# IS PER-AGENT POLICY COMPOSITION SAFE? RETHINKING SUCCESSOR-FEATURE TRANSFER IN COOPERATIVE MULTI-AGENT REINFORCEMENT LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Many reinforcement learning systems, from fleet management to traffic signal control, must serve an objective that changes dynamically after deployment, and retraining a policy for each new objective is prohibitively expensive. For a single agent, this problem is well understood: successor features with generalized policy improvement, together with their universal extension, recombine a library of learned policies into a policy for any new objective, with a guarantee that the result is never worse than any policy in the library. However, multi-agent transfer has received far less attention, and the common practice of letting each agent recombine its own library independently inherits the recipe but not the guarantee. We prove that this independent composition can produce joint behavior strictly worse than every policy in the library, because recombining teammates changes the environment each agent faces and invalidates the values it relies on, a failure with no single-agent counterpart. We further show that the only unconditionally safe fixed rule is synchronized composition, which moves the whole team to one jointly trained policy but cannot serve objectives that assign different goals to different agents. To attain safety and flexibility at once, we propose multi-agent universal successor feature approximators (MA-USFA), a hierarchical method with two layers: a lower layer of universal successor feature approximators that predicts each agent's successor features while conditioned on its teammates' objectives, and an upper composer that selects, across agents, which library entry each agent should follow and supplies the cross-agent correction a per-agent value cannot represent. Trained once over the distribution of objectives, it is applied at deployment with no per-task adaptation. On a controlled grid world and a real-world city-scale traffic signal control problem, MA-USFA matches or exceeds every fixed composition rule and recovers the performance of policies retrained from scratch. The code of this paper is provided at anonymous repository `https://anonymous.4open.science/r/MA-USFA-1566`.

## 1 INTRODUCTION

Reinforcement learning is often deployed in settings where the environment and the available actions stay fixed while the objective changes repeatedly. For example, a traffic signal controller regulates the same intersections with the same phases every day, yet the goal it should optimize shifts with the hour (e.g. peak hour and off-peak hour), and different intersections in the same network may weigh throughput, delay, and queue length differently at the same moment (e.g. main road and branchway). The practical difficulty is that the two conventional responses each pay a price. One trains a dedicated policy for every objective, as cooperative multi-agent methods do when they factorize value for a single fixed team reward (Sunehag et al., 2018; Rashid et al., 2020b; Wei et al., 2019). This is accurate for the objective it targets but must be redone whenever the preference changes, and the cost grows with the number of agents. The other trains a single policy on one or a few objectives and reuses it on the rest, trading away per-objective performance for coverage (Da Silva & Costa, 2019). The real goal is neither of these but to serve an objective that keeps changing without paying either price.

For a single agent this problem is well understood: successor features (SF) and generalized policy improvement (GPI) decouple the dynamics of a policy from the objective it serves, so that a library



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.

of policies learned for a few objectives can be recombined by a closed-form rule into a policy for any new objective in the same family, with a guarantee that the recombined policy is never worse than any policy already in the library (Barreto et al., 2017; 2020), and universal successor feature approximators (USFA), which inherit the goal-conditioned value modeling of universal value function approximators (UVFA) (Schaul et al., 2015), extend the same rule from a finite library to a whole distribution of objectives (Borsa et al., 2018). Cooperative multi-agent transfer, where the cost of retraining grows with the number of agents, has received far less attention. The natural way to carry the single-agent recipe into a team is to let each agent apply the recombination rule to its own library, selecting its own component of the joint action independently. Several recent works follow exactly this template (de Almeida et al., 2024; Liu et al., 2022; Nigam et al., 2025), on the implicit assumption that the single-agent guarantee carries over to the team. Some acknowledge that the multi-agent improvement guarantee is not established, or allow a few steps of fine-tuning at deployment, but none characterizes when the assumption holds and when it fails. (A detailed discussion of related work is deferred to Appendix A.)

We supply the missing characterization, and it has two sides. Independent per-agent composition is not safe in general: even in cooperative tasks whose rewards are fully separable, the independently composed policy can be strictly worse than every policy in the shared library, because each agent's values were learned while its teammates followed their old policies, and recombining the library changes the environment each agent faces. This failure has no counterpart in single-agent problems, where the environment is fixed. The rule that does carry a guarantee is synchronized composition, in which the whole team switches together to a single jointly trained policy indexed by one shared choice. We prove that it is unconditionally safe, never worse than any joint policy in the library for any objective, and without any assumptions on the reward or the dynamics. We further give conditions, checkable from the library alone, under which independent composition regains its guarantee, so the field is left with both a correct safe baseline and a precise test for when the cheaper rule may still be used.

Synchronized composition, however, guarantees only that the team performs no worse than the library, because it can only reproduce joint policies stored as a whole, and it cannot serve objectives that are heterogeneous across the team, assigning different goals to different agents. We therefore propose multi-agent universal successor feature approximators (MA-USFA), a hierarchical method with two layers. The lower layer is a per-agent USFA that predicts each agent's successor features while conditioned on a compact summary of what the teammates are being asked to do, so that no agent's value assumes a fixed set of teammates. The upper layer is a small learned composer that reads the joint state and selects each agent's library component, learning exactly the cross-agent correction that a per-agent value cannot represent by itself. The composer is trained once over the distribution of objectives anticipated before deployment and applied with no per-task adaptation. On a controlled grid world and a real-world city-scale traffic signal problem with nearly two hundred agents, MA-USFA improves throughput and delay over both fixed rules and recovers the performance of policies retrained separately for each objective. Together these results turn independent per-agent composition, the field's de facto but unexamined heuristic, into a well-posed problem: a precise diagnosis of when it fails, a provably safe baseline to fall back on, and a learned method that is safe and flexible at once.

## 2 Preliminaries and Problem Formulation

### 2.1 Single-agent transfer: successor features, generalized policy improvement, and universal value modeling

Consider a Markov decision process with state space $S$, action space $A$, transition kernel $P$, discount factor $\gamma$, and a feature function $\phi : S \times A \to \mathbb{R}^d$. The reward is linear in the features, $r_w(s,a) = \phi(s,a)^\top w$, so different objectives in the same family differ only in the weight vector $w$. The successor features of a policy $\pi$ are the expected discounted feature sums

$$\psi^\pi(s,a) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t \phi(s_t,a_t) \,\Big|\, s_0 = s,\ a_0 = a,\ \pi\Big], \tag{1}$$

and the value of $\pi$ under any objective $w$ is the linear function $V_w^\pi(s) = \psi^\pi(s,\pi(s))^\top w$: a single successor feature model prices a policy under any $w$ without new rollouts.

GPI turns this pricing rule into a composition rule over a library of policies $\{\pi^1, \ldots, \pi^K\}$ with successor features $\{\psi^1, \ldots, \psi^K\}$: the composed policy acts greedily with respect to the best library value,

$$\pi(s) \in \arg\max_a \max_k \psi^k(s,a)^\top w_{\text{test}}, \tag{2}$$

and satisfies $V^{\pi}_{w_{\text{test}}}(s) \geq \max_k V^{\pi^k}_{w_{\text{test}}}(s)$ for every state $s$: the composed policy is never worse than any policy in the library (Barreto et al., 2017; 2020). Two premises make this machinery work: the dynamics are fixed, so the successor features stored in the library remain valid whatever the objective. There is a single decision maker, so the rule in Eq. (2) is executed on the only action space there is.

UVFA make the value model a function of the objective itself, $V(s,g)$, so that one network represents a whole family of value functions and generalizes to unseen objectives by interpolation rather than by retraining a separate value function for each (Schaul et al., 2015). USFA combine this idea with successor features in a single model $\tilde{\psi}(s,a,z,w)$ that carries two axes: a policy axis $z$ that indexes which library policy is being evaluated, and a task axis $w$ that shapes the behavior during training and prices any objective at test time. After one training run over a distribution of objectives, composition at test time is therefore a dot product over a candidate set, with no per-objective adaptation (Borsa et al., 2018). With a single goal as the candidate set USFA reduces to UVFA, and with a finite library to SF and GPI.

Appendix B recalls the pricing identity and the GPI improvement guarantee, and explains why UVFA and USFA generalize across objectives and what motivates the USFA policy-task factorization. Our multi-agent analysis builds directly on these single-agent facts.

### 2.2 Cooperative multi-agent setting

We now lift the single-agent setup of Section 2.1 to a team, and the notation carries an agent index $i$ throughout. We formalize the team as a multi-agent Markov decision process: a set of $N$ agents shares a state space $S$, agent $i$ takes action $a_i$, and the joint action is $a = (a_1, \ldots, a_N)$, with $a_{-i} = (a_j)_{j \neq i}$ the teammates' actions. The transition kernel $P(s' \mid s, a)$ and the features depend on the joint action.

Our scope is cooperative transfer, where each agent carries its own objective but the team shares one criterion. Agent $i$ has features $\phi_i(s,a)$ and a task weight $w_i$, giving a per-agent reward $r_i(s,a) = \phi_i(s,a)^\top w_i$ and, under a joint policy $\pi$, a per-agent reward-to-go $V_i^\pi(s) = \psi_i^\pi(s,\pi(s))^\top w_i$ built from the per-agent successor features $\psi_i^\pi$ of Eq. (1). The system objective is the sum of these reward-to-go values,

$$V^\pi(s) \;=\; \sum_{i=1}^{N} V_i^\pi(s) \;=\; \psi^\pi(s,\pi(s))^\top w, \qquad \psi^\pi = \sum_i \psi_i^\pi, \tag{3}$$

so the team maximizes total reward-to-go, the cooperative criterion. A homogeneous task assigns all agents one shared weight $w$, and a heterogeneous task assigns a per-agent vector $(w_1, \ldots, w_N)$, the case single-agent transfer has no analogue for. The objective distribution the team is trained and deployed on spans both types. This matters for composition: a synchronized joint policy commits the whole team to one shared weight, so it can be trained on and can serve only the homogeneous objectives, whereas the heterogeneous objectives are reachable only by composing per-agent policies.

Two differences from the single-agent case of Section 2.1 drive everything that follows, and each breaks one of its two premises. First, execution is decentralized: $N$ agents each select their own $a_i$, so the single decision maker of Eq. (2) is replaced by $N$ of them acting in parallel. Second, the teammates are part of each agent's environment: the marginal transition of agent $i$ is $P_i(s_i' \mid s, a_i, a_{-i})$ and its features may be $\phi_i(s_i, a_i, a_{-i})$, so recomposing the library changes $a_{-i}$ and hence the very process against which agent $i$'s successor features were measured. The fixed-dynamics premise and the single-decision-maker premise therefore both fail, and Section 3 traces every safety question back to these two failures.

### 2.3 Libraries and composition rules

A policy library $\Pi = \{\pi^1, \ldots, \pi^K\}$ is trained before deployment and contains two kinds of entries: synchronized entries, joint policies trained on homogeneous tasks so the whole team can switch

to $\pi^k$ through a single index $k$, and independent entries, assembled from per-agent policies $\pi^k = (z_1^k, \ldots, z_N^k)$ trained for their own task contexts, which span combinations no synchronized entry can represent. Each entry carries a joint successor feature $\psi^k$, measured directly for synchronized entries and summed from per-agent features when the features decompose additively. For each agent $i$ we also define the per-agent successor feature $\psi_i^k(s, a_i)$ of entry $k$: the expected discounted features of agent $i$ when it takes $a_i$ while the teammates follow entry $k$ and everyone continues along entry $k$ afterwards, a snapshot quantity measured against a specific version of the teammates. Requirement 2 will turn precisely this dependence into a problem. A composition rule $\mathrm{R}$ maps the library and a deployment objective to a joint policy, with no gradient updates at deployment time. This paper studies exactly three, which differ only in how each agent's library component is selected: two fixed rules, analyzed in Section 3, and the learned rule we propose in Section 4. (Table 4 in Appendix E summarizes the three rules.)

The first fixed rule, *synchronized composition*, selects one library entry per state and replays its joint action,

$$\pi^{\mathrm{sync}}(s) \;=\; \pi^{k^*}(s), \qquad k^* \in \arg\max_k \psi^k\big(s, \pi^k(s)\big)^\top w_{\mathrm{test}}, \tag{4}$$

the composition rule of Eq. (2) restricted to the joint actions the entries themselves take. Under valid values the score in Eq. (4) is the value of the entry itself, $\psi^k(s, \pi^k(s))^\top w_{\mathrm{test}} = V^{\pi^k}(s)$, so the synchronized rule always replays the entry that is best at the current state.

The second fixed rule, *independent composition*, lets each agent apply the single-agent rule to its own library,

$$\pi_i^{\mathrm{ind}}(s) \in \arg\max_{a_i} \max_k \psi_i^k(s, a_i)^\top w_i, \tag{5}$$

which is how the recent multi-agent transfer literature carries GPI into a team (de Almeida et al., 2024; Liu et al., 2022; Nigam et al., 2025).

The third learned rule, *MA-USFA*, is our proposed method: it keeps the decision space of independent composition but replaces the fixed argmax in Eq. (5) with a learned upper-layer composer over a per-agent successor-feature layer (Section 4). The deployment goal is zero per-task adaptation, amortized over the preference distribution: all learning happens once, before deployment, and at deployment the rule acts on the frozen library with no adaptation to the particular $w_{\mathrm{test}}$.

## 3 Fixed Methods: Synchronized and Independent Compositions

### 3.1 Design goals: safety and flexibility beyond the library

To evaluate a composition rule, we consider two criteria, and no fixed rule delivers both. The first is *safety*: the composed policy should never be worse than the best policy already in the library.

**Definition 1** (Safety). *A composition rule* $\mathrm{R}$ *is safe on* $(\Pi, W_{\mathrm{test}})$ *if the composed policy* $\pi^{\mathrm{R}}$ *satisfies, for every* $w_{\mathrm{test}} \in W_{\mathrm{test}}$ *and every state* $s$*,*

$$V_{w_{\mathrm{test}}}^{\pi^{\mathrm{R}}}(s) \geq \max_k V_{w_{\mathrm{test}}}^{\pi^k}(s), \tag{6}$$

*where the value is the team objective* $V = \sum_i V_i$ *of Eq.* (3) *and the maximum is over library entries with valid joint successor features.*

The second goal is *flexibility* beyond the library: a heterogeneous objective can ask each agent for something that no single stored joint policy contains, so we want a rule that can serve such objectives and, where the task structure allows, improve on every stored entry rather than merely tie the best one.The two goals pull against each other, and the two fixed rules sit at opposite corners. Synchronized composition, next, meets the safety goal but forfeits flexibility. Independent composition reaches for flexibility but forfeits safety. MA-USFA (Section 4) is built to meet both.

### 3.2 Synchronized composition is unconditionally safe

**Proposition 1** (Synchronized composition is unconditionally safe). *Let* $\Pi$ *be any library,* $\pi^{\mathrm{sync}}$ *the synchronized composition of Eq.* (4)*, and* $w_{\mathrm{test}}$ *any deployment objective. For every state* $s$ *and every library entry* $k$*,*

$$V_{w_{\mathrm{test}}}^{\pi^{\mathrm{sync}}}(s) \;\geq\; V_{w_{\mathrm{test}}}^{\pi^k}(s). \tag{7}$$

*Proof sketch.* At every state the synchronized rule scores each entry by the value of replaying it, $\psi^k(s, \pi^k(s))^\top w_{\text{test}} = V^{\pi^k}(s)$, and plays the action of the highest-scoring entry. Comparing against any entry $\pi^k$, its action scores at least as high at every state, so the one-step advantage telescopes into $V^{\pi^{\text{sync}}}_{w_{\text{test}}}(s) \geq V^{\pi^k}_{w_{\text{test}}}(s)$ (Appendix C). □

The structural reason is that synchronized switching never changes anyone's environment, so every joint successor feature stays valid. Synchronized composition is therefore the safe floor any claimed improvement must beat. What it cannot do, taken up in Section 3.4, is coverage: its composition space is only the $K$ library entries.

### 3.3 Independent composition: unsafe in general, safe under two conditions

Independent composition can express the heterogeneous objectives synchronized composition cannot, but its safety is conditional on two requirements, one repairing each of the two single-agent premises that a team breaks (Section 2.2). Decentralized execution breaks the single-decision-maker premise, so the first requirement constrains which joint actions the per-agent choices select. Teammates entering each agent's environment breaks the fixed-dynamics premise, so the second requirement asks whether the stored per-agent values are still correct after recomposition.

**Requirement 1** (Selection alignment)**.** *The joint actions optimal for the test objective are reachable by per-agent greedy choices: with $A^*(s) = \arg\max_a V^*_{w_{\text{test}}}(s, a)$, alignment requires $A^*(s) = \prod_i A^*_i(s)$, the Individual-Global-Max (IGM) condition of value-decomposition theory (Sunehag et al., 2018; Rashid et al., 2020b; Son et al., 2019) carried from training time to composition time.*

**Requirement 2** (Value validity)**.** *The per-agent successor features stored in the library remain the true successor features of the composed joint policy: each $\psi^k_i$ was measured while the teammates followed entry $k$, and once composition changes their behavior $\psi^k_i$ prices a policy that is no longer being executed.*

**Proposition 2** (The two requirements suffice)**.** *Fix a library and a test objective $w_{\text{test}}$. If Requirement 1 and Requirement 2 both hold, then independent composition is safe: $V^{\pi^{\text{ind}}}_{w_{\text{test}}}(s) \geq \max_k V^{\pi^k}_{w_{\text{test}}}(s)$ for every state $s$.*

The argument is direct: validity makes each per-agent score $\psi^k_i(s, a_i)^\top w_i$ the true value of the composed policy, and alignment makes the per-agent greedy choice coincide with the joint-GPI action, so independent composition executes the joint-GPI rule with correct values and inherits its single-agent guarantee (Appendix C). The two requirements are thus the whole story. They are fed by two independent channels of coupling: the reward channel (cross-features in $\phi$) governs Requirement 1, and the transition channel (teammate actions in the dynamics) governs Requirement 2. We show first that a benign-looking task can break validity even when alignment holds, then give the exact condition on each channel under which the independent rule is not merely safe but no worse than synchronized composition.

**It can fail even when rewards are fully separable.**

**Lemma 3** (Independent composition is not always safe)**.** *For every number of agents $N \geq 2$ there exists a cooperative multi-stage task whose rewards are fully separable (each agent's reward depends only on its own action), such that the alignment condition (Requirement 1) holds at the test objective and every library entry is suboptimal, yet*

$$V^{\pi^{\text{ind}}}_{w_{\text{test}}}(s_0) \;<\; \max_k V^{\pi^k}_{w_{\text{test}}}(s_0), \tag{8}$$

*so the independently composed policy is strictly worse than every policy in the library, while the centralized joint-GPI rule over the same library attains the joint optimum $V^*_{w_{\text{test}}}(s_0)$. The failure is therefore specific to decentralization.*

The construction is a multi-stage task with per-agent feature tables (Appendix C). The failure is not misalignment (Requirement 1 holds) but validity (Requirement 2 fails). Each agent acts exactly as its library values tell it to, but the values are stale: agent $i$ scores its choice against the features it would accrue if the teammates kept the library's actions, while their independent choices send the

team down a different branch and the accrued features fall short. A single stale rating pushes the team below its own best library entry, the precise sense in which teammates constitute the environment. Whether the failure is statistically visible depends on how the library was built, so safety must be characterized structurally rather than by example.

**The reward channel: supermodularity and the weight cone.**

**Proposition 4** (Supermodular values make independent composition no worse than synchronized composition)**.** *Let $g_s(a) = \max_k \psi^k(s,a)^\top w_{\text{test}}$ be the joint GPI value of the library. Suppose that for every state $s$, $g_s$ is supermodular on the lattice $\{0,1\}^N$, and ties are broken consistently across agents. Then the independent composition* (5) *satisfies, for every state $s$,*

$$V^{\pi^{\text{ind}}}_{w_{\text{test}}}(s) \;\geq\; V^{\pi^{\text{sync}}}_{w_{\text{test}}}(s). \tag{9}$$

**Mechanism.** Supermodularity of $g_s$ says that the marginal gain of one agent improving its action increases in the actions of the other agents. By Topkis's monotone comparative statics (Topkis, 1998) the argmax set is then a sublattice of the action lattice, and with a consistent tie-break the per-agent greedy choices coincide with the joint greedy choice. Requirement 1 therefore holds as a condition on the value structure rather than on an assumed decomposition, the structural generalization of value decomposition, which guarantees alignment by construction during training (Sunehag et al., 2018; Rashid et al., 2020b; Son et al., 2019) whereas supermodularity grants it at composition time.

**Corollary 5** (The weight cone)**.** *Assume the per-pair value differences of the library policies are linear in the features. Then the per-state supermodularity condition of Proposition 4 holds if and only if the test weight lies in the cone*

$$K_\phi \;=\; \big\{w : \textstyle\sum_d w_d\, \Delta\phi_d(a,b) \geq 0 \;\text{ for every pair of joint actions } a, b\big\}, \tag{10}$$

*where $\Delta\phi_d(a,b) = \phi_d(a \vee b) + \phi_d(a \wedge b) - \phi_d(a) - \phi_d(b)$ is the supermodularity gap of feature $d$ on the pair. A sufficient and easy-to-use condition: every feature supermodular and $w \geq 0$.*

The cone is checkable from library statistics: membership of $w_{\text{test}}$ is a dot product, so whether the free region applies to a given task can be read off the library and the test weight without running the policy.

**The transition channel: factorization is necessary, not sufficient.**

**Proposition 6** (Factored transitions with per-agent rewards)**.** *Suppose the transition kernel factorizes per agent, $s'_i = f_i(s_i, a_i)$, so that no agent's dynamics depend on teammate actions, and the reward features decompose per agent, $\phi(s,a) = \sum_i \phi_i(s_i, a_i)$. Then the per-agent successor feature $\psi^k_i(s, a_i)$ is the discounted feature stream of agent $i$'s own dynamics, and it depends on neither the entry $k$ nor the composition. Requirement 2 therefore holds by construction, for every library and every recomposition. Under the supermodularity and tie-breaking conditions of Proposition 4, the independent composition then satisfies $V^{\pi^{\text{ind}}}_{w_{\text{test}}}(s) \geq V^{\pi^{\text{sync}}}_{w_{\text{test}}}(s)$ for every state $s$.*

Factorization neutralizes the transition channel only in the decomposed-reward case, where the per-agent values are exact marginal processes of the agent's own dynamics. When the reward features couple the agents, the cross-feature components are measured against the library's teammate behavior and recomposition invalidates them: even with factored transitions and per-state supermodularity, the independent rule can violate the safety bound under the exact semantics of Eq. (5).

### 3.4 Beyond the free region: when no fixed rule suffices

The two fixed rules cover complementary but incomplete parts of the problem, and a large class of tasks falls outside both. Synchronized composition is always safe but cannot serve heterogeneous objectives: it replays one library entry as a joint action, so its reachable behavior is exactly the $K$ entries, and a test objective that asks different agents to pursue different components of the library has no synchronized representative. Independent composition can express those heterogeneous choices, but only inside the free region of Propositions 4 and 6, and both free conditions can fail. Transition coupling breaks Requirement 2 outright: the counterexample of Lemma 3 is an exact failure in which the independent rule falls below the best library entry and below joint-GPI. Cross-feature rewards break Requirement 1, so even factored transitions do not by themselves restore safety.

The two conditions are structural properties of the task, not tunable knobs. For example, a traffic network couples each intersection to its upstream neighbors, so factorization fails by construction and real coupled problems live in the failure region, where synchronized composition is too rigid to serve heterogeneous demand and independent composition is no longer safe. This is the region a learned composition rule must cover, and it is the design target of MA-USFA (Section 4), which is applied to every objective and is built to be at least as good as either fixed rule everywhere, so its use never depends on a safety test. The theory of this section instead governs the fixed-rule-only regime: when a learned composer is unavailable, synchronized composition is the unconditionally safe default, and the independent rule may replace it precisely where its two requirements hold.

# 4 LEARNABLE METHOD: MA-USFA

## 4.1 OVERVIEW

Section 3 shows that each fixed rule is safe only on part of the objective space, so a single rule that is safe and flexible on every objective must learn the composition. MA-USFA is a hierarchy of two decision layers over the cooperative multi-agent MDP of Section 2.2 (Fig. 1). The lower layer acts in that MDP with the primitive per-agent actions $a_i$: a per-agent context-conditioned USFA predicts each agent's successor features, conditioned on a compact summary of the teammates' task context, so that no agent's value assumes a fixed set of teammates. The upper layer is a selection layer where composition happens. Its observation is the joint state $s$, the task vector $w$, and the lower layer's candidate valuations $\{\tilde{\psi}_i(s,\cdot,z_i)^\top w_i : z_i \in C_i\}$. Its action is a per-agent choice of library entry $g_i \in C_i$, whose primitive action the team then executes. The upper layer thus decides which library policy each agent follows, the cross-agent correction a per-agent value cannot express, while the lower layer supplies the values it chooses among. Because the composer is initialized at the independent rule and trained with the value layer frozen, moving only in directions that raise team value, on every objective MA-USFA starts from a fixed rule and never performs below it.

## 4.2 THE TWO LAYERS

**Lower layer: per-agent context-conditioned successor features.** Each agent $i$ learns a value model

$$\tilde{\psi}_i(s, a_i \mid z_i, w_{-i}), \tag{11}$$

where $z_i$ is the agent's own policy encoding and $w_{-i} = (w_j)_{j\neq i}$ collects the teammates' task weights. The two arguments play the two roles of USFA (Borsa et al., 2018): $z_i$ is the policy axis, a learned index over the agent's library entries, and the weights carry the task axis, linear in the value and applied only at test time. The conditioning on $w_{-i}$ is the new ingredient, the architectural answer to Requirement 2: the teammate task context $w_{-i}$ indexes a cluster of teammate policies, so agent $i$ learns expected successor features over the resulting teammate behavior distribution rather than a snapshot against one fixed teammate version. Without it the value is stale the moment the library holds more than one teammate policy, the failure Lemma 3 exposes. For large teams the context is summarized (mean or learned embedding of $w_{-i}$).

Training follows the USFA protocol: the TD target of $\tilde{\psi}_i$ is the feature vector $\phi$, so the update does not depend on $w$, which plays the standard roles of behavior anchor, library anchor ($z \sim D_z(\cdot \mid w)$), and test-time pricing. With per-agent additive features the joint successor feature of a synchronized entry decomposes as $\psi^{\text{joint},k} = \sum_i \psi_i^k$, so one network serves both the synchronized anchor and the per-agent library (Appendix G).

**Upper layer: the learned composer.** The composer is a collection of per-agent selectors $\Upsilon^\theta = (\Upsilon_1^\theta, \ldots, \Upsilon_N^\theta)$. Its input is the joint state, the task vector, and the candidate valuations $\{\tilde{\psi}_i(s,\cdot,z_i)^\top w_i : z_i \in C_i\}$ from the lower layer. Its output is a selection $g_i \in C_i$ per agent:

$$g_i \;=\; \Upsilon_i^\theta\Big(s,\, w \;\Big|\; \big\{\, \tilde{\psi}_j(s,\cdot,z_j)^\top w_j \,\big\}_{j,\, z_j \in C_j}\Big) \;\in\; C_i, \tag{12}$$

It learns the cross-agent value correction: the lower layer absorbs the expected part of the teammate dependence, predictable from the public task context, and the composer adds the residual that depends on the joint combination of candidate policies, which no per-agent value can price. It does not relearn

single-agent decision making, only corrects values across agents, which is why it is small and trainable on a fraction of the library's budget. Similar to the lower layer, for large teams the contexts ($w$ and $\{\tilde{\psi}_j(s,\cdot,z_j)^\top w_j\}_{j,\, z_j \in C_j}$) are summarized (e.g. via attention or communication mechanisms) to ensure scalability.

### 4.3 Training and deployment

Training has two phases, one per layer (Table 3 and Appendix D, with full pseudocode in Appendix D.2). Phase 1 trains the lower per-agent value layer, sampling tasks at random from the homogeneous and heterogeneous scenarios with the $w_{-i}$ conditioning, and then freezes it. This layer produces the candidates the composer selects among. Phase 2 trains the upper composer, initialized at the independent transfer policy with the value layer frozen. Wherever that policy is already optimal the value-increasing gradient vanishes and no parameters move, so the budget concentrates on the objectives whose coupling actually calls for a cross-agent correction, namely coupled dynamics and cross-feature rewards. This is what keeps the composer small, the difference between MA-USFA and an end-to-end composer.

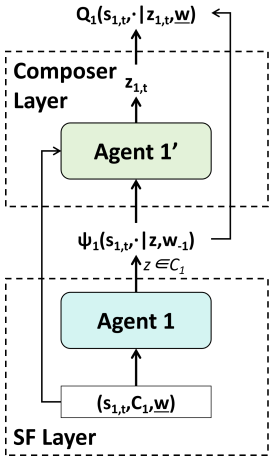



Figure 1: MA-USFA shown for a single agent.

Deployment is a single forward pass with no gradient updates, run on every objective: the lower layer prices the candidates by dot product and the composer selects, so any $w_{\text{test}}$ is served with no per-task adaptation. This is the entire method at test time. The composer is always the deciding rule, and the fixed rules of Section 3 take no part in it. The cost of learning to compose is paid once before deployment over the distribution of anticipated objectives, after which every objective needs only forward passes, the precise sense in which the method needs zero per-task adaptation.

## 5 Experiments

### 5.1 Experiment Setup

In this section, we aim to answer three questions: (i) Does independent composition fail exactly where the theory says it must, while synchronized composition holds its floor (Proposition 1, Lemma 3)? (ii) Does the learned composer recover per-objective retraining where the fixed rules break? (iii) And do both survive at large scale, where coupling is endogenous rather than dialed in? The controlled domain below answers the first two. The third is answered by a traffic signal control task on a Manhattan grid of $28 \times 7$ intersections (196 signal agents), run on a homogeneous task (one shared weight) and a heterogeneous task (each intersection with its own weight). Overall, MA-USFA reaches or exceeds per-task retraining while staying above both fixed rules. We leave the detail to Appendix F.

**Methods and baselines.** We compare five methods on every objective. The three composition rules of Section 2 are the objects of study: synchronized composition, independent composition, and MA-USFA. Two methods bracket them: joint-GPI, centralized generalized policy improvement over the joint action space (the prior-work baseline, feasible only for small teams), and per-task retraining, trained from scratch on the deployment reward including the out-of-basis collision penalty.

### 5.2 Controlled domain: SFWorld

SFWorld is a multi-agent grid world that exposes the two coupling channels of Section 3 as controllable parameters. $N$ agents move on a $5 \times 5$ grid with five actions over 40 steps ($\gamma = 0.95$). Features $\phi_i(s')$ are smooth bumps over $K$ resource cells, and the reward $r_i = \phi_i(s')^\top w_i - 2.0 \cdot \text{blocked}_i$ adds a collision penalty outside the feature basis that any closed-form pricing $\psi \cdot w$ is structurally blind to. The coupling parameter $\kappa$ is the probability that two agents targeting the same cell collide: at $\kappa = 0$ the dynamics factorize (Proposition 6). As $\kappa$ grows, each agent's kernel depends on the teammates' actions. In $A_{\text{distinct}}$ the optimal regions are disjoint so coupling never binds. In $B_{\text{overlap}}$

the agents contend for a shared region so collisions arise whenever the joint policy hybridizes. More details are provided at Appendix E.

Table 1: Controlled domain, $N = 2$, four-entry corner library. Team return over 40 steps, mean $\pm$ standard deviation over three evaluation seeds. $\mathrm{coll}_{\mathrm{ind}}$ is the collision rate of the independent rule. Per row, **best** is bold and second best underlined (used throughout the paper).

| task | $\kappa$ | Sync | Indep | joint-GPI | MA-USFA | retrain | $\mathrm{coll}_{\mathrm{ind}}$ |
|---|---|---|---|---|---|---|---|
| $A_{\mathrm{distinct}}$ | 0.00 | $23.42_{\pm 0.34}$ | $\mathbf{38.73}_{\pm 0.47}$ | $23.74_{\pm 0.29}$ | $38.40_{\pm 0.53}$ | $\underline{38.69}_{\pm 0.48}$ | 0.000 |
| $A_{\mathrm{distinct}}$ | 0.50 | $23.37_{\pm 0.16}$ | $\underline{38.23}_{\pm 0.01}$ | $23.63_{\pm 0.14}$ | $38.17_{\pm 0.22}$ | $\mathbf{38.35}_{\pm 0.10}$ | 0.000 |
| $A_{\mathrm{distinct}}$ | 1.00 | $23.44_{\pm 0.15}$ | $\underline{38.31}_{\pm 0.27}$ | $23.80_{\pm 0.23}$ | $38.25_{\pm 0.18}$ | $\mathbf{38.48}_{\pm 0.16}$ | 0.000 |
| $B_{\mathrm{overlap}}$ | 0.00 | $31.52_{\pm 0.26}$ | $\mathbf{38.77}_{\pm 0.14}$ | $32.59_{\pm 0.20}$ | $\underline{38.44}_{\pm 0.13}$ | $37.56_{\pm 0.25}$ | 0.000 |
| $B_{\mathrm{overlap}}$ | 0.25 | $31.36_{\pm 0.34}$ | $29.96_{\pm 0.32}$ | $\mathbf{31.71}_{\pm 0.24}$ | $\underline{31.52}_{\pm 0.16}$ | $30.56_{\pm 0.49}$ | 0.125 |
| $B_{\mathrm{overlap}}$ | 0.50 | $\mathbf{31.27}_{\pm 0.09}$ | $21.04_{\pm 0.94}$ | $30.65_{\pm 0.14}$ | $30.94_{\pm 0.30}$ | $\underline{31.08}_{\pm 0.25}$ | 0.225 |
| $B_{\mathrm{overlap}}$ | 0.75 | $\mathbf{31.15}_{\pm 0.19}$ | $13.67_{\pm 0.30}$ | $29.38_{\pm 0.29}$ | $\underline{30.71}_{\pm 0.25}$ | $30.67_{\pm 0.10}$ | 0.318 |
| $B_{\mathrm{overlap}}$ | 1.00 | $\mathbf{31.24}_{\pm 0.16}$ | $7.07_{\pm 0.20}$ | $28.62_{\pm 0.60}$ | $\underline{31.18}_{\pm 0.18}$ | $30.33_{\pm 0.31}$ | 0.369 |
| Average (all tasks) | | 28.35 | 28.22 | 28.02 | **34.70** | $\underline{34.46}$ | 0.130 |

Table 1 carries the whole controlled-domain story: its $B_{\mathrm{overlap}}$ block exhibits the four claims of the theory at once, and $A_{\mathrm{distinct}}$ isolates coverage from coupling. Down $B_{\mathrm{overlap}}$ the synchronized rule stays flat as the coupling climbs (Proposition 1), since synchronized switching never changes anyone's environment, whereas the independent rule falls monotonically as its collision rate rises (Lemma 3), pairing per-agent choices into hybrid joint policies whose collisions the harvest-only values cannot price. The two orderings cross: the independent rule wins in the free region but the order flips once coupling binds. MA-USFA instead reaches per-task retraining, exceeding it where the coupling binds, because transfer learning lets the composer reuse general structure already learned across tasks that retraining must rediscover. Joint-GPI degrades only mildly but never reaches retraining and is unavailable beyond small teams. In $A_{\mathrm{distinct}}$, where collisions never occur, independent composition is free as Proposition 6 predicts. Averaged over all eight tasks MA-USFA reaches the level of per-task retraining or above (34.70 versus 34.46) and is clearly above every fixed rule, with team-size and library-coverage sweeps in Appendix E confirming the pattern.

## 6 Conclusion

In this paper, we study transfer learning in cooperative multi-agent reinforcement learning. We first identify that independent per-agent composition (the approach adopted by most prior work) does not inherit the single-agent safety guarantee. In contrast, the only unconditionally safe fixed rule is synchronized composition, albeit at the cost of coverage. To balance safety and flexibility, we propose MA-USFA, a hierarchical solution that combines a frozen per-agent value layer with a learned cross-agent correction. Trained once over the preference distribution and deployed with zero per-task adaptation, MA-USFA attains both the safety of synchronized composition and the flexibility of independent recombination. Evaluated on a controlled grid-world scenario and a real-world large-scale traffic signal control task, MA-USFA matches or exceeds every fixed composition rule and reaches the level of per-task retraining.

## AI Use Statement

*In this work, we used generative AI tools for paper polishing, including grammar and wording checks, verification and support of mathematical derivations, and coding. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.*

## Ethics Statement

*This work adheres to the principles outlined in the ICLR Code of Ethics.*

## Reproducibility Statement

*The code for this paper is available at* `https://anonymous.4open.science/r/MA-USFA-1566`.

## References


Lucas Nunes Alegre, Ana Bazzan, and Bruno C Da Silva. Optimistic linear support and successor features as a basis for optimal policy transfer. In *International conference on machine learning*, pp. 394–413. PMLR, 2022.

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado Van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

André Barreto, Shaobo Hou, Diana Borsa, David Silver, and Doina Precup. Fast reinforcement learning with generalized policy updates. *Proceedings of the National Academy of Sciences*, 117 (48):30079–30087, 2020.

Diana Borsa, André Barreto, John Quan, Daniel Mankowitz, Rémi Munos, Hado Van Hasselt, David Silver, and Tom Schaul. Universal successor features approximators. *arXiv preprint arXiv:1812.07626*, 2018.

Felipe Leno Da Silva and Anna Helena Reali Costa. A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artificial Intelligence Research*, 64:645–703, 2019.

Vinícius A. de Almeida, Lucas N. Alegre, and Ana L. C. Bazzan. Knowledge transfer in multi-objective multi-agent reinforcement learning via generalized policy improvement. *Computer Science and Information Systems*, 2024.

Wenzhang Liu, Lu Dong, Dan Niu, and Changyin Sun. Efficient exploration for multi-agent reinforcement learning via transferable successor features. *IEEE/CAA Journal of Automatica Sinica*, 9 (9):1673–1686, 2022.

Rupal Nigam, Niket Parikh, Hamid Osooli, Mikihisa Yuasa, Jacob Heglund, and Huy T Tran. Zero-shot coordination in ad hoc teams with generalized policy improvement and difference rewards. *arXiv preprint arXiv:2510.16187*, 2025.

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.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pp. 1312–1320. PMLR, 2015.

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.

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 *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2018.

Donald M Topkis. *Supermodularity and complementarity*. Princeton university press, 1998.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1913–1922, 2019.

# APPENDIX CONTENTS

**A Related Works** **13**

**B Single-Agent Foundations** **13**

**C Proofs** **14**

C.1 Proof of Proposition 1 . . . . . 14

C.2 Proof of Proposition 2 . . . . . 14

C.3 The counterexample of Lemma 3: full construction . . . . . 15

C.4 Proof of Proposition 4 . . . . . 16

C.5 Proof of Corollary 5 . . . . . 17

C.6 Proof of Proposition 6 . . . . . 17

**D MA-USFA: Training and Deployment** **17**

D.1 The two training phases . . . . . 17

D.2 Pseudocode . . . . . 17

D.3 Composition rules studied as baselines . . . . . 19

**E Controlled Domain: SFWorld** **19**

E.1 Setup . . . . . 19

E.2 The crossover . . . . . 20

E.3 Team-size sweep . . . . . 20

E.4 Library-coverage sweep . . . . . 20

**F City-Scale Domain: Traffic Signal Control** **22**

F.1 Setup . . . . . 22

F.2 Main result . . . . . 24

F.3 Robustness sweeps . . . . . 24

**G Discussions** **24**

# A Related Works

**Single-agent transfer.** Successor features decouple the dynamics of a policy from the objective it serves: a single successor feature model prices any policy under any objective in a linear family by a dot product (Barreto et al., 2017). Generalized policy improvement turns this pricing rule into a composition rule with a guarantee: the greedy recombination of a library of policies is never worse than any policy in the library (Barreto et al., 2020). Universal value function approximators condition the value model on the goal directly and interpolate across goals (Schaul et al., 2015). Universal successor feature approximators merge the two: a model conditioned on a policy encoding and a task weight, trained once over a distribution of objectives, so that composition at test time is a closed-form dot product over a candidate set (Borsa et al., 2018). Optimistic linear support and successor features extend the same machinery to sequential policy transfer across a set of objectives (Alegre et al., 2022). The present paper takes the operating mode of this lineage, train once and compose with no per-task adaptation, and asks what survives in a team; the answer is that the guarantee does not survive independent per-agent execution.

**Multi-agent transfer.** The survey of da Silva and Costa (Da Silva & Costa, 2019) organizes multi-agent transfer along what is transferred, to whom, and with what mechanism, and documents that most proposals carry single-agent recipes into teams without re-examining their premises. The works closest to ours follow exactly the template we analyze: de Almeida et al. (2024) transfer knowledge across multi-objective multi-agent tasks by letting each agent keep its own library and apply generalized policy improvement independently, and do not establish a multi-agent improvement guarantee; Liu et al. (2022) transfer successor features per agent to accelerate exploration, again per-agent; Nigam et al. (2025) study zero-shot coordination in ad hoc teams with generalized policy improvement and difference rewards, and explicitly note that the single-agent improvement guarantee is not established for teams, the gap this paper fills with a precise negative result and a positive one. None of these works characterizes when the per-agent recipe is safe and when it fails; our Lemma 3 and Propositions 4 and 6 draw that boundary.

**Value decomposition.** The cooperative deep multi-agent literature aligns per-agent and joint value models at training time. VDN sums the per-agent values (Sunehag et al., 2018); QMIX and Weighted QMIX mix them through monotone networks and characterize the family of joint value functions representable under monotonicity (Rashid et al., 2020b;a); QTRAN formalizes the individual-global-max (IGM) condition under which per-agent greedy execution equals joint greedy execution (Son et al., 2019). These works guarantee alignment by construction during training, at the price of a fixed task. Our Requirement 1 imports the IGM condition to composition time, where the values are no longer trained together, and our Lemma 3 shows that even when the condition holds, composition can still be unsafe because the per-agent values are stale, the failure that Requirement 2 names and that no training-time credit assignment mechanism addresses.

# B Single-Agent Foundations

This appendix recalls the single-agent facts that Section 2.1 uses and that the multi-agent proofs of Appendix C reduce to. Throughout, features are $\phi(s,a) \in \mathbb{R}^d$, a reward is a linear readout $r_w(s,a) = \phi(s,a)^\top w$, and the successor features of a policy $\pi$ are the discounted feature stream $\psi^\pi(s,a) = \mathbb{E}^\pi\Big[\sum_{t\geq 0} \gamma^t \phi(s_t,a_t) \mid s_0 = s, a_0 = a\Big]$.

**The pricing identity.** Because expectation is linear and the reward is linear in the features, the value of $\pi$ under any objective $w$ is a dot product against a single, objective-independent successor feature model:

$$V_w^\pi(s,a) \;=\; \mathbb{E}^\pi\Big[\sum_{t\geq 0} \gamma^t\, \phi(s_t,a_t)^\top w \,\Big|\, s_0 = s, a_0 = a\Big] \;=\; \psi^\pi(s,a)^\top w. \tag{13}$$

The identity is what lets one model, learned once, price a policy under every objective in the linear family without re-estimating a value function per objective (Barreto et al., 2017). It is the single-agent form of the team identity (3), and every appearance of $\psi^\top w$ in the main text is an instance of it.

**The generalized policy improvement guarantee.** Given a library $\{\pi^k\}_{k=1}^K$ with successor features $\{\psi^k\}$ and a test objective $w$, the GPI policy acts greedily with respect to the best library value at each state, $\pi^{\mathrm{gpi}}(s) \in \arg\max_a \max_k \psi^k(s,a)^\top w$.

**Proposition 7** (GPI improvement, Barreto et al., 2017; 2020)**.** *For every state $s$ and every library entry $k$, $V_w^{\pi^{\mathrm{gpi}}}(s) \geq V_w^{\pi^k}(s)$.*

*Proof.* Let $Q(s,a) = \max_k \psi^k(s,a)^\top w$. For any $k$, the greedy action satisfies $\max_a Q(s,a) \geq Q(s,\pi^k(s)) \geq \psi^k(s,\pi^k(s))^\top w = V_w^{\pi^k}(s)$, so $\pi^{\mathrm{gpi}}$ has one-step advantage at least that of $\pi^k$ at every state. The advantage telescopes: writing $T^{\mathrm{gpi}}$ for the Bellman operator of $\pi^{\mathrm{gpi}}$, $V_w^{\pi^k} \leq T^{\mathrm{gpi}} V_w^{\pi^k} \leq (T^{\mathrm{gpi}})^n V_w^{\pi^k} \to V_w^{\pi^{\mathrm{gpi}}}$ by monotonicity and the $\gamma$-contraction of $T^{\mathrm{gpi}}$. Proposition 1 in the main text is the synchronized-composition specialization of this argument to a team that switches together. □

**Why UVFA and USFA generalize across objectives.** Universal value function approximators condition the value directly on the goal, $V(s,g)$, and share parameters across goals, so that a value learned for one goal transfers by function approximation to nearby goals rather than being retrained from scratch (Schaul et al., 2015). Universal successor feature approximators keep the pricing identity (13) but make the successor feature model itself universal, $\tilde{\psi}(s,a,z,w)$, with a policy axis $z$ that indexes which library policy is being priced and a task axis $w$ that shapes the training behavior distribution and prices the result at test time (Borsa et al., 2018). The motivation is precisely the operating mode this paper needs: separating the policy axis from the task axis lets a single network be trained once over a distribution of objectives and then, at deployment, evaluate a dot product over a candidate set with no per-task adaptation, because the objective enters only through the linear factor $w$ and never through retraining. MA-USFA is the multi-agent counterpart: it keeps this policy-task factorization per agent and adds a teammate-context axis, so that each agent's successor features remain valid as the teammates' objectives change. Appendix G states the reuse precisely.

# C Proofs

## C.1 Proof of Proposition 1

Fix the library $\Pi$, the test objective $w_{\mathrm{test}}$, and an initial state $s_0$. Write $v^k(s) = V_{w_{\mathrm{test}}}^{\pi^k}(s)$ for the value of entry $k$ under the test objective, and let $k^*(s) \in \arg\max_k v^k(s)$. The synchronized rule (4) plays $\pi^{k^*(s)}(s)$ at state $s$, provided the score $\psi^k(s,\pi^k(s))^\top w_{\mathrm{test}}$ equals the true value $v^k(s)$, this is the validity premise of the proposition, and it holds exactly under the linear feature model. We prove $V_{w_{\mathrm{test}}}^{\pi^{\mathrm{sync}}}(s) \geq v^k(s)$ for every state $s$ and every entry $k$.

For a finite horizon $T$, let $V_t^\pi(s)$ denote the value of policy $\pi$ with $t$ steps remaining, and $V_t^{\pi^k}(s)$ likewise. We show by induction on $t$ that $V_t^{\pi^{\mathrm{sync}}}(s) \geq V_t^{\pi^k}(s)$ for every $s$ and $k$. The base case $t = 0$ is trivial. For the step, at state $s$ the rule plays $a^{k^*} = \pi^{k^*(s)}(s)$, and

$$V_{t+1}^{\pi^{\mathrm{sync}}}(s) = \mathbb{E}\Big[\phi(s,a^{k^*})^\top w_{\mathrm{test}} + \gamma\, V_t^{\pi^{\mathrm{sync}}}(s')\Big] \tag{14}$$

$$\geq \mathbb{E}\Big[\phi(s,a^{k^*})^\top w_{\mathrm{test}} + \gamma\, V_t^{\pi^{k^*}}(s')\Big] = V_{t+1}^{\pi^{k^*}}(s), \tag{15}$$

where the inequality applies the induction hypothesis at the successor state $s'$, and the equality uses that $\psi^{k^*}(s,a^{k^*})^\top w_{\mathrm{test}} = V^{\pi^{k^*}}(s)$ by validity. Since $k^*(s)$ maximizes $v^k(s)$ over the library, $V_{t+1}^{\pi^{k^*}}(s) = \max_k V_{t+1}^{\pi^k}(s) \geq V_{t+1}^{\pi^k}(s)$, which closes the induction. Taking $T \to \infty$ with $\gamma < 1$ gives the infinite-horizon statement by standard monotone convergence (the finite-horizon values are increasing in $T$ and bounded by the discounted feature sums). No assumption on the reward structure beyond validity, and none on the dynamics, is used: the argument never requires the transition kernel to factor or the features to decompose.

## C.2 Proof of Proposition 2

We show that Requirements 1 and 2 together imply the safety bound $V_{w_{\mathrm{test}}}^{\pi^{\mathrm{ind}}}(s) \geq \max_k V_{w_{\mathrm{test}}}^{\pi^k}(s)$ for every state. Consider the joint-GPI value of the library, $g_s(a) = \max_k \psi^k(s,a)^\top w_{\mathrm{test}}$, whose greedy

policy $\pi^{\text{gpi}}(s) \in \arg\max_a g_s(a)$ satisfies the single-agent guarantee $V^{\pi^{\text{gpi}}}_{w_{\text{test}}}(s) \geq \max_k V^{\pi^k}_{w_{\text{test}}}(s)$ (Proposition 7 of Appendix B, applied to the joint action space). It suffices to show that under the two requirements the independently composed policy $\pi^{\text{ind}}$ of (5) selects a joint action in $\arg\max_a g_s(a)$ at every state and is scored by valid values.

Requirement 2 states that each stored $\psi^k_i$ is the true successor feature of the composed joint policy, so every per-agent score $\psi^k_i(s, a_i)^\top w_i$ equals the corresponding component of the true joint value, and the joint score $\sum_i \psi^k_i(s, a_i)^\top w_i$ equals $\psi^k(s, a)^\top w_{\text{test}}$. The values driving (5) are therefore the same values that define $g_s$, with no staleness. Requirement 1 states that the joint maximizer of $g_s$ is reachable by per-agent greedy choices, $A^*(s) = \prod_i A^*_i(s)$, so the coordinate-wise maximizers selected by (5) form a joint action in $\arg\max_a g_s(a)$ (a consistent tie-break resolves ties within the product set). Hence $\pi^{\text{ind}}$ coincides with a joint-GPI policy scored by valid values, and the guarantee of Proposition 7 transfers to it. The two requirements are exactly the two premises the single-agent argument needs once it is executed by $N$ decentralized maximizers rather than one: validity restores the fixed-dynamics premise at composition time, and alignment restores the single-decision-maker premise.

### C.3 The counterexample of Lemma 3: full construction

**Task.** Two agents, two stages, deterministic dynamics. Each agent has its own two-dimensional feature stream, and its reward depends only on its own action, so the team reward is fully separable: $r_i(s, a_i) = \phi_i(s, a_i)^\top w$ with a shared weight $w = (9, 8)$. The two coupling between the agents lives entirely in the transition: the second-stage state is $T(a_1, a_2)$ with $T(0, 0) = T(1, 1) = 0$ and $T(0, 1) = T(1, 0) = 1$. The feature values are collected in Table 2.

Table 2: Feature values of the Lemma 3 construction. Agent $i$'s reward at the first stage is $\phi_i(a_i)^\top w$, and at the second stage $\phi_i(s_1, a_i)^\top w$; only the transition depends on both agents' actions.

| stage | agent 1 | agent 2 |
|---|---|---|
| $s_0$, action 0 | $(1, 3)$ | $(0, 7)$ |
| $s_0$, action 1 | $(0, 6)$ | $(6, 5)$ |
| $s_1 = 0$, action 0 | $(4, 0)$ | $(7, 7)$ |
| $s_1 = 0$, action 1 | $(4, 3)$ | $(4, 6)$ |
| $s_1 = 1$, action 0 | $(5, 7)$ | $(3, 4)$ |
| $s_1 = 1$, action 1 | $(3, 0)$ | $(2, 2)$ |

**Library.** The library holds two entries, both suboptimal under $w$ (the joint optimum is 321, computed below). Entry $\pi^0$ plays $(0, 1)$ at $s_0$ and $(0, 0)$ at the second stage in both states; entry $\pi^1$ plays $(0, 1)$ at $s_0$ and $(0, 1)$ or $(1, 0)$ at the second stage. Their values are

$$V^{\pi^0}(s_0) = \big[(1, 3) + (6, 5) + (5, 7) + (3, 4)\big] \cdot w = (15, 19) \cdot (9, 8) = 287, \tag{16}$$

$$V^{\pi^1}(s_0) = \big[(1, 3) + (6, 5) + (3, 0) + (3, 4)\big] \cdot w = (13, 12) \cdot (9, 8) = 213, \tag{17}$$

so the best library entry is worth 287; the second-stage terms follow from $T(0, 1) = 1$ and the entries' second-stage actions. Every library entry is strictly suboptimal: $287 < 321$.

**The independent composition.** For each agent we compute the per-agent snapshot value of each action, the expected feature sum when the agent takes the action while the teammate follows the entry and the team continues along the entry. At $s_0$, agent 1's values are 134 for $a_1 = 0$ (achieved through entry $\pi^0$: teammate plays 1, the team transitions to $s_1 = 1$ along $T(0, 1)$, and agent 1 plays 0 there, so $[(1, 3) + (5, 7)] \cdot w = 134$) and 84 for $a_1 = 1$; agent 2's values are 175 for $a_2 = 0$ (through $\pi^0$: teammate plays 0, the team transitions to $s_1 = 0$ along $T(0, 0)$, and agent 2 plays 0 there, so $[(0, 7) + (7, 7)] \cdot w = 175$) and 153 for $a_2 = 1$. Neither agent faces a tie. The independent rule therefore plays $(0, 0)$ at $s_0$. The transition sends the team to $s_1 = 0$, where the per-agent values are the immediate feature rewards alone, $(36, 60)$ for agent 1 and $(119, 84)$ for agent 2, and the rule plays $(1, 0)$. The composed trajectory delivers

$$\big[(1, 3) + (0, 7) + (4, 3) + (7, 7)\big] \cdot w = (12, 20) \cdot (9, 8) = 268 < 287 = \max_k V^{\pi^k}(s_0). \tag{18}$$

The delivered per-agent shares are 93 for agent 1 (rated 134) and 175 for agent 2 (rated 175).

**Mechanism.** Agent 1's choice was justified by a value of 134, the features it would accrue if agent 2 kept the library's action at $s_0$ and the team continued down the branch $s_1 = 1$ along entry $\pi^0$. Agent 2's independent choice flips the transition to $s_1 = 0$, and agent 1 delivers 93 instead: its snapshot value described a teammate that is not the one the composition pairs it with. Agent 2's rating happened to be accurate, because agent 1's actual choice matches what the rating assumed; a single stale rating suffices to drag the team 19 points below its own best library entry. Both agents acted exactly as their values told them to: the failure is not a selection error, it is a validity failure (Requirement 2). The centralized joint-GPI rule over the same library selects $(1, 1)$ at $s_0$ (rated 297 through entry $\pi^0$'s continuation) and $(1, 0)$ at $s_1 = 0$, and delivers $[(0, 6) + (6, 5) + (4, 3) + (7, 7)] \cdot w = (17, 21) \cdot (9, 8) = 321$, the joint optimum: centralization protects the values, because the joint successor features remain valid descriptions of the joint policies they were measured for. The failure is therefore specific to decentralization.

**Alignment holds.** The joint actions optimal for the test objective are a singleton at every state: at $s_0$ the unique optimal joint action is $(1, 1)$ (value 321); at $s_1 = 0$ it is $(1, 0)$ (value 179, against 155, 120 and 144 for the other three actions); at $s_1 = 1$ it is $(0, 0)$ (value 160, against 135, 86 and 61). Singletons are trivially product sets, so the alignment condition of Requirement 1 holds at every state: no alignment story can explain the failure. What fails is Requirement 2: the per-agent values are snapshot quantities, measured against a specific version of the teammates, and recomposition invalidates them.

**A companion instance: staleness without violation.** The same mechanism need not always produce a violation; it always voids the guarantee. In a two-stage companion construction with joint features, weight $w = (0.6, 0.4)$, and a three-entry library whose best entry is worth 4.2 (the other two worth 3.0 and 3.2), the alignment condition again holds at the test objective, yet the independent rule delivers 6.2: above every library entry, so the safety inequality of Definition 1 holds, but below the joint optimum 6.6, which the centralized joint-GPI rule attains. The values that drove the composition were still wrong: agent 2's choice was justified by a snapshot value of 6.6, the value it would have if agent 1 followed the third entry, and the composition delivers 6.2 because agent 1 does not. The composition is safe by the letter of the definition and invalid by the mechanism that produces it.

### C.4 Proof of Proposition 4

We prove the alignment claim that carries the proposition, then the value comparison. Throughout, $g_s(a) = \max_k \psi^k(s, a)^\top w_{\text{test}}$ is the joint GPI value at state $s$.

**Topkis lemma.** If $g_s$ is supermodular on the lattice $\{0, 1\}^N$, its argmax set is a sublattice. For $a, b \in \arg\max g_s$, supermodularity gives $g_s(a \vee b) + g_s(a \wedge b) \geq g_s(a) + g_s(b) = 2g_s^*$, where $g_s^*$ is the maximum; since neither term can exceed $g_s^*$, both must equal it, so $a \vee b$ and $a \wedge b$ are also argmaxes, which is the defining property of a sublattice (Topkis, 1998). The same conclusion holds when $g_s$ is supermodular on a rectangular sublattice containing all of its argmax, by restricting the argument to that sublattice.

**Alignment.** The per-agent rule maximizes, per agent, the marginal value $h_i(a_i) = \max_k \psi_i^k(s, a_i)^\top w_i$. When the joint value decomposes over agents, $\psi^k(s, a)^\top w = \sum_i \psi_i^k(s, a_i)^\top w_i$, which holds under per-agent additive features, the per-agent greedy choices are the coordinate-wise maximizers of the decomposed library, and two facts combine: the argmax of $g_s$ is a sublattice by the lemma, and per-agent greedy execution with a consistent tie-break rule (the same resolution criterion applied by every agent, for example a shared index over entries) selects a vector in that sublattice. The composed policy then executes the joint GPI rule at every state, so $V^{\pi^{\text{ind}}}(s) \geq \max_k V^{\pi^k}(s)$ by the standard generalized policy improvement argument (the same telescoping as in Appendix C above), and $\max_k V^{\pi^k}(s) = V^{\pi^{\text{sync}}}(s)$ by Proposition 1. This gives $V^{\pi^{\text{ind}}}(s) \geq V^{\pi^{\text{sync}}}(s)$.

In the general case without an additive decomposition, the per-agent marginals and the joint value are no longer the same objects, so the argument covers the additive (value-decomposed) regime, the setting in which the per-agent greedy choices realize the joint maximizer.

### C.5 Proof of Corollary 5

Let $\Delta\phi_d(a,b) = \phi_d(a \vee b) + \phi_d(a \wedge b) - \phi_d(a) - \phi_d(b)$ be the supermodularity gap of feature $d$ on the pair of joint actions $(a, b)$. Under the hypothesis, each library entry prices joint actions by the same feature-linear function up to an additive constant, $\psi^k(s,a)^\top w_{\text{test}} = \phi(a)^\top w_{\text{test}} + c_k(s)$ with $c_k(s)$ independent of $a$, since the per-pair differences $\psi^k(s,a)^\top w_{\text{test}} - \psi^k(s,b)^\top w_{\text{test}} = (\phi(a) - \phi(b))^\top w_{\text{test}}$ do not depend on $k$. Taking the library maximum, $g_s(a) = \phi(a)^\top w_{\text{test}} + \max_k c_k(s)$, so $g_s$ equals the feature-linear value up to a state-constant. The constant cancels in every second difference, so for every pair of joint actions

$$g_s(a \vee b) + g_s(a \wedge b) - g_s(a) - g_s(b) = \textstyle\sum_d w_d^{\text{test}}\, \Delta\phi_d(a,b). \tag{19}$$

Supermodularity of $g_s$ is nonnegativity of the left-hand side over all pairs, and membership $w_{\text{test}} \in K_\phi$ is nonnegativity of the right-hand side over all pairs. These are the same inequalities, so $g_s$ is supermodular if and only if $w_{\text{test}} \in K_\phi$, and Proposition 4 applies exactly on the cone. The reasoning is not the loose "maximum of supermodular functions is supermodular" (false in general). It is the constant offset that makes the library maximum feature-linear. The sufficient condition stated in the corollary, every feature $\phi_d$ itself supermodular and $w \geq 0$, is immediate: each term $w_d\, \Delta\phi_d(a,b)$ is then nonnegative.

### C.6 Proof of Proposition 6

Assume the transition kernel factorizes per agent, $s_i' = f_i(s_i, a_i)$, and suppose first that the features also decompose per agent, so the per-agent successor feature $\psi_i^k(s, a_i)$ is the discounted feature stream of agent $i$'s own dynamics. Because the dynamics of agent $i$ never depend on the teammates, $\psi_i^k$ remains a valid description of agent $i$'s feature stream under any recomposition: whoever the other agents are, agent $i$'s marginal process is the same function of its own actions. Requirement 2 therefore holds by construction, for every library and every recomposition, not only at the test objective. Requirement 1 is then governed by the structural condition of Proposition 4, and the value comparison follows as in that proposition: $V^{\pi^{\text{ind}}}(s) \geq \max_k V^{\pi^k}(s) = V^{\pi^{\text{sync}}}(s)$.

For coupled rewards, the cross-feature components of the per-agent values are measured against the library's teammate behavior, and under recomposition they are stale: the per-agent value is a marginalized quantity whose validity is no longer exact, so the guarantee need not extend beyond the decomposed-reward case. The channels are therefore not independent: factorization pays off exactly where the reward decomposes.

## D MA-USFA: Training and Deployment

This appendix gives the full training and deployment procedure of MA-USFA (Section 4), stated independently of any experimental domain. It collects the two-phase protocol, the pseudocode for both training and the deployment forward pass, and the two design choices that keep the composer small, and it closes by placing MA-USFA next to the two composition rules it is analyzed against.

### D.1 The two training phases

The two phases of the MA-USFA protocol are summarized in Table 3; the per-team-size budgets are reported with the controlled-domain setup (Appendix E). Phase 1 trains the lower per-agent value layer $\tilde{\psi}_i$ of (11), sampling tasks at training set (tasks from the homogeneous and heterogeneous scenarios) with the teammate-context conditioning, and freezes it; Phase 2 trains the upper-layer composer on top. Both phases build components of MA-USFA itself: the first produces the frozen value layer, and the second produces the composer stacked on top of it. The two phases correspond one-to-one with the two layers of Section 4.

### D.2 Pseudocode

Algorithm 1 states the training procedure and Algorithm 2 the deployment-time forward pass, making concrete the two-phase protocol of Section 4 and the notation of (11). Training produces two artifacts that are never adapted per objective: the frozen per-agent value layer $\tilde{\psi}_i(s, a_i \mid z_i, w_{-i})$

Table 3: The two-phase training protocol of MA-USFA. Each phase builds one layer of the method: Phase 1 the frozen per-agent value layer, Phase 2 the composer on top of it.

| Phase | Data | Object learned | Role |
|---|---|---|---|
| 1. context-conditioned value layer | $W_{\text{homo}} \cup W_{\text{hetero}}$ | $\tilde{\psi}_i(\cdot \mid z_i, w_{-i})$ | lower layer, Req. 2 conditioning |
| 2. learned composer | tasks needing correction | upper selectors $\Upsilon^\theta$ | upper layer, Lem. 3, Props. 4, 6 |

and the composer $\Upsilon^\theta = (\Upsilon_1^\theta, \ldots, \Upsilon_N^\theta)$. Deployment runs the composer on every test objective $w_{\text{test}} = (w_1, \ldots, w_N)$: because it is initialized at the independent transfer policy and trained only in value-increasing directions with the value layer frozen, on every objective it is at least as good as that policy and improves on it wherever a cross-agent correction helps. The composer is always the deciding rule; the fixed rules take no part in deployment.

**Algorithm 1** MA-USFA training (run once before deployment)

**Require:** feature map $\phi$; training tasks $W_{\text{train}} = W_{\text{homo}} \cup W_{\text{hetero}}$; correction tasks $W_{\text{corr}}$ (coupled dynamics or weights outside $K_\phi$); discount $\gamma$

**Phase 1: per-agent context-conditioned value layer** (lower layer, Eq. 11)

**for** each $w = (w_1, \ldots, w_N)$ sampled at random from $W_{\text{train}}$ **do**
  sample policy encodings $z_i \sim D_z(\cdot \mid w_i)$; form the teammate context $w_{-i} = (w_j)_{j \neq i}$
  update each $\tilde{\psi}_i(s, a_i \mid z_i, w_{-i})$ toward $\phi_i + \gamma\, \tilde{\psi}_i(s', a_i' \mid z_i, w_{-i})$
**end for**
freeze the value layer $\{\tilde{\psi}_i\}$

**Phase 2: learned composer** (upper layer, Lem. 3, Props. 4, 6)

initialize the composer head $\theta$ at zero ▷ training starts exactly at the transfer policy
**for** each $w \in W_{\text{corr}}$ (weights outside the cone $K_\phi$, or coupled dynamics) **do**
  price candidates $q_i(z_i) \leftarrow \tilde{\psi}_i(s, \cdot, z_i)^\top w_i$ for $z_i \in C_i$ using the frozen value layer
  select $g_i \leftarrow \Upsilon_i^\theta\big(s,\, w,\, \{q_i(z_i)\}\big)$ for all $i$; execute the joint action; observe reward
  update $\theta$ by per-agent temporal difference (value layer frozen)
**end for**
**return** frozen value layer $\{\tilde{\psi}_i\}$ and composer $\Upsilon^\theta$

**Algorithm 2** MA-USFA deployment (single forward pass, no gradient updates)

**Require:** test objective $w_{\text{test}} = (w_1, \ldots, w_N)$; frozen $\{\tilde{\psi}_i\}$; composer $\Upsilon^\theta$; candidate sets $\{C_i\}$

**for** each decision state $s$ **do**
  price candidates $q_i(z_i) \leftarrow \tilde{\psi}_i(s, \cdot, z_i)^\top w_i$ for every $z_i \in C_i$ ▷ one dot product per candidate
  $a_i \leftarrow$ action of entry $g_i = \Upsilon_i^\theta\big(s,\, w_{\text{test}},\, \{q_i(z_i)\}\big)$ for all $i$ ▷ composer selects on every objective
  execute the joint action $(a_1, \ldots, a_N)$
**end for**

**Implementation choices.** Three choices make the composer small and stable, and they are shared across both experimental domains. The composer head is initialized at zero, so training starts exactly at the transfer policy; the value layer is frozen while the composer trains, protecting the library from drift; and for large teams the composer attends over the interaction graph rather than the full joint state, a one-hop graph attention pass per selector (Veličković et al., 2018), which is what makes the 196-agent traffic network of Appendix F feasible. The first two hold in every run; the third is the neighbor-limited instantiation used at city scale (Appendix F).

**Extension: deployment without a composer.** One extension lies outside the method proper. If the composer cannot be trained at all, for instance under a learning budget too small to fit it, deployment can use a fixed rule in its place, and the analysis of Section 3 says which one: synchronized

composition by default, and the independent rule only where its two requirements hold. This chooses among the rules we analyze rather than the one we propose. MA-USFA itself needs no such choice.

### D.3 Composition rules studied as baselines

MA-USFA is analyzed against two fixed composition rules, which are objects of study rather than components of the method. Table 4 places the three side by side, listing their mechanism, the set of joint policies each can produce, and the safety status established in Section 3.

Table 4: The coupling spectrum of composition rules. Composition space is the set of joint policies a rule can produce. Safety is with respect to (6).

| Rule | Mechanism | Composition space | Safety |
|---|---|---|---|
| Sync | synchronized shared index | $K$ library entries | unconditional (Prop. 1) |
| Indep | independent per-agent argmax | product $K^N$ | conditional (Prop. 4, 6) |
| MA-USFA | learned composer | product $K^N$ | learned (Sec. 4) |

## E Controlled Domain: SFWorld

This appendix reports the controlled-domain experiment in full: the environment and setup, the crossover result behind the main-body table, and the team-size and library-coverage sweeps.

### E.1 Setup

**Environment.** SFWorld is a $5 \times 5$ grid with $N \in \{2, 3, 4, 5\}$ agents, five actions (four moves and stay), horizon 40, discount $\gamma = 0.95$. The features $\phi_i(s')$ of agent $i$ are dense smooth bumps over $K$ resource cells, shaped after the congestion signals of the traffic domain; the reward of agent $i$ is $r_i = \phi_i(s')^\top w_i - 2.0 \cdot \mathrm{blocked}_i$, a task weight times the features minus a collision penalty that lives outside the feature basis, so any closed-form pricing $\psi \cdot w$ is structurally blind to it. The coupling parameter $\kappa$ governs conflicts: when two agents target the same cell, the conflict blocks all but one of them with probability $\kappa$. At $\kappa = 0$ the dynamics factorize exactly, the free region of Proposition 6 is in force, and the collision rate is zero at every $\kappa$ on $A_{\mathrm{distinct}}$. As $\kappa$ grows, each agent's effective kernel depends on the teammates' actions, which is transition coupling. Two task families separate the channels (Fig. 2): $A_{\mathrm{distinct}}$, where the agents' optimal regions are disjoint so coupling never binds, and $B_{\mathrm{overlap}}$, where the agents contend for a shared region so the same weights create collisions whenever the joint policy hybridizes.

**Library.** The library holds $K$ policies specialized to corner objectives. The main configuration uses the four corners of the weight simplex ($K = 4$); the library-content sweep (Fig. 5) varies both the size and the content: the four corners alone (P4), the corners plus the contested joint action $(0, 0)$ (P4+(0,0)), plus both $(0, 0)$ and $(1, 1)$ (P4+(0,0)+(1,1)), the corners plus four copies of one entry (P4+4 same), and libraries built around $(0, 0)$ alone or with one or two additional entries. The team-size sweep uses $K = 4$ for $N \leq 4$ and $K = 8$ for $N = 5$.

**Learning.** Successor features are trained with the USFA protocol: the TD target is the feature vector $\phi$, the weight plays the three roles of behavior anchor, library anchor ($z \sim D_z(\cdot \mid w)$), and test-time pricing vector. Training budgets scale with the team size: 8,000 episodes for $N = 2$, 6,000 for $N = 3$, 5,000 for $N = 4$, 4,000 for $N = 5$; the composer is trained for 1,000 episodes at $N = 2$, 2,000 at $N = 3$, 1,600 at $N = 4$, 1,400 at $N = 5$; per-task retraining runs 4,000 episodes. Evaluation averages 120 rollouts per configuration. The composer is warm-started from the transfer policy (zero-initialized correction head) and trained with per-agent temporal-difference updates with the feature backbone frozen; of the three composer instantiations, an attention head, a message-passing head, and a QMIX-style mixing head, the three variants are interchangeable in this domain because the optimal deferral is static, so the paper reports the best of the three. The coupling sweep uses $\kappa \in \{0, 0.25, 0.5, 0.75, 1.0\}$ in the main matrix and $\kappa \in \{0, 0.5, 1.0\}$ in the team-size sweep.

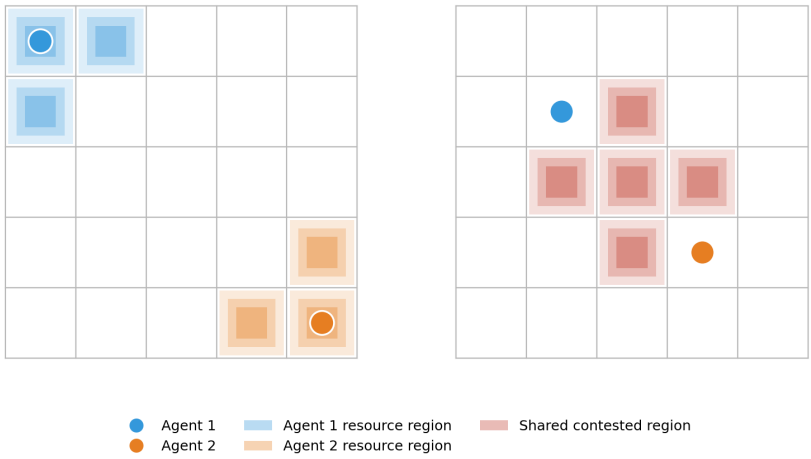



Figure 2: The SFWorld scene and its two task families, shown for $N = 2$ on the $5 \times 5$ grid. Each agent earns reward by occupying cells in its own resource region, whose value under a task is $\phi_i(s')^\top w_i$, and a collision penalty is charged whenever two agents target the same cell. Left ($A_{\text{distinct}}$): the two resource regions are disjoint. Right ($B_{\text{overlap}}$): a single contested region is valuable to both agents. The coupling probability $\kappa$ scales how often a shared-cell conflict blocks an agent.

**Baselines.** The synchronized rule is the composition rule of (4); the independent rule is the composition rule of (5); joint-GPI is the centralized generalized policy improvement rule over the joint action space, representing prior work on multi-agent GPI (computed only for $N \leq 3$, where the joint space can be enumerated; the joint-GPI entries for $N \geq 4$ in Table 5 are missing by design); per-task retraining optimizes the penalty-inclusive reward from scratch and is the oracle ceiling. All closed-form operators use successor features trained on the harvest features only, as theory prescribes; the collision penalty is deliberately outside the feature basis so that no closed-form rule can price it. The two-phase training protocol of MA-USFA is given in Appendix D (Table 3); the phase budgets per team size are those reported under Learning above.

### E.2 The crossover

Table 1 in the body reports the $N = 2$ main matrix. Fig. 3 plots its $B_{\text{overlap}}$ block, so the crossover of the two fixed rules and the tracking of MA-USFA against retraining are visible as curves in the coupling parameter $\kappa$.

### E.3 Team-size sweep

Table 5 and Fig. 4 sweep the team from $N = 2$ to 5 on both task families at $\kappa \in \{0, 0.5, 1.0\}$. The paid-region collapse of the independent rule deepens sharply with team size, from 7.07 at $N = 2$ to $-37.00$ at $\kappa = 1.0$ as its collision rate approaches 0.6. The synchronized rule stays flat and safe at every size, and MA-USFA recovers the retraining level at every $N$ and $\kappa$. On $A_{\text{distinct}}$ the collision rate is zero at all sizes and the independent rule is free, the null control of the design.

### E.4 Library-coverage sweep

Table 6 and Fig. 5 vary the library content on $B_{\text{overlap}}$ ($N = 2$), from a single entry to the corner library augmented with the contested joint action $(0, 0)$, and show that the value of coverage is regime-dependent. The mechanism is that Proposition 1 assumes the anchor price $\psi^k(s, a)^\top w_{\text{test}}$ equals the true value of entry $k$ under $w_{\text{test}}$, which is exact inside the linear feature model but becomes a transfer-blind estimate once the deployed reward carries a component outside the feature basis. In the free region ($\kappa = 0$) adding the contested vertex is pure gain, lifting the synchronized rule from 31.52 to 38.72, near the joint optimum; in the paid region ($\kappa = 1.0$) the same addition locks the team onto the entry the anchor overprices and crashes it to 7.08 while the library still contains entries

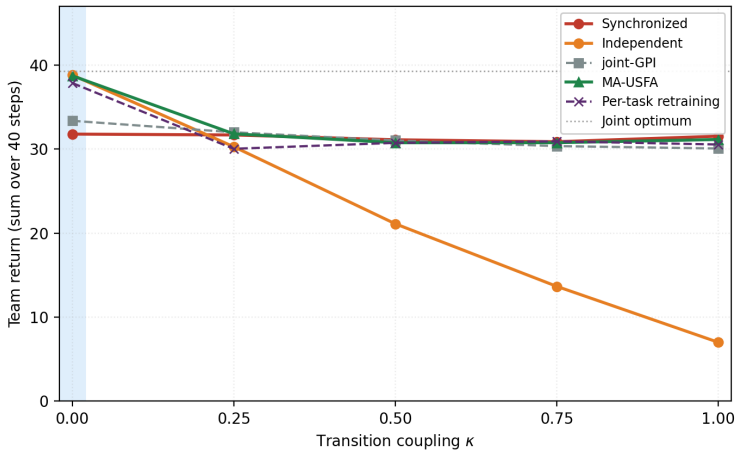



Figure 3: Crossover experiment on $B_{\text{overlap}}$ ($N = 2$, four-entry corner library), the $B_{\text{overlap}}$ block of Table 1 in visual form. Curves show team return (summed over the 40-step horizon) of the three composition rules of Section 2 (Synchronized, Independent, MA-USFA), the joint-GPI baseline of prior work, and per-task retraining, as the transition coupling $\kappa$ (the probability that two agents targeting the same cell collide) increases from 0 to 1.

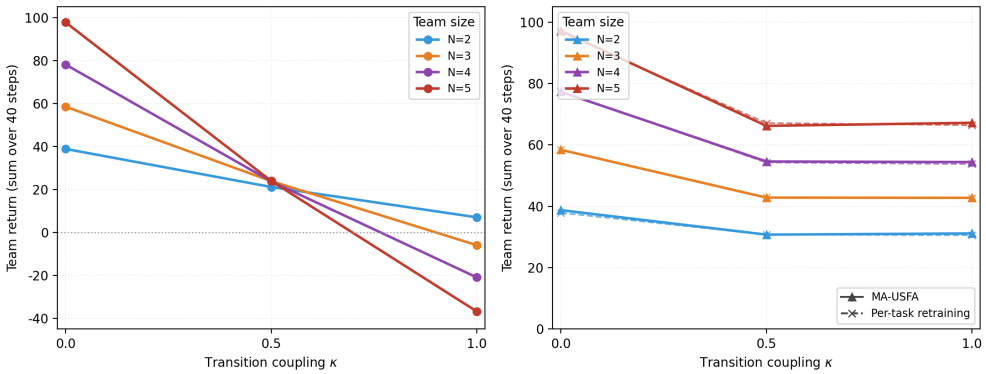



Figure 4: Team-size sweep on $B_{\text{overlap}}$ for $N = 2$ to 5 at $\kappa \in \{0, 0.5, 1\}$, where $\kappa$ is the collision probability and team return is summed over the 40-step horizon. Left: team return of independent composition, one curve per team size $N$. Right: MA-USFA (solid) and per-task retraining (dashed), one color per $N$. Synchronized composition is omitted.

worth 31.24, a signed gap of $-24.16$, and enlarging the library further does not cure it. The learned composer is immune to library content, returning the same 31.18 across every row of the paid region.

**The synchronized anchor must be valid.** Proposition 1 assumes that $\psi^k(s,a)^\top w_{\text{test}}$ equals the true value of entry $k$ under $w_{\text{test}}$; inside the linear feature model this is exact. When the deployed reward contains a component outside the feature basis, the estimate becomes a transfer-blind anchor that prices only what the features saw during training. Table 6 and Fig. 5 show the consequence: at $\kappa = 1$ the synchronized rule locks the team onto the contested entry $(0, 0)$ and falls to 7.08 while the library still contains entries worth 31.24, a signed gap of $-24.16$, and enlarging the library does not cure it, because the contested entry is exactly the one the anchor overprices. The learned composer is immune to library content (31.18 across every row of the paid region).

Table 5: Team-size sweep, corner library ($K = 4$ for $N \leq 4$, $K = 8$ for $N = 5$), all three coupling levels. Team return over 40 steps, mean $\pm$ standard deviation over three evaluation seeds. The joint-GPI baseline is omitted ($-$) for $N \geq 4$. The Average row under each team size gives each method's mean over the six task and $\kappa$ settings at that $N$.

| $N$ | task | $\kappa$ | Sync | Indep | joint-GPI | MA-USFA | retrain | $\text{coll}_{\text{ind}}$ |
|---|---|---|---|---|---|---|---|---|
| 2 | $A_{\text{distinct}}$ | 0.00 | 23.42±0.34 | **38.73**±0.47 | 23.74±0.29 | 38.40±0.53 | 38.69±0.48 | 0.000 |
| 2 | $A_{\text{distinct}}$ | 0.50 | 23.37±0.16 | 38.23±0.01 | 23.63±0.14 | 38.17±0.22 | **38.35**±0.10 | 0.000 |
| 2 | $A_{\text{distinct}}$ | 1.00 | 23.44±0.15 | 38.31±0.27 | 23.80±0.23 | 38.25±0.18 | **38.48**±0.16 | 0.000 |
| 2 | $B_{\text{overlap}}$ | 0.00 | 31.52±0.26 | **38.77**±0.14 | 32.59±0.20 | 38.44±0.13 | 37.56±0.25 | 0.000 |
| 2 | $B_{\text{overlap}}$ | 0.50 | **31.27**±0.09 | 21.04±0.94 | 30.65±0.14 | 30.94±0.30 | 31.08±0.25 | 0.225 |
| 2 | $B_{\text{overlap}}$ | 1.00 | **31.24**±0.16 | 7.07±0.20 | 28.62±0.60 | 31.18±0.18 | 30.33±0.31 | 0.369 |
| Average, $N$=2 | | | 27.38 | 30.36 | 27.17 | **35.90** | 35.75 | 0.099 |
| 3 | $A_{\text{distinct}}$ | 0.00 | 26.14±0.19 | **58.22**±0.48 | 27.70±0.21 | 57.98±0.51 | 58.14±0.49 | 0.000 |
| 3 | $A_{\text{distinct}}$ | 0.50 | 25.79±0.06 | 57.35±0.35 | 25.61±0.17 | 57.07±0.43 | **57.52**±0.11 | 0.000 |
| 3 | $A_{\text{distinct}}$ | 1.00 | 25.47±0.16 | 56.74±0.81 | 25.13±0.91 | 57.54±0.65 | **57.68**±0.19 | 0.000 |
| 3 | $B_{\text{overlap}}$ | 0.00 | 43.64±0.29 | **58.14**±0.38 | 48.23±0.07 | 57.91±0.37 | 57.85±0.36 | 0.000 |
| 3 | $B_{\text{overlap}}$ | 0.50 | 43.19±0.29 | 22.38±0.83 | 38.92±0.12 | **43.60**±0.21 | 42.86±0.24 | 0.329 |
| 3 | $B_{\text{overlap}}$ | 1.00 | 42.07±0.18 | −6.52±0.40 | 31.80±1.10 | **42.63**±0.14 | 42.22±0.17 | 0.504 |
| Average, $N$=3 | | | 34.38 | 41.05 | 32.90 | **52.79** | 52.71 | 0.139 |
| 4 | $A_{\text{distinct}}$ | 0.00 | 29.05±0.29 | **77.43**±0.45 | — | 77.05±0.49 | 77.29±0.46 | 0.000 |
| 4 | $A_{\text{distinct}}$ | 0.50 | 28.51±0.07 | **76.52**±0.12 | — | 75.93±0.35 | 76.46±0.73 | 0.000 |
| 4 | $A_{\text{distinct}}$ | 1.00 | 27.64±0.35 | 75.19±1.40 | — | 75.99±0.72 | **76.40**±0.86 | 0.000 |
| 4 | $B_{\text{overlap}}$ | 0.00 | 55.84±0.49 | **77.78**±0.40 | — | 77.31±0.41 | 77.48±0.39 | 0.000 |
| 4 | $B_{\text{overlap}}$ | 0.50 | 55.00±0.22 | 23.08±0.26 | — | **55.03**±0.09 | 54.40±0.60 | 0.349 |
| 4 | $B_{\text{overlap}}$ | 1.00 | 53.43±0.30 | −21.48±0.06 | — | **54.52**±0.69 | 53.79±0.44 | 0.556 |
| Average, $N$=4 | | | 41.58 | 51.42 | — | **69.31** | 69.30 | 0.151 |
| 5 | $A_{\text{distinct}}$ | 0.00 | 49.89±0.22 | **98.75**±0.41 | — | 98.22±0.42 | 98.67±0.40 | 0.000 |
| 5 | $A_{\text{distinct}}$ | 0.50 | 48.72±0.42 | 96.34±0.47 | — | 96.78±0.32 | **97.45**±0.20 | 0.000 |
| 5 | $A_{\text{distinct}}$ | 1.00 | 47.62±0.29 | 92.86±0.75 | — | 95.94±1.00 | **96.40**±1.44 | 0.003 |
| 5 | $B_{\text{overlap}}$ | 0.00 | 50.42±0.15 | **97.51**±0.40 | — | 96.70±0.51 | 96.99±0.43 | 0.000 |
| 5 | $B_{\text{overlap}}$ | 0.50 | 48.94±0.22 | 23.38±1.20 | — | 67.23±0.14 | **67.44**±0.29 | 0.364 |
| 5 | $B_{\text{overlap}}$ | 1.00 | 47.29±0.18 | −37.00±0.31 | — | **66.34**±0.35 | 66.11±0.14 | 0.601 |
| Average, $N$=5 | | | 48.81 | 61.97 | — | 86.87 | **87.18** | 0.161 |

# F City-Scale Domain: Traffic Signal Control

This appendix reports the traffic-signal experiment in full, in the same order as the controlled domain: the environment and setup, the main result, and the robustness sweeps.

## F.1 Setup

**Network and coupling.** The experiment runs on a Manhattan grid of $28 \times 7$ intersections, 196 signal agents, one per intersection. Road cells are shared between adjacent intersections, so spillback couples the state of each intersection to the decisions of its upstream neighbors: the transition coupling is endogenous, with no free parameter to control, and the factorization condition of Proposition 6 fails by construction. The feature model is the project's composite traffic reward, which is already of the form $\phi^\top w$: $\phi = [\text{queue}, \text{wait}, \text{pressure}, \text{speed}]_{\text{norm}}$, four normalized congestion features, so the composite reward is literally a successor-feature target and its weight vector is the task. All four features keep a fixed physical sign (queue, wait, and pressure are costs, speed is a gain); a task is a choice of the relative magnitudes on these four terms.

**Homogeneous and heterogeneous tasks.** The two tasks differ only in whether the target weight is shared across the city or varies by intersection. In the *homogeneous* task every intersection optimizes one shared weight vector, so the 196 agents pursue a single common goal. In the *heterogeneous* task each intersection draws its own weight independently, keeping the fixed signs but sampling the

Table 6: Library-content sweep ($N = 2$, $B_{\text{overlap}}$). Team return over 40 steps, mean $\pm$ standard deviation over three evaluation seeds. P4 is the four-entry corner library; the other rows add or replace entries around the contested joint action $(0, 0)$. The final row averages each method over all library variants and coupling levels.

| library | $\kappa$ | Sync | Indep | MA-USFA | retrain | $\text{coll}_{\text{ind}}$ | $\max_k V^k$ | Sync $-$ $\max_k V^k$ |
|---|---|---|---|---|---|---|---|---|
| $(0, 0)$ only | 0.00 | 38.72±0.16 | **38.77**±0.14 | 38.19±0.09 | 37.56±0.25 | 0.000 | 38.72 | +0.00 |
| $(0, 0)$ only | 0.50 | 21.18±0.34 | 21.04±0.94 | 28.34±0.42 | **31.08**±0.25 | 0.225 | 21.18 | +0.00 |
| $(0, 0)$ only | 1.00 | 7.08±0.05 | 7.07±0.20 | **31.18**±0.18 | 30.33±0.31 | 0.369 | 7.08 | +0.00 |
| $(0, 0) + (1, 2)$ | 0.00 | 38.72±0.16 | **38.77**±0.14 | 38.19±0.09 | 37.56±0.25 | 0.000 | 38.72 | +0.00 |
| $(0, 0) + (1, 2)$ | 0.50 | 21.18±0.34 | 21.04±0.94 | 28.34±0.42 | **31.08**±0.25 | 0.225 | 21.18 | +0.00 |
| $(0, 0) + (1, 2)$ | 1.00 | 7.08±0.05 | 7.07±0.20 | **31.18**±0.18 | 30.33±0.31 | 0.369 | 7.08 | +0.00 |
| $(0, 0) + (1, 2) + (2, 1)$ | 0.00 | 38.72±0.16 | **38.77**±0.14 | 38.19±0.09 | 37.56±0.25 | 0.000 | 38.72 | +0.00 |
| $(0, 0) + (1, 2) + (2, 1)$ | 0.50 | 21.18±0.34 | 21.04±0.94 | 28.34±0.42 | **31.08**±0.25 | 0.225 | 21.18 | +0.00 |
| $(0, 0) + (1, 2) + (2, 1)$ | 1.00 | 7.08±0.05 | 7.07±0.20 | **31.18**±0.18 | 30.33±0.31 | 0.369 | 14.83 | −7.75 |
| P4 corners | 0.00 | 31.52±0.26 | **38.77**±0.14 | 38.19±0.09 | 37.56±0.25 | 0.000 | 31.52 | +0.00 |
| P4 corners | 0.50 | **31.27**±0.09 | 21.04±0.94 | 28.34±0.42 | 31.08±0.25 | 0.225 | 31.27 | +0.00 |
| P4 corners | 1.00 | **31.24**±0.16 | 7.07±0.20 | 31.18±0.18 | 30.33±0.31 | 0.369 | 31.24 | +0.00 |
| P4+(0,0) | 0.00 | 38.72±0.16 | **38.77**±0.14 | 38.19±0.09 | 37.56±0.25 | 0.000 | 38.72 | +0.00 |
| P4+(0,0) | 0.50 | 21.18±0.34 | 21.04±0.94 | 28.34±0.42 | **31.08**±0.25 | 0.225 | 31.27 | −10.10 |
| P4+(0,0) | 1.00 | 7.08±0.05 | 7.07±0.20 | **31.18**±0.18 | 30.33±0.31 | 0.369 | 31.24 | −24.16 |
| P4+(0,0)+(1,1) | 0.00 | 38.72±0.16 | **38.77**±0.14 | 38.19±0.09 | 37.56±0.25 | 0.000 | 38.72 | +0.00 |
| P4+(0,0)+(1,1) | 0.50 | 21.18±0.34 | 21.04±0.94 | 28.34±0.42 | **31.08**±0.25 | 0.225 | 31.27 | −10.10 |
| P4+(0,0)+(1,1) | 1.00 | 7.08±0.05 | 7.07±0.20 | **31.18**±0.18 | 30.33±0.31 | 0.369 | 31.24 | −24.16 |
| P4+4 same | 0.00 | 38.72±0.16 | **38.77**±0.14 | 38.19±0.09 | 37.56±0.25 | 0.000 | 38.72 | +0.00 |
| P4+4 same | 0.50 | 21.18±0.34 | 21.04±0.94 | 28.34±0.42 | **31.08**±0.25 | 0.225 | 31.27 | −10.10 |
| P4+4 same | 1.00 | 7.08±0.05 | 7.07±0.20 | **31.18**±0.18 | 30.33±0.31 | 0.369 | 31.24 | −24.16 |
| Average (all rows) | | 23.61 | 22.29 | 32.57 | **32.99** | 0.198 | — | −5.26 |

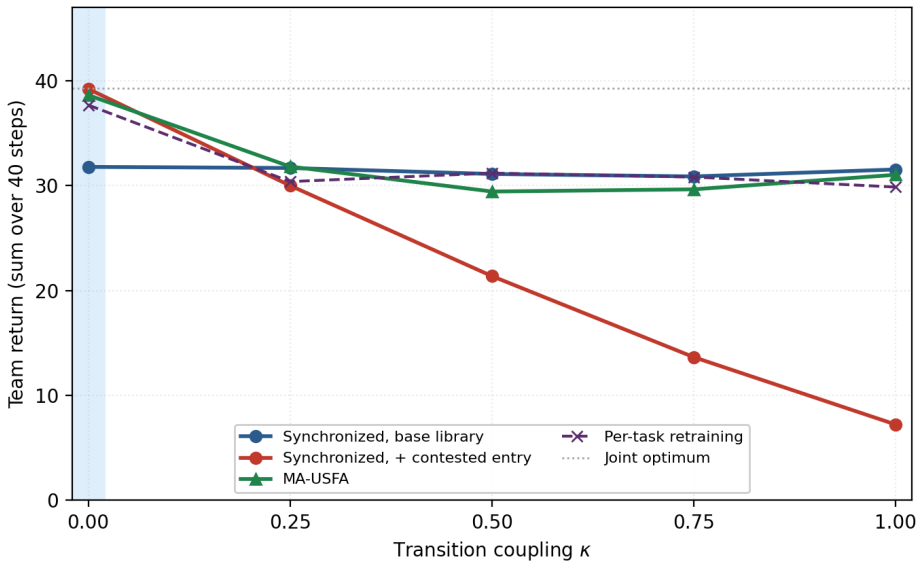



Figure 5: Library-content sweep on $B_{\text{overlap}}$ ($N = 2$). Team return (summed over the 40-step horizon) versus transition coupling $\kappa$ for synchronized composition under two libraries, the base four-entry corner library and the same library with the contested joint entry $(0, 0)$ added, compared against MA-USFA and per-task retraining.

four magnitudes uniformly in $[0.5, 1.5]$ times their defaults, so different corners of the city weigh queue, waiting, pressure, and speed differently at the same moment. The closed-form transfer must then serve every node's own $w$ simultaneously, and the composer can additionally condition on its neighbors' weights. The reported metrics are the task return $\sum_i \phi_i^\top w_i$, mean travel time over completed trips, throughput (completed trips), mean queue, mean waiting time, and mean speed.

**Composer.** The composer is the neighbor-limited instantiation of Section 4: a one-hop graph attention selector per intersection, $u_i = \psi_i^\top w_i + \rho \tanh(\mathrm{head}(\mathrm{GAT}_i(\mathrm{context})))$ with $\rho = 4.0$, zero-initialized so that training starts exactly at the transfer policy, with the feature backbone frozen. Global information reaches each decision only through direct neighbors, the strict locality constraint a 196-agent network imposes, following the neighbor-limited cooperative pattern of CoLight (Wei et al., 2019). The composer budget is 200 episodes, short relative to the toy but expensive per episode at this scale.

### F.2 Main result

**Results.** Table 7 reports the full numbers, and the picture is consistent across both task families: MA-USFA is best on every metric, and its margins over the two fixed rules are an order of magnitude larger than the gap between those rules, whose fine ordering we do not claim. On the homogeneous task the composer reaches return 2165 against 2009 (independent) and 1993 (synchronized), throughput 518 against 368 and 356, and mean waiting 6.60 s against 10.35 and 11.95 s. On the heterogeneous task it reaches return 1650 against 1515 and 1527, throughput 483 against 337 and 352, and waiting 5.48 s against 8.57 and 8.50 s. One entry needs reading with care: the composer's homogeneous travel time (288.2 s, the longest in its block) is a measurement artifact rather than a regression, since travel time is averaged over completed trips and the composer clears roughly 150 extra long-distance trips that the fixed rules leave stranded, which lengthens the mean it is credited with.

Table 7: Traffic signal control on the Manhattan $28 \times 7$ network (196 intersections), on the homogeneous and heterogeneous tasks. Return is the task objective $\sum_i \phi_i^\top w_i$ summed over the horizon; the remaining columns are physical traffic metrics.

| Task | Method | Return | Travel (s) | Throughput | Queue | Wait (s) | Speed (m/s) |
|---|---|---|---|---|---|---|---|
| Homo | Sync (anchor) | 1993 | 280.1 | 356 | 3.13 | 11.95 | 6.26 |
| Homo | Indep (transfer) | 2009 | **274.7** | 368 | 2.94 | 10.35 | 6.36 |
| Homo | MA-USFA (composer) | **2165** | 288.2 | **518** | **2.54** | **6.60** | **6.96** |
| Hetero | Indep (transfer) | 1515 | 273.1 | 337 | 2.81 | 8.57 | 6.49 |
| Hetero | Sync (anchor) | 1527 | 276.2 | 352 | 2.73 | 8.50 | 6.52 |
| Hetero | MA-USFA (composer) | **1650** | **270.2** | **483** | **2.27** | **5.48** | **6.98** |

### F.3 Robustness sweeps

Two sweeps on the homogeneous task support the headline numbers (Table 8). The composer already clears both fixed rules after 100 fine-tuning episodes (return 2156, throughput 456), and further episodes buy only small refinements (2165 at 200, 2174 at 500), so the 200-episode budget used above sits near the plateau rather than being a compute-heavy outlier. Enlarging the synchronized rule's source library from one to five entries does not help it and eventually hurts: return holds at 2032 for $K \in \{1, 2, 3\}$, dips to 2028 at $K = 4$, and falls to 1993 at $K = 5$, because added specialists rarely become the winning shared anchor and the last, contested entry drags the team down in the coupled regime, the same coverage liability the controlled domain isolates (Appendix E). The independent rule (2009) sits between the small and large synchronized libraries, and the composer clears every configuration.

## G Discussions

The proposed MA-USFA sits inside a lineage whose generality we rely on at three points, and this appendix states the facts precisely. Universal value function approximators condition the value model directly on the goal, $V(s, g)$, and interpolate across goals by function approximation (Schaul et al., 2015). Universal successor feature approximators combine the two ideas in a model $\tilde{\psi}(s, a, z, w)$ with a policy axis $z$ and a task axis $w$: $z$ indexes the policies in the library, and $w$ shapes the behavior distribution during training and prices the library at test time through a dot product (Borsa et al., 2018). With a single goal as the candidate set, USFA reduces to UVFA; with a finite library it reduces to successor features with generalized policy improvement. The operating mode we import into a team

Table 8: Supplementary traffic sweeps on the homogeneous task. Top: the MA-USFA composer as a function of the fine-tuning budget in episodes. Bottom: the synchronized rule as a function of source-library size $K$, with the independent rule listed for reference.

| Sweep | Setting | Return | Travel (s) | Throughput | Queue | Wait (s) | Speed (m/s) |
|---|---|---|---|---|---|---|---|
| Composer budget | MA-USFA, 100 ep. | 2156 | 268.2 | 456 | 2.71 | 6.73 | 6.90 |
| Composer budget | MA-USFA, 200 ep. | 2165 | 288.2 | 518 | 2.54 | 6.60 | 6.96 |
| Composer budget | MA-USFA, 500 ep. | 2174 | 277.5 | 493 | 2.40 | 6.70 | 6.94 |
| Library size | Sync, $K = 1$ | 2032 | 275.4 | 357 | 2.75 | 8.69 | 6.52 |
| Library size | Sync, $K = 3$ | 2032 | 275.4 | 357 | 2.75 | 8.69 | 6.52 |
| Library size | Sync, $K = 4$ | 2028 | 272.2 | 345 | 2.79 | 8.85 | 6.51 |
| Library size | Sync, $K = 5$ | 1993 | 280.1 | 356 | 3.13 | 11.95 | 6.26 |
| Library size | Indep (reference) | 2009 | 274.7 | 368 | 2.94 | 10.35 | 6.36 |

is the one USFA was designed for: train once over a distribution of objectives, and at deployment evaluate dot products only, with no per-task adaptation.

MA-USFA uses the same machinery per agent. Agent $i$'s value model $\tilde{\psi}_i(s, a_i \mid z_i, w_{-i})$ keeps $z_i$ as its own policy axis, whose self-referential TD target $a'_i = \arg\max_b \tilde{\psi}_i(s', b, z_i)^\top z_i$ has as its fixed point the optimal policy for the task $z_i$. The task axis is carried by the weights, linear in the value, appearing only at test time. The conditioning on the teammates' weights $w_{-i}$ is the new ingredient: each teammate's weight indexes a cluster of teammate policies, so the model learns the expected successor features over the teammate behavior distribution rather than a snapshot against one fixed version of the teammates, which is the architectural answer to Requirement 2 of Section 3.3. Two structural facts make the reuse exact rather than approximate. First, when the features are per-agent additive, the joint successor feature of a synchronized entry decomposes as $\psi^{\mathrm{joint},k} = \sum_i \psi_i^k$, so the same network serves both the synchronized anchor and the per-agent library. The sum is the value the synchronized rule prices. Second, the composer is initialized at the independent rule with the value layer frozen, and its correction is trained only to raise team value, so on every objective MA-USFA is at least as good as per-agent USFA and coincides with it wherever the independent rule is already optimal, without any runtime test.