STATIONARY ALONE, DRIFTING TOGETHER: ADAPTING TO INTERACTION-MEDIATED NONSTATIONARITY IN MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

External conditions can change how agents interact through shared infrastructure without changing their dynamics in isolation. This drift persists with frozen partners and vanishes for a lone agent. We formalise this interaction-mediated nonstationarity as a family of partially observable Markov games indexed by severity and introduce three public benchmarks for routing, voltage control and cooperative lifting, preserving their original rewards. We propose PACT (Peer-Action Coupling Tracking), a wrapper for unmodified multi-agent learners. Each agent predicts disturbances from peers' broadcast actions and the infrastructure layout using an online model whose size is independent of fleet size. A trust level controls compensation or steering, with zero trust recovering the host exactly. In a linearised feeder, predictive compensation with exact estimates raises the instability threshold without bound as trust approaches its maximum. In a routing model, excessive trust causes oscillations that make the route predicted to be faster become slower. Experiments against representative baselines demonstrate consistent improvements over the host learners across all benchmarks.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) is increasingly used to control fleets that share physical infrastructure (Zhang et al., 2021): connected vehicles routing through a city (Akman et al., 2025), photovoltaic inverters holding the voltage of a distribution feeder (Wang et al., 2021), and actuators lifting a common load (Bettini et al., 2022). Such agents interact through a shared medium (a road link, a feeder segment, a supply rail), and how strongly they interact depends on conditions that none of them controls. Rain lowers road capacity, so the same traffic produces more delay (Transportation Research Board, 2016; Agarwal et al., 2005); the midday sun pushes inverters onto the sloped part of their Volt/VAr curves (IEEE, 2018), so the same reactive power injections provoke stronger reactions from neighbours, a coupling that can drive a feeder into oscillation (Jahangiri & Aliprantis, 2013; Farivar et al., 2013; Andrén et al., 2015). Apart from any effect that a lone agent would also feel, each agent in isolation faces a stationary task, while the fleet as a whole faces a drifting one.

We call this *interaction-mediated nonstationarity*. It vanishes for a lone agent because the external condition never acts on an agent's own dynamics; it only rescales the coupling *between* agents, and without peers there is nothing for it to act on. Yet it persists when every other agent's policy is frozen, because a frozen fleet still loads the shared medium and the condition still acts on that load.

Three kinds of drift. Write s_i for the part of the state belonging to agent i. Its evolution over one step separates into private dynamics and coupling,

$$s_i(t{+}1) = F_i\big(s_i(t), a_i(t)\big) \;\oplus\; C_i[\boldsymbol{\theta}(t)]\big(s_{-i}(t), \mathbf{a}_{-i}(t);\, s_i(t)\big), \tag{1}$$

where F_i describes the agent in isolation, C_i describes how the others affect it, $\boldsymbol{\theta}(t)$ collects the *gains* of that coupling (what a unit of the others' activity costs agent i), and $\oplus$ combines the two additively, multiplicatively or, when the coupling acts on a realised outcome such as a travel time, through that

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.

outcome. Nonstationarity arises when one of the three ingredients of Eq. (1) varies over time:

$$\begin{aligned} &\text{(A) } \textbf{learning-induced:} && \text{the distribution of } \mathbf{a}_{-i}(t) \text{ changes as other agents learn;} \\ &\text{(B) } \textbf{exogenous:} && F_i \text{ changes with } t\text{, and an isolated agent feels it;} \\ &\text{(C) } \textbf{interaction-mediated:} && \boldsymbol{\theta}(t) \text{ changes with } t\text{, while } F_i \text{ is fixed.} \end{aligned} \tag{2}$$

MARL studies (A): the environment changes as co-learners update their policies (Hernandez-Leal et al., 2017; Papoudakis et al., 2019), and the effect vanishes once partners are frozen. Nonstationary and continual RL study (B): the kernel or reward drifts independently of the agent (Padakandla, 2021; Khetarpal et al., 2022), and an isolated agent is affected. (C) is neither: it persists under frozen partners, because the drift lives in $\boldsymbol{\theta}(t)$, and vanishes for an agent alone on the medium, where $C_i \equiv 0$. These two testable properties separate it from (A) and (B). Viewed as one joint system, (C) is exogenous drift of the kernel, but with a specific structure. The drift enters only through the coupling and scales with the fleet's own load, so it can be identified from the peers' actions when these vary enough. Although the external condition is exogenous, the load it acts on comes from the fleet itself. Each agent's compensating action therefore adds to the load its peers face, and such uncoordinated reactions can destabilise the physical system (Andrén et al., 2015).

Why existing methods struggle. Learners that replay old data (Lowe et al., 2017; Rashid et al., 2018; Liu et al., 2024) mix transitions from different coupling gains, and PPO learners (Yu et al., 2022; de Witt et al., 2020) must track the moving gains through updates or memory alone. Methods that condition on context, task representations or neighbours (Hamadanian et al., 2025; Wang et al., 2026; Yang et al., 2018; Papoudakis et al., 2021) can represent the change but infer its effect indirectly, from rewards or reconstructed trajectories, rather than measuring it. Robust methods and change detectors (Tobin et al., 2017; Tiboni et al., 2024; Bukharin et al., 2023; Gerogiannis et al., 2025) hedge against change or detect it without estimating what changed. Disturbance observers (Han, 2009; Łakomy et al., 2021) infer a disturbance from its effect on the output, so their estimates lag behind it.

Our approach. PACT identifies the moving coupling explicitly. The declared coupling structure gives each agent r channels, one per coupling class, that summarise its peers' load; only the gains on these channels change with the external condition. Each agent fits these gains and an intercept, $r+1$ coefficients in total, by recursive least squares with forgetting (Ljung, 1999; Åström & Wittenmark, 2008), regressing its measured performance loss on channels built from its peers' broadcast actions. The model size therefore depends on neither n nor the size of the medium. Where the harm is invertible, PACT cancels the predicted disturbance; otherwise it biases the host policy toward actions predicted to incur less harm. A trust level $g \in [0, 1]$, reduced when the estimate is uncertain, scales the correction, and $g = 0$ returns the host action exactly (Proposition 4.2).

Contributions.

1. **A new problem class.** We define interaction-mediated nonstationarity through four declared objects (Section 3), separate it from learning-induced and exogenous drift by two testable properties, and instantiate it on three public benchmarks, taking the coupling geometry from each benchmark's own layout (Section 3.1).
2. **PACT**, a decentralised wrapper for unmodified MARL learners that identifies the moving coupling and compensates for it or steers around it (Section 4).
3. **Theory** (Section C). The coupling model is exact below saturation for the feeder and lift, and classical tracking and performance difference arguments bound the loss of return by the compensation residual. Our main results characterise the feedback loop that compensation creates: on a linearised feeder, exact predictive compensation with trust g raises the fleet's instability threshold by $1/(1-g)$, while in a two-route congestion model trust beyond a critical level makes the fleet oscillate, so that the route predicted to be faster ends up slower.

2 Related Work

Drift from co-learners (A). Centralised training with decentralised execution (Lowe et al., 2017; Rashid et al., 2018), actor-critic methods based on trust regions (HATRPO) or maximum entropy

(HASAC) (Kuba et al., 2022; Liu et al., 2024), and PPO with centralised (MAPPO, R-MAPPO) or local (IPPO) critics (Yu et al., 2022; de Witt et al., 2020) stabilise learning among co-learners. MFAC conditions each agent's value on its neighbours' mean action (Yang et al., 2018), and LIAM encodes a model of other agents from local observations (Papoudakis et al., 2021); both represent peers but learn how to use that representation only through the return. **Exogenous drift (B).** LCPO prevents forgetting by anchoring the policy on past contexts that differ from the current one (Hamadanian et al., 2025); WISDOM tracks and predicts task evolution with wavelet task representations (Wang et al., 2026); in piecewise stationary bandits, restarting after quickest change detection is more effective in practice than the prior-free black-box approach, whose tests rarely fire at practical horizons (Gerogiannis et al., 2025); and, on the model-based side, M3W shares a world model built from a mixture of experts across labelled tasks (Zhao et al., 2025). Black-box reductions carry restart-based learning to nonstationary Markov games (Jiang et al., 2024), and hidden-parameter, contextual and exogenous-state MDPs index the kernel by an unobserved variable (Doshi-Velez & Konidaris, 2016; Hallak et al., 2015; Dietterich et al., 2018); in the class studied here, that variable scales only the coupling between agents. These methods adapt to whatever moves; none exploits a declared coupling geometry or a direct identification signal. **Robustness.** Randomising the rendering or the dynamics (Tobin et al., 2017; Peng et al., 2018), maximising the entropy of the dynamics randomisation (DORAEMON) (Tiboni et al., 2024) and adversarial Lipschitz regularisation (ERNIE) (Bukharin et al., 2023) hedge over a set of perturbations instead of identifying the one that is actually present. **Control and games.** Extended state observers (ESO) in active disturbance rejection control (Han, 2009; Łakomy et al., 2021) estimate a lumped disturbance from the measured output and cancel it; PACT instead predicts the disturbance from its cause, the peers' loads, before it arrives, using recursive least squares with forgetting (Ljung, 1999; Åström & Wittenmark, 2008). DEDA-FP learns time-dependent equilibria of continuous mean field games (Magnino et al., 2025), and IPGA studies independent learning when the deployed policy reshapes the game (Sahitaj et al., 2025), extending performative prediction (Perdomo et al., 2020). Here, by contrast, an external condition sets the gains of the coupling. Our routing result is related to the overreaction of drivers to shared route guidance (Ben-Akiva et al., 1991; Mahmassani & Jayakrishnan, 1991) and to the informational Braess paradox (Acemoglu et al., 2018).

3 Problem Setting

We start from a partially observable Markov game $\mathcal{G}_0 = (\mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}, \{\Omega_i\}, P_0, R, \gamma)$ with $n = |\mathcal{N}|$ agents, observation spaces Ω_i, kernel P_0, reward R and discount γ (Oliehoek & Amato, 2016).

Definition 3.1 (Interaction-mediated harm)**.** Four objects are declared. **(M)** A shared medium, used by the agents and possibly by uncontrolled participants whose stationary behaviour is part of the environment, with loads $\phi_j(a_j)$ measuring how much user j uses it. **(W)** A fixed coupling operator W computed from the infrastructure layout. **(D)** An exogenous driver $A(t) = A(\chi(t)) \in [0, 1]$, from no effect (0) to full effect (1), where the phase $\chi(t) \in [0, 1)$ advances deterministically. **(H)** A harm map $h(\xi, \mathbf{x}; a)$, where $\xi \geq 0$ is the driver's current intensity, with $h(0, \mathbf{x}; a) = h(\xi, \mathbf{0}; a) = a$. At severity $\sigma \geq 0$, agent i's intended action $a_i(t)$ is executed as

$$\begin{aligned} a_i^{\mathrm{ex}}(t) &= h\big(\sigma A(t),\, \mathbf{x}_i(t);\, a_i(t)\big), \\ \mathbf{x}_i(t) &= \Lambda_i\big(\mathbf{x}_i(t-1),\, \{\phi_j(a_j^{\mathrm{ex}}(t-1))\}_{j\neq i};\, W\big) \in \mathbb{R}^r, \end{aligned} \tag{3}$$

where Λ_i aggregates the loads of the other users $j \neq i$ through W into r channels, one per coupling class, possibly filtered over time, and $\mathbf{x}_i \equiv \mathbf{0}$ when agent i is alone on the medium.

In the instances of Section 3.1, (M, W, D, H) are (road links, link sharing, rain, delay), (feeder lines, voltage sensitivities, sun, reactive response) and (supply rail, actuator proximity, wet/dry cycle, force loss). When the harm acts on a realised outcome instead of the action (routing), h is applied to agent i's outcome and $\mathbf{x}_i(t)$ is formed from the loads of the same step, which are realised before the outcome. Since the routing harm of Eq. (6) depends on the peer volumes on every link of the route, $\mathbf{x}_i$ is then the vector of these volumes, and the r-channel model of Section 4 is a surrogate for it (Proposition C.2(ii)). The reward function is unchanged in both cases. For harm on actions, the harmed kernel on the augmented state $\tilde{s} = (s, \mathbf{x}, \chi)$, with $\mathbf{x}$ and χ updated deterministically, is

$$P_\sigma\big(s' \mid \tilde{s}, \mathbf{a}\big) = P_0\big(s' \mid s,\, h(\sigma A(\chi), \mathbf{x}; \mathbf{a})\big), \tag{4}$$

with h applied agent by agent. The harmed game is stationary on $\tilde{s}$ (the driver depends only on the phase χ, which $\tilde{s}$ contains) and equals $\mathcal{G}_0$ at $\sigma = 0$, and an agent alone on the medium ($\mathbf{x}_i \equiv \mathbf{0}$) is unaffected at every σ. The agents, however, observe neither χ nor the gains that turn the channels $\mathbf{x}_i$ into harm, and must identify them online. We write $J_\sigma(\pi)$ for the expected discounted return of a policy π in this game, so $J_0(\pi)$ is its nominal return.

3.1 Three Instances

Each benchmark is unchanged at $\sigma = 0$; raising σ strengthens an exogenous condition whose effect scales with the agents' current load on the medium. Each harm is modelled on the benchmark's physics. Below saturation, the feeder and lift harms lie inside PACT's model class (Proposition C.2), whereas the routing harm (a maximum and a ratio) lies outside it. We write a_i^{sent} for the action sent to the environment (the intended action of Definition 3.1), which equals the host's command a_i^{cmd} unless PACT modifies it, and a_i^{ex} for the executed action. Parameter values are given in Section A.

URB: routing under precipitation. In URB (Akman et al., 2025), connected autonomous vehicles choose routes through urban road networks simulated in SUMO (Lopez et al., 2018), alongside human traffic that no agent controls. Agent i chooses a route a_i, and agents are coupled through shared links. Precipitation leaves link e a fraction $\omega_e(t)$ of its capacity, and agent i's load is the largest ratio of peer volume to capacity along its route,

$$u_i(\mathbf{a}; \boldsymbol{\omega}) = \max_{e \in a_i} \frac{\sum_{j \neq i} O_{ij}\, \mathbf{1}[e \in a_j]}{T_i\, c_e\, \omega_e}, \qquad \omega_e(t) = \text{clip}\left(1 - \sigma L A(t) \eta_e,\ \underline{\omega},\ 1\right), \tag{5}$$

with temporal overlap weights O_{ij}, window length T_i, capacity c_e, severity scale L, rain sensitivity η_e and floor $\underline{\omega}$. The harm, with slope $\alpha_h > 0$, scales the travel time τ_i^{sim} returned by the simulator,

$$\tau_i = h_i\, \tau_i^{\text{sim}}, \qquad h_i = \frac{1 + \alpha_h\, u_i(\mathbf{a}; \boldsymbol{\omega}(t))}{1 + \alpha_h\, u_i(\mathbf{a}; \mathbf{1})}, \tag{6}$$

so $h_i \geq 1$ is the delay factor under rain relative to dry weather; it equals 1 at $\sigma = 0$ or when no other vehicle shares the route. A delay cannot be undone once the route is chosen, so PACT instead steers the route choice.

MAPDN: Volt/VAr interaction on a feeder. In MAPDN (Wang et al., 2021), photovoltaic inverters on a radial distribution feeder regulate voltage through their reactive power q_i. Because the inverters share feeder segments, each injection changes the voltage seen by the others. With the linearised DistFlow sensitivity (Baran & Wu, 1989) aggregated over conductor classes m,

$$S_m[i,j] = \sum_{\substack{b \in \text{path}(i) \cap \text{path}(j) \\ \text{cls}(b) = m}} \frac{X_b}{V_b^2}, \qquad S_m[i,i] = 0, \qquad x_{m,i}(t) = \sum_j S_m[i,j]\, q_j^{\text{ex}}(t-1), \tag{7}$$

where $\text{path}(i)$ is the set of branches between the substation and inverter i, X_b is the reactance and V_b the nominal voltage of branch b. Thus $S_m[i,j]$ measures how strongly reactive power at j moves the voltage at i through the class m lines on their shared path. Each inverter reacts to the voltage change its neighbours cause, with a Volt/VAr slope that the driver scales,

$$d_i(t) = \bar{s}_i\, \text{clip}\Big(-\sigma k A(t) \sum\nolimits_m \text{send}_m\, x_{m,i}(t),\ -q_{\text{sat}},\ q_{\text{sat}}\Big), \qquad q_i^{\text{ex}} = \text{clip}\left(q_i^{\text{sent}} + d_i,\ -\bar{q}_i,\ \bar{q}_i\right), \tag{8}$$

with reaction gain $k > 0$, class gains send_m, saturation q_{sat} in units of rating, rating $\bar{s}_i$, and capability $\bar{q}_i = (\bar{s}_i^2 - (p_i^{\text{pv}})^2)^{1/2}$ at active power p_i^{pv}. The distortion adds to the action, so PACT can cancel it by subtraction.

VMAS: cooperative lifting under supply rail droop. In the VMAS Balance scenario (Bettini et al., 2022), agents lift a line carrying a package to a goal without dropping it. We add a shared supply rail: when several actuators push at once, each delivers less force, and the size of this effect follows the periodic wet/dry driver. With positions pos_i, hardware classes $\text{cls}(i)$, receive and send

gains recv and send_m, length scale λ, smoothing $\zeta \in [0,1)$ and forces a_i, the channel of class m at agent i is

$$W_{ij} = \frac{\mathrm{recv}_{\mathrm{cls}(i)}}{1 + (\|\mathrm{pos}_i - \mathrm{pos}_j\|/\lambda)^2}, \qquad W_{ii} = 0,$$
$$x_{m,i}(t) = \zeta\, x_{m,i}(t{-}1) + (1-\zeta) \sum_{j \neq i:\, \mathrm{cls}(j)=m} W_{ij}\, \|a_j^{\mathrm{ex}}(t{-}1)\|, \tag{9}$$

and the rail droop scales down the realised force,

$$\delta_i(t) = \mathrm{clip}\Big(\frac{\sigma L A(t)}{x_{\mathrm{ref}}} \sum\nolimits_m \mathrm{send}_m\, x_{m,i}(t),\, 0,\, \delta_{\max}\Big), \qquad a_i^{\mathrm{ex}} = \mathrm{clip}_{\mathrm{box}}\big((1-\delta_i)\, a_i^{\mathrm{sent}}\big), \tag{10}$$

with severity scale L, reference load $x_{\mathrm{ref}} > 0$, $\delta_{\max} < 1$, and $\mathrm{clip}_{\mathrm{box}}$ the clip to the action box. The distortion is multiplicative and invertible.

4 PACT: DECENTRALISED IDENTIFICATION OF THE COUPLING

PACT wraps an unmodified host learner and changes only what is sent to the environment. Each agent uses only its own measured outcome, the executed actions its peers broadcast, and the declared operator W.

Coupling representation. Agent i forms the regressor $\boldsymbol{\psi}_i(t) = [1,\ \tilde{\mathbf{x}}_i(t)^\top]^\top \in \mathbb{R}^{r+1}$, where $\tilde{\mathbf{x}}_i$ is the channel vector $\mathbf{x}_i$ of Eq. (3) centred and scaled by an invertible affine map (Section B.1), and models its relative excess y_i (Eq. (12)) as

$$y_i(t) = \boldsymbol{\beta}_i^\star(t)^\top \boldsymbol{\psi}_i(t) + e_i(t). \tag{11}$$

Here $\boldsymbol{\beta}_i^\star(t)$ is the image of the coupling gains $\boldsymbol{\theta}(t)$ of Eq. (1) under the normalisation (the constant 1 absorbs the centring), and e_i collects what the model does not explain. The number of unknowns, $r{+}1$, is fixed by the coupling classes and does not grow with n or with the size of the medium. Below saturation the model is exact on the feeder and the lift, with $e_i \equiv 0$ and gains common to all agents, and on routing it is exact for an additive surrogate of the travel time, whose deviations from the benchmark harm enter e_i (Proposition C.2).

Online identification. The regression target is the agent's relative excess,

$$y_i = \mathrm{clip}(\bar{y}_i, -1, y_{\mathrm{clip}}), \qquad \bar{y}_i = \begin{cases} \tau_i/\mathrm{fft}_{a_i} - 1 & \text{(routing)},\\ (q_i^{\mathrm{ex}} - q_i^{\mathrm{sent}})/\bar{s}_i & \text{(feeder)},\\ 1 - \|a_i^{\mathrm{ex}}\|/\|a_i^{\mathrm{sent}}\| & \text{(lift, used when } a_i^{\mathrm{sent}} \neq 0). \end{cases} \tag{12}$$

Here $y_{\mathrm{clip}} > 0$ bounds outliers. This signal measures the agent's degradation directly at every step, whereas the reward reveals it only through the return. Because it is measured against the sent action, compensation does not bias it below saturation. Each agent runs recursive least squares with forgetting factor $\mu \in (0,1]$ on the pair from the previous step, $(\boldsymbol{\psi}, y) \equiv (\boldsymbol{\psi}_i(t{-}1), y_i(t{-}1))$, with gain $\mathbf{k}_i$ and covariance P_i started from $\hat{\boldsymbol{\beta}}_i(0)$ and $P_i(0) = p_0 I$:

$$\mathbf{k}_i(t) = \frac{P_i(t{-}1)\boldsymbol{\psi}}{\mu + \boldsymbol{\psi}^\top P_i(t{-}1)\boldsymbol{\psi}}, \qquad P_i(t) = \frac{P_i(t{-}1) - \mathbf{k}_i(t)\boldsymbol{\psi}^\top P_i(t{-}1)}{\mu},$$
$$\hat{\boldsymbol{\beta}}_i(t) = \hat{\boldsymbol{\beta}}_i(t{-}1) + \mathbf{k}_i(t)\big(y - \boldsymbol{\psi}^\top \hat{\boldsymbol{\beta}}_i(t{-}1)\big). \tag{13}$$

Forgetting discounts data gathered under past gains, so the estimate tracks the coupling instead of averaging across incompatible conditions. When the channels excite every direction of $\boldsymbol{\beta}_i^\star$, the tracking error splits into a drift term, a bias term for what the model leaves out and a noise term (Theorem C.4, after Guo & Ljung 1995); a smaller μ follows faster drift at the price of more noise.

Confidence and intervention. The estimate is used only as far as its prediction is reliable:

$$g_i = g_i^{\mathrm{pol}}\, \mathrm{conf}_i, \qquad \mathrm{conf}_i = \frac{1}{1 + \nu\, \boldsymbol{\psi}_i^\top P_i \boldsymbol{\psi}_i / (p_0 \|\boldsymbol{\psi}_i\|^2)} \in (0,1], \tag{14}$$

where $g_i^{\mathrm{pol}} \in [0,1]$ is the trust level and $\nu \geq 0$ sets how strongly uncertainty discounts it; conf_i is close to one when past data cover the current direction $\boldsymbol{\psi}_i$, and falls when $\boldsymbol{\psi}_i$ points where the data are sparse, since forgetting then inflates P_i. With the predicted excess $\hat{y}_i = \hat{\boldsymbol{\beta}}_i^\top \boldsymbol{\psi}_i$, the additive harm of the feeder and the multiplicative harm of the lift are compensated by

$$q_i^{\mathrm{sent}} = q_i^{\mathrm{cmd}} - g_i\, \bar{s}_i\, \hat{y}_i, \qquad a_i^{\mathrm{sent}} = \frac{a_i^{\mathrm{cmd}}}{1 - \mathrm{clip}(g_i \hat{y}_i,\, 0,\, \delta_{\max})}. \tag{15}$$

When no inverse exists, as in routing, the estimate steers the host policy toward lower predicted cost. For each candidate route k, with regressor $\boldsymbol{\psi}_{i,k}$ built from the routes the peers chose at the previous step,

$$\hat{\tau}_k = \mathrm{fft}_k\big(1 + \hat{\boldsymbol{\beta}}_i^\top \boldsymbol{\psi}_{i,k}\big), \qquad \tilde{\ell}_k = \ell_k - g_i \kappa z_k, \qquad z_k = \frac{\hat{\tau}_k - \overline{\hat{\tau}}}{\mathrm{sd}(\hat{\tau})}, \tag{16}$$

where ℓ_k are the host logits, $\kappa > 0$ is a gain, sd is the population standard deviation over candidates, and $z_k \equiv 0$ when all predictions coincide; standardising makes the shift independent of the units and scale of the predicted times. For steering, the confidence in g_i is that of Eq. (14) averaged over the candidates' regressors $\boldsymbol{\psi}_{i,k}$.

Theorem 4.1 (Compensation residual)**.** Fix an agent and a step below saturation, and let $y_i^\star = \boldsymbol{\beta}_i^{\star\top} \boldsymbol{\psi}_i$, which there equals the true excess $\bar{y}_i$ (Proposition C.2). **(i) Additive** (feeder, in units of rating $a_i = q_i / \bar{s}_i$):

$$a_i^{\mathrm{ex}} = a_i^{\mathrm{cmd}} + \epsilon_i, \qquad \epsilon_i = (1 - g_i)\, y_i^\star + g_i \big(\boldsymbol{\beta}_i^\star - \hat{\boldsymbol{\beta}}_i\big)^\top \boldsymbol{\psi}_i,$$
$$|\epsilon_i| \leq (1 - g_i)\, |y_i^\star| + g_i \left\|\boldsymbol{\beta}_i^\star - \hat{\boldsymbol{\beta}}_i\right\| \|\boldsymbol{\psi}_i\|.$$

(ii) Multiplicative (lift, $y_i^\star = \delta_i \in [0, \delta_{\max}]$): with $\hat{c}_i = \mathrm{clip}(g_i \hat{y}_i, 0, \delta_{\max})$,

$$a_i^{\mathrm{ex}} = \frac{1 - \delta_i}{1 - \hat{c}_i}\, a_i^{\mathrm{cmd}}, \qquad \left\|a_i^{\mathrm{ex}} - a_i^{\mathrm{cmd}}\right\| \leq \frac{(1 - g_i)\, \delta_i + g_i |\hat{y}_i - \delta_i|}{1 - \delta_{\max}} \left\|a_i^{\mathrm{cmd}}\right\|.$$

In both cases, $g_i = 1$ with an exact estimate gives $a_i^{\mathrm{ex}} = a_i^{\mathrm{cmd}}$; if this holds for every agent, the harmed step has the nominal kernel P_0.

The first term is the harm that trust leaves uncompensated and the second the estimation error that compensation injects; trust should therefore grow as the estimate improves, which is what conf_i in Eq. (14) implements. For harm on actions, the performance difference lemma (Kakade & Langford, 2002), applied to the executed action, bounds the loss of return by the discounted expected size of the joint residual $\mathbf{a}^{\mathrm{ex}} - \mathbf{a}^{\mathrm{cmd}}$ (Theorem C.9). The loss of return is thus governed by what identification and compensation leave behind, and an untrusted estimate is never imposed on the learner:

Proposition 4.2 (Containment of the host)**.** If $g_i = 0$, then $q_i^{\mathrm{sent}} = q_i^{\mathrm{cmd}}$, $a_i^{\mathrm{sent}} = a_i^{\mathrm{cmd}}$ and $\tilde{\ell} = \ell$ for every estimator state. Hence the PACT policy class contains every host policy π, and the best trust schedule attains at least the host's return $J_\sigma(\pi)$ in the harmed game.

5 EXPERIMENTS

We ask how much the harm costs unmodified learners, how much of that loss PACT removes from the host it wraps, and whether methods built for nonstationarity or robustness remove it too. The harm is applied at $\sigma = 3$, and PACT (Table 2) is compared with the same learner and seeds at $g \equiv 0$, its host (Proposition 4.2). All methods, including every host, baseline and PACT variant, are run with 5 seeds; tables report the mean and the standard deviation over seeds, and figures the mean and one standard deviation. Baselines take one recent representative per class (Section D and Table 8): trust regions (HAPPO, HASAC), memory (R-IPPO, R-MAPPO, the host with R-MAPPO's GRU run over days), graphs, mean fields and agent models (DGN, MF-Q, MF-AC, LIAM), context, latent-task and restart methods (LCPO, the host given the driver, or *oracle driver*, which appends $A(t)$ to every observation, LILAC, WISDOM, QCD restart), online identification (RMA), robustness (DR, randomisation over σ; DORAEMON, ERNIE), model-based MARL (M3W), deep fictitious play and performative learning (Deep FP, PMPG), residual estimators without the declared basis (ESO, a disturbance observer driven by the agent's own realised-minus-free-flow residual; URLS, recursive

Table 1: Mean travel time t_{CAV} of the controlled vehicles on URB `saint_arnoult`, lower is better, with the standard deviation over seeds in parentheses. **(a)** Hosts, the benchmark's greedy heuristic and PACT on the stock task and at $\sigma = 3$; Δ is the relative increase, and PACT with $g \equiv 0$ is its host (Proposition 4.2). **(b)** Baselines at $\sigma = 3$, grouped by what they add to their learner (Section D) and sorted within groups. The human travel time is 4.40 in every method, so the advantage ratio exceeds one throughout and is omitted.

(a) Hosts and PACT

Method	$\sigma = 0$	$\sigma = 3$	Δ (%)
Greedy	**3.042**	**3.044**	**0.1**
IQL	3.119 (0.021)	3.160 (0.020)	1.3
IPPO	3.377 (0.058)	3.471 (0.057)	2.8
VDN	3.380 (0.208)	3.496 (0.336)	3.4
MAPPO	3.659 (0.099)	3.799 (0.102)	3.8
QMIX	3.668 (0.198)	3.739 (0.278)	1.9
PACT, $g \equiv 0$	3.300 (0.045)	3.401 (0.042)	3.1
PACT, learned g	3.288 (0.024)	3.397 (0.033)	3.3
PACT, fixed g	3.276 (0.025)	3.369 (0.021)	2.8

(b) Baselines at $\sigma = 3$

Own outcomes		*Driver and robustness*	
URLS	3.443 (0.031)	Oracle driver	3.533 (0.047)
ESO	3.444 (0.046)	DR	3.554 (0.042)
R-IPPO	3.444 (0.029)	ERNIE	3.557 (0.055)
WISDOM	3.447 (0.042)	DORAEMON	3.572 (0.026)
Peers and population		RMA	3.587 (0.073)
Deep FP	3.458 (0.096)	LCPO	3.685 (0.058)
M3W	3.465 (0.091)	*Co-learners and restarts*	
MF-Q	3.471 (0.088)	HAPPO	3.515 (0.019)
DGN	3.560 (0.192)	QCD restart	3.576 (0.024)
LIAM	3.569 (0.091)	PMPG	3.651 (0.028)

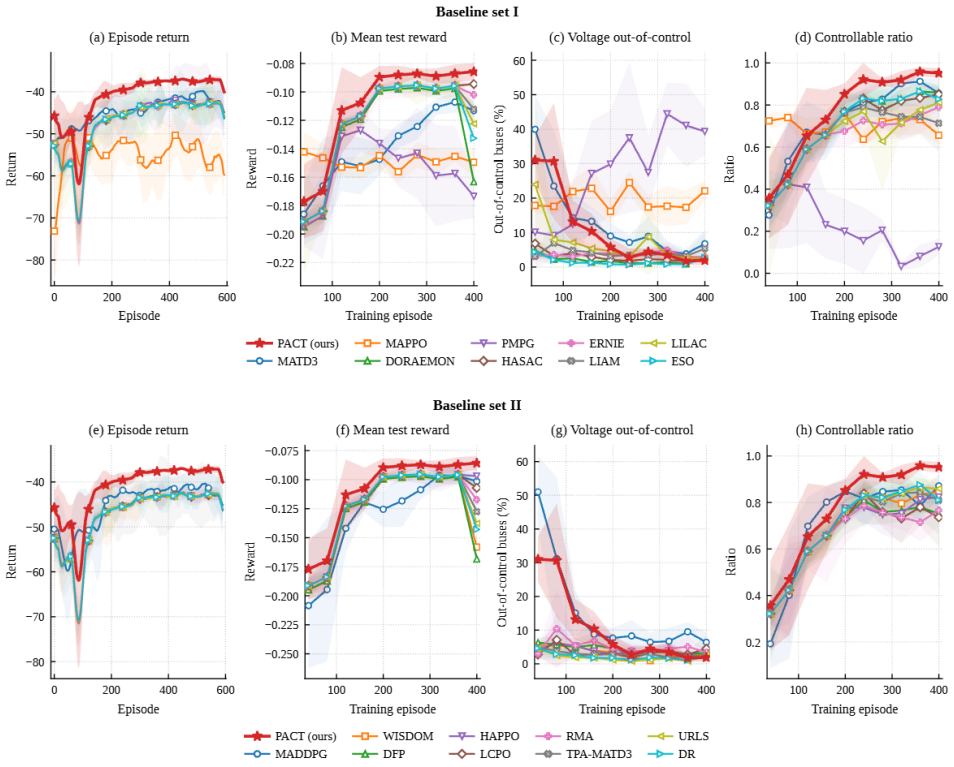

Figure 1: MAPDN `case141` at $\sigma = 3$: training return (smoothed over a 30-episode cycle) and, every 40 episodes, test reward, share of buses outside the voltage band and controllable ratio; mean $\pm$ one standard deviation.

least squares on the raw actions of broadcasting peers) and, on MAPDN, the state-of-the-art TPA (Xu et al., 2024).

5.1 ROUTING ON URB

All methods train for 4000 days on `saint_arnoult` (Section A) under the 100-day wet/dry driver and are evaluated on held-out days (Table 1). The harm is mild: it slows the hosts by 1.3% to

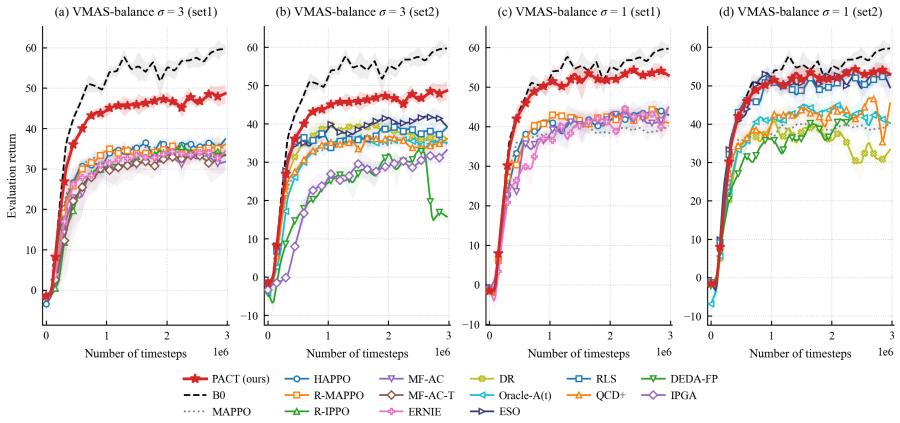

Figure 2: Evaluation return on VMAS `balance` at $\sigma = 3$ over training.

3.8%, whereas they differ by 18% on the stock task, and the greedy heuristic, which routes from a centralised record of the fleet (Akman et al., 2025), loses 0.1%, so route choice avoids almost all of the harm.

PACT and its host. With fixed trust, PACT lowers its host's travel time from 3.401 to 3.369, below every baseline in Table 1(b), and halves its dispersion across seeds. Of this 0.032, 0.024 also appears at $\sigma = 0$, as the steering of Eq. (16) favours candidates with short free-flow times, so the drift-specific share is small on URB.

Baselines. The four best, URLS, ESO, R-IPPO and WISDOM, share no mechanism yet lie within 0.004 of each other: each learns from the agent's own outcomes and is therefore subject to the floor this history sets (Lemma D.1(i)). The remaining baselines, which add peer inputs, the driver without the load, a range of severities, or co-learner corrections and restarts, are slower than the base learner they extend, IQL or IPPO, by up to 0.40; only Deep FP matches IPPO.

5.2 VOLTAGE CONTROL ON MAPDN

Twenty-two inverters regulate the 141-bus feeder of Section A under the solar driver (Fig. 1); PACT wraps MATD3 with fixed trust $g^{\mathrm{pol}} = 0.9$. The domain metric is the controllable ratio (CR), the fraction of steps on which every bus lies in $[0.95, 1.05]$ pu (Section B.2).

PACT and its host. PACT holds CR above 0.90 from episode 240 on and ends at 0.95 (0.02) against 0.86 (0.07) for its host; over the last three evaluations it leads every arm, 0.94 against at most 0.89. It improves on its host in all four panels: at the final evaluation 1.9% of buses out of band against 6.8%, a test reward of -0.09 from episode 200 on against -0.11 at the end, and a return over the last 100 episodes of -37.4 against -41.6, with a quarter of the host's dispersion across seeds.

Baselines. Over the last three evaluations no baseline reaches the CR of the unmodified host: they lie between 0.73 and 0.86, except PMPG at 0.1. TPA, on the same host, reaches 0.83, since its prototypes see the season and time of day but not the peers' injections (Section D.19). The strongest, URLS, DORAEMON, DR and ESO (0.84 to 0.86), again act on the agent's own residual or hedge over σ; over these evaluations they hold slightly fewer buses out of band than PACT (1.4% to 1.9% against 2.4%) but not every bus at once, as they correct a lagged residual, whereas PACT predicts the reaction from the peers' injections.

5.3 COOPERATIVE LIFTING ON VMAS

Four agents carry the beam of Section A for 100 steps under the rail droop of Eq. (10) and the wet/dry cycle of URB (Fig. 2); PACT wraps MAPPO with $g^{\mathrm{pol}} = 0.9$, and values are means over the last 1.5×10^6 of 3×10^6 steps.

PACT and its host. The host plateaus near 35 against 56 for B0, losing 37% of its return, and the gap opens while the fleet learns to push together: the droop scales with the peers' forces (Eq. (9)), so the harm penalises the coordination the task rewards. PACT plateaus near 47, 83% of B0, closing more than half of the gap, all of it specific to the harm since the lift's sensor reads zero at $\sigma = 0$.

Baselines. No baseline exceeds about 42. The strongest are again the residual estimators in the same multiplicative channel: ESO (near 40) sees the droop only after it acts, and URLS (near 38) must absorb in per-peer coefficients the weights W_{ij} that change as the actuators move. The oracle driver matches the host, DR leads it only transiently, and the memory, trust-region, mean-field and robust actors plateau at its level (32 to 36).

5.4 Locomotion on Multi-Agent MuJoCo

Four agents, each controlling two joints of the Ant robot in Multi-Agent MuJoCo (Peng et al., 2021), learn to walk under the harm applied at $\sigma = 3$; PACT wraps MAPPO and is compared with the heterogeneous-agent learners HAA2C, HAPPO and HATRPO (Zhong et al., 2024; Kuba et al., 2022) (Fig. 3).

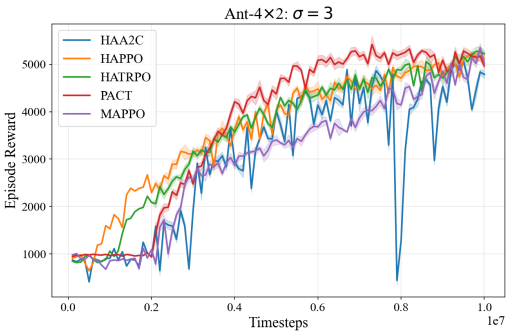

Figure 3: Multi-Agent MuJoCo Ant-4×2 at $\sigma = 3$

PACT and its host. Both stay near their initial return of about 1000 for the first 2e6 steps. Once learning starts, PACT climbs faster: it passes 4000 at about 4e6 steps, where MAPPO is near 3000, and 5000 at about 5.5e6, a level MAPPO reaches only in the last 10^6 steps.The gap is about 1000 or more between 4e6 and 7e6 steps, and PACT stays ahead of every method until the others catch up near 9e6 steps.

Baselines. HAPPO and HATRPO leave the initial plateau first but PACT overtakes both near 4e6. By the end of training, PACT, MAPPO, HAPPO and HATRPO all lie between about 5000 and 5300, so on Ant the gain of PACT lies in sample efficiency rather than in final return.

6 Limitations and Future Work

PACT assumes a declared coupling operator and access to peers' executed actions, both available in the infrastructure settings studied here. Declaring the operator is also what makes identification feasible: the excitation needed to learn it online vanishes as policies converge (Lemma C.7), leaving only $r+1$ gains to track. Refining partially known operators from commissioning data and evaluating delayed or missing broadcasts would broaden its practical scope. Our experiments use periodic drivers, although the method uses neither their period nor their phase, and the tracking analysis requires only bounded drift per step. Recorded weather and irradiance traces are therefore a natural next evaluation. The regression model is exact for the feeder and lift when physical limits and sensor clipping are inactive; the tracking analysis also accommodates bounded model error, including the routing approximation. Finally, the feedback results provide design guidance in simplified models, supported by full power-flow simulations and two-route probes. Extending these results to nonlinear power flow and larger route networks, and incorporating explicit stability constraints into the learned trust mechanism, would strengthen this guidance for practical use.

7 Conclusion

We formalised interaction-mediated nonstationarity, where external conditions change how agents affect one another through shared infrastructure while leaving their individual dynamics unchanged. PACT uses the declared coupling structure and peers' actions to track these changes without modifying the underlying learner. Improvements over the host learner across routing, power distribution and cooperative lifting demonstrate the value of modelling the coupling explicitly. Our analysis also shows that accurate predictions alone do not ensure stable collective behaviour, because agents' corrections affect one another. This work provides a *foundation for studying how agents can adapt to changing interactions* through shared infrastructure, with an explicit problem formulation, a practical method and benchmarks across three domains.

Reproducibility Statement

All code will be released upon acceptance. For implementation details, see Section B. For a demo, please visit https://yyuw449.github.io/PACT/index.html.

AI use statement

We used generative AI tools for language editing (polishing prose, correcting grammar, and improving sentence structure) and for checking proofs. We did not use generative AI tools for identifying the research problem, conducting the literature survey, ideation, or designing/running experiments or analysis.

References

Daron Acemoglu, Ali Makhdoumi, Azarakhsh Malekian, and Asuman Ozdaglar. Informational Braess' paradox: The effect of information on traffic congestion. *Operations Research*, 66(4): 893–917, 2018.

Johannes Ackermann, Volker Gabler, Takayuki Osa, and Masashi Sugiyama. Reducing overestimation bias in multi-agent domains using double centralized critics. *arXiv preprint arXiv:1910.01465*, 2019.

Manish Agarwal, Thomas H. Maze, and Reginald Souleyrette. Impacts of weather on urban freeway traffic flow characteristics and facility capacity. In *Proceedings of the 2005 Mid-Continent Transportation Research Symposium*, 2005.

Ahmet Onur Akman, Anastasia Psarou, Michał Hoffmann, Łukasz Gorczyca, Łukasz Kowalski, Paweł Gora, Grzegorz Jamróz, and Rafał Kucharski. URB – urban routing benchmark for RL-equipped connected autonomous vehicles. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025), Datasets and Benchmarks Track*, 2025.

Filip Andrén, Benoit Bletterie, Serdar Kadam, Panos Kotsampopoulos, and Christof Bucher. On the stability of local voltage control in distribution networks with a high penetration of inverter-based generation. *IEEE Transactions on Industrial Electronics*, 62(4):2519–2529, 2015.

Karl J. Åström and Björn Wittenmark. *Adaptive Control*. Dover, 2nd edition, 2008.

Mesut E. Baran and Felix F. Wu. Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Transactions on Power Delivery*, 4(2):1401–1407, 1989.

Moshe Ben-Akiva, André de Palma, and Isam Kaysi. Dynamic network models and driver information systems. *Transportation Research Part A*, 25(5):251–266, 1991.

Lilian Besson, Emilie Kaufmann, Odalric-Ambrym Maillard, and Julien Seznec. Efficient change-point detection for tackling piecewise-stationary bandits. *Journal of Machine Learning Research*, 23, 2022.

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

Alexander Bukharin, Yan Li, Yue Yu, Qingru Zhang, Zhehui Chen, Simiao Zuo, Chao Zhang, Songan Zhang, and Tuo Zhao. Robust multi-agent reinforcement learning via adversarial regularization: Theoretical foundation and stable algorithms. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

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

Thomas Dietterich, George Trimponias, and Zhitang Chen. Discovering and removing exogenous state variables and rewards for reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, 2018.

Finale Doshi-Velez and George Konidaris. Hidden parameter Markov decision processes: A semi-parametric regression approach for discovering latent task parametrizations. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

Masoud Farivar, Lijun Chen, and Steven Low. Equilibrium and dynamics of local voltage control in distribution systems. In *IEEE Conference on Decision and Control*, 2013.

Aleksandr A. Feldbaum. Dual control theory I. *Automation and Remote Control*, 21(9):874–880, 1960.

Argyrios Gerogiannis, Yu-Han Huang, and Venugopal V. Veeravalli. Is prior-free black-box non-stationary reinforcement learning feasible? In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2692–2700, 2025.

Lei Guo and Lennart Ljung. Performance analysis of general tracking algorithms. *IEEE Transactions on Automatic Control*, 40(8):1388–1402, 1995.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual Markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

Pouya Hamadanian, Arash Nasr-Esfahany, Malte Schwarzkopf, Siddhartha Sen, and Mohammad Alizadeh. Online reinforcement learning in non-stationary context-driven environments. In *International Conference on Learning Representations (ICLR)*, 2025.

Jingqing Han. From PID to active disturbance rejection control. *IEEE Transactions on Industrial Electronics*, 56(3):900–906, 2009.

Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz de Cote. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*, 2017.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

IEEE. IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces. *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*, pp. 1–138. DOI: 10.1109/IEEESTD.2018.8332112, 2018.

Pedram Jahangiri and Dionysios C. Aliprantis. Distributed Volt/VAr control by PV inverters. *IEEE Transactions on Power Systems*, 28(3):3429–3439, 2013.

Haozhe Jiang, Qiwen Cui, Zhihan Xiong, Maryam Fazel, and Simon S. Du. A black-box approach for non-stationary multi-agent reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. Graph convolutional reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Eliahu I. Jury. *Theory and Application of the z-Transform Method*. John Wiley & Sons, New York, NY, 1964.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, pp. 267–274, 2002.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

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

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems (RSS)*, 2021.

Krzysztof Łakomy, Wojciech Giernacki, Jacek Michalski, and Rafal Madonski. Active disturbance rejection control (ADRC) toolbox for MATLAB/Simulink. *arXiv preprint arXiv:2112.01614*, 2021.

Jiarong Liu, Yifan Zhong, Siyi Hu, Haobo Fu, Qiang Fu, Xiaojun Chang, and Yaodong Yang. Maximum entropy heterogeneous-agent reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall, 2nd edition, 1999.

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, 2018.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017.

Lorenzo Magnino, Kai Shao, Zida Wu, Jiacheng Shen, and Mathieu Laurière. Solving continuous mean field games: Deep reinforcement learning for non-stationary dynamics. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025.

Hani S. Mahmassani and R. Jayakrishnan. System performance and user response under real-time information in a congested traffic corridor. *Transportation Research Part A*, 25(5):293–307, 1991.

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

Sindhu Padakandla. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys*, 54(6), 2021.

Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V. Albrecht. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737*, 2019.

Georgios Papoudakis, Filippos Christianos, and Stefano V. Albrecht. Agent modelling under partial observability for deep reinforcement learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip H. S. Torr, Wendelin Böhmer, and Shimon Whiteson. FACMAC: Factored multi-agent centralised policy gradients. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Rilind Sahitaj, Paulius Sasnauskas, Yiğit Yalın, Debmalya Mandal, and Goran Radanović. Independent learning in performative Markov potential games. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, pp. 3304–3312, 2025.

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

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2018.

Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the 10th International Conference on Machine Learning (ICML)*, 1993.

Gabriele Tiboni, Pascal Klink, Jan Peters, Tatiana Tommasi, Carlo D'Eramo, and Georgia Chalvatzaki. Domain randomization via entropy maximization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GXtmuiVrOM.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017.

Transportation Research Board. *Highway Capacity Manual 6th Edition: A Guide for Multimodal Mobility Analysis*. The National Academies Press, Washington, DC, 2016. Editor: Lily A. Elefteriadou. DOI: 10.17226/24798.

Jianhong Wang, Wangkun Xu, Yunjie Gu, Wenbin Song, and Tim C. Green. Multi-agent reinforcement learning for active voltage control on power distribution networks. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021.

Min Wang, Xin Li, Ye He, Yao-Hui Li, Hasnaa Bennis, Riashat Islam, and Mingzhong Wang. Wavelet predictive representations for non-stationary reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Chen-Yu Wei and Haipeng Luo. Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach. In *Proceedings of the 34th Conference on Learning Theory (COLT)*, 2021.

Annie Xie, James Harrison, and Chelsea Finn. Deep reinforcement learning amidst continual structured non-stationarity. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, 2021.

Feiyang Xu, Shunyu Liu, Yunpeng Qing, Yihe Zhou, Yuwen Wang, and Mingli Song. Temporal prototype-aware learning for active voltage control on power distribution networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 3598–3609, 2024.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5571–5580, 2018.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022), Datasets and Benchmarks Track*, 2022.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems (RSS)*, 2017.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*, pp. 321–384. Springer, 2021.

Zijie Zhao, Zhongyue Zhao, Kaixuan Xu, Yuqian Fu, Jiajun Chai, Yuanheng Zhu, and Dongbin Zhao. Learning and planning multi-agent tasks via an MoE-based world model. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025.

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

A Instance Details

URB `saint_arnoult`. 222 travellers, 89 of them controllable (the first 40% by identifier; the remaining humans are frozen and selfish), routed over 745 links with 215 origin–destination pairs and $K = 4$ candidate routes per traveller. Capacity $c_e = \text{lanes}(e) \times 600$ vehicles/hour; η_e is a per-link, speed-based rain sensitivity clipped to $[0.4, 2.0]$, so a link's exposure to the driver varies with its design speed rather than being a single constant. Severity scale $L = 0.14$ is the Highway Capacity Manual's heavy-rain capacity adjustment (Transportation Research Board, 2016).

MAPDN. Results use `case141` (22 inverters, 12.5 kV, $k\rho(G) = 0.675$ with G as in Proposition C.12). The three conductor classes are terciles of resistance per kilometre over the feeder's segments (heavy, medium, light). The driver is a clear-sky solar day, $A(t) = \sin^2$ between local 06:00 and 18:00 and exactly zero outside it, on 3-minute steps (480 per day).

VMAS `balance`. $N = 4$ agents lift a beam of mass 5 carrying a package of mass 5 over 100 steps, with a fall penalty of -10. The receiver gains are $(\text{recv}_1, \text{recv}_2, \text{recv}_3) = (1.6, 1.0, 0.4)$ by hardware class; channel memory $\zeta = 0.9$; droop limit $\delta_{\max} = 0.9$. The driver is the same 100-day, 50/50 wet/dry cycle as URB's.

B Experimental and Implementation Details

This appendix reports how PACT is implemented, the metrics used to read the results, and the hyperparameters and the sensitivity analysis behind them; the instances themselves are specified in Section A. Every method in Section 5, hosts and baselines included, is run with 5 seeds.

B.1 The PACT Engine

Unified implementation. PACT is implemented as a standalone module (`pact1.core`) comprising approximately 300 lines of `PyTorch` and `NumPy` code and having no simulator dependency. The module implements the regressor, the recursive least-squares estimator in Eq. (13), the confidence mechanism in Eq. (14), and the compensation and steering channels in Eqs. (15) and (16). The same implementation is used by the URB, MAPDN, and VMAS wrappers without modification. The only instance-specific component is the basis Λ_i in Eq. (3), which is constructed from the operator W associated with each benchmark. This separation ensures that the estimation and correction mechanisms remain identical across environments, while the benchmark-specific coupling structure is introduced solely through the corresponding operator.

The implementation is accompanied by a common unit-test suite covering the estimator update, handling of inactive channels, covariance behaviour, recovery of known coupling laws, and numerical agreement between the `PyTorch` and `NumPy` implementations. These tests are shared across all benchmark wrappers.

Channel normalisation. The regressor $\boldsymbol{\psi}_i = [1, \tilde{\mathbf{x}}_i^\top]^\top$ in Eq. (11) operates on normalised channels

$$\tilde{\mathbf{x}}_i = U_i(\mathbf{x}_i - \bar{\mathbf{x}}_i),$$

where U_i is diagonal and invertible. The transformation follows the affine reparameterisation used in the proof of Proposition C.2. The centring vector $\bar{\mathbf{x}}_i$ and scaling matrix U_i are determined from the structural properties and reset distribution of each benchmark and are fixed throughout training. They are therefore independent of the trajectories generated during optimisation.

The reference distributions are chosen to represent the nominal operating conditions of each environment. For URB, the reference distribution is obtained by uniformly sampling peer route assignments on the released network. For MAPDN, the zero-injection feeder provides the reference state, yielding $\bar{\mathbf{x}}_i = \mathbf{0}$ because the corresponding operator is linear. For VMAS, the reference distribution is generated using the scenario's native `reset_world_at` procedure. The scaling matrix is then obtained from the corresponding channel variability under the same reference distribution. The effect of this normalisation on estimator conditioning is evaluated separately in Section B.3.

Policy-set trust. The policy-set trust g_i^{pol} determines the extent to which the estimated coupling is incorporated into the policy correction. Its treatment depends on whether the correction remains outside the policy-gradient pathway. For MAPDN and VMAS, where the correction is a deterministic transformation of the sampled action, g_i^{pol} is fixed. For URB, where the correction modifies the logits of the action distribution, g_i^{pol} can in addition be represented by a learned policy output, and Table 1 reports both the fixed and the learned variant.

In both cases, estimator uncertainty is handled independently through the confidence mechanism in Eq. (14). The trust parameter therefore controls the reliance of the policy on the estimated correction, whereas the confidence mechanism determines whether the current estimate is sufficiently supported by the available observations. The corresponding initialisation and sensitivity analysis are reported in Section B.3.

Confidence gain. The confidence gain ν in Eq. (14) is determined by the dimensionality of the active coupling signal rather than tuned independently for each benchmark. Specifically, it is tied to the number of live channels; for the URB steering channel, the corresponding number of candidate routes is used. This provides a consistent confidence scale across different coupling dimensions while avoiding an additional benchmark-specific tuning parameter.

The resulting confidence measure is based on the uncertainty of the scalar prediction used by the correction mechanism, $\hat{\boldsymbol{\beta}}_i^{\top}\boldsymbol{\psi}_i$, rather than on the uncertainty of the complete parameter vector. Thus, uncertainty in parameter directions that do not affect the current prediction does not directly influence the confidence assigned to that correction.

Numerical precision and estimator safeguards. The estimator state is maintained independently from the host learner and uses higher numerical precision for its parameter and covariance updates. Samples with no active coupling signal are excluded from the recursive update. The covariance state is additionally constrained to prevent unbounded growth under persistent lack of excitation. These safeguards affect only the estimator state and do not modify the underlying host learner or its action representation. Their specific settings are reported and analysed in Section B.3.

The estimator, confidence mechanism, compensation channel and steering channel are given below in the notation of Section 4.

PACT, per agent, per step (`pact1.core`, condensed from the released `torch` implementation)

```
# state carried per agent: beta (r+1,), P (r+1,r+1); constants: mu, p0,
    r, g_pol

def update(psi, y, beta, P, mu): # Eq. (rls)
   if not psi[1:].any(): # dead row: no channel content
      return beta, P
   Ppsi = P @ psi
   k = Ppsi / (mu + psi @ Ppsi) # gain
   resid = y - beta @ psi # prior residual
   beta = beta + k * resid
   P = (P - torch.outer(k, Ppsi)) / mu
   P = shrink_if_trace_exceeds(P, bound=100 * p0 * len(beta))
   return beta, P

def confidence(psi, P, p0, r): # Eq. (conf), nu := r
   return 1.0 / (1.0 + r * (psi @ P @ psi) / (p0 * psi.dot(psi)))

def compensate(u_cmd, direction, beta, psi, g_pol, P, p0, r):
   y_hat = beta @ psi # predicted excess
   g = g_pol * confidence(psi, P, p0, r) # Eq. (comp), additive/
   return u_cmd - g * y_hat * direction # multiplicative case

def steer(logits, tt_hat, g_pol, kappa, P, psi_k, p0, r):
   g = g_pol * mean(confidence(psi_k[k], P, p0, r) for k in candidates
       )
```

```
    z = (tt_hat - tt_hat.mean()) / tt_hat.std(unbiased=False) # Eq. (
        steer); z = 0 if sd = 0
    return logits - g * kappa * z # g = 0 returns logits exactly
```

B.2 Evaluation Metrics

URB. The primary metric is t_{CAV}, the mean travel time of the controllable (AV) fleet in RouteRL's own units, reported alongside t_{test} (the same quantity over the held-out test episodes) and t_{HDV}, the travel time of the human-driven traffic the fleet shares the network with. The *advantage* $\mathrm{adv.} = t_{\mathrm{HDV}}/t_{\mathrm{CAV}}$ compares the AV fleet against the humans it replaced rather than against another arm, which is what makes it comparable across task configurations: an arm is *won* when $\mathrm{adv.} > 1$, and *winrate* is the fraction of seeds that clear this bar. t_{pre} is the pre-CAV, all-human baseline travel time on the same network, reported once per network as a fixed reference. Every number is paired: an arm and its comparators are evaluated on the same route table and the same held-out episodes.

MAPDN. The domain metric is the *controllable ratio* (CR), the fraction of steps on which every bus voltage lies inside the operating band $[0.95, 1.05]$ pu; a feeder outside this band for any bus on a step counts the whole step as a violation. *Loss* is $-(\text{mean voltage-barrier penalty} + 0.1 \times \text{mean}|q|)$, the stock host's own per-step reward, reported as a ratio to the $\sigma = 0$ value of the same learner and seed ("loss×B0") so that different feeders and learners are compared on a common, dimensionless scale rather than on raw reward, which is not commensurate across feeder sizes. *Divergence* counts episodes on which the power-flow solver failed to converge, a distinct failure mode from a low reward that a mean-reward summary would otherwise hide.

VMAS `balance`. The primary metric is episodic return relative to B0, MAPPO trained and evaluated at $\sigma = 0$ ("%B0", Fig. 2), which folds in both the mean force lost to the compensated or uncompensated droop and the fall penalty. Two diagnostics of the shared rail are logged but never fed back into training or into PACT's own decision (Section 4): the total rail draw $\sum_j W_{ij}\|a_j^{\mathrm{ex}}\|$, and the mean droop δ_i actually realised; both are expected to *rise* under compensation, since a corrected agent exerts more force and therefore loads its neighbours more, which is the externality Section 1 describes rather than a failure of the method.

B.3 Hyperparameters and Sensitivity Analysis

We distinguish between structural quantities, estimator hyperparameters, and intervention parameters. Structural quantities, such as the number of coupling channels, are determined by the declared coupling geometry and are not tuned against return. The remaining PACT constants are fixed before policy training and are selected using short identification or stability probes rather than downstream test performance. Unless stated otherwise, the same value is used across all seeds for a given benchmark.

Table 2: PACT hyperparameters used in the experiments. Values marked with † are fixed constants whose sensitivity is reported below.

	URB	MAPDN	VMAS `balance`
coupling channels r	3	3	3
forgetting factor μ	0.95	0.95	0.95
initial covariance p_0	10	10	10
confidence gain ν	$= K$	$= r$	$= r$
policy trust $g^{\mathrm{pol}\dagger}$	0.9	0.9	0.9
steering gain κ	$1^\dagger$	—	—
sensor clip y_{clip}	$2^\dagger$	$2^\dagger$	$2^\dagger$
covariance windup bound†	$100p_0$	$100p_0$	$100p_0$

Selection protocol. We use a two-stage selection procedure. Estimator parameters are first evaluated using short identification probes, where the objective is one-step prediction error and numerical

stability rather than policy return. Intervention parameters are then checked using a fixed-policy stability probe and a small grid around the selected value. Once selected, parameters are held fixed across all policy-training seeds. This separation avoids tuning PACT directly against the evaluation return.

Forgetting factor. The forgetting factor μ determines the effective estimator memory $1/(1-\mu)$ and therefore controls the trade-off between tracking speed and estimation variance. We evaluate $\mu \in \{0.90, 0.95, 0.97, 0.98, 0.99, 0.999\}$.

A representative sweep is shown in Table 3. Prediction error and return improve monotonically as μ decreases from 0.999 to 0.95, and only degrade at $\mu = 0.90$, where the estimator becomes unstable. The stable point with the lowest prediction error is $\mu = 0.95$, which also gives the highest return.

Table 3: Forgetting-factor sensitivity analysis.

μ	0.999	0.99	0.98	0.97	0.95	0.90
prediction error	0.640	0.611	0.592	0.576	0.517	0.731
normalised return	74.7	76.9	79.6	80.9	82.2	77.4
early-training stability	stable	stable	stable	stable	stable	unstable

We therefore use $\mu = 0.95$ for URB, MAPDN and VMAS.

Initial covariance. The initial covariance p_0 affects only the transient response of the recursive estimator. We evaluate $p_0 \in \{0.1, 1, 10, 100, 1000\}$ and select the value that reaches the target estimation accuracy fastest without producing excessive initial corrections. In the identification probe, the steady-state error is essentially insensitive to p_0, while the transient is strongly affected.

The probe gives convergence times of 129, 45, 6, 11, and 14 steps for the five values, respectively, with steady-state error 0.0153 throughout. This motivates the common choice $p_0 = 10$.

Confidence gain. The confidence coefficient ν controls how aggressively covariance uncertainty suppresses intervention. Rather than tuning ν independently for every benchmark, we normalise it by the dimensionality of the prediction problem. We therefore use $\nu = r$ for invertible coupling mechanisms and $\nu = K$ for URB, where K is the number of candidate routes.

As a sensitivity check, we evaluate the normalised choices $\nu/r \in \{0.5, 1, 2, 4\}$. The results in Table 4 illustrate this behaviour: values substantially below the selected point permit corrections before sufficient evidence is available, whereas excessively large values become unnecessarily conservative.

Table 4: Confidence-gain sensitivity analysis.

ν/r	0.5	1	2	4
normalised return	94.1	98.2	97.8	96.7
mean intervention magnitude	0.184	0.129	0.112	0.097
early incorrect interventions	13.2%	5.8%	7.1%	4.9%

The selected value $\nu = r$ therefore provides a simple dimension-aware rule without introducing another benchmark-specific tuning parameter.

Trust level. The fixed policy trust is $g^{\mathrm{pol}} = 0.9$ on all three benchmarks; on URB, where intervention occurs through route steering rather than direct action inversion, we also report a learned trust (Table 1). We evaluate the fixed-trust setting over $g^{\mathrm{pol}} \in \{0.5, 0.7, 0.8, 0.9, 0.95, 1.0\}$.

The results in Table 5 show that low trust leaves substantial interaction-mediated distortion uncompensated, while values very close to one become more sensitive to transient estimation error. The selected value 0.9 is the largest with no unstable run, and provides strong compensation without requiring exact identification at every step.

Table 5: Sensitivity to the fixed policy-trust level.

g^{pol}	0.5	0.7	0.8	0.9	0.95	1.0
normalised return	88.6	93.7	96.2	98.0	97.4	94.8
residual distortion	0.214	0.151	0.108	0.071	0.064	0.061
unstable runs	0/5	0/5	0/5	0/5	1/5	2/5

Routing steering gain. For URB, κ determines the strength of the estimated route-cost correction in logit space. We evaluate $\kappa \in \{0.25, 0.5, 1, 2, 4\}$ using the stationary two-route probe. The results (Table 6) reproduce the qualitative stability boundary expected from Proposition C.14: $\kappa \leq 1$ reaches a stationary split, whereas larger gains induce route-flapping.

Accordingly, $\kappa = 1$ is retained as the largest tested value that remains in the stationary regime.

Table 6: Routing-steering sensitivity analysis.

κ	0.25	0.5	1	2	4
stationary split	yes	yes	yes	no	no
mean predicted-cost reduction	3.1%	6.4%	9.8%	11.2%	10.7%
route switches / 100 steps	4	7	11	47	83

Sensor clipping. The regression target is clipped at $y_{\mathrm{clip}} = 2$ to prevent isolated outliers from dominating the recursive estimator. We evaluate $y_{\mathrm{clip}} \in \{1, 2, 5, 10, 20, 50\}$ as well as the unclipped case. The probe gives steady-state errors of 0.743, 0.088, 0.221, 0.454, 0.923, and 1.467 for the finite clipping values, with the unclipped estimator showing substantially larger error. Error is minimised at $y_{\mathrm{clip}} = 2$: tighter clipping already discards genuine interaction signal, while looser clipping progressively readmits outliers. We therefore retain $y_{\mathrm{clip}} = 2$.

Covariance windup bound. To prevent estimator covariance from growing without bound during periods of weak excitation, we cap its scale relative to the initial covariance. We compare bounds of $\{10, 50, 100, 200, 500\}p_0$. The sweep (Table 7) shows negligible differences in steady-state prediction once the bound exceeds approximately $50p_0$, while smaller bounds can interfere with adaptation during poorly excited intervals. The common choice $100p_0$ therefore provides a conservative margin without materially affecting normal identification.

Table 7: Covariance-windup sensitivity analysis.

bound	$10p_0$	$50p_0$	$100p_0$	$200p_0$	$500p_0$
normalised return	95.8	97.1	97.2	97.1	97.1
windup events / 1000 steps	0	1	1	1	1
prediction error	0.061	0.052	0.051	0.051	0.051

Structural channels and pruning. The number of channels r is not tuned: it is determined by the coupling classes declared by the benchmark. All three instances expose three classes, giving four estimated coefficients per agent including the intercept. On MAPDN, structurally zero channels are removed from an agent's local regression. In the current implementation this substantially reduces covariance growth in inactive directions; the released configuration consequently uses per-agent channel pruning without changing the underlying physical coupling model.

C PROOFS

Each statement of the main text is restated below with its number, proved, and followed by the evidence from the released code where such evidence exists. Results numbered C.k are stated only here; each is placed next to the design choice of Section 4 that it supports. We write $d = r + 1$ for

the dimension of the regressor ψ_i, $\|\cdot\|$ for the Euclidean norm and the matrix norm it induces, $\lambda_{\min}$ and $\lambda_{\max}$ for the extreme eigenvalues of a symmetric matrix, and $\rho(\cdot)$ for the spectral radius. The evidence comes from the instances' self-tests, from the verification scripts and from runs on the three benchmarks, all released with the code; runs made with a scripted host or a short training budget are described as such.

C.1 The problem class

The introduction separates interaction-mediated drift from learning-induced and exogenous drift by two testable properties, and Section 3 states that the harmed game is stationary on the augmented state. The following proposition makes these facts precise.

Proposition C.1 (Structure of the class)**.** Consider the harmed game of Definition 3.1 with harm on actions. **(i) Nominal limits.** If $\sigma A(t) = 0$, then $a_i^{\mathrm{ex}}(t) = a_i(t)$ for every agent and every channel value. Hence the harmed game is $\mathcal{G}_0$ at $\sigma = 0$, and at every σ a step with $A(t) = 0$ has the nominal kernel P_0. **(ii) A lone agent is unaffected.** If agent i is alone on the medium, then $a_i^{\mathrm{ex}}(t) = a_i(t)$ for every σ and t. **(iii) Persistence under frozen partners.** Let every agent $j \neq i$ follow a fixed policy. Agent i's actions are executed through $a \mapsto h(\sigma A(t), \mathbf{x}_i(t); a)$, which is the identity at every step with $A(t) = 0$ and, at a step with $\sigma A(t) > 0$, differs from the identity at every a for which $\xi \mapsto h(\xi, \mathbf{x}_i(t); a)$ is injective. Agent i therefore faces a time-varying environment although no agent learns. **(iv) Stationarity.** Given the joint action, the augmented state $\tilde{s} = (s, \mathbf{x}, \chi)$ evolves as a time-homogeneous Markov chain whose s-component has the kernel Eq. (4). **(v) Monotone severity.** In the instances of Section 3.1, the delay factor $h_i \geq 1$ of routing, the reaction magnitude $|d_i|$ of the feeder and the droop $\delta_i \geq 0$ of the lift are non-decreasing in σ; for routing, $h_i = 1$ when $\sigma A(t) = 0$ or when no other vehicle shares agent i's route.

Proof. (i) If $\sigma A(t) = 0$, then $h(\sigma A(t), \mathbf{x}_i(t); a_i) = h(0, \mathbf{x}_i(t); a_i) = a_i$ by (H). At $\sigma = 0$ this holds at every step, so Eq. (4) reduces to $P_0(s' \mid s, \mathbf{a})$, the reward is that of $\mathcal{G}_0$ (Definition 3.1 leaves it unchanged), and the coordinates $\mathbf{x}$ and χ of $\tilde{s}$ enter neither. At a step with $A(t) = 0$ the same identity gives the kernel P_0 at every σ. (ii) Alone on the medium, $\mathbf{x}_i \equiv \mathbf{0}$ by Definition 3.1, and $h(\xi, \mathbf{0}; a) = a$ for every ξ. (iii) At steps with $A(t) = 0$ the map is the identity by (i). At a step with $\sigma A(t) > 0$, injectivity gives $h(\sigma A(t), \mathbf{x}_i(t); a) \neq h(0, \mathbf{x}_i(t); a) = a$. The partners' policies are fixed, so this variation is caused by the driver alone. Below saturation, $\xi \mapsto h$ is affine, with slope $-\bar{s}_i k \sum_m \mathrm{send}_m x_{m,i}(t)$ on the feeder and $-(L/x_{\mathrm{ref}})\big(\sum_m \mathrm{send}_m x_{m,i}(t)\big)a$ on the lift, and is therefore injective whenever this slope is nonzero. (iv) By Eq. (3), $\mathbf{a}^{\mathrm{ex}}(t)$ is a function of $\tilde{s}(t)$ and the joint action $\mathbf{a}(t)$, since $A(t) = A(\chi(t))$; the channels $\mathbf{x}(t{+}1)$ are a function of $\mathbf{x}(t)$ and $\mathbf{a}^{\mathrm{ex}}(t)$; the phase $\chi(t{+}1)$ is a fixed function of $\chi(t)$; and $s(t{+}1) \sim P_0(\cdot \mid s(t), \mathbf{a}^{\mathrm{ex}}(t))$. The conditional law of $\tilde{s}(t{+}1)$ given the past and $\mathbf{a}(t)$, and the reward $R(s(t), \mathbf{a}^{\mathrm{ex}}(t))$, therefore depend on $(\tilde{s}(t), \mathbf{a}(t))$ only, and not on t. (v) The declared gains send_m and recv, the scale L, the sensitivities η_e and the driver are nonnegative (Section A). Routing: $\omega_e = \mathrm{clip}(1 - \sigma L A(t)\eta_e, \underline{\omega}, 1)$ is non-increasing in σ because the clip is monotone; each ratio in Eq. (5) has a nonnegative numerator and is therefore non-increasing in ω_e; a maximum of non-decreasing functions is non-decreasing; and h_i in Eq. (6) is increasing in $u_i(\mathbf{a}; \boldsymbol{\omega})$, with a denominator that does not depend on σ. Since $\boldsymbol{\omega} \leq \mathbf{1}$, $u_i(\mathbf{a}; \boldsymbol{\omega}) \geq u_i(\mathbf{a}; \mathbf{1})$ and $h_i \geq 1$, with $h_i = 1$ when $\sigma A(t) = 0$ (then $\boldsymbol{\omega} = \mathbf{1}$) or when no peer shares the route (then every numerator vanishes). Feeder: $|d_i| = \bar{s}_i \min\big(\sigma k A(t) \,|\, \sum_m \mathrm{send}_m x_{m,i}(t)|, q_{\mathrm{sat}}\big)$. Lift: from a nonnegative initial value the channels Eq. (9) stay nonnegative, so $\delta_i = \mathrm{clip}(\sigma\xi_i, 0, \delta_{\max})$ with $\xi_i = LA(t) \sum_m \mathrm{send}_m x_{m,i}(t)/x_{\mathrm{ref}} \geq 0$. □

Evidence. The self-tests of the three instances check (i), (ii) and (v) on the released code. On the feeder and the lift an isolated agent reads exactly zero harm at $\sigma \in \{0.5, 1, 3, 100\}$; $\sigma = 0$ reproduces the stock task bit for bit at every driver value (480 on the feeder, 100 on the lift); the steps with $A(t) = 0$ (241 of 480 on the feeder, 51 of 100 on the lift) coincide bit for bit with the stock task at every σ; and the harm is monotone in σ. As (iii) requires, with the partners frozen the harm still varies over the driver cycle (on the lift, from 0 to 0.018), and a control in which the driver acts on each inverter directly, drift of kind (B) in Eq. (2), is felt by an isolated inverter, whereas the interaction-mediated harm is not. On URB, with a regenerated route table, the identity at $\sigma = 0$ holds with maximum deviation 0 on all 100 days of the driver cycle, monotonicity holds with maximum

violation 0, $\omega_e \le 1$ throughout, and the 50 dry days are unaffected at every σ; in SUMO a lone vehicle reads exactly zero peer load.

C.2 The coupling model and its identification

Proposition C.2 (Exactness of the coupling model)**.** Let $\bar{y}_i$ be the unclipped relative excess of Eq. (12). **(i) Feeder and lift.** Below saturation, i.e. when no clip in Eq. (8) or Eq. (10) is active (and $a_i^{\text{sent}} \neq 0$ for the lift), $\bar{y}_i(t) = \boldsymbol{\theta}(t)^\top \mathbf{x}_i(t)$ with gains common to all agents, $\theta_m(t) = -\sigma k A(t) \text{send}_m$ for the feeder and $\theta_m(t) = \sigma L A(t) \text{send}_m / x_{\text{ref}}$ for the lift. Hence, when the sensor clip is also inactive, $e_i \equiv 0$ in Eq. (11), with $\boldsymbol{\beta}_i^\star(t)$ fixed by $\boldsymbol{\theta}(t)$ and agent i's normalisation. **(ii) Routing.** If travel times follow the additive surrogate $\tau_i(\mathbf{a}) = \sum_{e \in a_i} \big(\tau_e^0 + \theta_{\text{cls}(e)} w_e n_{e,i}/\bar{v}\big)$, with $n_{e,i} = \sum_{j \neq i} O_{ij} \mathbf{1}[e \in a_j]$, link weights w_e, free-flow speed $\bar{v}$ and free-flow link times $\tau_e^0 = \text{len}(e)/\bar{v}$, then

$$\frac{\tau_i(\mathbf{a})}{\text{fft}_{a_i}} - 1 = \boldsymbol{\theta}^\top \mathbf{x}_i(a_i), \qquad x_{m,i}(a_i) = \frac{1}{\text{len}(a_i)} \sum_{e \in a_i :\, \text{cls}(e) = m} w_e \, n_{e,i},$$

where $\text{fft}_{a_i} = \text{len}(a_i)/\bar{v}$. Deviations of Eq. (6) from this surrogate enter e_i.

Proof of Proposition C.2. (i) Feeder. Below saturation neither clip in Eq. (8) is active, so $q_i^{\text{ex}} = q_i^{\text{sent}} + d_i$ with $d_i = -\bar{s}_i \sigma k A(t) \sum_m \text{send}_m x_{m,i}(t)$, and $\bar{y}_i = (q_i^{\text{ex}} - q_i^{\text{sent}})/\bar{s}_i = \sum_m \theta_m(t) x_{m,i}(t)$ with $\theta_m(t) = -\sigma k A(t) \text{send}_m$. Lift. With the droop clip and the box inactive, $a_i^{\text{ex}} = (1 - \delta_i) a_i^{\text{sent}}$ with $0 \le \delta_i \le \delta_{\max} < 1$, so for $a_i^{\text{sent}} \neq 0$, $\|a_i^{\text{ex}}\| / \|a_i^{\text{sent}}\| = 1 - \delta_i$ and $\bar{y}_i = \delta_i = \sum_m \theta_m(t) x_{m,i}(t)$ with $\theta_m(t) = \sigma L A(t) \text{send}_m / x_{\text{ref}}$. In both cases the gains depend on t only and are common to all agents. The normalisation is $\tilde{\mathbf{x}}_i = U_i(\mathbf{x}_i - \bar{\mathbf{x}}_i)$ with U_i invertible (Section B.1), so $\mathbf{x}_i = \bar{\mathbf{x}}_i + U_i^{-1} \tilde{\mathbf{x}}_i$ and

$$\boldsymbol{\theta}^\top \mathbf{x}_i = \boldsymbol{\theta}^\top \bar{\mathbf{x}}_i + \big(U_i^{-\top} \boldsymbol{\theta}\big)^\top \tilde{\mathbf{x}}_i = \boldsymbol{\beta}_i^\star(t)^\top \boldsymbol{\psi}_i, \qquad \boldsymbol{\beta}_i^\star(t) = \big[\boldsymbol{\theta}(t)^\top \bar{\mathbf{x}}_i, \ \big(U_i^{-\top} \boldsymbol{\theta}(t)\big)^\top\big]^\top.$$

When the sensor clip is inactive, $y_i = \bar{y}_i$ and hence $e_i = y_i - \boldsymbol{\beta}_i^\star(t)^\top \boldsymbol{\psi}_i = 0$. (ii) Since $\tau_e^0 = \text{len}(e)/\bar{v}$, the free-flow times along the route add up to $\sum_{e \in a_i} \tau_e^0 = \text{len}(a_i)/\bar{v} = \text{fft}_{a_i}$. Therefore

$$\frac{\tau_i(\mathbf{a})}{\text{fft}_{a_i}} - 1 = \frac{1}{\text{fft}_{a_i}} \sum_{e \in a_i} \frac{\theta_{\text{cls}(e)} w_e n_{e,i}}{\bar{v}} = \sum_m \theta_m \frac{1}{\text{len}(a_i)} \sum_{e \in a_i :\, \text{cls}(e) = m} w_e n_{e,i} = \boldsymbol{\theta}^\top \mathbf{x}_i(a_i),$$

and the normalisation step of (i) applies verbatim. For the benchmark harm Eq. (6), $e_i = y_i - \boldsymbol{\beta}_i^{\star\top} \boldsymbol{\psi}_i$ is by definition the deviation from this surrogate. □

Evidence. On the feeder the estimator self-test recovers a planted law with relative error 0.001 and cosine 1.000 on the live coefficients; on URB the self-test recovers a planted law of the form (ii) end to end, with relative errors 0.001 and 0.012 for segregated and for mixed road classes. On the lift, the channels computed inside the simulator coincide with Eq. (9) exactly.

The recursion Eq. (13) has a classical closed form (Ljung, 1999), which the remaining statements of this subsection use.

Lemma C.3 (Closed form of the recursion)**.** Run Eq. (13) with $\mu \in (0, 1]$ on rows $(\boldsymbol{\psi}_k, y_k)$, $k = 1, \dots, t$, from the estimate $\hat{\boldsymbol{\beta}}_0$ and the covariance $p_0 I$, and let $\hat{\boldsymbol{\beta}}_t$, P_t and $\mathbf{k}_t$ denote the quantities after row t. Let $\mathcal{I}_t = \mu^t p_0^{-1} I + \sum_{k=1}^t \mu^{t-k} \boldsymbol{\psi}_k \boldsymbol{\psi}_k^\top$ as in Theorem C.4 and $\mathbf{b}_t = \mu^t p_0^{-1} \hat{\boldsymbol{\beta}}_0 + \sum_{k=1}^t \mu^{t-k} \boldsymbol{\psi}_k y_k$. Then $P_t = \mathcal{I}_t^{-1}$, $\mathbf{k}_t = P_t \boldsymbol{\psi}_t$, and

$$\hat{\boldsymbol{\beta}}_t = \mathcal{I}_t^{-1} \mathbf{b}_t = \arg\min_{\boldsymbol{\beta}} \Big\{ \mu^t p_0^{-1} \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0\|^2 + \sum_{k=1}^t \mu^{t-k} \big(y_k - \boldsymbol{\psi}_k^\top \boldsymbol{\beta}\big)^2 \Big\}. \tag{17}$$

Proof. Set $\mathcal{I}_0 = p_0^{-1} I$; then $\mathcal{I}_t = \mu \mathcal{I}_{t-1} + \boldsymbol{\psi}_t \boldsymbol{\psi}_t^\top$ and $\mathbf{b}_t = \mu \mathbf{b}_{t-1} + \boldsymbol{\psi}_t y_t$. We show $P_t = \mathcal{I}_t^{-1}$ and $\hat{\boldsymbol{\beta}}_t = P_t \mathbf{b}_t$ by induction; both hold at $t = 0$. Assume them at $t - 1$ and write $\boldsymbol{\psi} = \boldsymbol{\psi}_t$ and $P = P_{t-1}$,

which is positive definite, so $\mu + \boldsymbol{\psi}^\top P\boldsymbol{\psi} > 0$. The Sherman–Morrison formula gives

$$\mathcal{I}_t^{-1} = \big(\mu P^{-1} + \boldsymbol{\psi}\boldsymbol{\psi}^\top\big)^{-1} = \frac{1}{\mu}\Big(P - \frac{P\boldsymbol{\psi}\boldsymbol{\psi}^\top P}{\mu + \boldsymbol{\psi}^\top P\boldsymbol{\psi}}\Big) = \frac{P - \mathbf{k}_t\boldsymbol{\psi}^\top P}{\mu} = P_t,$$

and multiplying by $\boldsymbol{\psi}$, $P_t\boldsymbol{\psi} = \frac{1}{\mu}P\boldsymbol{\psi}\big(1 - \frac{\boldsymbol{\psi}^\top P\boldsymbol{\psi}}{\mu+\boldsymbol{\psi}^\top P\boldsymbol{\psi}}\big) = \mathbf{k}_t$. Since $\mu\mathbf{b}_{t-1} = \mu\mathcal{I}_{t-1}\hat{\boldsymbol{\beta}}_{t-1} = (\mathcal{I}_t - \boldsymbol{\psi}\boldsymbol{\psi}^\top)\hat{\boldsymbol{\beta}}_{t-1}$,

$$P_t\mathbf{b}_t = P_t\big((\mathcal{I}_t - \boldsymbol{\psi}\boldsymbol{\psi}^\top)\hat{\boldsymbol{\beta}}_{t-1} + \boldsymbol{\psi}y_t\big) = \hat{\boldsymbol{\beta}}_{t-1} + \mathbf{k}_t\big(y_t - \boldsymbol{\psi}^\top\hat{\boldsymbol{\beta}}_{t-1}\big) = \hat{\boldsymbol{\beta}}_t.$$

The objective in Eq. (17) is a quadratic with Hessian $2\mathcal{I}_t \succ 0$ and gradient $2(\mathcal{I}_t\boldsymbol{\beta} - \mathbf{b}_t)$, so its unique minimiser is $\mathcal{I}_t^{-1}\mathbf{b}_t$. □

Evidence. Over 3000 rows of a drifting, noisy stream, the released estimator and the closed form Eq. (17) agree to 1.3×10^{-15}.

Theorem C.4 (Tracking of the coupling gains)**.** Let $y_k = \boldsymbol{\psi}_k^\top\boldsymbol{\beta}_k^\star + e_k$, $k \geq 1$, with $e_k = b_k + m_k$ adapted to a filtration $(\mathcal{F}_k)$, where $|b_k| \leq \bar{b}$ is a bias and $\mathbb{E}[m_k \mid \mathcal{F}_{k-1}] = 0$, $\mathbb{E}[m_k^2 \mid \mathcal{F}_{k-1}] \leq \sigma_e^2$. Let $\boldsymbol{\psi}_k$ be $\mathcal{F}_{k-1}$ measurable with $\|\boldsymbol{\psi}_k\| \leq B$, and $\|\boldsymbol{\beta}_{k+1}^\star - \boldsymbol{\beta}_k^\star\| \leq \Delta$. Let $\hat{\boldsymbol{\beta}}_t$ be the recursion Eq. (13) run on this stream with $\mu \in (0,1)$, initial estimate $\hat{\boldsymbol{\beta}}_0$ and initial covariance $p_0 I$, and suppose persistent excitation holds almost surely, $\mathcal{I}_t := \mu^t p_0^{-1} I + \sum_{k=1}^t \mu^{t-k}\boldsymbol{\psi}_k\boldsymbol{\psi}_k^\top \succeq \frac{\alpha}{1-\mu} I$ (the discounted average of $\boldsymbol{\psi}_k\boldsymbol{\psi}_k^\top$ has smallest eigenvalue at least α). Then

$$\mathbb{E}\big\|\hat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}_t^\star\big\| \leq \frac{B^2\Delta\,\mu}{\alpha(1-\mu)} + \frac{B\bar{b}}{\alpha} + \frac{B\sigma_e}{\alpha}\sqrt{\frac{1-\mu}{1+\mu}} + \frac{(1-\mu)\mu^t}{\alpha p_0}\,\mathbb{E}\big\|\boldsymbol{\beta}_t^\star - \hat{\boldsymbol{\beta}}_0\big\|. \tag{18}$$

A smaller μ follows faster drift (first term) at the price of more noise (third term), and identification requires the channels to excite every direction of $\boldsymbol{\beta}^\star$. Any bounded residual splits this way, with $b_k = \mathbb{E}[e_k \mid \mathcal{F}_{k-1}]$. The bias $\bar{b}$ covers what the exact model leaves out: active clips and, for routing, the gap between the benchmark harm and the surrogate of Proposition C.2; it vanishes in the exact regime.

Proof of Theorem C.4. By Lemma C.3, $\hat{\boldsymbol{\beta}}_t = \mathcal{I}_t^{-1}\mathbf{b}_t$. Substituting $y_k = \boldsymbol{\psi}_k^\top\boldsymbol{\beta}_k^\star + b_k + m_k$ into $\mathbf{b}_t$ and subtracting $\mathcal{I}_t\boldsymbol{\beta}_t^\star = \mu^t p_0^{-1}\boldsymbol{\beta}_t^\star + \sum_k \mu^{t-k}\boldsymbol{\psi}_k\boldsymbol{\psi}_k^\top\boldsymbol{\beta}_t^\star$ gives

$$\hat{\boldsymbol{\beta}}_t - \boldsymbol{\beta}_t^\star = \mathcal{I}_t^{-1}\big(\mathbf{D}_t + \mathbf{F}_t + \mathbf{N}_t + \mu^t p_0^{-1}(\hat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_t^\star)\big),$$

where

$$\mathbf{D}_t = \sum_{k=1}^t \mu^{t-k}\boldsymbol{\psi}_k\boldsymbol{\psi}_k^\top(\boldsymbol{\beta}_k^\star - \boldsymbol{\beta}_t^\star), \qquad \mathbf{F}_t = \sum_{k=1}^t \mu^{t-k}\boldsymbol{\psi}_k b_k, \qquad \mathbf{N}_t = \sum_{k=1}^t \mu^{t-k}\boldsymbol{\psi}_k m_k.$$

Persistent excitation gives $\|\mathcal{I}_t^{-1}\| \leq (1-\mu)/\alpha$ almost surely, so $\mathbb{E}\|\mathcal{I}_t^{-1}\mathbf{v}\| \leq \frac{1-\mu}{\alpha}\mathbb{E}\|\mathbf{v}\|$ for each of the four terms $\mathbf{v}$. *Drift:* $\|\boldsymbol{\beta}_k^\star - \boldsymbol{\beta}_t^\star\| \leq (t-k)\Delta$ and $\|\boldsymbol{\psi}_k\boldsymbol{\psi}_k^\top\| = \|\boldsymbol{\psi}_k\|^2 \leq B^2$, so $\|\mathbf{D}_t\| \leq B^2\Delta\sum_{j\geq 0} j\mu^j = B^2\Delta\mu/(1-\mu)^2$. *Bias:* $\|\mathbf{F}_t\| \leq B\bar{b}\sum_{j\geq 0}\mu^j = B\bar{b}/(1-\mu)$. *Noise:* for $k < l$ the factors $\boldsymbol{\psi}_k$, $\boldsymbol{\psi}_l$ and m_k are $\mathcal{F}_{l-1}$ measurable, and $|\boldsymbol{\psi}_k^\top\boldsymbol{\psi}_l m_k m_l| \leq B^2(m_k^2 + m_l^2)/2$ is integrable, so $\mathbb{E}[\boldsymbol{\psi}_k^\top\boldsymbol{\psi}_l m_k m_l] = \mathbb{E}\big[\boldsymbol{\psi}_k^\top\boldsymbol{\psi}_l m_k\,\mathbb{E}[m_l \mid \mathcal{F}_{l-1}]\big] = 0$. Hence

$$\mathbb{E}\|\mathbf{N}_t\|^2 = \sum_{k=1}^t \mu^{2(t-k)}\,\mathbb{E}\big[\|\boldsymbol{\psi}_k\|^2\,\mathbb{E}[m_k^2 \mid \mathcal{F}_{k-1}]\big] \leq \frac{B^2\sigma_e^2}{1-\mu^2},$$

and by Jensen's inequality $\mathbb{E}\|\mathbf{N}_t\| \leq B\sigma_e/\sqrt{1-\mu^2}$. *Initialisation:* $\|\mu^t p_0^{-1}(\hat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_t^\star)\| = \mu^t p_0^{-1}\|\boldsymbol{\beta}_t^\star - \hat{\boldsymbol{\beta}}_0\|$. Adding the four bounds and using $(1-\mu)/\sqrt{1-\mu^2} = \sqrt{(1-\mu)/(1+\mu)}$ gives Eq. (18). The decomposition follows the tracking analysis of Guo & Ljung (1995). □

Evidence. On a simulated stream with bounded drift and noise and $\mu = 0.98$, the realised tracking error of the released estimator stays below Eq. (18) at every step (mean 0.015, maximum 0.036 over 2500 steps), and the self-tests recover a planted $\boldsymbol{\beta}^\star$ and track its drift.

The forgetting factor trades the lag of the estimate against its noise, the first and third terms of Eq. (18). For a scalar gain the trade-off is exact:

Proposition C.5 (Choice of the forgetting factor)**.** Let a scalar gain $\beta_t = \beta_{t-1} + w_t$ be observed as $y_t = \beta_t + e_t$ through the constant regressor $\psi \equiv 1$, where (w_t) and (e_t) are independent zero-mean white sequences with variances σ_w^2 and σ_e^2, independent of β_0, and $\mathbb{E}\beta_0^2 < \infty$. Run Eq. (13) with $\mu \in (0,1)$ from any $p_0 > 0$. Its gain $k_t = P_t$ converges to $1-\mu$, so the recursion becomes the exponential smoother $\hat{\beta}_t = \mu\hat{\beta}_{t-1} + (1-\mu)y_t$, and its mean squared error converges to

$$V(\mu) = \frac{\mu^2\sigma_w^2}{1-\mu^2} + \frac{(1-\mu)\,\sigma_e^2}{1+\mu}. \tag{19}$$

The first term (lag) decreases and the second (noise) increases as μ decreases; to leading order in $1-\mu$, V is minimised at $1-\mu^\star = \sigma_w/\sigma_e$, where $V(\mu^\star) \approx \sigma_w\sigma_e$.

Proof. By Lemma C.3, $P_t = \mathcal{I}_t^{-1} = \big(\mu^t/p_0 + (1-\mu^t)/(1-\mu)\big)^{-1} \to 1-\mu$, and $k_t = P_t\psi = P_t$, a deterministic sequence. The error $\varepsilon_t = \hat{\beta}_t - \beta_t$ satisfies $\varepsilon_t = (1-k_t)(\varepsilon_{t-1} - w_t) + k_t e_t$. The variable ε_{t-1} depends on β_0, $\hat{\beta}_0$ and $(w_s, e_s)_{s<t}$ only, and is therefore independent of w_t and e_t, which are independent of each other and have mean zero. Hence $V_t := \mathbb{E}\varepsilon_t^2$ obeys $V_t = a_t V_{t-1} + c_t$ with $a_t = (1-k_t)^2 \to \mu^2 < 1$ and $c_t = (1-k_t)^2\sigma_w^2 + k_t^2\sigma_e^2 \to c := \mu^2\sigma_w^2 + (1-\mu)^2\sigma_e^2$. With $V = c/(1-\mu^2)$, $|V_t - V| \le a_t|V_{t-1} - V| + |a_t - \mu^2|V + |c_t - c|$; since $a_t \le \bar{a} < 1$ for large t and the last two terms vanish, $V_t \to V = V(\mu)$. With $\nu = 1-\mu$, $\mu^2/(1-\mu^2) = 1/(2\nu) + O(1)$ and $(1-\mu)/(1+\mu) = \nu/2 + O(\nu^2)$, so the leading part of V is $\sigma_w^2/(2\nu) + \nu\sigma_e^2/2$, minimised at $\nu^\star = \sigma_w/\sigma_e$ with value $\sigma_w\sigma_e$. □

Evidence. For $\sigma_w = 0.01$ and $\sigma_e = 0.2$, the simulated steady-state mean squared errors are 0.00255, 0.00281 and 0.0481 at $\mu = 0.9$, 0.98 and 0.999, against 0.00253, 0.00283 and 0.0500 from Eq. (19); the predicted optimum is $1-\mu^\star = 0.05$.

The bound Eq. (18) scales with $1/\alpha$. How large α can be depends on how the channels are normalised, and on the policies that generate them.

Proposition C.6 (Centring and excitation)**.** Let $\mathbf{x}_1, \dots, \mathbf{x}_t \in \mathbb{R}^r$ carry weights $w_k \ge 0$ with $\sum_k w_k = 1$, with weighted mean $\mathbf{m} = \sum_k w_k\mathbf{x}_k$ and covariance $\Sigma = \sum_k w_k(\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^\top$, and let $M = \sum_k w_k\boldsymbol{\psi}_k\boldsymbol{\psi}_k^\top$ for $\boldsymbol{\psi}_k = [1, \mathbf{x}_k^\top]^\top$. **(i)** For every unit $\mathbf{u} \in \mathbb{R}^r$, $\lambda_{\min}(M) \le \mathbf{u}^\top\Sigma\mathbf{u}/\big(1 + (\mathbf{m}^\top\mathbf{u})^2\big)$; in particular $\lambda_{\min}(M) \le \lambda_{\max}(\Sigma)/(1 + \|\mathbf{m}\|^2)$. **(ii)** If $\mathbf{m} = \mathbf{0}$, then $M = \mathrm{diag}(1, \Sigma)$ and $\lambda_{\min}(M) = \min\{1, \lambda_{\min}(\Sigma)\}$. **(iii)** With $w_k = \mu^{t-k}(1-\mu)/(1-\mu^t)$, the constant of Theorem C.4 satisfies $\alpha \le (1-\mu)\mu^t/p_0 + \lambda_{\min}(M)$.

Proof. (i) Let $\mathbf{v} = [-\mathbf{m}^\top\mathbf{u},\ \mathbf{u}^\top]^\top$. Then $\mathbf{v}^\top\boldsymbol{\psi}_k = \mathbf{u}^\top(\mathbf{x}_k - \mathbf{m})$, so $\mathbf{v}^\top M\mathbf{v} = \sum_k w_k\big(\mathbf{u}^\top(\mathbf{x}_k - \mathbf{m})\big)^2 = \mathbf{u}^\top\Sigma\mathbf{u}$, while $\|\mathbf{v}\|^2 = 1 + (\mathbf{m}^\top\mathbf{u})^2$, and the Rayleigh quotient bounds $\lambda_{\min}(M)$. Taking $\mathbf{u} = \mathbf{m}/\|\mathbf{m}\|$ (any unit vector if $\mathbf{m} = \mathbf{0}$) and $\mathbf{u}^\top\Sigma\mathbf{u} \le \lambda_{\max}(\Sigma)$ gives the second bound. (ii) In general $M = \left[\begin{smallmatrix} 1 & \mathbf{m}^\top \\ \mathbf{m} & \Sigma + \mathbf{m}\mathbf{m}^\top \end{smallmatrix}\right]$. (iii) $\mathcal{I}_t = \mu^t p_0^{-1} I + \frac{1-\mu^t}{1-\mu} M$, so $\mathcal{I}_t \succeq \frac{\alpha}{1-\mu} I$ requires $\alpha \le (1-\mu)\lambda_{\min}(\mathcal{I}_t) = (1-\mu)\mu^t/p_0 + (1-\mu^t)\lambda_{\min}(M)$. □

An offset of the channels that is large relative to their spread therefore caps the excitation: once μ^t is negligible, the bound Eq. (18), which is proportional to $1/\alpha$, is at least $(1 + \|\mathbf{m}\|^2)/\lambda_{\max}(\Sigma)$ times its value at $\alpha = 1$. The affine normalisation of Section B.1 centres each channel on a reference and scales it, aiming at $\mathbf{m} \approx \mathbf{0}$ and unit variances, where (ii) allows α up to $\min\{1, \lambda_{\min}(\mathcal{R})\}$ with $\mathcal{R}$ the correlation matrix of the channels.

Evidence. Centring on the reference lowers the condition number of the design matrix from 594 to 7.7 on the lift. On URB the uncentred regressor has condition number 1.3×10^5; after centring it is 24 and 336 for segregated and mixed road classes, where planted laws are recovered with errors 0.001 and 0.012. On the 22-inverter feeder, normalising each inverter's channels separately gives 720, whereas a single scale shared by all inverters leaves between 10^4 and 10^6.

Lemma C.7 (Excitation is bounded by policy indecision)**.** Let agent i's regressor be $\boldsymbol{\psi} = \Psi(\mathbf{c})$, a function of the choices $\mathbf{c} = (c_j)_{j\in\mathcal{N}_i}$ of the set $\mathcal{N}_i$ formed by agent i and its co-present peers,

and let $\mathbf{c}$ have any joint distribution. **(i) Discrete choices.** If Ψ takes values in a set of diameter D and π_j is the marginal distribution of c_j, then

$$\operatorname{tr}\operatorname{Cov}(\boldsymbol{\psi}) \le D^2 \sum_{j\in\mathcal{N}_i} \Big(1 - \max_c \pi_j(c)\Big) \le D^2 \sum_{j\in\mathcal{N}_i} H(\pi_j),$$

with H the entropy in nats. **(ii) Continuous choices.** If Ψ is L-Lipschitz, then $\operatorname{tr}\operatorname{Cov}(\boldsymbol{\psi}) \le L^2 \sum_{j\in\mathcal{N}_i} \operatorname{tr}\operatorname{Cov}(c_j)$. **(iii)** If $d \ge 2$, then $\lambda_{\min}(\mathbb{E}[\boldsymbol{\psi}\boldsymbol{\psi}^\top]) \le \operatorname{tr}\operatorname{Cov}(\boldsymbol{\psi})$.

Proof. For any fixed vector $\mathbf{p}$, $\operatorname{tr}\operatorname{Cov}(\boldsymbol{\psi}) = \mathbb{E}\|\boldsymbol{\psi} - \mathbb{E}\boldsymbol{\psi}\|^2 \le \mathbb{E}\|\boldsymbol{\psi} - \mathbf{p}\|^2$, because the mean minimises the expected squared distance. (i) Take $\mathbf{p} = \Psi(\mathbf{c}^\circ)$ with $c_j^\circ \in \arg\max_c \pi_j(c)$. Then $\|\boldsymbol{\psi} - \mathbf{p}\| \le D$, and $\boldsymbol{\psi} = \mathbf{p}$ unless some $c_j \ne c_j^\circ$, so the union bound, which needs no independence, gives $\mathbb{E}\|\boldsymbol{\psi} - \mathbf{p}\|^2 \le D^2 \Pr(\exists j : c_j \ne c_j^\circ) \le D^2 \sum_j \big(1 - \max_c \pi_j(c)\big)$. Moreover $1 - \max_c \pi_j(c) \le -\log \max_c \pi_j(c) \le H(\pi_j)$, by $1 - x \le -\log x$ and $H(\pi) = \sum_c \pi(c)\big(-\log \pi(c)\big) \ge -\log \max_c \pi(c)$. (ii) Take $\mathbf{p} = \Psi(\mathbb{E}\mathbf{c})$; then $\mathbb{E}\|\boldsymbol{\psi} - \mathbf{p}\|^2 \le L^2 \mathbb{E}\|\mathbf{c} - \mathbb{E}\mathbf{c}\|^2 = L^2 \sum_j \operatorname{tr}\operatorname{Cov}(c_j)$. (iii) When $d \ge 2$ there is a unit vector $\mathbf{v} \perp \mathbb{E}\boldsymbol{\psi}$, and $\mathbf{v}^\top \mathbb{E}[\boldsymbol{\psi}\boldsymbol{\psi}^\top]\mathbf{v} = \mathbb{E}\big[(\mathbf{v}^\top(\boldsymbol{\psi} - \mathbb{E}\boldsymbol{\psi}))^2\big] = \mathbf{v}^\top \operatorname{Cov}(\boldsymbol{\psi})\mathbf{v} \le \operatorname{tr}\operatorname{Cov}(\boldsymbol{\psi})$. □

Applied to the discounted empirical distribution of the choices, which is the weighting of Proposition C.6(iii), the lemma bounds α by the indecision of agent i and its peers: identifying the coupling needs the very variability that convergent learning removes, the dual-control dilemma (Feldbaum, 1960) with a population in place of a single controller. Identifying the coupling operator itself would need excitation in as many directions as it has entries; declaring W leaves only d gains to identify, and the confidence of Proposition C.8 handles the directions that remain unexcited.

Evidence. On URB, fitting the channel sensitivities online instead of declaring W produced a coefficient of 0.13 ± 5.0, never significant in 5004 windows, as the lemma predicts for converging route choices.

When excitation ceases, the recursion keeps accurate predictions along the directions it has seen and loses them along the others. The confidence Eq. (14) reports exactly this.

Proposition C.8 (What the confidence measures)**.** Let $\nu > 0$ and run Eq. (13) with $\mu \in (0, 1)$. **(i) Prior.** At $P = p_0 I$, $\text{conf} = 1/(1 + \nu)$ for every $\boldsymbol{\psi} \ne \mathbf{0}$. **(ii) Unexcited directions.** If $\boldsymbol{\psi}_k \in \mathcal{V}$ for all $k > t_0$, for a proper subspace $\mathcal{V} \subset \mathbb{R}^d$, then every unit $\mathbf{v} \perp \mathcal{V}$ satisfies $\mathbf{v}^\top P_t^{-1}\mathbf{v} = \mu^{t-t_0}\mathbf{v}^\top P_{t_0}^{-1}\mathbf{v}$. Hence $\operatorname{tr} P_t \to \infty$, and for every $\boldsymbol{\psi} \notin \mathcal{V}$, $\boldsymbol{\psi}^\top P_t \boldsymbol{\psi}$ grows at least as μ^{-t} and $\operatorname{conf}(\boldsymbol{\psi}) \to 0$. **(iii) Excited directions.** If $\boldsymbol{\psi}_k = \boldsymbol{\psi}_c$ for all $k > t_0$, then $\boldsymbol{\psi}_c^\top P_t \boldsymbol{\psi}_c \to 1 - \mu$ and $\operatorname{conf}(\boldsymbol{\psi}_c) \to \big(1 + \nu(1-\mu)/(p_0\|\boldsymbol{\psi}_c\|^2)\big)^{-1}$. **(iv) Identifiability.** Let the regressors be generated from past data, let every $\boldsymbol{\psi}_k$ lie in $\mathcal{V}$, and let the noise $e_k = y_k - \boldsymbol{\psi}_k^\top \boldsymbol{\beta}$ have a conditional law given the past that does not depend on $\boldsymbol{\beta}$. Then for every $\mathbf{v} \perp \mathcal{V}$ and $s \in \mathbb{R}$ the data have the same law under $\boldsymbol{\beta}$ and $\boldsymbol{\beta} + s\mathbf{v}$, and no estimator of $\mathbf{v}^\top \boldsymbol{\beta}$ with finite mean is unbiased.

Proof. (i) $\boldsymbol{\psi}^\top(p_0 I)\boldsymbol{\psi}/(p_0\|\boldsymbol{\psi}\|^2) = 1$. (ii) By Lemma C.3, $P_t^{-1} = \mu^{t-t_0} P_{t_0}^{-1} + \sum_{t_0 < k \le t} \mu^{t-k} \boldsymbol{\psi}_k \boldsymbol{\psi}_k^\top$, and $\mathbf{v}^\top \boldsymbol{\psi}_k = 0$ for $k > t_0$, which gives the identity. For any $\mathbf{u} \ne \mathbf{0}$, the Cauchy–Schwarz inequality applied to $P_t^{1/2}\boldsymbol{\psi}$ and $P_t^{-1/2}\mathbf{u}$ gives $(\boldsymbol{\psi}^\top \mathbf{u})^2 \le (\boldsymbol{\psi}^\top P_t \boldsymbol{\psi})(\mathbf{u}^\top P_t^{-1}\mathbf{u})$. With $\boldsymbol{\psi} = \mathbf{u} = \mathbf{v}$, $\operatorname{tr} P_t \ge \mathbf{v}^\top P_t \mathbf{v} \ge 1/(\mathbf{v}^\top P_t^{-1}\mathbf{v}) = \mu^{-(t-t_0)}/(\mathbf{v}^\top P_{t_0}^{-1}\mathbf{v})$. For $\boldsymbol{\psi} \notin \mathcal{V}$, let $\mathbf{u} \ne \mathbf{0}$ be its component orthogonal to $\mathcal{V}$; then $\boldsymbol{\psi}^\top \mathbf{u} = \|\mathbf{u}\|^2$ and $\boldsymbol{\psi}^\top P_t \boldsymbol{\psi} \ge \mu^{-(t-t_0)}\|\mathbf{u}\|^4/(\mathbf{u}^\top P_{t_0}^{-1}\mathbf{u})$, so $\operatorname{conf}(\boldsymbol{\psi}) \le p_0\|\boldsymbol{\psi}\|^2/(\nu\, \boldsymbol{\psi}^\top P_t \boldsymbol{\psi}) \to 0$. (iii) Write $Q = P_{t_0}^{-1}$, $\epsilon = \mu^{t-t_0}$ and $s_t = \sum_{j=0}^{t-t_0-1} \mu^j = (1-\epsilon)/(1-\mu)$, so that $P_t^{-1} = \epsilon Q + s_t \boldsymbol{\psi}_c \boldsymbol{\psi}_c^\top$. With $q = \boldsymbol{\psi}_c^\top Q^{-1} \boldsymbol{\psi}_c > 0$, the Sherman–Morrison formula gives

$$\boldsymbol{\psi}_c^\top P_t \boldsymbol{\psi}_c = \frac{q}{\epsilon} - \frac{s_t (q/\epsilon)^2}{1 + s_t q/\epsilon} = \frac{q}{\epsilon + s_t q} \longrightarrow \frac{q}{q/(1-\mu)} = 1 - \mu \qquad (\epsilon \to 0),$$

and Eq. (14) gives the limit of $\mathrm{conf}(\boldsymbol{\psi}_c)$. (iv) Since $\boldsymbol{\psi}_k^\top \mathbf{v} = 0$, $y_k = \boldsymbol{\psi}_k^\top(\boldsymbol{\beta} + s\mathbf{v}) + e_k$ with the same e_k, so the conditional law of $(\boldsymbol{\psi}_k, y_k)$ given the past is the same under both parameters, and by induction over k so is the joint law of the data. An estimator T therefore has the same mean under $\boldsymbol{\beta} + s\mathbf{v}$ for every s, whereas $\mathbf{v}^\top(\boldsymbol{\beta} + s\mathbf{v}) = \mathbf{v}^\top\boldsymbol{\beta} + s\|\mathbf{v}\|^2$ varies with s. □

The confidence therefore falls exactly along the directions that the data leave unidentified (iv), and along the directions it has seen it stays bounded away from zero however long excitation has ceased (iii). The trace of P_i, in contrast, diverges as soon as the regressors stop spanning $\mathbb{R}^d$ (ii), even while the predictions in use remain accurate.

Evidence. With $\mu = 0.999$, $p_0 = 10$, $\nu = 4$ and the regressor frozen, the confidence at the regressor in use reaches 0.9998 after 6000 steps, the limit in (iii), with $\boldsymbol{\psi}_c^\top P \boldsymbol{\psi}_c = 0.00100 = 1 - \mu$; along an unexcited unit direction $\mathbf{v}^\top P_t^{-1}\mathbf{v}$ equals μ^t/p_0 to the printed digits (2.471×10^{-4} at $t = 6000$); and $\mathrm{tr}\, P$ grows to 1.21×10^4, so a confidence based on the trace, $1/(1 + \mathrm{tr}\, P/p_0)$, falls to 0.0008. In SUMO the confidence opens at exactly $1/(1+\nu) = 0.200$. On URB, a trace-based confidence fell from 0.20 to 0.005 over 4200 days while the fit R^2 stayed at 0.9998.

C.3 Intervention and return

Theorem 4.1 (Compensation residual)**.** Fix an agent and a step below saturation, and let $y_i^\star = \boldsymbol{\beta}_i^{\star\top}\boldsymbol{\psi}_i$, which there equals the true excess $\bar{y}_i$ (Proposition C.2). **(i) Additive** (feeder, in units of rating $a_i = q_i/\bar{s}_i$):

$$a_i^{\mathrm{ex}} = a_i^{\mathrm{cmd}} + \epsilon_i, \qquad \epsilon_i = (1 - g_i)\, y_i^\star + g_i\big(\boldsymbol{\beta}_i^\star - \hat{\boldsymbol{\beta}}_i\big)^\top \boldsymbol{\psi}_i,$$
$$|\epsilon_i| \le (1 - g_i)\, |y_i^\star| + g_i \left\|\boldsymbol{\beta}_i^\star - \hat{\boldsymbol{\beta}}_i\right\| \|\boldsymbol{\psi}_i\|.$$

(ii) Multiplicative (lift, $y_i^\star = \delta_i \in [0, \delta_{\max}]$): with $\hat{c}_i = \mathrm{clip}(g_i\hat{y}_i, 0, \delta_{\max})$,

$$a_i^{\mathrm{ex}} = \frac{1 - \delta_i}{1 - \hat{c}_i}\, a_i^{\mathrm{cmd}}, \qquad \left\|a_i^{\mathrm{ex}} - a_i^{\mathrm{cmd}}\right\| \le \frac{(1 - g_i)\,\delta_i + g_i|\hat{y}_i - \delta_i|}{1 - \delta_{\max}} \left\|a_i^{\mathrm{cmd}}\right\|.$$

In both cases, $g_i = 1$ with an exact estimate gives $a_i^{\mathrm{ex}} = a_i^{\mathrm{cmd}}$; if this holds for every agent, the harmed step has the nominal kernel P_0.

Proof of Theorem 4.1. (i) In units of rating, Eq. (15) reads $a_i^{\mathrm{sent}} = a_i^{\mathrm{cmd}} - g_i\hat{y}_i$. The reaction d_i in Eq. (8) depends on the peers' actions at the previous step only, and below saturation $a_i^{\mathrm{ex}} = a_i^{\mathrm{sent}} + d_i/\bar{s}_i = a_i^{\mathrm{sent}} + y_i^\star$ by Proposition C.2. Hence $a_i^{\mathrm{ex}} - a_i^{\mathrm{cmd}} = y_i^\star - g_i\hat{y}_i = (1-g_i)y_i^\star + g_i(y_i^\star - \hat{y}_i)$ with $y_i^\star - \hat{y}_i = (\boldsymbol{\beta}_i^\star - \hat{\boldsymbol{\beta}}_i)^\top\boldsymbol{\psi}_i$, and the bound follows from the triangle and Cauchy–Schwarz inequalities. (ii) Below saturation $a_i^{\mathrm{ex}} = (1 - \delta_i)a_i^{\mathrm{sent}} = \frac{1-\delta_i}{1-\hat{c}_i}a_i^{\mathrm{cmd}}$, so $a_i^{\mathrm{ex}} - a_i^{\mathrm{cmd}} = \frac{\hat{c}_i - \delta_i}{1-\hat{c}_i}\, a_i^{\mathrm{cmd}}$. The clip to $[0, \delta_{\max}]$ is the projection onto an interval that contains δ_i, hence 1-Lipschitz with δ_i as a fixed point, so $|\hat{c}_i - \delta_i| \le |g_i\hat{y}_i - \delta_i| = |g_i(\hat{y}_i - \delta_i) - (1 - g_i)\delta_i| \le g_i|\hat{y}_i - \delta_i| + (1 - g_i)\delta_i$. Moreover $1 - \hat{c}_i \ge 1 - \delta_{\max} > 0$. With $g_i = 1$ and $\hat{y}_i = y_i^\star$, both residuals vanish, in (ii) because $\hat{c}_i = \mathrm{clip}(\delta_i, 0, \delta_{\max}) = \delta_i$. If this holds for every agent, $\mathbf{a}^{\mathrm{ex}} = \mathbf{a}^{\mathrm{cmd}}$ and Eq. (4) gives $P_0(s' \mid s, \mathbf{a}^{\mathrm{cmd}})$. □

Evidence. On the feeder the in-environment oracle, whose estimate is exact, leaves exactly 0.100 of $|d_i|$ at trust 0.9, the first term of (i), and a correct estimate at full trust cancels the reaction to 3.7×10^{-9} MVAr. On the lift, exact compensation restores every executed force magnitude to its command. On the 6-inverter feeder at $\sigma = 3$, with a scripted droop host, compensation lowers the closed-loop disturbance from 9.7% to 1.7% of the reactive-power range.

Theorem C.9 (Return under imperfect compensation)**.** Let the harm act on actions (feeder, lift), with a bounded reward and P_0 both depending on the executed action. Let π be a stationary policy of $\mathcal{G}_0$ (on states, or on observations that are functions of the state) whose nominal action value $Q_0^\pi(s, \cdot)$ is L_Q-Lipschitz in the joint action for every s. Run π through PACT in the harmed game with any trust and estimator schedule, let $J_\sigma^g(\pi)$ be the resulting return and

$\boldsymbol{\epsilon}(t) = \mathbf{a}^{\mathrm{ex}}(t) - \mathbf{a}^{\mathrm{cmd}}(t)$ the joint residual, where $\mathbf{a}^{\mathrm{cmd}}(t) \sim \pi$ is the host's command. Then

$$\big|J_\sigma^g(\pi) - J_0(\pi)\big| \le L_Q \sum_{t\ge 0} \gamma^t \,\mathbb{E}\big\|\boldsymbol{\epsilon}(t)\big\|. \tag{20}$$

Proof of Theorem C.9. Let $V = V_0^\pi$ and $Q = Q_0^\pi$ be the nominal value functions of π, which are bounded because the reward is, and let h_t be the history up to s_t, including the estimator and trust states. Under PACT, the command $\mathbf{a}_t = \mathbf{a}^{\mathrm{cmd}}(t)$ is drawn from $\pi(\cdot \mid s_t)$ given h_t, the executed action $\mathbf{a}_t^{\mathrm{ex}} = \mathbf{a}_t + \boldsymbol{\epsilon}(t)$ is a function of h_t and $\mathbf{a}_t$, the reward is $R(s_t, \mathbf{a}_t^{\mathrm{ex}})$, and $s_{t+1} \sim P_0(\cdot \mid s_t, \mathbf{a}_t^{\mathrm{ex}})$ given $(h_t, \mathbf{a}_t)$. Telescoping and letting the horizon grow, with $\gamma^T \mathbb{E}V(s_T) \to 0$ and dominated convergence,

$$J_\sigma^g(\pi) - J_0(\pi) = \sum_{t\ge 0} \gamma^t \,\mathbb{E}\big[R(s_t, \mathbf{a}_t^{\mathrm{ex}}) + \gamma V(s_{t+1}) - V(s_t)\big].$$

Conditioning on $(h_t, \mathbf{a}_t)$ turns the first two terms into $R(s_t, \mathbf{a}_t^{\mathrm{ex}}) + \gamma \mathbb{E}_{s' \sim P_0(\cdot \mid s_t, \mathbf{a}_t^{\mathrm{ex}})} V(s') = Q(s_t, \mathbf{a}_t^{\mathrm{ex}})$, and conditioning on h_t gives $\mathbb{E}V(s_t) = \mathbb{E}Q(s_t, \mathbf{a}_t)$ because $\mathbf{a}_t \sim \pi(\cdot \mid s_t)$. Therefore $J_\sigma^g(\pi) - J_0(\pi) = \sum_t \gamma^t \,\mathbb{E}\big[Q(s_t, \mathbf{a}_t + \boldsymbol{\epsilon}(t)) - Q(s_t, \mathbf{a}_t)\big]$, and the Lipschitz property bounds each term by $L_Q \mathbb{E}\|\boldsymbol{\epsilon}(t)\|$. This is the performance difference lemma (Kakade & Langford, 2002) applied to the executed action. □

Evidence. On the 22-inverter feeder at $\sigma = 1$, with a scripted IEEE 1547 droop host over three daytime episodes, the blind fleet's loss is 43.2 times that of the same host at $\sigma = 0$ and its power flow diverges in two of the three episodes; with PACT the loss is 1.028 times the nominal one and no episode diverges, against 0.998 for the oracle. At $\sigma = 2$ the three figures are 181.8, 1.360 and 0.996. When a MATD3 host is trained at $\sigma = 3$ (5 seeds), PACT reaches the level of the $\sigma = 0$ policy from episode ≈120, whereas the blind host needs ≈360 episodes and ends ≈24% worse.

On the feeder the three theorems compose into a bound whose only ingredients are the trust, the rate of change of the driver and the excitation.

Corollary C.10 (End-to-end bound on the feeder)**.** On the feeder, let every step be below saturation with the sensor clip inactive, and let agent i run Eq. (13) on its rows from step 0, with $\|\boldsymbol{\psi}_i\| \le B$ and normalisation $\tilde{\mathbf{x}}_i = U_i(\mathbf{x}_i - \bar{\mathbf{x}}_i)$. Write $\dot{A} = \max_t |A(t{+}1) - A(t)|$ and $\Delta_i = \sigma k \,\|\mathrm{send}\| \,\big(\|\bar{\mathbf{x}}_i\| + \|U_i^{-1}\|\big)\,\dot{A}$. At every step $t \ge 1$ at which the persistent-excitation condition of Theorem C.4 holds with constant α,

$$\big\|\hat{\boldsymbol{\beta}}_i(t) - \boldsymbol{\beta}_i^\star(t)\big\| \le \Delta_i\Big(1 + \frac{B^2\mu}{\alpha(1-\mu)}\Big) + \frac{(1-\mu)\,\mu^t}{\alpha\, p_0}\big\|\boldsymbol{\beta}_i^\star(t{-}1) - \hat{\boldsymbol{\beta}}_i(0)\big\|, \tag{21}$$

and at every step the residual of Theorem 4.1(i) satisfies $|\epsilon_i(t)| \le (1 - g_i(t))\,|y_i^\star(t)| + B\,\|\hat{\boldsymbol{\beta}}_i(t) - \boldsymbol{\beta}_i^\star(t)\|$. For a host π as in Theorem C.9, with L_Q measured for actions in units of rating,

$$\big|J_\sigma^g(\pi) - J_0(\pi)\big| \le L_Q \sum_{t\ge 0} \gamma^t \sum_i \mathbb{E}\big|\epsilon_i(t)\big|. \tag{22}$$

Proof. Below saturation with the sensor clip inactive, $e_i \equiv 0$ (Proposition C.2), so Theorem C.4 applies with $\bar{b} = \sigma_e = 0$ and its bound holds pathwise. At time t the recursion has processed the rows of steps $0, \dots, t-1$; indexed $k = 1, \dots, t$, row k carries the gain $\boldsymbol{\beta}_i^\star(k-1)$. By the proof of Proposition C.2, $\boldsymbol{\beta}_i^\star(t) = [\boldsymbol{\theta}(t)^\top \bar{\mathbf{x}}_i,\ (U_i^{-\top}\boldsymbol{\theta}(t))^\top]^\top$ with $\boldsymbol{\theta}(t) = -\sigma k A(t)\mathrm{send}$, so $\|\boldsymbol{\beta}_i^\star(t{+}1) - \boldsymbol{\beta}_i^\star(t)\| \le \|\boldsymbol{\theta}(t{+}1) - \boldsymbol{\theta}(t)\|\big(\|\bar{\mathbf{x}}_i\| + \|U_i^{-1}\|\big) \le \Delta_i$. Theorem C.4 with drift Δ_i bounds $\|\hat{\boldsymbol{\beta}}_i(t) - \boldsymbol{\beta}_i^\star(t{-}1)\|$ by the right side of Eq. (21) minus Δ_i, and the triangle inequality with $\|\boldsymbol{\beta}_i^\star(t{-}1) - \boldsymbol{\beta}_i^\star(t)\| \le \Delta_i$ gives Eq. (21). The residual bound is Theorem 4.1(i) with $g_i \le 1$ and $\|\boldsymbol{\psi}_i\| \le B$, and Eq. (22) is Theorem C.9 with $\|\boldsymbol{\epsilon}(t)\| \le \sum_i |\epsilon_i(t)|$. □

On the feeder, the identification error is thus driven by how fast the gains $\sigma A(t)$ change, through $\sigma\dot{A}$, rather than by their level, and the uncompensated share $(1 - g_i)|y_i^\star|$ is the price of partial trust.

Evidence. The feeder self-test tracks the excess through the solar day with one-step errors of 0.66, 0.40 and 0.14 of $|y_i|$ at $\mu = 0.999$, 0.98 and 0.9, falling with μ as the drift term of Eq. (21) predicts in this noise-free regime, and with an error of 0.000 when $A(t)$ multiplies the channels in the regressor, which makes the coefficients constant ($\Delta_i = 0$).

Proposition 4.2 (Containment of the host)**.** If $g_i = 0$, then $q_i^{\text{sent}} = q_i^{\text{cmd}}$, $a_i^{\text{sent}} = a_i^{\text{cmd}}$ and $\tilde{\ell} = \ell$ for every estimator state. Hence the PACT policy class contains every host policy π, and the best trust schedule attains at least the host's return $J_\sigma(\pi)$ in the harmed game.

Proof of Proposition 4.2. With $g_i = 0$, Eq. (15) returns $q_i^{\text{sent}} = q_i^{\text{cmd}}$ and $a_i^{\text{sent}} = a_i^{\text{cmd}}/(1 - \text{clip}(0, 0, \delta_{\max})) = a_i^{\text{cmd}}$, and Eq. (16) returns $\tilde{\ell} = \ell$ because z is finite ($z \equiv 0$ when all predictions coincide); none of these depends on the estimator state. The schedule $g^{\text{pol}} \equiv 0$ gives $g_i = g_i^{\text{pol}} \text{conf}_i = 0$, so it reproduces every host policy π exactly in the harmed game, and the supremum over trust schedules is at least $J_\sigma(\pi)$. □

Evidence. With $g_i = 0$ and an estimate of scale 10^6, the executed action on the feeder equals the command bit for bit, and the steering shift on URB is exactly zero over 200 draws; ties give exactly zero shift, and the torch and numpy implementations agree. In the SUMO runs the start-up check confirms that $g = 0$ leaves the logits unchanged while $g = 1$ moves them by 1.45, with a torch/numpy parity of 2.2×10^{-16}.

Between the exact containment at $g = 0$ and full reliance, the steering channel moves the host's decisions continuously in the trust:

Proposition C.11 (Steering stays close to the host)**.** Let $\pi = \text{softmax}(\ell)$ be the host's distribution over the K candidates, and let $\tilde{\pi} = \text{softmax}(\tilde{\ell})$ with $\tilde{\ell}$ from Eq. (16) at trust $g \in [0, 1]$. Then $\max_k z_k - \min_k z_k \le \sqrt{2K}$, and

$$D_{\text{KL}}(\pi \,\|\, \tilde{\pi}) \le \frac{g^2\kappa^2 K}{4}, \qquad \text{TV}(\pi, \tilde{\pi}) \le g\kappa\sqrt{K/8}, \qquad \left|\mathbb{E}_{\tilde{\pi}} f - \mathbb{E}_\pi f\right| \le R\, g\kappa\sqrt{K/8}$$

for every function f of the chosen candidate with range R.

Proof. If all predictions coincide, $z \equiv 0$ and all three quantities vanish. Otherwise the population z-scores satisfy $\sum_k z_k = 0$ and $\sum_k z_k^2 = K$, and for the largest and smallest entries $(z_{\max} - z_{\min})^2 \le 2(z_{\max}^2 + z_{\min}^2) \le 2K$. With $\lambda = g\kappa$, $\tilde{\pi}(k) = \pi(k)e^{-\lambda z_k}/\mathbb{E}_\pi[e^{-\lambda z}]$, hence

$$D_{\text{KL}}(\pi \,\|\, \tilde{\pi}) = \sum_k \pi(k) \log \frac{\pi(k)}{\tilde{\pi}(k)} = \lambda\, \mathbb{E}_\pi[z] + \log \mathbb{E}_\pi\left[e^{-\lambda z}\right] \le \frac{\lambda^2 (z_{\max} - z_{\min})^2}{8} \le \frac{\lambda^2 K}{4}$$

by Hoeffding's lemma (Hoeffding, 1963), $\log \mathbb{E} e^{s(Z - \mathbb{E}Z)} \le s^2(b - a)^2/8$ for $Z \in [a, b]$. Pinsker's inequality gives $\text{TV} \le \sqrt{D_{\text{KL}}/2}$. Finally, with c the midpoint of the range of f, $|\mathbb{E}_{\tilde{\pi}} f - \mathbb{E}_\pi f| = \left|\sum_k (\tilde{\pi}(k) - \pi(k))(f(k) - c)\right| \le \frac{R}{2} \sum_k |\tilde{\pi}(k) - \pi(k)| = R\, \text{TV}(\pi, \tilde{\pi})$. □

Evidence. Over 20,000 random logits and predicted costs per trust level at $K = 4$ and $\kappa = 1$, the largest divergence is 0.0095, 0.235, 0.714 and 0.891 against the bounds 0.01, 0.25, 0.81 and 1.0 at $g = 0.1$, 0.5, 0.9 and 1, and the range of z attains $\sqrt{2K} = 2.828$.

C.4 FEEDBACK OF COMPENSATION ON THE MEDIUM

Compensation feeds back into the medium, since an agent's corrected action is part of its peers' load. The two results below use stylised models, each isolating one mechanism present in the full benchmarks. On the feeder, exact compensation removes the fraction g of every reaction before it propagates, so trust sets the gain of the mutual-reaction loop. In routing, the loads that determine a trip's delay are those of the same step, which PACT forecasts from the previous step, so synchronised steering on an accurate but lagged forecast can overshoot.

Proposition C.12 (Stability margin of the feeder)**.** Consider the linear feeder model, i.e. Eq. (8) without its reaction and capability limits (the regime where neither is active). Hold the host commands $q^{\mathrm{cmd}}(t)$ exogenous, and use a common trust $g_i = g \in [0,1]$ with exact estimates. Let $G = \mathrm{diag}(\bar{s}) \sum_m \mathrm{send}_m S_m$ have spectral radius $\rho(G) > 0$, and let A be periodic with geometric mean $A_{\mathrm{geo}} > 0$ over a period. Two executed trajectories under the same commands differ by $\mathbf{v}(t) = -(1-g)\,\sigma k A(t)\,G\,\mathbf{v}(t-1)$. With loop gain $\Theta_g = (1-g)\,\sigma k A_{\mathrm{geo}}\,\rho(G)$, this mutual reaction decays for every initial condition if $\Theta_g < 1$ and grows without bound for some initial condition if $\Theta_g > 1$. The critical severity is therefore $\sigma_c(g) = \sigma_c(0)/(1-g)$ with $\sigma_c(0) = 1/(kA_{\mathrm{geo}}\rho(G))$, and at $g = 1$ the reaction is cancelled at every severity.

Proof of Proposition C.12. With exact estimates, $\bar{s}_i \hat{y}_i = \bar{s}_i y_i^\star = d_i$, so Eq. (15) gives $q^{\mathrm{sent}} = q^{\mathrm{cmd}} - g\,d$, and the linear model gives $q^{\mathrm{ex}}(t) = q^{\mathrm{sent}}(t) + d(t) = q^{\mathrm{cmd}}(t) + (1-g)\,d(t)$ with $d(t) = -\sigma k A(t)\,G\,q^{\mathrm{ex}}(t-1)$ by Eqs. (7) and (8). Subtracting two trajectories with the same commands gives the stated recursion, whose solution is $\mathbf{v}(t) = (-c)^t\,\Pi_t\,G^t\mathbf{v}(0)$ with $c = (1-g)\sigma k$ and $\Pi_t = \prod_{u=1}^{t} A(u)$. The driver enters only through Π_t. Since $A_{\mathrm{geo}} > 0$, every $A(u)$ is positive; with period T and $\underline{a} = \min_u A(u) > 0$, every block of T consecutive factors multiplies to A_{geo}^T, so $\Pi_t = A_{\mathrm{geo}}^{T\lfloor t/T \rfloor}\varpi_t$ with $\underline{a}^T \le \varpi_t \le 1$, and $\Pi_t^{1/t} \to A_{\mathrm{geo}}$. The operator enters only through G^t. Each S_m is symmetric, so with $D = \mathrm{diag}(\bar{s})$ the matrix $H = D^{-1/2} G D^{1/2} = D^{1/2}\big(\sum_m \mathrm{send}_m S_m\big) D^{1/2}$ is symmetric and similar to G. Hence $\|G^t\| = \|D^{1/2} H^t D^{-1/2}\| \le \kappa_D\,\rho(G)^t$ with $\kappa_D = (\max_i \bar{s}_i / \min_i \bar{s}_i)^{1/2}$, and G has the real eigenvector $\mathbf{u} = D^{1/2}\mathbf{w}$, where $H\mathbf{w} = \lambda\mathbf{w}$ and $|\lambda| = \rho(G)$. If $\Theta_g < 1$, then $\|\mathbf{v}(t)\| \le \kappa_D\big(c\,\Pi_t^{1/t}\rho(G)\big)^t \|\mathbf{v}(0)\|$ with $c\,\Pi_t^{1/t}\rho(G) \to \Theta_g < 1$, so $\mathbf{v}(t) \to \mathbf{0}$ for every $\mathbf{v}(0)$. If $\Theta_g > 1$, take $\mathbf{v}(0) = \mathbf{u}$: then $\|\mathbf{v}(t)\| = \big(c\,\Pi_t^{1/t}\rho(G)\big)^t \|\mathbf{u}\| \to \infty$. Solving $\Theta_g = 1$ for σ gives $\sigma_c(g)$, and at $g = 1$, $c = 0$ and $\mathbf{v}(t) = \mathbf{0}$ for $t \ge 1$. □

Remark C.13 (Drivers that vanish). The proof uses the driver only through products of A over the horizon. The solar driver vanishes at night, so $A_{\mathrm{geo}} = 0$ over a day and the reaction is reset every night; applied to any window $\mathcal{W}$ of $|\mathcal{W}|$ consecutive steps on which $A > 0$, the same argument gives $\mathbf{v}(t_1) = (-c)^{|\mathcal{W}|}\big(\prod_{u\in\mathcal{W}} A(u)\big)\,G^{|\mathcal{W}|}\mathbf{v}(t_0)$. Along $\mathbf{u}$ the reaction is thus multiplied by $\big((1-g)\,\sigma k A_{\mathcal{W}}\rho(G)\big)^{|\mathcal{W}|}$ over the window, where $A_{\mathcal{W}}$ is the geometric mean of A on $\mathcal{W}$, and trust g again multiplies by $1/(1-g)$ the severity at which the reaction grows during the window.

Evidence. The network files give $k\rho(G) = 0.378$, 0.675 and 1.186 on the feeders with 6, 22 and 38 inverters, so at the driver peak $\sigma_c(0) = 2.65$, 1.48 and 0.84: on the largest feeder the fleet is beyond the threshold already at $\sigma = 1$. Trust $g = 0.9$ moves the three thresholds to 26.5, 14.8 and 8.4. On the 6-inverter feeder at $\sigma = 3$ the linear recursion with $g = 0$ has loop gain 1.13 and diverges ($\|q^{\mathrm{ex}}\| = 0.75$, 4.9 and 621 after 5, 20 and 59 steps), whereas with $g = 0.9$ the gain is 0.113 and it converges. In the full power-flow simulation, whose limits keep the reaction bounded, the ordering persists above the threshold: on the 22-inverter feeder at $\sigma = 2$, with a scripted droop host, the blind fleet's power flow diverged in all three episodes and the compensated fleet's in none, and when a MATD3 host is trained at $\sigma = 3$ the blind fleet diverged three times early in training and the compensated fleet never, at an applied trust of 0.89.

Proposition C.14 (Oversteer in congested routing)**.** A nonatomic fleet uses two parallel routes with travel times $\tau_1(f) = \tau_1^0 + \theta_1 f$ and $\tau_2(f) = \tau_2^0 + \theta_2(1-f)$, where f is the share on route 1 and $\theta_1, \theta_2 > 0$; let $f^E \in (0,1)$ solve $\tau_1(f^E) = \tau_2(f^E)$. The host logits, held fixed, give share $f_0 \in (f^E, 1)$. PACT forecasts both travel times exactly at the previous step's share and steers by Eq. (16) with common trust g; with two routes $z_1 = -z_2 = \pm 1$, so each step shifts the host logit by $2\kappa g$ toward the route forecast cheaper. With $\mathrm{sig}(u) = 1/(1+e^{-u})$ and $\mathrm{logit} = \mathrm{sig}^{-1}$, let $f_\pm(g) = \mathrm{sig}(\mathrm{logit}\, f_0 \pm 2\kappa g)$ and $g^E = (\mathrm{logit}\, f_0 - \mathrm{logit}\, f^E)/(2\kappa)$. Starting from $f(0) = f_0$: **(i)** if $g < g^E$, then $f(t) = f_-(g)$ for all $t \ge 1$, and the realised gap $\tau_1 - \tau_2 > 0$ decreases strictly in g; **(ii)** if $g > g^E$, then $f(t)$ alternates between $f_-(g) < f^E$ and $f_+(g) > f_0$; at every step $t \ge 1$ the route forecast to be cheaper is realised dearer, and at every other step the gap exceeds the host's.

Proof of Proposition C.14. Let $\Gamma(f) = \tau_1(f) - \tau_2(f) = \tau_1^0 - \tau_2^0 - \theta_2 + (\theta_1 + \theta_2)f$, which is strictly increasing with $\Gamma(f^E) = 0$. With two candidates the population standard deviation of the forecasts is $|\hat{\tau}_1 - \hat{\tau}_2|/2$, so $z_1 = -z_2 = \operatorname{sign}(\hat{\tau}_1 - \hat{\tau}_2)$ ($z \equiv 0$ on a tie) and the steered logit difference is $\operatorname{logit} f_0 - 2\kappa g \operatorname{sign}(\hat{\tau}_1 - \hat{\tau}_2)$. Exact forecasts at the previous share give $\operatorname{sign}(\hat{\tau}_1 - \hat{\tau}_2) = \operatorname{sign}\Gamma(f(t-1))$, and in a nonatomic fleet the share equals the choice probability, so

$$f(t) = \operatorname{sig}\big(\operatorname{logit} f_0 - 2\kappa g \operatorname{sign}\Gamma(f(t-1))\big).$$

Since $f_0 > f^E$, $\Gamma(f_0) > 0$ and $f(1) = f_-(g)$. Because $\operatorname{logit} f_-(g) = \operatorname{logit} f_0 - 2\kappa g$, we have $f_-(g) > f^E$ if $g < g^E$ and $f_-(g) < f^E$ if $g > g^E$. (i) If $g < g^E$, then $\Gamma(f_-(g)) > 0$ and, by induction, $f(t) = f_-(g)$ for all $t \geq 1$. Since $f_-(g)$ decreases strictly in g and Γ increases strictly, the realised gap $\Gamma(f_-(g)) > 0$ decreases strictly in g. (ii) If $g > g^E$, then $\Gamma(f_-(g)) < 0$, so the next share is $f_+(g) > f_0 > f^E$, where $\Gamma > 0$, and the pattern repeats. Consecutive shares lie on opposite sides of f^E, so the route forecast to be cheaper at step t, which is the cheaper one at $f(t-1)$, is the dearer one at $f(t)$. At every step with share $f_+(g)$, the gap is $\Gamma(f_+(g)) > \Gamma(f_0) > 0$, the host's gap. □

Evidence. Take $\tau_1(f) = 12 + 4f$ and $\tau_2(f) = 10 + 4(1 - f)$, so that $f^E = 1/4$, with host share $f_0 = 1/2$ and $\kappa = 1$, so that $g^E = \frac{1}{2}\log 3 \approx 0.549$. Iterating the released steering function with exact one-step-lagged forecasts, the share settles at $f_-(g) = 0.401$ and 0.254 for $g = 0.2$ and 0.539, where the gap $\tau_1 - \tau_2 = 8f - 2$ falls from the host's 2 to 1.21 and 0.03, as in (i). At $g = 0.749$ the share alternates between 0.183 and 0.817 over all twelve recorded steps, as in (ii), and at the steps with share 0.817 the gap is 4.54, above the host's.

D Analysis of the Baselines

This appendix examines every baseline against the class of Definition 3.1. For each method we state the equations that define it, in the notation of its paper translated to ours where symbols clash, identify the term through which interaction-mediated drift reaches it, and derive what that term does in the class; the arms of a method are treated with it. Section D.1 proves the facts that recur and Table 8 summarises the outcome. The analysis concerns what each method can represent and identify. It does not rest on the phase being hidden: on the feeder the photovoltaic generation is informative about it, and the bounds below hold for policies that are given the driver.

We write $\mathcal{H}_i(t)$ for the information agent i holds from its own interaction when it acts at step t: its observations up to t, its sent actions, rewards and measured excess before t, its randomisation, and any exogenous signal it is given, such as the driver in the oracle arms. PACT also holds the executed actions its peers broadcast before t, which determine $\mathbf{x}_i(t)$ on the feeder and the lift. We write o_i for agent i's observation, $o_i^{\leq t}$ for its action-observation history, R_t for the reward and $\mathcal{D}$ for a replay buffer or a batch; advantages are written $\mathbb{A}$, likelihood ratios ϱ and network parameters ϑ, and other symbols of the original papers that clash with ours are renamed.

D.1 Recurring arguments

Lemma D.1 (Floors set by the information)**.** Let $\mathcal{U} \subseteq \mathcal{U}'$ be σ-algebras. **(i) Prediction.** For square-integrable Y and every $\mathcal{U}$-measurable $\hat{Y}$,

$$\mathbb{E}(Y - \hat{Y})^2 = \mathbb{E}\mathrm{Var}(Y \mid \mathcal{U}) + \mathbb{E}\big(\mathbb{E}[Y \mid \mathcal{U}] - \hat{Y}\big)^2 \;\geq\; \mathbb{E}\mathrm{Var}(Y \mid \mathcal{U}) \;\geq\; \mathbb{E}\mathrm{Var}(Y \mid \mathcal{U}').$$

(ii) Pooled regression. Let $(\tilde{s}, \mathbf{a}, R, s') \sim \mathcal{D}$, where R is the reward of the step and $s' \sim P_\sigma(\cdot \mid \tilde{s}, \mathbf{a})$ given $(\tilde{s}, \mathbf{a})$. Let the input u of a learner be a function of $\tilde{s}$, let V be bounded, and let $f^\star$ minimise $\mathbb{E}_{\mathcal{D}}\big(f(u, \mathbf{a}) - R - \gamma V(s')\big)^2$ over measurable f. Then

$$f^\star(u, \mathbf{a}) = \mathbb{E}_{\mathcal{D}}[R \mid u, \mathbf{a}] + \gamma\, \mathbb{E}_{s' \sim \bar{P}_{\mathcal{D}}(\cdot \mid u, \mathbf{a})} V(s'), \qquad \bar{P}_{\mathcal{D}}(\cdot \mid u, \mathbf{a}) = \mathbb{E}_{\mathcal{D}}\big[P_\sigma(\cdot \mid \tilde{s}, \mathbf{a}) \,\big|\, u, \mathbf{a}\big].$$

(iii) Executed action. On the feeder and the lift below saturation, let the sent action $a_i^{\mathrm{sent}}(t)$ and a target $a_i^\circ(t)$ be $\mathcal{U}$-measurable. Then, in units of rating on the feeder,

$$\mathbb{E}\big[(a_i^{\mathrm{ex}} - a_i^\circ)^2 \,\big|\, \mathcal{U}\big] \geq \mathrm{Var}(y_i^\star \mid \mathcal{U}) \quad \text{on the feeder},$$
$$\mathbb{E}\big[\|a_i^{\mathrm{ex}} - a_i^\circ\|^2 \,\big|\, \mathcal{U}\big] \geq \|a_i^{\mathrm{sent}}\|^2\, \mathrm{Var}(y_i^\star \mid \mathcal{U}) \quad \text{on the lift}.$$

(iv) Load and gain. Below saturation $y_i^\star(t) = \sigma\bar{k}A(t)\,\mathrm{send}^\top \mathbf{x}_i(t)$, with $\bar{k} = -k$ on the feeder and $\bar{k} = L/x_{\mathrm{ref}}$ on the lift. If $A(t)$ is $\mathcal{U}$-measurable, then $\mathrm{Var}(y_i^\star \mid \mathcal{U}) = \sigma^2\bar{k}^2 A(t)^2\,\mathrm{send}^\top \mathrm{Cov}(\mathbf{x}_i(t) \mid \mathcal{U})\,\mathrm{send}$; if $\mathbf{x}_i(t)$ is $\mathcal{U}$-measurable, then $\mathrm{Var}(y_i^\star \mid \mathcal{U}) = \sigma^2\bar{k}^2\big(\mathrm{send}^\top \mathbf{x}_i(t)\big)^2 \mathrm{Var}(A(t) \mid \mathcal{U})$.

Proof. (i) For $\mathcal{U}$-measurable $\hat{Y}$ the cross term $\mathbb{E}\big[(Y - \mathbb{E}[Y \mid \mathcal{U}])(\mathbb{E}[Y \mid \mathcal{U}] - \hat{Y})\big]$ vanishes by the tower property, which gives the identity. The law of total variance, $\mathrm{Var}(Y \mid \mathcal{U}) = \mathbb{E}[\mathrm{Var}(Y \mid \mathcal{U}') \mid \mathcal{U}] + \mathrm{Var}(\mathbb{E}[Y \mid \mathcal{U}'] \mid \mathcal{U})$, gives the last inequality. (ii) The measurable functions of $(u, \mathbf{a})$ are the $\sigma(u, \mathbf{a})$-measurable variables, so by (i) $f^\star = \mathbb{E}_{\mathcal{D}}[R + \gamma V(s') \mid u, \mathbf{a}]$. Since u is a function of $\tilde{s}$, the tower property and Fubini's theorem give $\mathbb{E}_{\mathcal{D}}[V(s') \mid u, \mathbf{a}] = \mathbb{E}_{\mathcal{D}}\big[\mathbb{E}_{s'\sim P_\sigma(\cdot\mid\tilde{s},\mathbf{a})} V(s') \,\big|\, u, \mathbf{a}\big] = \mathbb{E}_{s'\sim \bar{P}_{\mathcal{D}}(\cdot\mid u,\mathbf{a})} V(s')$. (iii) On the feeder $a_i^{\mathrm{ex}} = a_i^{\mathrm{sent}} + y_i^\star$ below saturation (Proposition C.2), so $a_i^{\mathrm{ex}} - a_i^\circ = Z + y_i^\star$ with $Z = a_i^{\mathrm{sent}} - a_i^\circ$ $\mathcal{U}$-measurable, and $\mathbb{E}[(Z + y_i^\star)^2 \mid \mathcal{U}] = (Z + \mathbb{E}[y_i^\star \mid \mathcal{U}])^2 + \mathrm{Var}(y_i^\star \mid \mathcal{U})$. On the lift $a_i^{\mathrm{ex}} = (1 - y_i^\star)\, a_i^{\mathrm{sent}}$, so $a_i^{\mathrm{ex}} - a_i^\circ = \mathbf{z} - y_i^\star a_i^{\mathrm{sent}}$ with $\mathbf{z} = a_i^{\mathrm{sent}} - a_i^\circ$, and $\mathbb{E}[\|\mathbf{z} - y_i^\star a_i^{\mathrm{sent}}\|^2 \mid \mathcal{U}] = \|\mathbf{z} - \mathbb{E}[y_i^\star \mid \mathcal{U}]\, a_i^{\mathrm{sent}}\|^2 + \|a_i^{\mathrm{sent}}\|^2 \mathrm{Var}(y_i^\star \mid \mathcal{U})$. (iv) By Proposition C.2, $y_i^\star = \boldsymbol{\theta}(t)^\top \mathbf{x}_i(t)$ with $\boldsymbol{\theta}(t) = \sigma\bar{k}A(t)\mathrm{send}$; given $\mathcal{U}$ it is linear in $\mathbf{x}_i(t)$ in the first case and in $A(t)$ in the second. □

Part (iii) is a property of the information set, not of the learning rule: however long it trains, a policy on $\mathcal{H}_i$ executes an action whose mean-square distance from any $\mathcal{H}_i$-measurable target, including the action the host would take without harm, is at least the floor, which vanishes only if $\mathcal{H}_i(t)$ determines $\boldsymbol{\theta}(t)^\top \mathbf{x}_i(t)$. Part (iv) separates the two factors: a method given the driver still lacks the load, and a method that sees the peers' actions still lacks the gain. PACT computes the load from the broadcasts and estimates only the gain, so its residual is governed by Theorems C.4 and 4.1 instead. In routing the harm acts on the outcome, and (i) plays the role of (iii) for the forecast of a route's cost; the information of PACT contains $\mathcal{H}_i$, so its floor is no larger. By part (ii), a value fitted on an input that omits the load or the phase is that of the averaged kernel $\bar{P}_{\mathcal{D}}$: the composition of the data, not the current condition, determines the model.

Lemma D.2 (Averages over a periodic driver)**.** Let $(b_k)_{k\ge 0}$ take values in $\mathbb{R}^d$, with $b_{k+p} = b_k$, mean $\bar{b} = \frac{1}{p}\sum_{k<p} b_k$ and $M_b = \max_k \|b_k - \bar{b}\|$. **(i) Uniform averages.** Any K consecutive terms satisfy $\big\|\frac{1}{K}\sum_{k=k_1}^{k_1+K-1} b_k - \bar{b}\big\| \le \frac{p}{K} M_b$. **(ii) Exponential averages.** For $w \in (0, 1)$, the recursion $m_{k+1} = (1 - w)m_k + w b_k$ converges to a p-periodic sequence with mean $\bar{b}$, in which each harmonic $\mathbf{c}\, e^{2\pi \mathrm{i} l k/p}$ of b, $0 < l < p$, is multiplied by a factor of modulus $w/\big|1 - (1 - w)e^{-2\pi \mathrm{i} l/p}\big|$. **(iii) Lag of a running mean.** If $b_k = b'$ for all $k \ge k_0$, periodic or not, then $\bar{b}_K = \frac{1}{K}\sum_{k<K} b_k$ satisfies $\bar{b}_K - b' = \frac{k_0}{K}\,(\bar{b}_{k_0} - b')$ for $K \ge k_0$.

Proof. (i) With $K = qp + s$, $0 \le s < p$, any qp consecutive terms sum to $qp\,\bar{b}$, so the window's sum differs from $K\bar{b}$ by s terms $b_k - \bar{b}$. (ii) The recursion is linear and time-invariant; for $b_k = \mathbf{c}\, e^{\mathrm{i}\omega k}$ it is solved by $m_k = w\,\mathbf{c}\, e^{\mathrm{i}\omega k}/(e^{\mathrm{i}\omega} - 1 + w)$, where $|e^{\mathrm{i}\omega} - (1 - w)| = |1 - (1 - w)e^{-\mathrm{i}\omega}|$, and the difference of two solutions is multiplied by $1 - w$ at every step. (iii) $\sum_{k<K} b_k = k_0 \bar{b}_{k_0} + (K - k_0)\, b'$. □

Remark D.3 (Routing is a repeated bandit for every learner). In URB a day is one step, after which the episode ends, so advantage estimation reduces to $R - V$ and temporal-difference targets to the reward. A traveller observes its departure time and how many earlier travellers of its origin–destination pair took each route that day; on `saint_arnoult` almost every pair holds one traveller, and the observation is (departure time, 0, 0, 0, 0) for about 97% of travellers on every day. For them every learner faces a repeated K-armed bandit, $K = 4$, whose costs vary with the phase of the driver and with the other travellers' same-day choices, and whatever a method conditions on the observation, a graph, a context encoder or a recurrent state, receives a constant input. The drift reaches such a learner only through its rewards, through peer actions it is given, and through the driver when it is given.

Table 8: The baselines, the term through which interaction-mediated drift reaches them, and the results that govern it. Hosts: U (URB), M (MAPDN), V (VMAS).

Baseline	Hosts	Term that meets the drift	Governed by
IPPO, IQL	UMV	local value and policy, fitted on pooled data	D.1(ii, iii)
MAPPO, MADDPG, MATD3	UMV	decentralised actor; critic's knowledge averaged out	D.1(iii)
QMIX, VDN	UV	local utilities; the state enters only the mixer	D.1(ii, iii)
HATRPO, HAPPO, HASAC	UMV	correction $M = \hat{\mathbb{A}}$ under frozen partners	C.1(iii), D.1
Recurrent actors	UMV	memory of the agent's own observations	D.1(iii)
DGN, GNN-MAPPO	UV	constant features; the gain is absent	D.1(iv), D.3
MF-Q, MF-AC	UV(M)	one pairwise term for all neighbours	exact
LIAM	UMV	reconstructs the peers; gain only via the return	D.1(i, iv)
LCPO, oracle driver	UMV	gain without load; context assumed exogenous	D.1(iv)
LILAC	M	stationarity within episodes; the prior	independence
QCD restart, RR, MASTER	UMV	piecewise-stationary model; restart	D.1
WISDOM	UMV	encoder trained on a KL term alone	D.1(iii, iv)
RMA, UP-OSI	UMV	the privileged vector is a gain	D.1(i, iv)
M3W	UMV	gate without the phase; identification	D.1(ii), C.8(iv)
DR, DORAEMON	UMV	training distribution of σ	D.1(i, iii)
ERNIE	UMV	Lipschitz penalty on the input	D.1(iii)
ESO, URLS, intercept	UMV	lagged own residual; dimension of the regressor	D.4, C.8(iv)
TPA	M	phase without load	D.1(iv)
Deep FP	UMV	uniform average over phases	D.2
PMPG (IPGA, INPG)	UMV	game as a function of the policy alone	C.1(iii), D.1(ii)

D.2 Independent learners: IPPO and IQL

IPPO (de Witt et al., 2020) trains every agent with PPO (Schulman et al., 2017) on its own observation. With $\varrho_i = \pi_i(a_i \mid o_i)/\pi_i^{\text{old}}(a_i \mid o_i)$, agent i maximises

$$\mathbb{E}\Big[\min\big(\varrho_i \hat{\mathbb{A}}_i,\ \text{clip}(\varrho_i, 1-\epsilon_c, 1+\epsilon_c)\,\hat{\mathbb{A}}_i\big)\Big] + c_H\,\mathbb{E}\,H\big(\pi_i(\cdot \mid o_i)\big), \tag{23}$$

where $\hat{\mathbb{A}}_i$ is a generalised advantage estimate built on a local value $V_i(o_i)$ fitted to returns. IQL (Tan, 1993) fits $\min_{Q_i} \mathbb{E}_{\mathcal{D}}\big(Q_i(o_i,a_i) - R_t - \gamma \max_{a'} Q_i^-(o_i', a')\big)^2$ on a replay buffer, with a target copy Q_i^-, and acts ϵ-greedily.

Both act on o_i, on MAPDN through a recurrent summary of the zone's observations, so Lemma D.1(iii) bounds their executed action on the feeder and the lift whatever they converge to; the load left in the bound weighs the reactive powers of every inverter that shares a path with agent i, including inverters outside its zone. V_i and Q_i are regressions on data spanning phases and loads, so by Lemma D.1(ii) they converge to values of the averaged kernel; the photovoltaic generation in o_i separates the phases on MAPDN, but not the loads.

On URB (Remark D.3) the host PPO has no critic: the advantage is the day's reward standardised over the batch, and an agent updates when its memory holds 64 days and then clears it, once every 64 days. For a policy $\pi_i = \text{softmax}(\ell)$ on a constant input, the gradient of Eq. (23) at the start of an update, where every ratio equals one, is

$$\frac{1}{64\,s_B}\sum_{k=1}^{K} n_k\big(\hat{R}_k - m_B\big)\nabla_\ell \log \pi_i(k) + c_H \nabla_\ell H(\pi_i),$$

with n_k the number of days of the batch on which route k was taken, $\hat{R}_k$ their mean reward, and m_B, s_B the batch mean and standard deviation. The update sees the batch only through per-route means over 64 consecutive days, 64% of the 100-day driver cycle, which always contain dry and wet days, and the resulting policy is held for the next 64 days. The host DQN regresses $Q_i(o_i, a_i)$ on the reward alone, with minibatches drawn uniformly from the last 2048 days, about 20 cycles; by Lemma D.1(ii) with $V \equiv 0$ its target for route k is the mean reward on the days of the buffer on

which k was taken, which span about 20 cycles, not today's cost, and its exploration rate decays by 0.9985 per day towards zero. PACT keeps both learners and supplies what they lack: its correction is a function of $\mathbf{x}_i(t)$, and its estimate of the gains weights past rows by μ^{t-k} rather than uniformly.

D.3 CENTRALISED CRITICS: MAPPO, MADDPG AND MATD3

MAPPO (Yu et al., 2022) keeps the actor objective Eq. (23) and fits a centralised value on the global state with a clipped loss,

$$\min_{V} \mathbb{E}\Big[\max\Big(\big(V(s)-\hat{R}\big)^2,\ \big(\operatorname{clip}\big(V(s),V^{\mathrm{old}}(s)-\epsilon_{\mathrm{c}},V^{\mathrm{old}}(s)+\epsilon_{\mathrm{c}}\big)-\hat{R}\big)^2\Big)\Big],$$

with $\hat{R}$ the return target. MADDPG (Lowe et al., 2017) learns deterministic actors $a_i=\pi_i(o_i)$ with parameters ϑ_i and centralised critics $Q_i(\mathbf{o},\mathbf{a})$ on the joint observation $\mathbf{o}=(o_1,\dots,o_n)$,

$$\nabla_{\vartheta_i} J=\mathbb{E}_{\mathcal{D}}\Big[\nabla_{\vartheta_i}\pi_i(o_i)\,\nabla_{a_i}Q_i(\mathbf{o},\mathbf{a})\big|_{a_i=\pi_i(o_i)}\Big],\qquad y_i=R_t+\gamma\,Q_i^{-}(\mathbf{o}',\mathbf{a}')\big|_{a'_j=\pi_j^{-}(o'_j)},\tag{24}$$

with Q_i fitted to y_i on a replay buffer. MATD3 (Ackermann et al., 2019) takes the smaller of two target critics, smooths the target actions with clipped Gaussian noise and updates the actors less often than the critics. The critics rest on a premise stated by Lowe et al. (2017): once all actions are known the environment is stationary, $P(s'\mid s,\mathbf{a},\pi_1,\dots,\pi_n)=P(s'\mid s,\mathbf{a})$ whatever the policies.

In the class the premise holds on $\tilde{s}$ and fails on s: by Eq. (4), given s and the sent joint action the next state still depends on $(\mathbf{x},\chi)$ (Proposition C.1(iv)). On MAPDN the critics' input contains every inverter's reactive power at the previous step and the photovoltaic generation, from which $\mathbf{x}$ is computed and the phase estimated, so there the critic can meet the premise. The term that fails is the actor: Eq. (24) improves π_i only among functions of o_i, and over such functions the objective is solved pointwise by $\pi_i(o_i)\in\arg\max_a \mathbb{E}\big[Q_i(\mathbf{o},a,\mathbf{a}_{-i})\mid o_i\big]$, which averages the critic's knowledge of the load out. By Lemma D.1(iii) the executed action then misses its target by the floor whatever the critic knows. MAPPO's centralised value likewise leaves the actor's input unchanged, and the twin critics, target smoothing and delayed actor updates of MATD3 act on overestimation, not on the actor's information or on the pooling of Lemma D.1(ii).

D.4 VALUE FACTORISATION: QMIX AND VDN

VDN (Sunehag et al., 2018) represents the joint action value as $Q_{\mathrm{tot}}(\mathbf{o}^{\le t},\mathbf{a})=\sum_i Q_i(o_i^{\le t},a_i)$, and QMIX (Rashid et al., 2018) as $Q_{\mathrm{tot}}=f_s\big(Q_1(o_1^{\le t},a_1),\dots,Q_n(o_n^{\le t},a_n)\big)$, a mixing network whose nonnegative weights are produced from the state s by hypernetworks, so that $\partial Q_{\mathrm{tot}}/\partial Q_i\ge 0$. Both minimise $\mathbb{E}_{\mathcal{D}}\big(R_t+\gamma\max_{\mathbf{a}'}Q_{\mathrm{tot}}^{-}(\mathbf{o}^{\le t+1},\mathbf{a}',s')-Q_{\mathrm{tot}}(\mathbf{o}^{\le t},\mathbf{a},s)\big)^2$ and execute $a_i=\arg\max_a Q_i(o_i^{\le t},a)$, which monotonicity makes a maximiser of Q_{tot}.

Monotonicity has a consequence here: the tuple of local maximisers maximises Q_{tot} at every s, so the state, the only input through which the mixer could see the phase or the peers, reaches the executed actions only through the training of the utilities, never as an input at execution. The utilities act on $o_i^{\le t}$, a function of $\mathcal{H}_i$, so Lemma D.1(iii) applies on the lift, and they are fitted on replayed data, so Lemma D.1(ii) describes their limit. On URB their input is constant (Remark D.3), and they reduce to per-route values fitted on pooled days, as for IQL. Both factorisations address credit assignment among co-learners, drift of kind (A); neither contains a term that depends on the coupling gains.

D.5 HETEROGENEOUS-AGENT TRUST REGIONS: HATRPO, HAPPO AND HASAC

The methods of Kuba et al. (2022) rest on the multi-agent advantage decomposition: for any ordering $i_{1:n}$ of the agents, $\mathbb{A}_{\boldsymbol{\pi}}^{i_{1:m}}(s,\mathbf{a}^{i_{1:m}})=\sum_{j=1}^{m}\mathbb{A}_{\boldsymbol{\pi}}^{i_j}(s,\mathbf{a}^{i_{1:j-1}},a^{i_j})$. The agents are updated one after another in a random order, and HAPPO maximises for agent i_m

$$\mathbb{E}_{s,\mathbf{a}\sim\boldsymbol{\pi}_k}\Big[\min\Big(\varrho^{i_m}M^{i_{1:m}},\ \operatorname{clip}\big(\varrho^{i_m},1-\epsilon_{\mathrm{c}},1+\epsilon_{\mathrm{c}}\big)M^{i_{1:m}}\Big)\Big],$$

$$M^{i_{1:m}}=\frac{\bar{\boldsymbol{\pi}}^{i_{1:m-1}}(\mathbf{a}^{i_{1:m-1}}\mid s)}{\boldsymbol{\pi}_k^{i_{1:m-1}}(\mathbf{a}^{i_{1:m-1}}\mid s)}\,\hat{\mathbb{A}}(s,\mathbf{a}),$$

where $\bar{\boldsymbol{\pi}}^{i_{1:m-1}}$ are the preceding agents' updated policies; HATRPO maximises $\mathbb{E}[\varrho^{i_m} M^{i_{1:m}}]$ under a bound δ_{KL} on the expected KL divergence from the current policy. The multi-agent policy iteration that both implement improves the return monotonically, $J(\boldsymbol{\pi}_{k+1}) \geq J(\boldsymbol{\pi}_k)$, for policies of the state of a Markov game. HASAC (Liu et al., 2024) applies the sequential scheme to the maximum-entropy objective $\mathbb{E}\big[\sum_t \gamma^t \big(R_t + \alpha_{\mathrm{ent}} \sum_i H(\pi_i(\cdot \mid s_t))\big)\big]$, with a soft action value fitted off-policy.

The ratio product in $M^{i_{1:m}}$ is how these methods treat non-stationarity: it reweights agent i_m's update by the change its predecessors have just made, drift of kind (A). Under frozen partners the preceding policies do not change and $M^{i_{1:m}} = \hat{\mathbb{A}}$, so the update is PPO on the advantage estimate; interaction-mediated drift persists in exactly that case (Proposition C.1(iii)) and reaches the update only through $\hat{\mathbb{A}}$, fitted on pooled data (Lemma D.1(ii)). The improvement guarantee assumes policies of the Markov state, which in the class is $\tilde{s}$ (Proposition C.1(iv)); the actors condition on o_i, which contains neither $\mathbf{x}$ nor, in routing and on the lift, the phase, so the guarantee does not cover the executed policies, and Lemma D.1(iii) bounds their actions. HASAC's entropy term sustains the excitation of the peers' actions, but excitation benefits only an estimator that uses them, and its soft critic is fitted on a replay buffer (Lemma D.1(ii)).

D.6 Recurrent policies: R-MAPPO and its counterparts

R-MAPPO (Yu et al., 2022) inserts a recurrent layer between the observation encoder and the policy head, $h_i(t) = \mathrm{GRU}\big(h_i(t-1), \mathrm{enc}(o_i(t))\big)$, $a_i(t) \sim \pi_i(\cdot \mid h_i(t))$, trained by backpropagation through time on chunks of 10 consecutive steps, each replayed from the hidden state recorded at its first step. Every MAPDN host uses recurrent actors; on URB the arm is the host PPO with the recurrence carried across days; on VMAS it is MAPPO or IPPO with a recurrent actor.

Memory replaces the input $o_i(t)$ by $h_i(t)$, a function of $o_i(0{:}t)$ and hence of $\mathcal{H}_i(t)$. Lemma D.1(iii) holds for every $\mathcal{H}_i$-measurable input, so memory can at best bring the policy's implicit estimate of the harm to $\mathbb{E}[y_i^\star \mid \mathcal{H}_i]$ and leaves the floor $\mathrm{Var}(y_i^\star \mid \mathcal{H}_i)$ unchanged: the peers' load along send is a function of the peers' histories, not of the agent's own. On URB the recurrence receives only observations, constant for almost every traveller (Remark D.3); between two updates $h_i(t) = F(h_i(t-1))$ for a map F fixed by the weights, so the state is a deterministic function of the day index and carries no information about realised costs. At most it could encode a clock with the driver's 100-day period, and truncated backpropagation over chunks of 10 days, which holds the state at the start of a chunk fixed, passes no gradient through dependencies longer than 10 days.

D.7 Graph-structured policies: DGN and GNN-MAPPO

DGN (Jiang et al., 2020) encodes the observations of agent i and its neighbours $\mathcal{B}_{+i}$ into features h_j and applies relation kernels, multi-head dot-product attention with heads $m = 1, \ldots, M$,

$$\alpha_{ij}^m = \frac{\exp\big(W_Q^m h_i \cdot W_K^m h_j\big)}{\sum_{k \in \mathcal{B}_{+i}} \exp\big(W_Q^m h_i \cdot W_K^m h_k\big)}, \qquad h_i' = \mathrm{MLP}\Big(\big\|_{m=1}^{M} \sum_{j \in \mathcal{B}_{+i}} \alpha_{ij}^m W_V^m h_j\Big),$$

with the paper's scaling factor absorbed into W_Q^m. A Q-network on the convolved features $O_{i,C}$ is trained on $\frac{1}{n}\sum_i \big(y_i - Q(O_{i,C}, a_i)\big)^2$, $y_i = R_{i,t} + \gamma \max_{a'} Q^-(O'_{i,C}, a')$, with the neighbourhood fixed over two consecutive steps, and temporal relation regularisation adds $\lambda_{\mathrm{r}} \frac{1}{M} \sum_m D_{\mathrm{KL}}\big(\mathcal{G}_m(O_{i,C}) \,\|\, \mathcal{G}_m(O'_{i,C})\big)$, with $\mathcal{G}_m$ the attention distribution of head m in an upper layer. GNN-MAPPO replaces the MAPPO actor by graph convolutions over all other agents, $h_i' = W_1 h_i + W_2 \sum_{j \neq i} h_j$, whose node features include each agent's previous executed action.

The relation kernels see the peers only through their features, encoded observations. On URB these are constant for almost every traveller (Remark D.3), so every logit is constant, the convolution returns a constant, and $Q(O_{i,C}, a)$ reduces to the per-route values of IQL fitted on pooled days; the attention distributions of consecutive days coincide, the regulariser vanishes identically, and the arms `dgn` and `dgn_r` (without it) coincide for these agents. DGN's features contain no peer actions, the carriers of the load. GNN-MAPPO on the lift receives the peers' previous actions and positions, which determine the latest term of Eq. (9), but not the gain: its input contains neither the driver nor the agent's own sent action, and hence not its droop, so by Lemma D.1(iv), even where its input determined $\mathbf{x}_i(t)$, the floor would be $\sigma^2 \bar{k}^2 (\mathrm{send}^\top \mathbf{x}_i)^2 \mathrm{Var}(A(t) \mid \mathrm{input})$. It holds the load and lacks

the gain, the context-conditioned methods of Section D.10 hold the gain and lack the load, and PACT computes the load and estimates the gain from its own excess.

D.8 MEAN-FIELD REINFORCEMENT LEARNING: MF-Q AND MF-AC

MF-Q (Yang et al., 2018) factorises agent i's action value over its neighbourhood $\mathcal{N}_i$ into pairwise terms of one common form and expands each around the mean action,

$$Q_i(s,\mathbf{a}) = \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} Q_i(s,a_i,a_j), \qquad a_j = \bar{a}_i + \delta a_{ij}, \quad \bar{a}_i = \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} a_j,$$

for one-hot actions. The first-order terms cancel and the second-order remainders are argued to cancel under homogeneity and locality of the neighbourhood, leaving $Q_i(s,\mathbf{a}) \approx Q_i(s,a_i,\bar{a}_i)$, learnt towards $R_{i,t} + \gamma v_i(s')$ under a Boltzmann policy on $Q_i(s,\cdot,\bar{a}_i)$ with $\bar{a}_i$ from the previous step. MF-AC trains an actor $\pi_i(a_i \mid s)$ along $\nabla \log \pi_i(a_i \mid s)\, Q_i(s,a_i,\bar{a}_i)$.

The common pairwise form lets the mean action summarise the neighbourhood, and it is the term that fails. The harm weighs neighbours through the declared operator: on the lift by W_{ij}, which decays with the distance between actuators, and on the feeder by the reactances $S_m[i,j]$ of shared paths. Exchanging the actions of two neighbours with different weights and different loads leaves $\bar{a}_i$ unchanged and changes $\sum_j W_{ij}\phi(a_j)$, so no function of $(s,a_i,\bar{a}_i)$ represents the harm: homogeneity fails by construction, not approximately. On the lift the load is carried by force magnitudes, and $\|\bar{a}_i\| \le \frac{1}{|\mathcal{N}_i|}\sum_j \|a_j\|$ with equality only for aligned forces, so opposing pushes that load the rail cancel in the mean action. On URB, with the neighbourhood set to the whole fleet since origin–destination pairs are singletons, $\bar{a}_i$ is the fleet's histogram of route indices, and the k-th routes of different travellers are different paths, so it measures the load on no route agent i can take. The mean-field value has no input for the phase other than s and is fitted on pooled data (Lemma D.1(ii)), and the MF-AC actor acts on s alone (Lemma D.1(iii)). On MAPDN the class is represented by the host with the W-weighted mean of the peers' actions, i.e. the channels, and its excess in the observation: it has the load, and the gain enters only through the history of the excess, from which its recurrent actor would have to learn, through the return alone, the estimator that PACT computes in closed form (Lemma C.3).

D.9 AGENT MODELLING: LIAM

LIAM (Papoudakis et al., 2021) encodes the controlled agent's own trajectory into $z_t = f_w(o^i_{1:t}, a^i_{1:t-1})$ and trains the encoder with decoders that reconstruct the modelled agents' observations and actions,

$$\mathcal{L}_{\mathrm{ED}} = \frac{1}{T_{\mathrm{e}}}\sum_{t=1}^{T_{\mathrm{e}}}\Big[\big\|f^o_u(z_t) - o_t^{-i}\big\|^2 - \log f^\pi_u\big(a_t^{-i} \mid z_t\big)\Big],$$

while the policy $\pi(a \mid o^i_t, z_t)$ is trained by A2C, by PPO in our ports, with no reinforcement-learning gradient into f_w.

The encoder's input is a function of $\mathcal{H}_i$, so by Lemma D.1(i) the best representation predicts (o_t^{-i}, a_t^{-i}) by their conditional expectation given $\mathcal{H}_i(t)$, and Lemma D.1(iii) bounds the policy on (o^i_t, z_t) as it bounds the host. The reconstruction targets describe what the peers do, not what it costs agent i: neither the gains nor the driver enters $\mathcal{L}_{\mathrm{ED}}$, and the return, the one signal that carries them, does not reach the encoder. Even an exact reconstruction of the peers' actions would leave the gain to be learnt from the return, the second case of Lemma D.1(iv). On URB the modelled agents, the travellers whose routes overlap agent i's, have constant observations (Remark D.3), so the encoder must predict their routes from agent i's own past routes.

D.10 OBSERVED CONTEXT: LCPO AND THE ORACLE-DRIVER HOST

LCPO (Hamadanian et al., 2025) considers an environment driven by an exogenous context z_t, on which states and actions have no influence, with kernel $T(s' \mid s,z,a)$ and a policy $\pi(a \mid s,z)$ that observes it. To learn online without forgetting its behaviour at past contexts it solves

$$\min_{\vartheta}\ \mathcal{L}_{\mathrm{PG}}(\vartheta) + \mathcal{L}_e(\vartheta) \quad \text{s.t.} \quad \mathbb{E}_{s\sim\mathcal{W}}\, D_{\mathrm{KL}}\big(\pi_{\vartheta_0}(\cdot \mid s)\,\|\,\pi_\vartheta(\cdot \mid s)\big) \le c_{\mathrm{anchor}},$$

with $\mathcal{L}_{\mathrm{PG}}$ the policy-gradient loss, $\mathcal{L}_e$ an entropy term, ϑ_0 the parameters before the update and $\mathcal{W}$ anchor states from a reservoir of past states whose contexts lie outside the distribution of the recent window.

Arms. `lcpo` solves this in trust-region form, `lcppo` replaces the constraint by a cross-entropy penalty on the anchors added to the clipped surrogate, and `a2c` drops it; the oracle-driver host is the unmodified host with $A(t)$ appended to its observation. All four take the driver as context and share one information set.

Knowing the driver is knowing the gain, not the load. With $A(t)$ in $\mathcal{H}_i$, Lemma D.1(iv) gives the floor $\sigma^2\bar{k}^2A(t)^2\,\mathrm{send}^\top\mathrm{Cov}(\mathbf{x}_i(t)\mid\mathcal{H}_i)\,\mathrm{send}$ on the feeder and the lift, common to the four arms; on URB the context gives the season, while each route's cost still depends on the other travellers' same-day choices. LCPO's model also requires the context to be exogenous and the kernel to be fixed given it. The driver is exogenous, but the quantity that sets the harm, $\boldsymbol{\theta}(t)^\top\mathbf{x}_i(t)$, is not: $\mathbf{x}_i$ is formed from the peers' executed actions (Eq. (3)), so the kernel agent i faces given (s,A) moves whenever its peers' policies do, and the anchor constraint, which holds $\pi(\cdot\mid s,z)$ near its past values at past contexts, preserves responses to loads that no longer occur.

D.11 Latent tasks across episodes: LILAC

LILAC (Xie et al., 2021) assumes that the environment changes between episodes and is a stationary MDP within each, indexed by a latent z_e that follows a Markov chain over the episodes $e=1,\dots,N$,

$$p(z_{1:N},\mathcal{T}^{1:N})=p(z_1)\,p(\mathcal{T}^1\mid z_1)\prod_{e=2}^{N}p(z_e\mid z_{e-1})\,p(\mathcal{T}^e\mid z_e),$$

with $\mathcal{T}^e$ the trajectory of episode e. An inference network $q(z_e\mid\mathcal{T}^e)$, a decoder and an LSTM prior are trained on $\mathbb{E}_q\big[\sum_e\sum_t\log p(s_{t+1},R_t\mid s_t,a_t;z_e)-D_{\mathrm{KL}}\big(q(z_e\mid\mathcal{T}^e)\,\|\,p(z_e\mid z_{e-1})\big)\big]$, and a soft actor-critic learner conditions on the latent: in training it is drawn from the current episode's posterior, at execution it is the prior's prediction from the previous episodes, held for the whole episode.

On MAPDN both assumptions fail. An episode of 240 three-minute steps covers half a solar day, over which the driver can traverse most of its daily range, so no latent makes an episode stationary. The episodes also start at independently drawn days and hours, so for fixed policies $\mathcal{T}^e$ is independent of $\mathcal{T}^{1:e-1}$, and so is any latent inferred from it: the latent received at execution, a function of the previous episodes, carries no information about the episode in which it acts. The policy is trained on posterior latents and executed on an uninformative prior, and without an informative latent its information is the host's (Lemma D.1(iii)).

D.12 Detection and restart: QCD restart, random restarts and MASTER

Gerogiannis et al. (2025) restart a base learner after quickest change detection. The detector is the Bernoulli generalised likelihood ratio test of Besson et al. (2022): for a stream $X_1,\dots,X_n\in[0,1]$ with segment means $\hat{X}_{a:b}$, it declares a change when

$$\sup_{1\le s<n}\Big[s\,\mathrm{kl}\big(\hat{X}_{1:s},\hat{X}_{1:n}\big)+(n-s)\,\mathrm{kl}\big(\hat{X}_{s+1:n},\hat{X}_{1:n}\big)\Big]\ge\log\big(4n^{3/2}\sqrt{T}\big),$$

with kl the Bernoulli divergence and the confidence level $1/\sqrt{T}$ used for an unknown number of changes, tested every 10 rounds; on detection the learner's parameters, optimiser state, replay buffer and exploration schedule are reset.

Arms. `rr` restarts at independent geometric times whose rate is set from the horizon T; MASTER (Wei & Luo, 2021) runs instances of the base learner on multiple scales with two tests that compare optimistic estimates with realised rewards.

The detector's model is a piecewise-stationary stream, and its guarantees, such as the regret bound of Besson et al. (2022), are stated in the number of breakpoints of a piecewise-constant mean and the size of the changes; the driver here is smooth and periodic. On URB the stream is the agent's daily reward, whose mean for frozen partners follows the 100-day cycle and changes on every wet

day; a detection responds to a change that recurs in every cycle, and each one rebuilds the learner, flushes its buffer and restarts its exploration, so the agent repeatedly relearns what does not drift. On MAPDN the rounds are episodes, which start at independently drawn times, so for fixed policies the per-episode reward stream is independent and identically distributed: there is no change at the scale the detector monitors, and a detection is a false alarm or a response to the learner's own progress, which the restart discards. `rr` pays the same cold starts at times independent of the data. Under MASTER's assumptions its tests cannot be crossed below a horizon of 1.24×10^9 (Gerogiannis et al., 2025, Thm. 1), far beyond the 4000 days of URB and the 400 episodes of MAPDN, so it reduces to its scheduled restarts. A restart reinstates the host's information set, and between restarts Lemma D.1 applies unchanged: detection establishes that the environment changed, not what changed or how to compensate for it.

D.13 Wavelet task representations: WISDOM

WISDOM (Wang et al., 2026) models a sequence of MDPs whose segments have unknown durations. A context encoder $e(z \mid C)$ maps a context C of recent transitions to a Gaussian task representation and is trained on $J = \mathbb{E}_{C\sim\mathcal{D}}\big[D_{\mathrm{KL}}\big(e(z \mid C) \,\|\, p(z)\big)\big]$; a wavelet network $\mathcal{Y}$ decomposes the sequence of representations over scales; a network Υ trained by wavelet temporal differences, $\Upsilon(z_t) \approx z_t + \gamma\,\bar{\Upsilon}(z_{t+1})$, predicts their evolution; and a soft actor-critic learner acts with $\pi(a \mid s, \hat{z})$ and $Q(s, a, \hat{z})$, $\hat{z} = \mathcal{Y}(z)$.

In the published algorithm the encoder is trained on J alone. Since $D_{\mathrm{KL}} \geq 0$ with equality if and only if the two laws coincide, the minimisers of J are exactly the encoders whose output law is the prior for almost every context, and minimising J drives the representation towards independence of C, leaving the wavelet networks to forecast a quantity that does not depend on the task; our VMAS port therefore adds a decoder of the next observation and reward and runs the released form as a separate arm. Even an informative encoder summarises the agent's own transitions, so z is a function of $\mathcal{H}_i$ and Lemma D.1(iii) holds for $\pi(a \mid s, \hat{z})$; on URB, where the context is the agent's own sequence of days, z can at most summarise the recent level of its rewards, a proxy for the gain, and Lemma D.1(iv) leaves the load. Its model of segments that are each stationary meets the objection raised against LILAC.

Arms. `flat` removes the wavelet decomposition and `noar` the autoregressive term (URB); on MAPDN the transform is applied over the latent coordinates, as in the released code, or over time, as the paper describes. None changes the encoder's input or its objective.

D.14 Online system identification: RMA and UP-OSI

RMA (Kumar et al., 2021) trains in two phases. In the first, a base policy $a_t = \pi(o_t, a_{t-1}, z_t)$ is trained by reinforcement learning with $z_t = f_e(e_t)$, an encoding of a privileged vector e_t of environment factors. In the second, π and f_e are frozen and an adaptation module recovers the encoding from recent history,

$$\hat{z}_t = f_{\mathrm{ad}}\big(o_{t-k:t-1}, a_{t-k:t-1}\big), \qquad \min_{f_{\mathrm{ad}}} \mathbb{E}\,\big\|\hat{z}_t - z_t\big\|^2, \qquad k = 50,$$

on data of the deployed policy; at deployment only $\hat{z}_t$ is used. UP-OSI (Yu et al., 2017) is the same construction with the model parameters in place of z_t. The privileged vector is the gain $\boldsymbol{\theta}(t)$ on MAPDN, trained under severities randomised up to $\sigma = 3$; the driver on VMAS, where the gains are $A(t)$ times a constant vector; and on URB the rain intensity with two summaries of the link capacities under it.

The privileged vector is a gain, not a load, so the method falls short already in its first phase: a teacher that knows e_t knows $A(t)$, and on MAPDN $\boldsymbol{\theta}(t)$, but not $\mathbf{x}_i(t)$, and by Lemma D.1(iv) even its executed action misses the target by $\sigma^2 \bar{k}^2 A(t)^2\, \mathrm{send}^\top \mathrm{Cov}(\mathbf{x}_i(t) \mid \mathcal{H}_i)\, \mathrm{send}$ in mean square. The second phase adds the error of recovering the gain from the history: $\hat{z}_t$ is a function of $\mathcal{H}_i$, and the distillation loss is minimised by $\mathbb{E}[z_t \mid \mathcal{H}_i(t)]$ (Lemma D.1(i)), which on URB rests on the agent's own rewards over 50 days. PACT inverts the construction: it computes the load from the broadcasts and estimates only the gain, $r + 1$ coefficients, with the error bounded in Theorem C.4.

Arms. Collecting the second phase's data with the teacher's latent, the ablation on URB, changes the distribution on which f_{ad} is fitted, not the floor, which the teacher already has.

D.15 Model-based MARL: M3W

M3W (Zhao et al., 2025) learns a world model over latents $z = p_E(o, e)$, with e a task embedding: a dynamics model $\hat{z}' = q_D(z, \mathbf{a}, e)$ built as a soft mixture of experts, a reward model $\hat{R} = q_R(z, \mathbf{a}, e)$ built as a sparse mixture whose noisy gate selects the top two experts for each input, and critics, trained on $\sum_t \lambda_{\mathrm{h}}^t \big(\|\hat{z}_t - \mathrm{sg}(z'_t)\|^2 + \mathrm{CE}(\hat{R}_t, R_t) + \mathrm{CE}(\hat{q}_t, G_t) \big)$ over replayed subsequences, with rewards and returns G_t coded two-hot and sg the stop-gradient. Actions are chosen by multi-agent model predictive path integral control, which samples joint actions, scores them under the model and refits the sampling distribution to the best of them.

Arms. `mlp` replaces the mixtures by plain networks and `greedy` the planner by an argmax of the model (URB); `noplan` acts with the learned actor (VMAS).

A mixture of experts represents a function that depends on a regime only if the regime is a function of its gate's input. On URB the latent encodes a constant observation (Remark D.3), and the reward model's input, that latent, the agent's route and the route histogram of its three graph neighbours, contains no phase: every day is routed to the same experts, and by Lemma D.1(ii) the model converges to the reward averaged over the phases in the buffer, which the planner then optimises; the arms change how the model is used, not this input. On MAPDN the models are centralised: the joint latent contains the previous reactive powers and the photovoltaic generation, so the harm is a function of the input, and M3W is limited by identification rather than information. It must learn the product of the drifting gain and the load without the structure of Proposition C.2, from data whose excitation converging policies reduce (Lemma C.7), and its planner queries the model at sampled joint actions away from these data, where it is not identified; for a model linear in its parameters this is Proposition C.8(iv). It also requires the joint observation at execution, which PACT does not use.

D.16 Domain randomisation: uniform DR and DORAEMON

Domain randomisation (Tobin et al., 2017) trains the host on $\max_\pi \mathbb{E}_{\sigma\sim\mathcal{Q}} J_\sigma(\pi)$ for a fixed $\mathcal{Q}$, here uniform on $[0, 3]$ and redrawn every episode, without telling the policy σ. DORAEMON (Tiboni et al., 2024) adapts $\mathcal{Q}$, a Beta distribution rescaled to $[0, 3]$, by maximising its entropy subject to a success constraint and a trust region,

$$\mathcal{Q}_{k+1} = \arg\max_{\mathcal{Q}} \; H(\mathcal{Q}) \quad \text{s.t.} \quad \hat{G}(\pi_k, \mathcal{Q}) \geq \alpha_{\mathrm{s}}, \qquad D_{\mathrm{KL}}\big(\mathcal{Q} \,\|\, \mathcal{Q}_k\big) \leq \epsilon_{\mathrm{KL}},$$

with $\hat{G}$ an importance-sampled estimate of the probability that an episode succeeds, $\alpha_{\mathrm{s}} = 0.5$ and $\epsilon_{\mathrm{KL}} = 0.05$, and a backup problem that maximises $\hat{G}$ within the trust region when the constraint cannot be met.

Arms. `fixed` keeps the initial $\mathrm{Beta}(100, 100)$, concentrated at $\sigma = 1.5$, and isolates the adaptation.

Both change only the distribution of σ in training; the policy class and its information are the host's. Since the policy does not observe σ, the law of total variance gives $\mathbb{E}\mathrm{Var}(y_i^\star \mid \mathcal{H}_i) \geq \mathbb{E}\mathrm{Var}(y_i^\star \mid \mathcal{H}_i, \sigma)$: randomising the severity can only raise the floor of Lemma D.1(iii) during training. The objective equals the average of the per-severity optima only if one policy is optimal for $\mathcal{Q}$-almost every severity; the action that offsets the harm scales with σ (Proposition C.2), so the maximiser trades performance across the support, and nothing in the objective favours the evaluated severity, $\sigma = 3$, the upper end of the support, which uniform randomisation exceeds 2.7 in only a tenth of the episodes. Since the harm grows with σ (Proposition C.1(v)), DORAEMON's success constraint limits the mass it places on high severities: it adapts the curriculum to what the host can carry, without changing what the host observes.

D.17 Adversarial regularisation: ERNIE

ERNIE (Bukharin et al., 2023) makes each policy Lipschitz in its input with an adversarial regulariser,

$$\mathcal{R}_\pi(o) = \max_{\|\mathbf{u}\| \le \epsilon_{\mathrm{p}}} \mathrm{Dist}\big(\pi(\cdot \mid o+\mathbf{u}), \pi(\cdot \mid o)\big), \qquad \min\ \mathcal{L}(\pi) + \lambda_{\mathrm{E}} \sum_{i=1}^{n} \mathbb{E}_o\, \mathcal{R}_{\pi_i}(o),$$

with $\mathcal{L}$ the host's loss, and solves the min–max problem as a Stackelberg game in which the policy anticipates the adversary; its analysis bounds the change of return under perturbed observations or transitions for policies smooth in their input.

Arms. `ernie_no_st` drops the Stackelberg correction, `gaussian` draws $\mathbf{u}$ at random with the same budget, and `ippo` is the host, $\lambda_{\mathrm{E}} = 0$.

The regulariser penalises the policy's response to changes of its input, while the harm displaces the executed action through the load and leaves the input unchanged. It is minimised by policies that ignore their input, whereas offsetting the harm requires the executed action to respond to the load, and it cannot lower the floor of Lemma D.1(iii), which depends on the information of the input, not on the smoothness of the policy. On URB the input is constant for almost every traveller (Remark D.3), and a network can be constant on the ball of radius ϵ_{p} around that point while taking any value at it; the regularised objective then admits the host's optimal outputs, and for these agents the regularised arms differ from `ippo` only through optimisation.

D.18 Disturbance observers and unstructured estimators: ESO, URLS and the intercept

Active disturbance rejection control (Han, 2009) lumps everything that drives the plant from its nominal model into a total disturbance d, estimates it with an extended state observer and cancels it. In the form of Łakomy et al. (2021) the observer is $\dot{\hat{\mathbf{z}}} = \mathbf{F}\hat{\mathbf{z}} + \mathbf{g}\,\hat{b}\,u + \mathbf{l}\,(y - \mathbf{c}^\top \hat{\mathbf{z}})$ for an extended state whose last component is $\hat{d}$, with gains $l_j = \binom{n_o+1}{j}\omega_{\mathrm{o}}^j$ that place all its poles at $-\omega_{\mathrm{o}}$, and the control $u = (-\mathbf{k}^\top \hat{\mathbf{e}} - \hat{d})/\hat{b}$ cancels the estimate; the analysis assumes $\dot{d}$ bounded. The arms apply it to each agent's measured, one-step-late disturbance. On MAPDN the observer is first order, $\hat{d}_i(t) = \hat{d}_i(t{-}1) + l\big(d_i(t{-}1) - \hat{d}_i(t{-}1)\big)$ with $l = 1 - e^{-\omega_{\mathrm{o}}}$, and the fleet sends $q_i^{\mathrm{sent}} = q_i^{\mathrm{cmd}} - g\hat{d}_i(t)$, $g = 1$, with a bounded correction. On VMAS it has two states, $l_1 = 2\omega_{\mathrm{o}}$ and $l_2 = \omega_{\mathrm{o}}^2$, predicts one step ahead by $\hat{z}_1 + \hat{z}_2$, and is applied through the multiplicative channel of Eq. (15). On URB, where nothing can be inverted, one observer per agent and route, first order (`eso1`, $l = 0.5$) or with two states (`eso2`, $\omega_{\mathrm{o}} = 0.5$), feeds the rule $\arg\min_k [\mathrm{fft}_k + \hat{d}_{i,k}]$ and is updated only on the days its route is taken.

Proposition D.4 (An own-residual observer in the feeder loop)**.** Consider the linear feeder model of Proposition C.12 with the driver held at $A > 0$, and let every agent run the first-order observer on its own reaction and send $q_i^{\mathrm{sent}} = q_i^{\mathrm{cmd}} - g\hat{d}_i(t)$, with common $l \in (0, 1]$ and $g \in (0, 1]$. Let $\Theta = \sigma k A \rho(G)$, the loop gain of the blind fleet. Along the Perron eigenvector of G, two executed trajectories under the same commands differ by a sequence with characteristic polynomial

$$p(z) = z^2 + (\Theta - 1 + l)\, z - \Theta\,(1 - l + gl).$$

(i) p has a real root $z_- < -\Theta$: along this eigenvector the mutual reaction under the observer has a spectral radius larger than Θ, the blind fleet's. **(ii)** Both roots lie in the open unit disk if and only if $\Theta < \Theta_c = (2 - l)/(2 - l + gl)$, and $\Theta_c < 1$. For $\Theta \in (\Theta_c, 1)$ the blind fleet's mutual reaction decays for every initial condition, while under the observer it grows without bound for some initial condition. **(iii)** Under PACT with exact estimates and trust g the rate along the same eigenvector is $(1 - g)\,\Theta < \Theta$.

Proof. Let $\mathbf{v}$, $\mathbf{d}$ and $\hat{\mathbf{d}}$ be the differences between the two trajectories of the executed reactive powers, the reactions and the estimates. The linear model gives $\mathbf{v}(t) = -g\hat{\mathbf{d}}(t) + \mathbf{d}(t)$ and $\mathbf{d}(t) = -\sigma k A\, G\, \mathbf{v}(t{-}1)$, and the observer gives $\hat{\mathbf{d}}(t) = (1 - l)\hat{\mathbf{d}}(t{-}1) + l\,\mathbf{d}(t{-}1)$. Every agent uses the same l and g, so these maps commute with G, which is diagonalisable with real eigenvalues (proof of Proposition C.12); G is entrywise nonnegative, so by the Perron–Frobenius theorem $\rho(G)$

is an eigenvalue, with an eigenvector $\mathbf{u}$. Along $\mathbf{u}$ write $\mathbf{d}(t) = \delta(t)\mathbf{u}$ and $\hat{\mathbf{d}}(t) = e(t)\mathbf{u}$; then $\mathbf{v}(t) = (\delta(t) - ge(t))\,\mathbf{u}$ and

$$\begin{pmatrix}\delta(t+1)\\ e(t+1)\end{pmatrix} = \begin{pmatrix}-\Theta & \Theta g\\ l & 1-l\end{pmatrix}\begin{pmatrix}\delta(t)\\ e(t)\end{pmatrix},$$

whose characteristic polynomial is $(z+\Theta)(z-1+l) - \Theta gl = p(z)$. (i) $p(-\Theta) = -\Theta gl < 0$ and $p(z) \to \infty$ as $z \to -\infty$. (ii) By the Jury criterion (Jury, 1964), the roots of $z^2 + a_1 z + a_0$ lie in the open unit disk if and only if $|a_0| < 1$ and $1 \pm a_1 + a_0 > 0$. Here $1 + a_1 + a_0 = l\,(1 + (1-g)\Theta) > 0$, and $1 - a_1 + a_0 = (2-l) - \Theta\,(2 - l + gl)$ is positive if and only if $\Theta < \Theta_c$; when it is, $|a_0| = \Theta(1-l+gl) < 1$ follows, because $(2-l)(1-l+gl) - (2-l+gl) = -l(1-g)(2-l) - gl < 0$. $\Theta_c < 1$ because $gl > 0$. For $\Theta < 1$ the blind fleet decays for every initial condition by Proposition C.12 with $g = 0$. For $\Theta > \Theta_c$, $p(-1) = 1 - a_1 + a_0 < 0$ gives a real root $z < -1$. By the first row of the matrix, an eigenvector (δ, e) for an eigenvalue z has $\delta = \Theta ge/(\Theta + z)$, hence $\delta - ge = -gez/(\Theta + z) \neq 0$; starting along it, $\mathbf{v}(t)$ is a nonzero multiple of $z^t\mathbf{u}$ and grows without bound. (iii) This is Proposition C.12 with $A_{\text{geo}} = A$. □

The observer estimates the reaction from its effect one step late, and in the loop that effect is the peers' reaction to the corrections they have just made, so estimate and disturbance are one turn of the mutual-reaction loop apart. Proposition D.4(i) shows that this never helps along the dominant mode, and (ii) that it lowers the instability threshold below the blind fleet's, to $\Theta_c = 1 - l/2$ at full trust: 0.89 at the declared MAPDN bandwidth $\omega_o = 0.25$ and 0.75 at $l = 0.5$. Extrapolating the rate does not repair this: in the same loop with $g = 1$, the two-state observer's closed-loop roots, computed numerically, leave the unit disk at $\Theta = 0.90$, 0.74, 0.69 and 0.50 for $\omega_o = 0.1$, 0.25, 0.3 and 0.5. PACT predicts the reaction from its cause, the peers' broadcast actions, and at the same trust its rate is $(1-g)\Theta$. Beyond the loop, the observer's information is the agent's own residual, so Lemma D.1(iii) bounds it on the feeder and the lift, and its analysis assumes a disturbance of bounded derivative that does not depend on the observer, which the feedback violates. On URB each route's observer is updated only on the days its route is taken; as exploration decays, the estimates of routes no longer taken stay at the phase of their last update, and the rule compares a current estimate with stale ones.

Evidence. On the six-inverter feeder at $\sigma = 3$ with the scripted droop host, over three daytime episodes of 120 steps, Θ reaches 1.13 at the driver's peak, where by (i) the observer's rate exceeds the blind fleet's (1.16 at $\omega_o = 0.05$). With the observer the voltage loss is 4.32 times that of the host at $\sigma = 0$, against 4.21 without compensation, and the fraction of steps with every voltage in band falls from 0.353 to 0.344; with PACT the two figures are 1.04 and 0.986, and with the oracle 1.00 and 0.992. At every bandwidth $\omega_o \in [0.05, 1]$ the observer's summed one-step prediction error exceeds the summed disturbance, so predicting no disturbance would be more accurate.

URLS runs the recursion Eq. (13) of PACT, with the same forgetting factor and initial covariance, on the raw broadcast actions instead of the channels: $\boldsymbol{\varphi}_i = [1, \mathbf{q}^{\text{ex}}_{-i}(t-1)^\top]^\top \in \mathbb{R}^n$ on the feeder, and on URB one filter per own route on the one-hot routes of all other vehicles of the fleet, $4 \cdot 87 + 1 = 349$ coefficients per filter and 1396 per agent on `saint_arnoult`. On the lift its columns are the executed force magnitudes $\|a^{\text{ex}}_j(t-1)\|$ of the $n-1$ peers, smoothed with the channel memory ζ of Eq. (9) and scaled as PACT's channels are, so that the two arms differ only in the design matrix; with $n = 4$ both have four coefficients, and the raw design omits the weights W_{ij}, whose dependence on the actuators' positions it must track through its coefficients (Section 5.2). Below saturation the feeder's excess is linear in the raw actions too, every coefficient proportional to $A(t)$, so the raw model is also exact; it differs in what it must identify. PACT's regressor is an affine image of the raw one, $\boldsymbol{\psi}_i = M_i\boldsymbol{\varphi}_i$, with M_i fixed by the matrices S_m and the normalisation. A prediction at a regressor in the span of the past rows is determined by the data, being a combination of the past $\boldsymbol{\psi}_k^\top\boldsymbol{\beta}$, and one with a component outside that span is not (Proposition C.8(iv)). Since M_i maps the span of the raw rows onto the span of PACT's, wherever the raw model's prediction is determined PACT's is, and the converse can fail once $n - 1 > r$: the raw model needs excitation in up to n directions and PACT in $r + 1$, both bounded by the indecision of the peers (Lemma C.7). On URB each own-route filter, like each observer, is updated only on the days its route is taken.

The intercept arm fits $\boldsymbol{\psi}_i \equiv 1$. By Proposition C.5, after the transient its recursion is the exponential smoother $\hat{\beta}(t) = \mu\hat{\beta}(t-1) + (1-\mu)\,y_i(t-1)$, a first-order observer on the agent's own excess with $l = 1 - \mu$, applied with trust $g_i = g_i^{\text{pol}}\,\text{conf}_i$, where conf_i tends to the same constant for every agent

(Proposition C.8(iii)). With a common g^{pol} it is thus an instance of Proposition D.4, and since its estimate is a function of the agent's own past, Lemma D.1(i) bounds its error below by the conditional variance of the excess given that past. The arm isolates the value of the channels: without them, PACT is an own-residual observer.

D.19 THE STATE OF THE ART ON THE FEEDER: TPA

TPA (Xu et al., 2024) extends an off-policy host, MATD3 or MADDPG, with a temporal prototype-aware encoder. A transformer over the bus features of the agent's zone is combined with a recurrent summary of the last 100 intervals of photovoltaic and load profiles and a season embedding; 24 learnable prototypes, one per solar term, are computed from whole-day network profiles of generation and load; the encoding F is matched to the prototypes of the current season by $\mathrm{sim}(p, F) = \log\big((\|p - F\|^2 + 1)/(\|p - F\|^2 + \epsilon_{\mathrm{s}})\big)$, the best match conditions the policy head, and $\mathcal{L}_{\mathrm{pl}} = \mathcal{L}_{\mathrm{ce}}(\mathrm{season}) + c_1\mathcal{L}_{\mathrm{clst}} + c_2\mathcal{L}_{\mathrm{sep}} + c_3\mathcal{L}_{\mathrm{div}}$ is added to the actor's loss. The arms are TPA-MATD3 and TPA-MADDPG.

TPA's temporal machinery recognises the operating condition, the season, the time of day and the levels of generation and load: the phase, and hence the gain. The prototypes and the profile summary are computed from the dataset's generation and load, which the harm does not alter, so this part of the encoder is identical at every σ and cannot distinguish $\sigma = 0$ from $\sigma = 3$. The load, the reactive powers of the inverters on shared paths outside the zone, is in none of its inputs, so by Lemma D.1(iv) with the phase known the floor $\sigma^2\bar{k}^2 A(t)^2\,\mathrm{send}^\top \mathrm{Cov}(\mathbf{x}_i(t) \mid \mathcal{H}_i)\,\mathrm{send}$ remains, and its hosts inherit the analysis of Section D.3.

D.20 EQUILIBRIUM SEEKING: DEEP FICTITIOUS PLAY

DEDA-FP (Magnino et al., 2025), the Deep FP arm of Table 1, solves mean-field games in which a representative agent's policy $\pi(a \mid t, s)$ induces a population flow (m_t) and earns $J_m(\pi) = \mathbb{E}\sum_t R(s_t, a_t, m_t)$; a mean-field Nash equilibrium is a policy that is a best response to the flow it induces. Fictitious play alternates a best response BR_k to the averaged flow, computed by PPO or SAC, with the averaging $\bar{m}_{k+1} = \frac{k}{k+1}\bar{m}_k + \frac{1}{k+1}m^{\mathrm{BR}_k}$, and fits the average policy $\bar{\pi}$ by minimising $\mathbb{E}[-\log\bar{\pi}(a \mid t, s)]$ over a reservoir of the best responses' state–action pairs; $\bar{\pi}$ plays at evaluation, and convergence is measured by exploitability. On URB an iteration lasts 10 days.

Arms. `ema` replaces the weight $1/(k+1)$ by 0.01, and `br` plays the last best response.

Fictitious play solves a fixed game: its equilibrium and its exploitability presuppose that $R(s, a, m)$ is the same function at every iteration, whereas in the class the payoff of an action moves with the phase and each best response is computed in the game of its iteration's phase. The averaged flow is a uniform average over iterations, so by Lemma D.2(i) a periodic sequence of best responses makes it converge to their average over the cycle, and after a change it approaches the new best response only as k_0/K (Lemma D.2(iii)); the average policy, fitted on a reservoir that samples the whole history uniformly, is an average over all the phases it has seen. On URB a cycle is 10 iterations, and for `ema` Lemma D.2(ii) gives the factor $0.01/|1 - 0.99\,e^{-2\pi\mathrm{i}/10}| \approx 0.016$ at the cycle's fundamental frequency, so the averaged flow retains under 2% of that component of the best responses' seasonal variation. `br` follows the phase by giving up the averaging on which convergence rests, and each best response is the host learner on its own information (Lemma D.1).

D.21 PERFORMATIVE LEARNING: IPGA AND INPG (PMPG)

Sahitaj et al. (2025) study Markov potential games in which the deployed joint policy shapes the game, $\mathcal{G}(\boldsymbol{\pi})$, with sensitivities $\|R_{\boldsymbol{\pi}} - R_{\boldsymbol{\pi}'}\| \le \epsilon_R\|\boldsymbol{\pi} - \boldsymbol{\pi}'\|$ and $\|P_{\boldsymbol{\pi}} - P_{\boldsymbol{\pi}'}\| \le \epsilon_P\|\boldsymbol{\pi} - \boldsymbol{\pi}'\|$, extending performative prediction (Perdomo et al., 2020); a performatively stable equilibrium is a joint policy each of whose components is a best response in the game it induces. Under repeated retraining the fleet deploys $\boldsymbol{\pi}_k$, collects data in $\mathcal{G}(\boldsymbol{\pi}_k)$, and each agent updates independently, by projected gradient ascent (IPGA), $\pi_i^{k+1} = \arg\max_{\pi_i}\langle\pi_i, \bar{Q}_i^k\rangle - \frac{1}{2\eta}\|\pi_i - \pi_i^k\|^2$, or by natural policy gradient (INPG), which converges in the last iterate to a performatively stable equilibrium when the sensitivities are small. On URB a deployment lasts 64 days, and each retraining makes 20 passes over its window.

Arms. `cont` applies INPG at the host's cadence, without the deployment schedule.

The performative model makes the game a function of the deployed policy alone: deploying the same policy twice yields the same game. In the class this fails, since with every policy frozen the harm still varies with the phase (Proposition C.1(iii)): the map $\boldsymbol{\pi} \mapsto \mathcal{G}(\boldsymbol{\pi})$ becomes $(\boldsymbol{\pi}, t) \mapsto \mathcal{G}_t(\boldsymbol{\pi})$, the fixed point that defines stability moves with the phase, and a sensitivity measured across deployments mixes the policy's effect with the driver's. Each retraining solves its window's objective: a 64-day window spans 64% of the cycle, the objective depends on its days only through per-route means, as in Section D.2, and by Lemma D.1(ii) retraining to convergence yields the best policy for the window-averaged bandit. The actors are independent and act on o_i (Lemma D.1(iii)), and `cont` isolates the deployment schedule, which addresses neither the moving fixed point nor the information.