 3×3 neighborhood. The sensor-blackout evaluation zeroes the neighborhood features while retaining self-features.

C.3 Exploratory Radius Ladder

Figure 4 in the main text shows collapse at low radius and failure to learn at high radius, with $\rho = 0.5$ between them. This exploratory cohort predates deterministic environment seeding; Table 1 provides the seeded comparisons at $\rho \in \{0, 0.5, 1\}$. Figure 7 resolves the same runs by environment, exposing differences hidden by the pooled curve.

C.4 Fault Injection over Time

Figure 8 extends the static observation views into time: the fault set is held fixed after injection, while unaffected robots continue to receive their neighborhood observations and act.

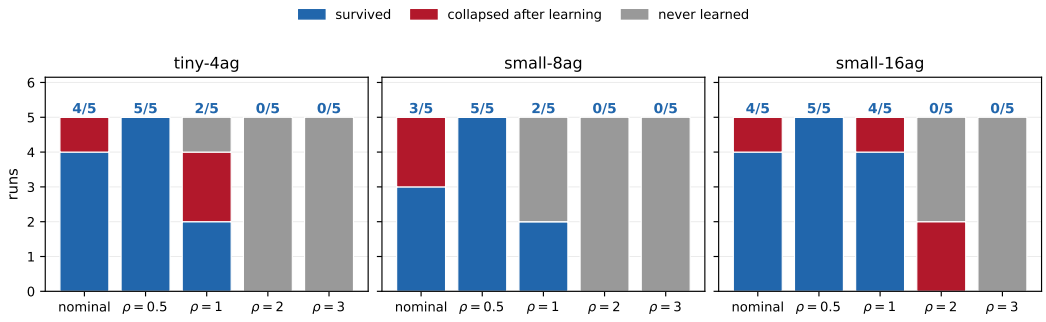

Figure 7: Per-environment outcome composition for the exploratory radius ladder in Figure 4, with five runs per cell. The labels above the bars report alive runs out of five. At $\rho = 1$, all failures on `small-8ag` are failures to learn, whereas `tiny-4ag` and `small-16ag` still include post-learning collapses. Because this cohort predates environment seeding, the figure is descriptive; inferential comparisons use the seeded cohort in Table 1.

Table 3: Seeded 2×2 training factorial on `small-8ag`, five runs per cell and 10^7 steps.

	nominal training		actuator-fault training	
Backup	alive	delivery	alive	delivery
nominal	4/5	53.0	3/5	25.4
KW-DRO, $\rho = .5$	5/5	84.2	5/5	82.2

C.5 Independent Replication of Late Collapse

Before the seeded control cohort, nominal VDN appeared as a reference arm in three independent experimental batches: the K–ρ sweep, the actuator factorial, and the QMIX calibration study. Together these batches contain 15 `small-8ag` runs, of which 10 finish alive and 5 collapse after learning; the corresponding `tiny-4ag` count is 14 alive and 1 collapsed. Because these runs predate deterministic environment seeding, they establish recurrence across independent batches but do not estimate a collapse rate. They are not pooled with the later seeded cohort in Table 1, which contains 10 runs per arm and environment from a single controlled batch.

C.6 Actuator-Training Factorial

We independently vary the Bellman backup and actuator-only perturbations during training, evaluating all policies nominally.

The seeded actuator-training contrast is not resolved at five runs per cell ($p = 0.444$ on `small-8ag`; pooled across two environments, $10/10$ versus $7/10$, $p = 0.211$). The result is underpowered for a backup-by-fault interaction and is reported as a supplementary analysis.

C.7 Illustrative Collapse Diagnostics

Checkpoint rollouts from one collapsed nominal run, one surviving nominal run, and one KW-DRO run do not separate the outcomes by action-value spread, action concentration, agent agreement, or stalled-step frequency. Both surviving and collapsing runs enter low-delivery regions; the observed distinction is whether they recover. Because these diagnostics cover three selected runs, they are illustrative and do not identify a mechanism.

C.8 Reference and Observation-Mask Studies

SEAC learns `small-8ag` on $3/3$ seeds, whereas its independent actor–critic control with sharing weight $\lambda = 0$ remains at zero delivery on $0/3$; a four-rate learning-rate sweep of the independent

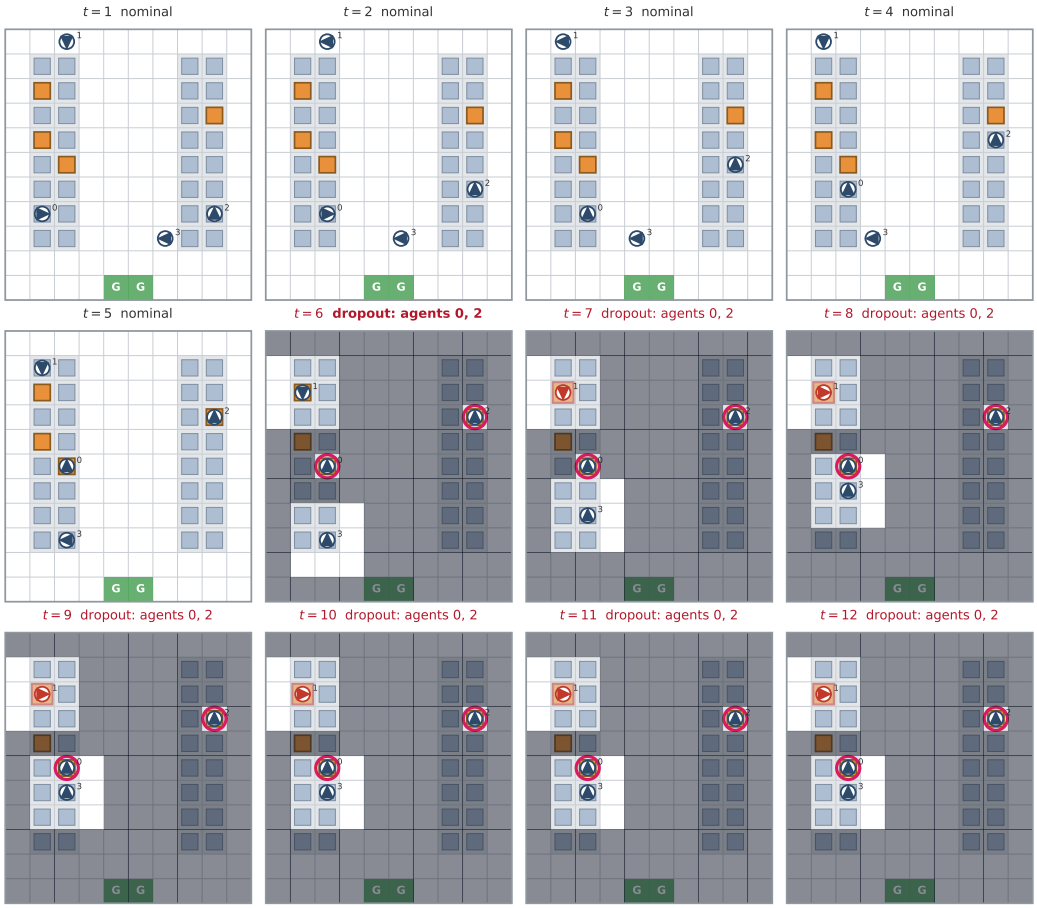

Figure 8: Illustration of actuator dropout and observation masking over one deterministic, scripted rollout on `tiny-4ag`. During steps 1–5, all four robots act normally. From step 6 onward, the fixed set $\{0, 2\}$ ($K = 2$) is forced to `noop`(pink rings), while agents 1 and 3 continue executing the controller. The translucent overlay marks cells unavailable through the surviving neighborhood observations: each failed robot retains only its own 8 self-features, while its remaining 63 observation entries are zeroed. Across the seven displayed faulted states, agents 0 and 2 never change pose; at least one non-failed agent moves in two of the six intervening transitions. This sequence visualizes the perturbation semantics and is not a learned-policy robustness result.

control remains at $0/24$. This reproduces the difficulty reported for independent actor–critic training without making it the premise of KW-DRO.

We evaluate training-time observation dropout in 72 runs: two environments, three algorithms, four rates, and three seeds. No masked cell exceeds its unmasked baseline on final return, peak return, or delivery rate, and a mask probability of 0.05 reduces final return by 75–97%. By contrast, post-training evaluation under the combined sensor-blackout and forced-`noop`condition increases measured return on $16/18$ unmasked runs. These interventions answer different questions and support the withdrawal of the original adversarial-evaluation interpretation.

C.9 Actuator-Only Deployment Evaluation

We evaluate all 120 final checkpoints under no faults, i.i.d. actuator faults, and faults targeted at bottleneck agents. Observation masking is disabled. Faults occur with probability 0.3 over 20 paired episodes of 500 steps per checkpoint. The primary evaluation permits one concurrent fault; a second evaluation uses three, matching the training budget of the KW-DRO arms.

No arm exhibits a significant throughput reduction under either fault condition (paired Wilcoxon tests over 10 seeds, $p \geq 0.38$). With one concurrent fault, the mean change pooled over all checkpoints is $+0.47$ deliveries per 1000 steps. With three concurrent faults, the pooled changes are $+0.39$ under i.i.d. faults and $+0.11$ under targeted faults. Independent evaluations of the two budgets agree to a mean absolute difference of 0.05 deliveries per 1000 steps across the 36 shared cells.

The earlier combined protocol is not suitable for ranking robustness, and a larger sweep isolates the reason. Targeted actuator faults and the combined protocol realize the *same* fault intensity — 1.204 against 1.207 forced stalls per step on `small-8ag`, and 1.204 against 1.204 on `tiny-4ag` — so the pair differs in exactly one factor, the observation mask. Switching the mask on raises throughput in all twelve arm–environment cells:

Table 4: Deliveries per 1000 steps under targeted actuator faults with the observation mask off, then on, at matched realized fault rate (≈ 1.20 stalls/step). Means over 10 seeds.

	nominal	$\rho = 0.5$	$\rho = 1$	0.36%	0.72%	double Q
`small-8ag` off	5.51	6.88	6.15	3.75	11.62	9.94
`small-8ag` on	9.31	12.98	7.99	6.08	16.53	14.51
`tiny-4ag` off	10.04	11.55	1.58	11.38	14.06	14.02
`tiny-4ag` on	13.70	16.64	2.12	17.59	19.69	18.22

The ratios range from $1.30\times$ to $1.89\times$, with the same direction in every cell. At these fleet densities, forced inactivity can relieve corridor contention, and the observation mask further raises measured throughput. The combined protocol therefore confounds fault robustness with a beneficial masking effect and is excluded from method comparisons.

For targeted faults, the nominal `failure_prob` argument does not equal the realized intensity. This path considers only the top-K threat-ranked agents and floors the per-agent probability at $\max(\texttt{failure_prob}, 0.4)$, so settings 0.1 and 0.3 both produce 1.204 stalls per step. We therefore match targeted conditions by the realized rate recorded by the evaluation driver. The i.i.d. path has no such floor and draws over all N agents before applying the budget cap, producing 0.597 and 1.668 stalls per step at the same settings; we do not compare the i.i.d. and targeted conditions directly.

At $\rho = 0.5$, the actuator-only evaluation does not resolve a difference between KW-DRO and its matched 0.36% constant (9.16 versus 8.04 deliveries per 1000 steps, paired Wilcoxon $p = 0.37$, $n = 20$). At $\rho = 1$, KW-DRO performs worse than its matched 0.72% constant (3.91 versus 13.26, $p = 0.002$), reproducing the unfavorable higher-radius regime observed during training.

C.10 Inner-Solver Verification Against a Reference LP

Section 5.1 compares Algorithm 2 with a general-purpose linear-program solver on 2000 random instances, 400 in each of five regimes: unit fault costs with $\rho \leq \max_m c_m$; unit costs with a slack radius; the duplicated costs $\{0, 1, 1, 2, 2, 3\}$ of an unfiltered $K = 3$ enumeration; continuous

heterogeneous costs; and identical candidate values. The regimes are not interchangeable. Duplicated *and* unequal costs are the only ones that exercise the tie rule in the hull construction, which continuous costs never reach because continuous draws never repeat; and the identical-value regime produces a degenerate hull — a single horizontal segment on which every feasible distribution is optimal — so an endpoint artifact there would not surface as infeasibility. The duplicated vector is an unfiltered candidate list. The greedy candidate construction retains one pattern per cardinality and therefore yields costs $\{0, 1, 2, 3\}$ at $K = 3$; duplicated unequal costs test the generic hull solver's tie handling.

The maximum absolute deviation is sample-dependent: across random seeds for the 2000-instance validation set, it ranges from 1.7×10^{-7} to 4.4×10^{-7}. We therefore report 4.5×10^{-7} as an empirical upper bound. Ratio-greedy allocation is suboptimal in the two unequal-cost regimes and exact in the three others, which is what the equal-cost reduction to fractional knapsack predicts.

Figure 2 in Section 5.1 complements this synthetic LP verification with a probe of candidate values and active atom occupancy produced by a trained policy.

C.11 Greedy versus exhaustive fault selection on trained QMIX

Corollary 2 bounds the greedy chain's robust damage against the exact full-support optimum by a factor $\alpha = \min_{1 \le k \le K}(1 - (1 - 1/k)^k)$, 0.7037 at $K = 3$, and concerns QMIX's non-linear monotone mixer specifically. The primary KW-DRO learning comparisons use VDN, for which greedy is exact by construction (Appendix A.8) and the corollary has $\alpha = 1$. Here we evaluate the corollary directly on trained QMIX hypernetworks by comparing greedy selection with exhaustive enumeration of the full K-sparse support.

At $N = 8$, $K = 3$ the admissible support $\{S : |S| \le 3\}$ has $\binom{8}{0} + \binom{8}{1} + \binom{8}{2} + \binom{8}{3} = 93$ subsets, small enough to enumerate exactly by batching the subset dimension in one mixer evaluation and compare against the greedy chain's $O(NK) = 24$ mixer evaluations. We measure three conditions, 512 successors each:

- **trained**: three QMIX checkpoints (seeds 1–3) that reached sustained 97–100% delivery on `small-8ag`, each evaluated on 512 successor states sampled from its greedy rollout.
- **random**: a freshly initialized QMIX of the same shape, rolled the same way – the architectural control for whether training changes the ratio at all.
- **nominal**: the same trained checkpoint (seed 1), but with no rollout: 512 independent environment resets stand in for successor states, isolating the successor distribution from the checkpoint identity.

Table 5: Greedy versus exhaustive fault selection at $N = 8$, $K = 3$. The trained condition pools three seeds with 512 successors each; the random and nominal conditions contain 512 successors each. Approximation ratio is $(v_{\text{nom}} - v_{\text{greedy},k})/(v_{\text{nom}} - v_{\text{exhaustive},k})$, pooled over $k \in \{1, 2, 3\}$; target gap is $|\text{hull(greedy)} - \text{hull(exhaustive)}|$ after both pass through the hull solver, maximized over $\rho \in \{0.5, 1, 2, 3\}$.

Condition	ratio min	ratio mean	target gap max	top-set agreement	mixer evals (gr./exh.)
trained (3 seeds)	1.0000	1.0000	0	1.0000	24 / 93
random	1.0000	1.0000	0	1.0000	24 / 93
nominal	1.0000	1.0000	0	1.0000	24 / 93

Greedy selects the same top-K set as exhaustive enumeration for every successor, condition, and cardinality $k \le 3$. Consequently, the approximation ratio is 1.0 and the post-hull target gap is 0 for every tested $\rho \in \{0.5, 1, 2, 3\}$; these are equivalent consequences of set agreement rather than independent measurements. A zero gap is the lower endpoint of the interval $[0, (1 - \alpha)\mathcal{D}^\star(\rho)]$ allowed by Corollary 2. No run contains a negative $\delta_i = q_i - \tilde{q}_i$: the trained condition has $0/12{,}288$ negative agent drops across three seeds, and each remaining condition has $0/4096$.

Near-modularity at the tested scale. Define the pairwise non-modularity $\eta(i, j) = |g(\{i, j\}) - g(\{i\}) - g(\{j\})|$, where $g(S) = v_{\text{nom}} - v_S$ is the damage of stalling exactly the agents in S. When

Table 6: Non-modularity at the native (unamplified) fault-drop scale, i.e. the actual perturbation an actuator fault produces. Ratio is $\max \eta / \text{mean } g(\{i\})$. The `trained` row pools three seeds; the parenthetical value reports the largest per-seed ratio.

Condition	mean $g(\{i\})$	max $g(\{i\})$	max $\eta(i,j)$	max η / mean $g(\{i\})$
trained (3 seeds)	9.96×10^{-3}	1.37×10^{-1}	3.53×10^{-5}	0.35% (worst seed 0.41%)
random	2.44×10^{-4}	1.65×10^{-2}	0	0%
nominal	1.51×10^{-2}	1.48×10^{-1}	4.96×10^{-5}	0.33%

$\max_{i,j} \eta(i,j)$ is negligible relative to singleton damage, the selection problem is numerically close to sorting even for a nonlinear mixer.

In every condition, the largest measured non-modularity is under half a percent of the mean singleton damage and is numerically negligible at float32 precision. The `random` condition measures exactly zero and also has the smallest perturbations, with mean singleton damage $41\times$ below the trained mixers. The trained and nominal conditions have comparable damage scales and support the relevant conclusion: at the native fault scale, the selection problem is numerically modular.

Sensitivity to the fault-drop scale. To locate where non-modularity matters, we scale the fault drop $q - \tilde{q}$ by factors up to $500\times$ its measured value and compare greedy with exhaustive selection at each scale. This isolates the mixer rather than simulating RWARE faults: factors of $50\times$ or more produce utility drops outside the native actuator-fault range. Table 7 reports the trained checkpoint for seed 1; per-seed results are available in the anonymous repository.

Table 7: Sensitivity probe on the trained QMIX mixer (seed 1), *not* a claim about RWARE's actual perturbation scale. The drop $q - \tilde{q}$ is scaled by the factor shown; ratio is $(v_{\text{nom}} - v_{\text{greedy},K})/(v_{\text{nom}} - v_{\text{exhaustive},K})$ at $K = 3$.

Scale	mean $g(\{i\})$	max $\eta(i,j)$	ratio (min)	ratio (mean)
$\times 1$	1.27×10^{-2}	4.77×10^{-6}	1.000000	1.000000
$\times 10$	1.27×10^{-1}	5.04×10^{-4}	1.000000	1.000000
$\times 50$	6.28×10^{-1}	2.65×10^{0}	0.999900	0.999998
$\times 100$	1.20×10^{0}	4.96×10^{0}	0.999285	0.999990
$\times 500$	3.86×10^{0}	7.55×10^{0}	0.999815	0.999999

Greedy first departs from exactness near $\times 50$ for this checkpoint, when non-modularity becomes comparable to the damage scale. The precise threshold varies across seeds, while at the native actuator-fault scale ($\times 1$) non-modularity remains far below it.

At RWARE's native perturbation scale, QMIX's selection problem is numerically indistinguishable from VDN's additive case: greedy uses 24 mixer evaluations rather than 93 exhaustive evaluations and returns the exact selected sets. Proposition 12 establishes that non-trivial approximation gaps remain possible for monotone submodular damage in general.

Cost scaling in N and K. The preceding comparison is about *quality*; this one is about cost. RWARE offers at most 16 agents, so we time both paths on synthetic QMIX mixers with random hypernetwork weights over $N \in \{8, 16, 32, 64\}$ and $K \in \{1, 2, 3\}$, batch 512. The two evaluation counts are exact and match their closed forms in every cell: greedy performs NK mixer evaluations, exhaustive enumeration performs $\sum_{k \le K} \binom{N}{k}$.

Where both paths are computable, greedy is 2.5 to $4.2\times$ faster at $K \geq 2$, and the gap widens with K. At $K = 1$ the two coincide to within timing noise, as they must: a budget of one admits $N + 1$ patterns and greedy enumerates the same set, so the chain saves nothing and the guarantee is vacuous rather than merely tight. The separation is therefore a statement about $K \geq 2$, which is the regime the submodularity theorem is written for.

At $K = 3$, exhaustive enumeration grows from 93 candidate patterns at $N = 8$ to 43,745 at $N = 64$, compared with 24 and 192 greedy evaluations. Greedy runtime grows from 0.007s to 0.263s over

Table 8: Mixer evaluations and wall-clock (minimum of 15 repeats, single-threaded CPU) for greedy chain construction against exhaustive enumeration of the K-sparse set. Exhaustive is not run above 2×10^5 candidate–successor pairs.

		mixer evaluations		seconds	
N	K	greedy	exhaustive	greedy	exhaustive
8	1	8	9	0.003	0.003
8	2	16	37	0.005	0.013
8	3	24	93	0.007	0.029
16	2	32	137	0.019	0.065
16	3	48	697	0.025	—
32	3	96	5,489	0.075	—
64	3	192	43,745	0.263	—

the same range. We omit exhaustive timings once the number of candidate–successor pairs exceeds 2×10^5; the accompanying script generates the remaining configurations.

C.12 Training KW-DRO with QMIX

The preceding subsection evaluates greedy selection on trained QMIX mixers; this one trains with it. The main learning comparisons use VDN, so we ran the QMIX arm the theory is written for: 30 runs at 10^7 environment steps, $\rho = 0.5$ and $K = 3$ against a nominal QMIX backup, 10 seeds per arm on `small-8ag` and 5 on `tiny-4ag`. Every setting matches the seeded VDN control cohort except the algorithm and the target-update interval, which is 2000 rather than 200 because QMIX does not learn at 200 (Table 2). All 30 runs completed; none was excluded.

Table 9: KW-DRO with QMIX at 10^7 steps. Alive follows the same sustained-delivery endpoint as Table 1; sustained rate is the mean over seeds of each run's final five evaluations.

Environment	Backup	Alive	Sustained delivery (%)
`small-8ag`	nominal	10/10	98.7
`small-8ag`	KW-DRO $\rho = 0.5$, $K = 3$	10/10	99.6
`tiny-4ag`	nominal	5/5	92.6
`tiny-4ag`	KW-DRO $\rho = 0.5$, $K = 3$	5/5	99.2

Every KW-DRO-QMIX run reaches and holds a working policy, showing that the greedy chain and hull solver can be used throughout training at this scale. The nominal arm reaches the same survival endpoint, and no QMIX run collapses under either backup. The QMIX and VDN cohorts also use different target-update intervals (2000 and 200; Table 2), precluding an architectural comparison of their stability.

The survival endpoint is saturated at this budget — a single delivery in 500 steps scores a run as delivering, and both arms sit near 99% — so we also evaluate the same checkpoints on deliveries per 1000 steps.

Table 10: Deliveries per 1000 steps for the QMIX arms, means over seeds. Paired Wilcoxon by seed within each environment; $n = 10$ on `small-8ag` and $n = 5$ on `tiny-4ag`.

Condition	Environment	nominal	KW-DRO	p
no faults	`small-8ag`	29.0	35.7	0.084
no faults	`tiny-4ag`	17.3	41.3	0.313
actuator $p = 0.3$	`small-8ag`	27.5	33.1	0.084
actuator $p = 0.3$	`tiny-4ag`	15.6	36.1	0.188

The throughput point estimates favor KW-DRO in all four comparisons, by approximately 19% on `small-8ag` and by a factor of 2.4 on `tiny-4ag`. None of the paired per-environment tests is

significant at the 0.05 level, and the `tiny-4ag` comparisons contain only five seeds. These estimates are descriptive and do not resolve either improvement or equivalence.

The QMIX cohort establishes computational applicability during training and reaches the same saturated survival endpoint under both backups. The exhaustive comparisons in Appendix C.11 separately establish agreement between greedy and exhaustive candidate selection on the sampled states.

C.13 Experimental Scope

The experiments verify the finite-support inner optimization, document late VDN collapse, and show stabilization under moderate pessimism together with a two-sided radius window. For QMIX, they establish trainability of the robust backup and agreement between greedy and exhaustive selection on the sampled states. The saturated survival endpoint and unresolved throughput comparisons do not show an additional performance benefit.

The actuator-only evaluation is an external stress test rather than an in-ball test of the Wasserstein model: its wrapper law is not constrained by ρ, and targeted faults are selected before the realized successor rather than by the conditional inner optimizer. The current evidence neither identifies the collapse mechanism nor separates mixer architecture from the different target-update intervals used by the VDN and QMIX cohorts.

D Expanded Comparison with Distributionally Robust MARL

The main text highlights DrIGM because it is the closest prior connection between distributional robustness and value factorization in cooperative MARL (Qu et al., 2026). The methods nevertheless robustify different random objects and use factorization for different technical purposes. DrIGM places history–action rectangular ambiguity sets around transition kernels, including contamination and total-variation constructions, and couples individual worst-case models into a joint model whose robust backup preserves alignment between individual and joint greedy actions. KW-DRO retains the nominal stochastic environment kernel rather than ambiguity-robustifying it and instead models a conditional fault channel, indexed by $(s, \boldsymbol{a}, s')$, whose outcomes are actuator-availability patterns. Under the Hamming metric and a no-fault Dirac nominal distribution, the Wasserstein radius bounds expected Hamming distance—equivalently, the expected number of failed actuators—whereas $\mathrm{supp}(P) \subseteq \Xi_K$ caps every realized pattern at K faults.

This distinction changes the inner optimization and the scope of the guarantees. DrIGM uses robust factorization to preserve decentralized greedy-action consistency under transition uncertainty. KW-DRO uses an existing monotone factorization to rank sparse fault patterns: under the stated QMIX mixer conditions, the induced damage is monotone submodular and greedy candidate construction is approximation-guaranteed; for additive VDN, selection is exact. Once candidates are supplied, lower-hull probability allocation is exact, so this exactness does not extend to the full combinatorial QMIX search. Thus, DrIGM supplies the closest robust-factorization precedent, whereas KW-DRO contributes a structured solver for budgeted actuator uncertainty.

The group-DRO warehouse formulation of Liu et al. (2025) studies a different operational scale. It treats package-induction regimes as empirical groups and learns a predictor for the worst operating distribution, addressing exogenous demand shift in facility-level dispatch. KW-DRO treats robot actuators as the uncertainty units and searches over subsets that may fail concurrently during execution. The learned group predictor therefore scales over candidate operating distributions, while the submodular construction scales over robot subsets and carries an approximation guarantee tied to the monotone mixer.

Table 11: Technical scope of the closest distributionally robust MARL settings.

Method	Uncertainty object	Role of factorization	Inner computation
DrIGM (Qu et al., 2026)	history–action rectangular transition kernel	robust greedy-action consistency	joint worst-case model + DrIGM-consistent backup
Warehouse group DRO (Liu et al., 2025)	package-induction group	facility-level policy under demand shift	learned worst-group prediction
KW-DRO (ours)	K-sparse actuator-fault channel	structured fault damage	guaranteed greedy search + exact conditional hull allocation

E EVALUATION AUDIT

This appendix documents the evaluation issues summarized in Section C.13. They concern benchmark semantics and metric design and therefore apply beyond a particular implementation.

Two defects compounded in the withdrawn evaluation.

The deadlock detector measured wall-pushing. The harness classified an agent as blocked by comparing its position before and after a step for agents whose recorded action was `FORWARD`. But the simulator overwrites that field to `noop`for exactly those agents whose forward move was refused, so after the step the only agents still marked `FORWARD` are those that moved successfully. The agents counted as "blocked" were instead those at a grid boundary facing outward, whose forward target clamps to their current cell. The metric therefore reported how often the policy pushed into a wall. Driving the harness with constant actions and no learning of any kind reproduced the previously reported headline numbers exactly: a constant-`FORWARD` policy scored a 90% deadlock rate and every other constant action scored 0%. A reported improvement from 90% to 0% is thus fully explained by which of five constant actions an untrained network's $\arg\max$ happened to select.

Returns were zero everywhere, which the metric rewarded. Every configuration, including a 3,000-episode run, delivered no shelves at all. Under a deadlock-only criterion, a policy that never moves is optimal (attaining a 0% deadlock rate), which constitutes the freezing-robot failure mode that robust training is supposed to avoid. Deadlock rate is therefore not a self-sufficient robustness metric on this benchmark, and we do not treat it as one below.

RWARE reports every episode ending as a termination, and there are no terminations. The warehouse has no goal state: nothing an agent can do ends an episode. `Warehouse.step` raises `done` only when the step budget or the inactivity budget runs out, and hardcodes `truncated` to `False`. Both limits are external clocks, so under the Gymnasium contract both are truncations and `terminated` should never be `True` at all. A training loop that follows the usual idiom and stores `done or truncated` (as ours did) then multiplies its bootstrap by $(1 - \texttt{done})$ at every episode boundary, which asserts $V(s_{500}) = 0$: that the world is worth nothing the instant the clock stops.

At $\gamma = 0.99$, the false terminal value propagates over an approximately 100-step horizon and affects all four algorithms. We correct the interface by reporting `terminated` as false and treating either upstream end flag as truncation. This removes a systematic downward bias from the value targets. Results from implementations that treat the time limit as a terminal state are therefore not directly comparable with ours.

Prior work reports near-zero value-decomposition performance on RWARE at budgets from 4×10^6 to 1.4×10^8 environment steps (Papoudakis et al., 2021; Christianos et al., 2020). Our calibration in Table 2 shows that a flat-zero QMIX curve is not by itself diagnostic: target-network configuration separates zero delivery from above 97% sustained delivery at the same training budget.

F REPRODUCIBILITY

Evaluation protocol. The primary seeded comparisons use `rware-tiny-4ag-v2` and `rware-small-8ag-v2`, parameter sharing, and 500-step episodes. Each run lasts 2×10^4 episodes, or 10^7 environment steps. Every 500 episodes, we evaluate the greedy policy for 20 episodes, yielding 40 evaluations per run. Both the learner and environment are seeded, and survival comparisons use nominal evaluation.

For each run, the *sustained delivery rate* is the mean percentage of episodes with at least one delivery over its final five evaluations. A run is *alive* when this rate exceeds 5%; otherwise, it is *collapsed* if an earlier evaluation exceeded 50% and *never learned* if none did. These thresholds were fixed before the seeded control cohort was launched. Primary survival comparisons use exact tests stratified by environment; pooled counts are descriptive.

Shared configuration. Unless a per-result exception is stated below, the off-policy runs reported in this paper use an MLP agent (one hidden layer of 64 units, the one-hot agent index appended to the current local observation), Adam at 3×10^{-4}, $\gamma = 0.99$, replay of 2.5×10^5 transitions with 5000 warm-up steps, batch 64, one gradient step every 4 environment steps, and ϵ annealed linearly $1.0 \to 0.05$ over the first 5% of the step budget. Episodes are capped at 500 steps and every

run is 2×10^4 episodes, i.e. 10^7 environment steps. Greedy evaluation is 20 episodes every 500 training episodes, giving 40 evaluation points per run. Deliveries are counted from the *unshaped* environment reward, and the environment is seeded at every `reset` together with the learner. All headline VDN and QMIX comparisons use `train_env_mode=nominal`: replay transitions are collected without injected actuator faults or observation masks, and nominal and KW-DRO arms differ only in the successor backup. This explicit setting overrides the command-line default `auto`, under which selecting the DRO learner would also select the fault-injected training wrapper. Factorial studies that deliberately vary the rollout environment are identified separately and are not used for the backup-only comparison.

The per-agent networks are memoryless, so the histories τ_i used in the general notation of Sections 3–4 reduce in the implementation to current local observations o_i. The QMIX mixer alone receives the centralized state during training; decentralized action selection uses only each agent's observation and one-hot identifier.

Per-result configuration. The primary training settings that vary by result are listed below. The target-update interval is listed per row rather than once, because the whole content of Section C.2 is that QMIX needs 2000 where VDN is unaffected at 200.

Result	Algorithm	Environments	K, ρ	Target int.	Seeds
Table 2	QMIX	`tiny-4ag`, `small-8ag`	nominal	200/2000	5/cell
Fig. 3	VDN	`small-8ag`	nominal/K=3, ρ=$\frac{1}{2}$	200	10/arm
Fig. 4	VDN	`tiny-4ag`, `small-8ag`, `small-16ag`	K=3, $\rho \in \{0, \frac{1}{2}, 1, 2, 3\}$	200	5/cell
Sec. 5.2	VDN	`tiny-4ag`, `small-8ag`	K=3, $\rho \in \{0, \frac{1}{2}, 1\}$	200	10/arm
Sec. C.8	SEAC	six `rware` layouts	n/a	n/a	3
Sec. 5.1	VDN	`small-8ag` ckpt	$K \in \{1, 3\}$	n/a	n/a

Two configurations differ from the shared block. SEAC and IAC are on-policy and use rollout buffers, so replay capacity, replay warm-up, replay-update frequency, and ϵ-greedy exploration do not apply. Their budget is 3×10^6 steps rather than 10^7 because the SEAC update is $O(N^2)$ in the number of agents. The solver verification is offline and evaluates a trained checkpoint without further training.

We use an MLP rather than a GRU because the available recurrent implementation propagates hidden state during rollout but evaluates replayed transitions with the hidden state reset, creating a rollout–update distribution mismatch. We therefore exclude recurrent agents from all reported algorithms.

Implementation complexity. Algorithm 1 is shared by VDN and QMIX and performs K batched rounds of N trials, hence $O(NK)$ candidate evaluations in the released implementation. VDN's modular objective admits the simpler $O(N \log N)$ sorting specialization proved in Corollary 11, but the current code does not implement that optimization; it returns the same prefixes through the generic routine. Algorithm 2 receives the distinct integer costs $(0, 1, \ldots, K)$ on every VDN/QMIX training path. Its equal-cost tie rule and general heterogeneous-cost tests validate the reusable solver but are not exercised by these production candidate chains.

These workloads are CPU-bound on small tensors, so we use one intra-operation thread per training process. On 32 cores, 30 processes achieved 142 environment steps/s with one thread each, compared with 3.6 steps/s with four threads each. Thread counts should therefore be fixed explicitly when reproducing the reported runtime.

Determinism and the exactness of the collapse test. Section 5.1 asserts agreement between configurations exactly rather than within a tolerance. This is meaningful because runs sharing a seed and differing only in a parameter the backup ignores draw the same random numbers in the same order, so they are bit-identical when the collapse holds. Any non-zero difference is therefore a real violation and not accumulated floating-point error, which is a strictly stronger test than one thresholded at 10^{-6}.