FULCRUM: A PREFERENCE-CONDITIONED HIERARCHICAL GRAPH REINFORCEMENT LEARNING FOR SUSTAINABLE MULTI-DATA-CENTER CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Geo-distributed data-center fleets must trade off electricity costs, carbon emissions, water use, and quality of service (QoS) across sites with varying prices, grids, and weather conditions. Learned controllers for this problem are trained for a single objective weighting and a single fleet size, so any change in operator priorities or footprint requires retraining. We present FULCRUM, a single preference-conditioned graph policy: a Graph Pattern Machine whose objective-aligned attention heads are feature-wise linear modulation (FiLM) conditioned on the operator preference $\boldsymbol{\omega}$, followed by cross-attention from data-center nodes to preference-scaled objective tokens. All parameters are shared across nodes and edges, so the same weights act on any fleet. Under a budget-matched protocol with paired tests over 50 evaluation episodes, repeated across three training seeds, FULCRUM attains the best mean rank across five operator preferences among ten controllers (1.6 ± 0.2), matches or exceeds every preference-conditioned baseline across all seven held-out preferences never used in training (54 practical wins, no losses against preference-conditioned baselines), and at one-hot preferences reaches the lowest cost, carbon, water, and QoS penalty of any baseline. It steers the fleet at inference time from -81% water to -39% cost relative to its balanced operating point and, trained once at 3 data centers, outperforms a Green-DCC controller retrained from scratch at each of 3, 5, 7, and 10 data centers by 6.6–8.2% reward, where preference-conditioned MLP and GNN baselines cannot be loaded at all. One training run replaces five per-preference specialists at $5.0\times$ lower compute for a 1.3% reward cost. We also report where FULCRUM does not lead: at intermediate preference weights, conditioned baselines use 2–7% less water, one of three seeds emits 3% more carbon at the carbon corner, and the cross-attention and FiLM components are not individually significant at this budget.

1 INTRODUCTION

Data centers consumed roughly 460 TWh of electricity in 2023, close to 2% of global demand, and this figure is projected to double by 2026 (International Energy Agency, 2024; Masanet et al., 2020). Water for evaporative cooling has become a first-order concern in its own right (Mytton, 2021; Li et al., 2025). Neither footprint is uniform: carbon intensity (Henderson et al., 2020), wet-bulb temperature (Li et al., 2025), water stress, and electricity prices vary by large factors across regions, so operators running fleets of many sites (Radovanovic et al., 2023) can shift load, storage use, and cooling mode across the fleet in ways a single-site controller cannot.

A learned controller for such a fleet must be sustainable twice over: it should reduce the operational footprint of the sites it controls, and it should not have to repay its own training cost every time priorities or the fleet change; the Green AI literature (Schwartz et al., 2020; Strubell et al., 2019; Patterson et al., 2021) makes the point that *repeated* retraining, not one run, dominates the lifecycle footprint of deployed models. Our framework is a single lever that pivots between cost, carbon, water, and QoS through a runtime preference vector $\boldsymbol{\omega}$; hence the name FULCRUM[1].

[1] A fulcrum is the pivot of a lever: the preference dial re-routes the whole fleet without retraining.

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.

Limitations of current learned policies. SustainDC (Wang et al., 2024) introduced a benchmark using per-site MAPPO agents, and Green-DCC (Naug et al., 2024; Sarkar et al., 2025) coordinates multiple sites via a centralized MLP. Both are strong at the configuration they were trained for and share three limitations:

(L1) Fixed scalarization. The reward is a pre-committed linear combination of objectives; moving from cost to water during a drought requires a fresh training run.

(L2) Fixed fleet size. Flat MLPs have input and output dimensions tied to N; adding one site invalidates every weight.

(L3) Unaccounted retraining. Neither method reports the training compute implied by (L1)–(L2) over a deployment lifetime, so the controller's own footprint is never included in the accounting.

Contributions.

1. **A preference-conditioned graph policy.** FULCRUM is a hierarchical policy $\pi_\theta(a \mid s, \boldsymbol{\omega})$: a Graph Pattern Machine (GPM) with $K = 4$ objective-aligned attention heads whose logits are FiLM-modulated by $\boldsymbol{\omega}$ (Perez et al., 2018), a preference cross-attention block in which every site attends over $\boldsymbol{\omega}$-scaled objective tokens, and $\boldsymbol{\omega}$-conditioned edge- and node-level action heads. Because no weight is dimensioned by N, the policy is permutation-equivariant and size-agnostic (Proposition 5).
2. **A budget-matched, statistically controlled evaluation.** Ten controllers, including three preference-conditioned baselines with the same inputs and trainer, are trained for the same 10^6 environment steps and compared with paired Wilcoxon tests, Holm correction, Cliff's δ, and a 1% practical-significance floor, over three training seeds. FULCRUM has the best mean rank across five preferences (1.6 ± 0.2 among ten methods), a higher reward than every baseline at each of seven held-out preferences (54 practical wins, 9 ties, no losses against the conditioned baselines), the lowest cost, carbon, water, and QoS penalty of any baseline at the one-hot corners, and, trained at $N = 3$, beats a Green-DCC controller retrained from scratch at each $N \in \{3, 5, 7, 10\}$ by 6.6–8.2% with 1.5–2.4$\times$ less water, where the fixed-dimension conditioned baselines cannot be evaluated at all.
3. **Lifecycle accounting and negative results.** One FULCRUM run costs $5.0\times$ less training compute than five same-architecture per-preference specialists and gives up 1.3% mean reward. We report that cross-attention and FiLM are not individually significant at this budget, that a Boltzmann–Tchebycheff auxiliary critic *hurts* by 11.6%, that at intermediate preference weights conditioned baselines use less water than FULCRUM on the named metric, and that on real Google/Alibaba traces at the balanced preference the learned methods are within 2% of each other.

2 Related Work

RL for data-center sustainability. Model-based RL reduced cooling energy in a single facility (Li et al., 2020). SustainDC (Wang et al., 2024) moved to per-site multi-agent control with MAPPO (Yu et al., 2022); Green-DCC (Sarkar et al., 2025; Naug et al., 2024) coordinates several sites through a centralized MLP and reports gains over MAPPO at a fixed fleet size. Related lines study battery dispatch (Sun et al., 2023), carbon-aware load shifting (Radovanovic, 2020), and water-constrained load balancing (Islam et al., 2016). None exposes the objective weighting at inference time or transfers across fleet sizes, and none reports the retraining budget those restrictions imply (App. I.1).

Preference-conditioned MORL. Multi-objective RL approximates a Pareto front (Roijers et al., 2013; Hayes et al., 2022); preference-conditioned methods train one policy $\pi(a \mid s, \boldsymbol{\omega})$ over a distribution of weight vectors (Abels et al., 2019; Yang et al., 2019; Xu et al., 2020; Basaklar et al., 2023). We follow this route in a graph-structured multi-site control problem and compare against flat and graph-preference-conditioned PPO baselines with the same inputs, curriculum, and budget, thereby separating the effect of the architecture from that of conditioning itself.

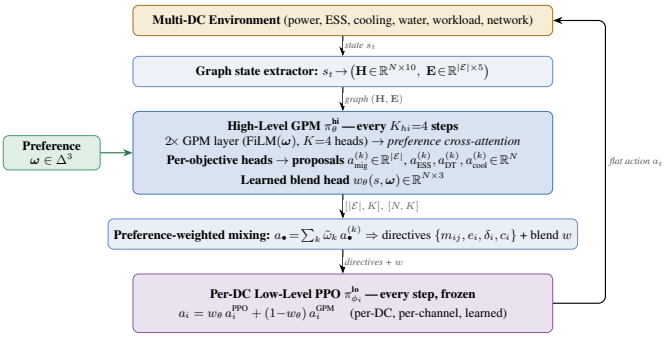


Figure 1: **FULCRUM information flow.** The preference $\boldsymbol{\omega}$ enters twice: as FiLM parameters on the $K = 4$ objective-aligned attention heads inside each GPM layer, and as scaling of the objective tokens that every site attends to. Edge and node heads then read the preference-mixed embeddings. No weight has a dimension that depends on N, so the same θ acts on any fleet.

Graph policies. Message-passing networks (Kipf and Welling, 2017; Veličković et al., 2018; Battaglia et al., 2018) are permutation-equivariant and generalize to unseen graph sizes (Xu et al., 2019), a property used in traffic (Li et al., 2018), power-grid (Donon et al., 2020), and swarm (Tolstaya et al., 2020) control. FULCRUM adds two preference pathways to such a network, FiLM on per-objective attention logits and cross-attention to objective tokens, and tests whether they help beyond preference concatenation.

3 Problem Formulation

We write the fleet as a multi-objective MDP (Roijers et al., 2013; Hayes et al., 2022); notation is collected in App. A.

Definition 1 (Multi-DC graph)**.** A fleet is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $N = |\mathcal{V}|$ data centers and $|\mathcal{E}| = N(N-1)$ directed migration links. Node v_i carries $\mathbf{h}_i \in \mathbb{R}^{d_n}$ ($d_n = 34$): utilization, storage state and headroom, ambient and wet-bulb temperature, price, carbon intensity, water stress, PUE, pending and eligible deferrable load, in-flight migrations, time of day, and four-step forecasts of load, price, temperature, and carbon intensity. Edge (v_i, v_j) carries $\mathbf{e}_{ij} \in \mathbb{R}^{d_e}$ ($d_e = 6$): last-step link usage, latency, log transfer cost, and the water-stress, price, and carbon-intensity differences between endpoints.

Definition 2 (Multi-objective MDP)**.** The fleet evolves as a finite-horizon MOMDP $(\mathcal{S}, \mathcal{A}, P, \mathbf{r}, \gamma, H)$ with $H = 24$ hourly steps and vector reward

$$\mathbf{r}_t = \big[\, -(E_t + \mathrm{Net}_t),\ -C_t,\ -W_t,\ -Q_t \big]^\top \in \mathbb{R}^K, \qquad K = 4, \tag{1}$$

where E_t is electricity cost (\$), Net_t the wide-area transfer cost of migrated work (\$), C_t carbon (kg CO_2), W_t site water use (L), and Q_t a penalty on deferred and late work. Folding Net_t into the first component makes "cost" the operating cost an operator pays, so a policy cannot lower its electricity bill by migrating at an unpriced network cost.

Definition 3 (Joint action and preference scalarization)**.** At every step, the controller emits, per site, a deferrable-task release $\delta_i \in [0, 1]$, a storage command $e_i \in [-1, 1]$, and a cooling mode $c_i \in \{\text{air}, \text{evaporative}\}$, and per directed link a migration fraction $m_{ij} \in [0, 0.5]$ of site i's deferrable work; the flat action has dimension $3N + N(N-1)$. An operator preference is $\boldsymbol{\omega} \in \Delta^{K-1}$, and the scalar reward is $R_t(\boldsymbol{\omega}) = \boldsymbol{\omega}^\top (\mathbf{r}_t \oslash \mathbf{s})$ with a fixed per-objective scale $\mathbf{s} = [50, 130, 2200, 120]$ that brings the four channels to comparable magnitude.

Problem 4 (Preference-adaptive, size-agnostic fleet control)**.** Find one parameter vector θ such that (i) $\theta \in \arg\max\, \mathbb{E}_{\boldsymbol{\omega}\sim\mu} \mathbb{E}_{\pi_\theta(\cdot|s,\boldsymbol{\omega})} \big[\sum_t \gamma^t R_t(\boldsymbol{\omega})\big]$ for a preference distribution μ covering Δ^{K-1}; (ii) $\pi_\theta(\sigma \cdot s, \boldsymbol{\omega}) = \sigma \cdot \pi_\theta(s, \boldsymbol{\omega})$ for every node permutation σ; and (iii) π_θ is well-defined for any fleet size $N' \neq N$ without new parameters.

4 METHOD: FULCRUM

4.1 GRAPH PATTERN MACHINE WITH PREFERENCE FILM

Node features are embedded by a two-layer MLP to $\mathbf{h}_i \in \mathbb{R}^d$ ($d{=}256$). Each of the two GPM layers performs edge-aware attention with K heads, one per objective. For node i and neighbour j, head k has logit

$$\ell_{ij}^{(k)} = \mathbf{a}_k^\top \,\mathrm{LeakyReLU}\big(\big[\mathbf{W}_k\mathbf{h}_i \,\|\, \mathbf{W}_k\mathbf{h}_j \,\|\, \mathbf{U}_k\mathbf{e}_{ij}\big]\big), \tag{2}$$

which a preference encoder $\psi(\boldsymbol{\omega}) = (\boldsymbol{\gamma}(\boldsymbol{\omega}), \boldsymbol{\beta}(\boldsymbol{\omega})) \in \mathbb{R}^{2K}$ modulates before the softmax:

$$\hat{\ell}_{ij}^{(k)} = \gamma_k(\boldsymbol{\omega})\,\ell_{ij}^{(k)} + \beta_k(\boldsymbol{\omega}), \qquad \alpha_{ij}^{(k)} = \mathrm{softmax}_{j\in\mathcal{N}(i)}\,\hat{\ell}_{ij}^{(k)}. \tag{3}$$

Messages $\sum_j \alpha_{ij}^{(k)}\mathbf{W}_k\mathbf{h}_j$ from the K heads are concatenated, projected, and added residually with layer normalization. The intent is that head k learns routing patterns useful for objective k (for instance, attending to low-carbon neighbors) and that $\boldsymbol{\omega}$ decides how much each pattern is trusted at inference time.

Preference cross-attention. After the GPM layers, node embeddings attend over K *objective tokens* $\mathbf{z}_k = \omega_k\mathbf{u}_k$, where $\mathbf{u}_k \in \mathbb{R}^d$ are learned and ω_k is the operator's weight on objective k. With $\mathbf{Q} = \mathbf{H}\mathbf{W}_Q$, $\mathbf{K} = \mathbf{Z}\mathbf{W}_K$, $\mathbf{V} = \mathbf{Z}\mathbf{W}_V$,

$$\tilde{\mathbf{H}} = \mathrm{LayerNorm}\Big(\mathbf{H} + \mathbf{W}_O\,\mathrm{softmax}\big(\mathbf{Q}\mathbf{K}^\top/\sqrt{d}\big)\mathbf{V}\Big). \tag{4}$$

FiLM changes *how sites attend to each other*; cross-attention changes *which objective each site emphasizes*. Whether these two pathways add anything over concatenating $\boldsymbol{\omega}$ to the inputs is an empirical question that Sec. 6.5 answers.

Action and value heads. A migration head reads $[\tilde{\mathbf{h}}_i\|\tilde{\mathbf{h}}_j\|\mathbf{e}_{ij}\|\boldsymbol{\omega}]$ for every directed edge and emits one logit z_{ij}; storage, deferral, and cooling heads read $[\tilde{\mathbf{h}}_i\|\boldsymbol{\omega}]$ for every node. All heads are Gaussian with state-independent log-standard deviations. Migration fractions are decoded as $m_{ij} = 0.5\,\sigma(z_{ij} - 3)$, so untrained policies migrate almost nothing and must learn to move work; storage and deferral use affine $\tanh$; cooling thresholds are a continuous logit at execution, while PPO log-probabilities are computed on the continuous value. A sum-pooled value head outputs a scalar V and a per-objective vector $\mathbf{v} \in \mathbb{R}^K$; both are trained, $\boldsymbol{\omega}^\top\mathbf{v}$ is the bootstrap target. The full network has 1.11M parameters.

Proposition 5 (Permutation equivariance and size transfer)**.** *For any permutation σ of $\mathcal{V}$ and any $\boldsymbol{\omega}$, $\pi_\theta(\sigma\cdot(\mathbf{H},\mathbf{E}),\boldsymbol{\omega}) = \sigma\cdot\pi_\theta((\mathbf{H},\mathbf{E}),\boldsymbol{\omega})$, and the same θ is well-typed for every $N' \geq 2$.*

The proof (App. B) checks that every block is either node-wise, edge-wise, or a permutation-invariant aggregation, and that $\boldsymbol{\omega}$ enters only through N-independent tensors. Equivariance is a structural property, not a performance guarantee; Sec. 6.4 measures what transfer actually delivers. Figure 1 shows one decision step.

4.2 TRAINING

FULCRUM is trained with PPO (Schulman et al., 2017) for 10^6 environment steps on $N{=}3$. Each update collects 96 episodes whose preferences are drawn from a curriculum μ mixing Dirichlet samples with concentration annealed to 0.5, the five canonical preferences of Sec. 5, and repeated balanced copies, so that corners and interior of the simplex are both visited. The loss is the clipped surrogate with GAE advantages on $R_t(\boldsymbol{\omega})$ plus $0.5\,\|V - \hat{R}\|^2 + 0.7\,\|\mathbf{v} - \hat{\mathbf{r}}\|^2$ and an entropy bonus annealed from 0.01 to 0; the learning rate $3{\times}10^{-4}$ decays linearly after a 5% warm-up, and the final checkpoint is evaluated. Add-ons that we studied and left *off* in the deployed model (a Boltzmann–Tchebycheff auxiliary critic, a preference-sensitivity margin, per-preference advantage normalization) are evaluated in Sec. 6.5; full hyperparameters are in App. H.

5 EXPERIMENTAL SETUP

Simulator and data. We use a multi-site simulator with IT power, lithium-ion storage (round-trip efficiency 0.90), air versus evaporative cooling with wet-bulb physics and regional water-stress multipliers (CA 2.6, TX 2.0, VA 1.0), a two-queue workload model with deadlines, and inter-site migration with per-link bandwidth, latency, and priced transfer. The training fleet has $N = 3$ sites in California, Texas, and Virginia with heterogeneous prices, carbon intensity, weather, and water stress. Training uses synthetic traces. We evaluate on (i) held-out synthetic episodes and (ii) real Google (Reiss et al., 2011) and Alibaba (Liu and Yu, 2018) workload traces joined with real hourly carbon-intensity and weather series on independent days, with no fine-tuning, so the real-trace results are a transfer test for every learned method. Scalability fleets of $N \in \{5, 7, 10\}$ sites are drawn as prefixes of a fixed ten-region portfolio with a sparse wide-area topology (App. I.2).

Preferences. Five canonical preferences are used throughout: COST-MIN, CARBON-MIN, WATER-MIN, and QOS-MAX place weight 0.7 on the named objective and 0.1 on each other, and BALANCED is $\mathbf{1}/4$. Each preference has a *named target metric* (operating cost, kg CO_2, liters, QoS penalty, and scalar reward, respectively). We report both the scalar reward and the named metric because a controller can raise the former by trading the secondary objectives while losing on the latter.

Controllers. Preference-*blind* capability baselines: **Random**; a price- and wet-bulb-aware **Heuristic**; **MAPPO** (Wang et al., 2024; Yu et al., 2022) with per-site actors $[64, 64]$ and no migration; **Green-DCC** (Naug et al., 2024) with a centralized MLP $[256, 256]$ over the concatenated fleet observation ($\approx$105k parameters). Balanced *specialists*: **Per-DC PPO**, one Stable-Baselines3 PPO controller per site (Raffin et al., 2021) trained in a single-site simulator on the balanced reward (10^6 steps each; no migration), and **FULCRUM (Balanced specialist)**, our architecture trained only at $\boldsymbol{\omega} = \mathbf{1}/4$. Preference-*conditioned* baselines with the same inputs, trainer, curriculum, and budget as FULCRUM: **Pref-MLP-PPO**, an MLP $[256, 256]$ over the flattened graph tensors and $\boldsymbol{\omega}$ (107k parameters); **Pref-GNN-PPO**, an edge-aware attention GNN with $\boldsymbol{\omega}$ concatenated to node features (91k); and **Pref-MLP-PPO (vector critic)**, with a K-dimensional value head. **FULCRUM (no migration)** forces migration to zero. All learned methods receive 10^6 environment steps; MAPPO and Green-DCC use their published batch of 16 episodes per update.

Protocol and statistics. Every method is evaluated using deterministic actions on the same 50 episode seeds per setting, which are disjoint from any training or checkpoint-selection episodes. Differences to FULCRUM are tested with a paired two-sided Wilcoxon signed-rank test over episodes, Holm-corrected within each family (Holm, 1979), with Cliff's δ (Cliff, 1993) as effect size. Because 50 paired episodes make sub-percent differences "significant," a difference is *practical* only if $|\text{relative margin}| \geq 1\%$ and $|\delta| \geq 0.147$; smaller significant differences are ties. The direct, multi-preference, and eval-only comparisons are repeated over three training seeds (42, 43, 44) for every learned method; scalability, ablations, and lifecycle use seed 42. Compute is converted with 250 W board power, 387 gCO_2/kWh (U.S. Environmental Protection Agency, 2024), and 11.2 L/kWh (Shehabi et al., 2016).

6 RESULTS

6.1 DIRECT COMPARISON AT THE BALANCED PREFERENCE

Table 1 is the setting in which the preference-blind baselines were designed to operate. On synthetic held-out episodes, FULCRUM posts the highest reward of any single general policy: $+8.4\%$ over MAPPO and $+10.9\%$ over Green-DCC (both $\delta = 1.0$), and a small but practical $+1.0\%$ over the per-site PPO specialists ($\delta = 0.28$). The gain is not on every named metric: Green-DCC has the lowest carbon and the heuristic the lowest QoS penalty, while FULCRUM runs the fleet with $2.3\times$ less water than either pref-blind method at moderate cost; at Green-DCC's QoS level the interpolated FULCRUM preference curve uses 82% less water for $+2\%$ cost and $+14\%$ carbon (App. D). On real traces the picture is a tie: FULCRUM and Green-DCC are indistinguishable ($p = 0.78$), MAPPO and the per-site specialists are 1.6–1.7% ahead ($|\delta| \approx 0.17$), all learned methods lie within 2% of each other, and FULCRUM again operates at $2.3\times$ lower water (6346 vs. 14373 L for MAPPO) and half the QoS penalty (18.6 vs. 39.4). The Balanced specialist of our own architecture is the strongest

Table 1: **Balanced preference, $N=3$, 50 episodes, training seed 42** (mean $\pm$ std over episodes). Left: scalar reward on synthetic held-out episodes and on real Google/Alibaba traces (no fine-tuning); right: named metrics on synthetic episodes. †: FULCRUM practically better; ‡: baseline practically better; unmarked learned rows are ties. Best per column in **bold**; the two balanced *specialists*, trained only for this preference, are excluded from bolding. Across three training seeds the synthetic reward is -17.77 ± 0.12 (FULCRUM), -19.09 ± 0.22 (MAPPO), -19.60 ± 0.27 (Green-DCC).

Method	Reward (synth.) ↑	Reward (real) ↑	Cost ($) ↓	Carbon (kg) ↓	Water (L) ↓	QoS pen. ↓
Random	$-21.8 \pm 0.5^{\dagger}$	$-23.2 \pm 1.4^{\dagger}$	1820	7793	16677	64.7
Heuristic	$-21.1 \pm 0.3^{\dagger}$	-21.9 ± 1.5	1550	7164	27636	**21.2**
MAPPO (pref-blind)	$-19.3 \pm 0.3^{\dagger}$	$\mathbf{-21.3} \pm 1.3^{\ddagger}$	1640	7169	18656	33.2
Green-DCC (pref-blind)	$-19.9 \pm 0.3^{\dagger}$	-21.6 ± 1.3	1720	**6802**	18738	57.5
Per-DC PPO (specialist)	$-17.9 \pm 0.4^{\dagger}$	$-21.3 \pm 1.3^{\ddagger}$	1599	7660	11872	23.1
FULCRUM (no migration)	$-18.1 \pm 0.3^{\dagger}$	-21.6 ± 1.3	**1546**	7542	9470	59.8
FULCRUM (Balanced specialist)	-17.6 ± 0.4	$-20.9 \pm 1.3^{\ddagger}$	1713	7747	6624	23.9
FULCRUM (ours, one policy)	$\mathbf{-17.7} \pm 0.4$	-21.6 ± 1.3	1641	7470	**8250**	35.8

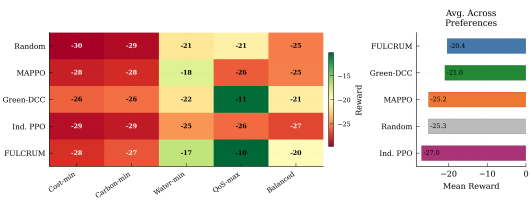
Table 2: **Multi-preference evaluation** ($N=3$, 50 episodes per preference and training seed; mean $\pm$ std across three training seeds). Columns give the *named target metric* of each preference; ranks (lower is better) are averaged over the five preferences by scalar reward and by named metric. Pref-blind methods emit the same policy for every column. Best per column in **bold**. On the named metric, seed 42, FULCRUM has practical wins over each conditioned baseline at COST-MIN, QOS-MAX, BALANCED, practical losses at CARBON-MIN (all three) and WATER-MIN (Pref-GNN), and ties elsewhere.

Method	Cost-min cost ($) ↓	Carbon-min CO_2 (kg) ↓	Water-min water (L) ↓	QoS-max QoS pen. ↓	Balanced reward ↑	Mean rank reward	Mean rank named
Random	1824	7766	16688	65.2	-21.79	9.9	9.5
MAPPO (pref-blind)	1656 ± 68	7325 ± 276	15807 ± 2528	44.1 ± 21.8	-19.09 ± 0.22	7.9	7.9
Green-DCC (pref-blind)	1717 ± 39	7036 ± 280	15932 ± 4709	65.9 ± 29.7	-19.61 ± 0.28	8.2	8.3
Per-DC PPO (specialist)	1603	7628	11624	23.8	-17.84	5.7	6.2
Pref-MLP-PPO	1052 ± 7	$\mathbf{5236} \pm 50$	1832 ± 33	5.3 ± 0.2	-18.01 ± 0.08	2.9	3.2
Pref-GNN-PPO	1071 ± 15	5271 ± 69	$\mathbf{1790} \pm 37$	5.5 ± 0.2	-17.96 ± 0.07	3.9	3.1
Pref-MLP-PPO (vector critic)	1098 ± 30	5247 ± 52	1858 ± 35	5.4 ± 0.3	-18.20 ± 0.17	4.4	4.1
FULCRUM (no migration)	$\mathbf{1028} \pm 9$	5250 ± 212	1815 ± 18	9.3 ± 0.1	-18.12 ± 0.09	3.1	3.1
FULCRUM (ours)	1035 ± 9	5288 ± 194	1841 ± 33	$\mathbf{4.5} \pm 0.4$	$\mathbf{-17.76} \pm 0.12$	$\mathbf{1.6} \pm 0.2$	$\mathbf{2.3} \pm 0.1$

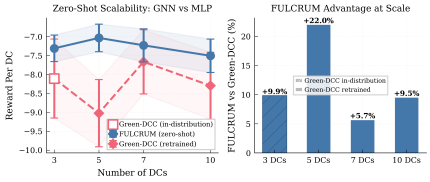
single controller on both trace sets (-17.6; -20.9), which prices generality at 0.3% (synthetic) to 3.4% (real) of reward.

6.2 ONE POLICY ACROSS OPERATOR PREFERENCES

Table 2 and Fig. 2(a) evaluate the property that the pref-blind baselines lack. **First, conditioning is what matters most.** Every preference-conditioned controller reaches roughly \$1050 at COST-MIN, 5250 kg at CARBON-MIN, 1800 L at WATER-MIN, and a QoS penalty near 5 at QOS-MAX, whereas the pref-blind controllers stay at \$1650–1720, 7000–7300 kg, 15800 L, and 44–66: a single dial moves the fleet by 37% of cost, 26% of carbon, 78% of water, and 87% of QoS penalty relative to its balanced operating point (App. C), with no retraining. **Second, FULCRUM has the best rank but does not dominate at these corners.** Its mean reward rank is 1.6 ± 0.2 against 2.9 for the strongest baseline, and its named-metric rank 2.3 ± 0.1 is the best but within one rank of Pref-GNN and Pref-MLP: it is ahead at COST-MIN, QOS-MAX, and BALANCED and behind at CARBON-MIN (5288 vs. 5236 kg, $+1.0\%$) and WATER-MIN (1841 vs. 1790 L, $+2.8\%$), all conditioned methods lying within $\pm 3\%$ of each other. Sec. 6.3 shows that these two deficits belong to the 0.7 scalarization, not to the policy's reach. **Third, migration buys QoS and water, not cost or carbon.** Disabling migration doubles the QOS-MAX penalty (9.3 vs. 4.5) and worsens the balanced reward by 2.1%, but slightly *lowers* cost and carbon at their corners; the WAN cost and transit delay of moving work are not recovered by cheaper or cleaner electricity at $N=3$.



(a) Scalar reward per preference (seed 42).



(b) Zero-shot FULCRUM vs. retrained Green-DCC.

Figure 2: **(a)** Preference-conditioned methods separate cleanly from pref-blind ones at every corner; among the former, FULCRUM has the highest reward at four of five preferences. **(b)** FULCRUM trained once at $N=3$ and evaluated zero-shot at $N \in \{5, 7, 10\}$ against Green-DCC retrained from scratch at each N with the same 10^6-step budget.

Table 3: **One-hot corners and held-out preferences** ($N = 3$, 50 episodes per setting, mean $\pm$ std across three training seeds; stored checkpoints, no retraining). Left: named metric at $\boldsymbol{\omega} = \mathbf{e}_k$, where the scalar objective *is* the named metric; rank over the four corners among the nine rows of Table 2. Right: scalar reward averaged over seven held-out preferences never used as a training corner (Table 9); rank over the seven. W/T/L: practical wins/ties/losses of FULCRUM against the row over the 4×3 or 7×3 setting–seed pairs. The four preference-blind rows of Table 2 emit the same policy at every preference (values unchanged up to Random's sampling noise), and each loses all 12 corner–seed pairs. Best per column in **bold**.

	One-hot corners (named metric)						**Held-out preferences** (reward)		
Method	cost ($) ↓	CO_2 (kg) ↓	water (L) ↓	QoS pen. ↓	rank	W/T/L	mean of 7 ↑	rank	W/T/L
Pref-MLP-PPO	1031 ± 7	5096 ± 11	1642 ± 66	5.6 ± 0.4	2.8	7/3/2	-16.80 ± 0.05	3.4	19/2/0
Pref-GNN-PPO	1048 ± 13	5114 ± 49	1738 ± 62	8.3 ± 2.3	4.0	11/0/1	-16.74 ± 0.04	3.0	15/6/0
Pref-MLP-PPO (vector critic)	1076 ± 30	5114 ± 20	1762 ± 68	5.9 ± 0.1	3.9	11/0/1	-17.02 ± 0.17	4.4	20/1/0
FULCRUM (no migration)	$\mathbf{1013} \pm 4$	5071 ± 177	$\mathbf{1578} \pm 109$	9.3 ± 0.0	2.4	3/7/2	-16.69 ± 0.04	3.0	11/10/0
FULCRUM (ours)	1018 ± 4	$\mathbf{5060} \pm 158$	1592 ± 118	$\mathbf{4.7} \pm 0.1$	$\mathbf{1.9} \pm 0.3$	—	$\mathbf{-16.48} \pm 0.06$	$\mathbf{1.3} \pm 0.0$	—

6.3 One-hot corners, held-out preferences, and the preference dial

The named-metric deficits in Table 2 arise at preferences that leave 0.3 of the weight on the other three objectives, where trading those objectives more aggressively is rewarded. Table 3 removes the ambiguity with the stored checkpoints. **One-hot corners.** At $\boldsymbol{\omega} = \mathbf{e}_k$ the scalar objective is the named metric up to a constant, and FULCRUM has the lowest three-seed mean of all four among the baselines ($1018 vs. $1031, 5060 vs. 5096 kg, 1592 vs. 1642 L, QoS penalty 4.7 vs. 5.6 against the best conditioned baseline) and the best rank (1.9 ± 0.3 vs. 2.8), with 29 practical wins, 3 ties and 4 losses over the 36 corner-seed pairs. The losses sit in single seeds, seed 42 at the carbon corner (2.5–3.1% more carbon than every conditioned baseline) and seed 44 at the water corner (1.8% more water than Pref-MLP), hence the large seed stds in those columns; one-hot corners are equally out of distribution for every method (training corners have weight 0.7). **Held-out preferences.** At seven preferences never used as a training corner (weight 0.55 on one objective, 0.4/0.4 two-objective mixes, a near-uniform vector), FULCRUM has a higher reward than every baseline in every seed (rank 1.3; only its own no-migration variant is ahead, at the two cost-weighted preferences), with 54 practical wins, 9 ties and no losses against the conditioned baselines and margins of 0.7–2.7% over the best of them. **Preference dial.** Sweeping one weight over $\{0.40, 0.55, 0.70, 0.85, 1.00\}$ (Table 10) gives the same picture on the scalar objective (120 wins, 8 ties, 7 losses over 135 point–seed–baseline pairs) but not on the named metric at intermediate weights, where FULCRUM uses 1.6–6.8% more water and emits 1.0–2.4% more carbon than the best baseline because it spends the remaining weight elsewhere. The dial does what the scalar objective asks; whether the named metric follows depends on how much weight remains.

6.4 Zero-shot transfer across fleet sizes

Table 4 and Fig. 2(b) test (L2). The scalability fleets are heterogeneous in capacity and use a sparse WAN, so the $N = 3$ row differs from Table 1; the comparison within each row is like-for-like. **(i)** FULCRUM, with weights trained at $N = 3$ and never updated, beats a Green-DCC controller retrained from scratch at every N by 6.6–8.2% reward, with a per-site reward that stays flat (-7.5,

-7.0, -7.4, -7.5 at $N=3, 5, 7, 10$) while Green-DCC's degrades (-8.2, -7.5, -8.0, -8.1). The gain comes mostly from cooling and QoS: FULCRUM keeps evaporative cooling to 4–5 of 24 hours against 10–15 for Green-DCC, using 1.5–2.4$\times$ less water at 14–28% lower QoS penalty, while Green-DCC has 3–9% lower electricity cost at $N=5$ and 10 (parity at $N=7$). **(ii)** The preference-conditioned baselines are not size-agnostic: their action heads are dimensioned by $3N + N(N-1)$, so the $N = 3$ checkpoints fail to load at $N \geq 5$; at $N = 3$ on the heterogeneous fleet they trail FULCRUM by 1.5–2.5%. Size transfer, not in-distribution reward, is where the graph structure pays off. **(iii)** The transfer is not free of drift: the per-site reward at $N=10$ is 0.4% below that at $N=3$, and a budget-matched retrain of FULCRUM at each N, which we did not run, would presumably do better still, so these numbers are a floor for the architecture.

6.5 Component ablations

Table 5 is a negative result that we consider important to report. Averaged over the five preferences, none of the architectural components moves the reward by a practical amount at this budget: removing cross-attention costs 1.0% and removing FiLM 0.3%, both with negligible effect sizes, and a plain attention GNN or a 107k-parameter preference-conditioned MLP is statistically indistinguishable from the 1.11M-parameter full model on in-distribution reward. The only practical change is the Boltzmann–Tchebycheff auxiliary critic, a component of an earlier design, which *lowers* reward by 11.6% ($\delta = 0.40$) by collapsing the WATER-MIN and BALANCED operating points (-12.3 vs. -8.2 and -21.5 vs. -17.9). The case for FULCRUM therefore rests on conditioning, on rank across preferences, and above all on size transfer (Sec. 6.4), which the flat baselines cannot attempt; not on FiLM or cross-attention, which we keep because they are cheap and do not hurt. Paired tests at each preference (App. F) are more sensitive and give the full model 16 practical wins, 33 ties and one loss on scalar reward against the ten variants, with cross-attention worth 1.1–3.5% at three of five preferences; but the full model is the best variant at no single preference, and several variants reach 1–4% lower carbon or water at the two green corners on this seed.

Table 4: **Fleet-size transfer** (50 episodes, balanced $\boldsymbol{\omega}$, mean $\pm$ std over episodes). FULCRUM uses the $N=3$ weights unchanged; Green-DCC is retrained from scratch at each N with the same 10^6-step budget. Pref-MLP/GNN checkpoints cannot be loaded at $N \geq 5$ (action head dimension). Last column: Green-DCC water divided by FULCRUM water.

N	FULCRUM (0-shot)	Green-DCC (retrained)	Δ	Water ratio
3	-22.5 ± 0.4	-24.5 ± 0.4	+8.2%	2.2$\times$
5	-35.1 ± 0.5	-37.6 ± 0.4	+6.6%	2.4$\times$
7	-51.5 ± 0.6	-55.8 ± 0.6	+7.7%	1.5$\times$
10	-75.3 ± 0.7	-81.1 ± 0.7	+7.1%	1.5$\times$

6.6 Lifecycle compute

Table 6 addresses (L3) with measured wallclock rather than projections. Serving five preferences with one preference-conditioned run costs 5.0$\times$ less than five same-architecture specialists (4.8$\times$ for

Table 5: **Ablations at matched budget** (10^6 steps each, seed 42, 50 episodes $\times$ five preferences). Mean scalar reward over the 250 preference–episode pairs; margin is (full $-$ variant) relative to |full| (positive: full model better); paired Wilcoxon over the 250 pairs, Holm-corrected; "practical" requires margin $\geq 1\%$ and $|\delta| \geq 0.147$. The extras-off floor keeps the GPM but disables FiLM, cross-attention, and every add-on.

Group	Variant	5-pref. reward	Margin	Cliff's δ	Practical?
	FULCRUM (full)	-15.06	—	—	—
Architecture	no preference cross-attention	-15.21	+0.99%	0.07	no
	no FiLM	-15.11	+0.33%	0.02	no
	extras-off GPM floor	-15.19	+0.87%	0.05	no
	standard attention GNN in place of GPM	-15.04	-0.09%	-0.01	no ($p=0.23$)
Same trainer	Pref-GNN-PPO (91k parameters)	-15.32	+1.72%	0.12	no ($\lvert\delta\rvert < 0.147$)
	Pref-MLP-PPO (107k parameters)	-15.04	-0.09%	0.00	no ($p=0.26$)
Mechanism	migration disabled	-15.14	+0.53%	0.04	no
Add-ons (off in full)	+ Boltzmann–Tchebycheff auxiliary critic	-17.04	+11.6%	0.40	**yes (hurts)**
	+ preference-sensitivity margin	-15.05	-0.02%	0.01	no ($p=1.0$)
	+ per-preference advantage normalization	-15.01	-0.30%	-0.01	no

Table 6: **Measured training compute for serving five preferences** (single GPU, 10^6 steps per run, seed 42; 250 W, 387 gCO_2/kWh). "Mean reward": five-preference mean on 25 held-out episodes. Ratios compare one-shot vs. five specialists of the same architecture.

	Strategy	Runs	Wallclock	kWh	gCO_2	Mean reward	Ratio
A	FULCRUM, one run, all preferences zero-shot	1	1.05 h	0.26	102	−15.06	—
C	FULCRUM architecture, one specialist per preference	5	5.29 h	1.32	512	−14.86	5.0× A
B	Pref-MLP, one run, all preferences zero-shot	1	0.77 h	0.19	74	−15.04	—
D	Pref-MLP, one specialist per preference	5	3.64 h	0.91	352	−14.89	4.8× B

the MLP), and the specialists recover only 1.3% (1.0%) of mean reward for that cost. For a pref-blind controller such as Green-DCC, the ratio is the number of (preference, fleet-size) configurations served, 20 runs against one here. A single run is cheap (≈ 0.26 kWh); the argument concerns the multiplier, which grows with each preference revision and fleet change.

7 Discussion

What the evidence supports. FULCRUM is one policy that (a) serves any operator preference at inference time, with the best mean rank among ten controllers at the canonical corners, a higher reward than every baseline at all seven held-out preferences, and the lowest cost, carbon, water, and QoS penalty of any baseline at the one-hot corners, (b) transfers unchanged to fleets more than three times its training size and there beats a controller retrained at each size, and (c) costs one fifth of the training compute of per-preference specialists, for about 1% of reward; (b) is what separates it from the flat conditioned baselines.

What it does not support. FULCRUM does not dominate every baseline on every metric. At the 0.7 training corners the conditioned baselines emit 1% less carbon and use 3% less water, and at intermediate dial weights 2–7% less water and 1–2% less carbon; one of three seeds emits 3% more carbon than every conditioned baseline at the one-hot carbon corner; on real traces MAPPO and the per-site specialists are 1.6% ahead; FiLM and cross-attention are not individually significant; and a 107k-parameter MLP matches a 1.11M-parameter graph policy at the training fleet size. The named-metric deficits share a cause: whenever weight is left on the other objectives, FULCRUM spends it, raising the scalar reward it is trained on and the named metric with it; at the one-hot corners, where nothing is left to trade, they vanish on the three-seed mean but not in every seed.

Limitations. The simulator is calibrated on real traces but is not a production system, and real-trace evaluation covers workload, carbon, and weather but not real prices or grid constraints. Green-DCC and MAPPO are architecture-faithful re-implementations trained in our simulator and could be under-tuned; the scalability comparison retrains Green-DCC but not FULCRUM, so a retrained FULCRUM is an unmeasured upper bound; and attention weights, though inspectable, are not claimed to be explanations. Also, although Pref-MLP-PPO and Pref-GNN-PPO isolate the effects of preference conditioning and graph structure under a matched PPO budget, we do not evaluate a dedicated state-of-the-art MORL algorithm from the broader MORL literature.

8 Conclusion

FULCRUM is a preference-conditioned graph reinforcement learning framework for multi-data-center control. Under a budget-matched evaluation over three training seeds, it achieves the best mean rank across five canonical preferences and the highest reward across seven held-out preferences among the evaluated controllers. At the one-hot preference corners, it also achieves the lowest cost, carbon, water, and QoS penalty among the evaluated baselines. Trained at three sites, FULCRUM transfers without retraining to fleets with five, seven, and ten sites and outperform a Green-DCC controller retrained for each size under the same budget. A single FULCRUM training run requires 5× less training compute than five per-preference specialists, at a 1.3% lower mean reward. These results support preference adaptation and zero-shot fleet-size transfer as the main benefits of the proposed approach, while the remaining gaps and limitations discussed in the paper indicate directions for future work.

AI USE STATEMENT

We used generative AI tools for limited language-related assistance, including grammar correction, proofreading, improving clarity and readability, and minor suggestions for wording and presentation. Generative AI tools were not used to generate the core technical ideas, methodology, experimental design, results, or conclusions of this work. All AI-assisted content was reviewed and edited by the authors, and the final manuscript was checked for correctness and consistency. The authors take full responsibility for the final content of the work, including all claims, text, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work presents a computational reinforcement learning framework for sustainable multi-data-center control and evaluates it in a simulated environment using public workload, carbon intensity, and weather traces. The study does not involve human subjects or personal data. The use of the framework can shift operational priorities among cost, carbon, water, and QoS, so deployed preferences should be explicitly specified and logged. The authors are not aware of any additional ethical concerns arising from the methods, data, or experiments presented in this work.

REPRODUCIBILITY STATEMENT

An anonymized version of the code is provided as supplementary material, along with scripts and configuration files to reproduce the reported experiments. The main paper describes the problem formulation, method, and experimental protocol, while the supplementary material provides additional implementation details, dataset and trace specifications, training procedures, hyperparameters, statistical procedures, and information on the computational environment. Result files also include algorithm fingerprints to identify the training and environment configurations used to generate them.

REFERENCES

Axel Abels, Diederik M Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 11–20, 2019.

Toygun Basaklar, Suat Gumussoy, and Umit Y Ogras. PD-MORL: Preference-driven multi-objective reinforcement learning algorithm. In *International Conference on Learning Representations (ICLR)*, 2023.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Norman Cliff. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114(3):494–509, 1993.

Balthazar Donon, Rémi Clément, Benjamin Donnot, Antoine Marot, Isabelle Guyon, and Marc Schoenauer. Neural networks for power flow: Graph neural solver. In *Power Systems Computation Conference (PSCC)*, 2020.

Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

International Energy Agency. Electricity 2024: Analysis and forecast to 2026. 2024. https://www.iea.org/reports/electricity-2024.

Mohammad A. Islam, Shaolei Ren, A. Hasan Mahmud, and Gang Quan. Water-constrained geographic load balancing in data centers. *IEEE Transactions on Cloud Computing*, 2016. doi: 10.1109/TCC.2015.2453982.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Pengfei Li, Jianyi Yang, Mohammad A. Islam, and Shaolei Ren. Making AI less thirsty: Uncovering and addressing the secret water footprint of AI models. *Communications of the ACM*, 68(7):54–61, 2025.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)*, 2018.

Yuanlong Li, Yonggang Wen, Dacheng Tao, and Kyle Guan. Transforming cooling optimization for green data center via deep reinforcement learning. *IEEE Transactions on Cybernetics*, 50(5): 2002–2013, 2020.

Qiliang Liu and Zhibin Yu. Imbalance in the cloud: An analysis on Alibaba cluster trace. In *IEEE International Conference on Big Data*, pages 2884–2886, 2018.

Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. Recalibrating global data center energy-use estimates. *Science*, 367(6481):984–986, 2020.

David Mytton. Data centre water consumption. *npj Clean Water*, 4(1):11, 2021.

Avisek Naug, Soumyendu Sarkar, Antonio Guillen, Vineet Gundecha, Ricardo Luna Gutiérrez, Sahand Ghorbanpour, Sajad Mousavi, Desik Rengarajan, Ashwin Ramesh Babu, and Cullen Bash. SustainCluster: A benchmark and reference implementation for sustainable geo-distributed data-center clusters (green-dcc), 2024. Companion codebase for Sarkar et al. (2025); https://github.com/HewlettPackard/sustain-cluster.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.

Ana Radovanovic. Our data centers now work harder when the sun shines and wind blows. *Google Sustainability Blog*, 2020. https://blog.google/inside-google/infrastructure/data-centers-work-harder-sun-shines-wind-blows.

Ana Radovanovic, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duber, Bhaskar Roy, Sean Xiao, Ming Harik, Nishant Harris, Robin Rapp, et al. Carbon-aware computing for datacenters. *IEEE Transactions on Power Systems*, 38(2):1270–1280, 2023.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Max Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Charles Reiss, John Wilkes, and Joseph L Hellerstein. Google cluster-usage traces: Format + schema. *Google Inc., White Paper*, 2011.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

Soumyendu Sarkar, Avisek Naug, Antonio Guillen, Vineet Gundecha, Ricardo Luna Gutiérrez, Sahand Ghorbanpour, Sajad Mousavi, Ashwin Ramesh Babu, Desik Rengarajan, and Cullen Bash. Hierarchical multi-agent framework for carbon-efficient liquid-cooled data center clusters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 29694–29696, 2025.

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

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green AI. *Communications of the ACM*, 63(12):54–63, 2020.

Arman Shehabi, Sarah Smith, Dale Sartor, Richard Brown, Magnus Herrlin, Jonathan Koomey, Eric Masanet, Nathaniel Horner, Inês Azevedo, and William Lintner. United states data center energy usage report. Technical Report LBNL-1005775, Lawrence Berkeley National Laboratory, 2016.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Jun Sun, Shibo Chen, Pengcheng You, Qinmin Yang, and Zaiyue Yang. Battery-assisted online operation of distributed data centers with uncertain workload and electricity prices. *IEEE Transactions on Cloud Computing*, 11(1):599–610, 2023. doi: 10.1109/TCC.2021.3132174.

Ekaterina Tolstaya, Fernando Gama, James Paulos, George Pappas, Vijay Kumar, and Alejandro Ribeiro. Learning decentralized controllers for robot swarms with graph neural networks. In *Conference on Robot Learning (CoRL)*, pages 671–682, 2020.

U.S. Environmental Protection Agency. eGRID 2023 summary tables, 2024. `https://www.epa.gov/egrid`.

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

Avisek Naug Wang, Nicholas Christianson, Chia-Hao Chu, Fan Chen, Lian Ling, Yuqian Wen, Amritanshu Bose, Dian Li, Yanming Li, Jim Gao, et al. SustainDC: Benchmarking for sustainable data center control. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, volume 37, 2024.

Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International Conference on Machine Learning (ICML)*, pages 10607–10616, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

Runzhe Yang, Xingyou Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

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 (NeurIPS)*, volume 35, 2022.

A NOTATION

Table 7: Notation summary.

Symbol	Meaning
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Multi-DC graph (Definition 1); $N = \lvert\mathcal{V}\rvert$, $\lvert\mathcal{E}\rvert = N(N-1)$
$\mathbf{h}_i \in \mathbb{R}^{34}$, $\mathbf{e}_{ij} \in \mathbb{R}^6$	Node and edge features
$\mathbf{H} \in \mathbb{R}^{N\times d}$, $\tilde{\mathbf{H}}$	Node embeddings after GPM layers / after preference cross-attention ($d = 256$)
$\boldsymbol{\omega} \in \Delta^{K-1}$	Operator preference; $K = 4$ objectives (cost, carbon, water, QoS)
$\mathbf{r}_t$, $R_t(\boldsymbol{\omega})$	Vector reward (Eq. 1); scalarized reward $\boldsymbol{\omega}^\top(\mathbf{r}_t \oslash \mathbf{s})$
$\mathbf{s} = [50, 130, 2200, 120]$	Per-objective reward scales
$\gamma_k(\boldsymbol{\omega}), \beta_k(\boldsymbol{\omega})$	FiLM scale and shift for objective head k (Eq. 3)
$\mathbf{u}_k$, $\mathbf{z}_k = \omega_k \mathbf{u}_k$	Learned objective embedding and preference-scaled objective token
m_{ij}, e_i, δ_i, c_i	Migration fraction, storage command, deferral release, cooling mode
V, $\mathbf{v} \in \mathbb{R}^K$	Scalar and per-objective value estimates

B PROOFS

B.1 PROOF OF PROPOSITION 5

Let σ be a bijection on $\mathcal{V}$ and write $\sigma\cdot(\mathbf{H}, \mathbf{E})$ for the graph with rows of $\mathbf{H}$ and edge index pairs permuted accordingly. (i) The node encoder applies one MLP to each row of $\mathbf{H}$ independently, so it commutes with σ. (ii) Each GPM logit (Eq. 2) is a function of $(\mathbf{h}_i, \mathbf{h}_j, \mathbf{e}_{ij})$ only, and the FiLM parameters (Eq. 3) depend on $\boldsymbol{\omega}$ only; the softmax over $\mathcal{N}(i)$ and the aggregation $\sum_j \alpha_{ij}^{(k)} \mathbf{W}_k \mathbf{h}_j$ are invariant to the ordering of $\mathcal{N}(i)$. Hence each GPM layer is equivariant. (iii) The cross-attention (Eq. 4) is applied row-wise to $\mathbf{H}$ against keys and values $\mathbf{Z}$ that do not depend on nodes, so it commutes with σ. (iv) Node heads act row-wise on $[\tilde{\mathbf{h}}_i \| \boldsymbol{\omega}]$; the edge head acts pair-wise on $[\tilde{\mathbf{h}}_i \| \tilde{\mathbf{h}}_j \| \mathbf{e}_{ij} \| \boldsymbol{\omega}]$; the value head sum-pools over nodes, which is permutation-invariant. Composing, $\pi_\theta(\sigma\cdot(\mathbf{H}, \mathbf{E}), \boldsymbol{\omega}) = \sigma \cdot \pi_\theta((\mathbf{H}, \mathbf{E}), \boldsymbol{\omega})$. No parameter tensor has a dimension equal to N or $\lvert\mathcal{E}\rvert$ ($\mathbf{W}_k, \mathbf{U}_k, \mathbf{a}_k, \mathbf{W}_{Q,K,V,O}, \mathbf{u}_k$ and all head MLPs are dimensioned by d, d_n, d_e, K only), so the same θ is well-typed for any $N' \geq 2$. □

C STEERING: FULCRUM'S OPERATING POINT UNDER EACH PREFERENCE

Table 8: FULCRUM (seed 42, 50 episodes) under each canonical preference (weight 0.7, top) and at the one-hot corners (bottom). Parentheses: change relative to the BALANCED row. Every preference lowers its own named metric; the cost of doing so appears in the other columns.

Preference	Cost ($)	Carbon (kg)	Water (L)	QoS pen.
BALANCED	1652	7445	8180	34.8
COST-MIN	1036 (−37%)	5354 (−28%)	19840 (+143%)	288.5 (+729%)
CARBON-MIN	1261 (−24%)	5502 (−26%)	15056 (+84%)	230.1 (+561%)
WATER-MIN	1728 (+5%)	7828 (+5%)	1810 (−78%)	63.2 (+82%)
QOS-MAX	1802 (+9%)	8118 (+9%)	6115 (−25%)	4.4 (−87%)
$\boldsymbol{\omega} = \mathbf{e}_{\text{cost}}$	**1015** (−39%)	5112 (−31%)	22670 (+177%)	294.1 (+745%)
$\boldsymbol{\omega} = \mathbf{e}_{\text{carbon}}$	1182 (−28%)	**5243** (−30%)	17259 (+111%)	278.7 (+701%)
$\boldsymbol{\omega} = \mathbf{e}_{\text{water}}$	1466 (−11%)	7052 (−5%)	**1564** (−81%)	218.7 (+528%)
$\boldsymbol{\omega} = \mathbf{e}_{\text{qos}}$	1883 (+14%)	8508 (+14%)	3951 (−52%)	**4.6** (−87%)

The mechanism is visible in the cooling and migration statistics: evaporative-cooling fraction is 0.81 at COST-MIN, 0.60 at CARBON-MIN, 0.00 at WATER-MIN, 0.14 at QOS-MAX, and 0.22 at BALANCED (0.98, 0.72, 0.00, 0.07 at the one-hot corners); migrated work per episode is 0.8, 6.1, 5.2, 4.3, and 5.3 units respectively (0.5, 3.6, 5.0, 2.8 at the one-hot corners). Under COST-MIN, the policy accepts a large QoS penalty by deferring work to cheap hours; the operator who does not want that should not place 0.7 or more of the weight on cost.

D EQUAL-QOS COMPARISON

Because controllers settle at different QoS levels, we interpolate FULCRUM's preference curve (the five canonical operating points, seed 42) for each baseline's QoS penalty and compare the remaining objectives at matched QoS levels. At Green-DCC's QoS (57.5), FULCRUM uses 3412 L of water against 18738 L (-82%), at \$1668 against \$1634 electricity ($+2.1\%$) and 7764 against 6802 kg ($+14\%$). At MAPPO's QoS (33.2), FULCRUM uses 8078 L against 18656 L (-57%), \$1622 against \$1640 (-1.1%), and 7527 against 7169 kg ($+5.0\%$). At the per-site specialists' QoS (23.1), FULCRUM uses 7420 L against 11872 L (-37%) at $+4.7\%$ cost and $+1.1\%$ carbon. FULCRUM therefore buys large water reductions with small cost increases and moderate carbon increases at equal QoS; whether that is the right trade is a matter of $\boldsymbol{\omega}$, which is the point of exposing it.

E EVAL-ONLY EXPERIMENTS: HELD-OUT PREFERENCES AND THE PREFERENCE DIAL

All experiments in this section and in Sec. 6.3 re-evaluate the stored 10^6-step checkpoints of every method (three training seeds) with the evaluation path of the multi-preference study: the same environment configuration, the same 50 episode seeds (700–749), deterministic actions, and the paired Wilcoxon/Holm/Cliff's-δ procedure of App. G within each preference. Nothing is retrained, and no fingerprinted module is touched. The one-hot corners $\boldsymbol{\omega} = \mathbf{e}_k$ and the seven held-out preferences below were never a corner of the training curriculum for any method (the curriculum uses weight 0.7 corners, the centroid, and Dirichlet samples), so they are mildly out of distribution for all methods alike.

Table 9: **Held-out preferences**: scalar reward (mean $\pm$ std across three training seeds; 50 episodes each) of the preference-conditioned controllers at seven preferences that were never a training corner. Column headers give $\boldsymbol{\omega}$ in the order (cost, carbon, water, QoS). Rank is the mean rank over the seven. Best per column in **bold**.

Method	Mild-Cost (.55,.15,.15,.15)	Mild-Carbon (.15,.55,.15,.15)	Mild-Water (.15,.15,.55,.15)	Mild-QoS (.15,.15,.15,.55)	Cost–Water (.4,.1,.4,.1)	Carbon–QoS (.1,.4,.1,.4)	Near-Equal (.3,.25,.25,.2)	Rank
Pref-MLP-PPO	-21.60 ± 0.05	-20.81 ± 0.01	-11.62 ± 0.04	-11.16 ± 0.10	-17.23 ± 0.03	-15.63 ± 0.08	-19.55 ± 0.13	3.4
Pref-GNN-PPO	-21.49 ± 0.05	-20.73 ± 0.21	-11.68 ± 0.11	-11.17 ± 0.10	-17.11 ± 0.04	-15.61 ± 0.08	-19.43 ± 0.05	3.0
Pref-MLP-PPO (vector critic)	-22.10 ± 0.31	-21.00 ± 0.17	-11.74 ± 0.10	-11.23 ± 0.13	-17.47 ± 0.25	-15.82 ± 0.11	-19.76 ± 0.21	4.4
FULCRUM (no migration)	$\mathbf{-21.09} \pm 0.04$	-20.54 ± 0.12	-11.73 ± 0.11	-11.30 ± 0.07	$\mathbf{-16.78} \pm 0.01$	-16.06 ± 0.11	-19.35 ± 0.10	3.0
FULCRUM (ours)	-21.19 ± 0.07	$\mathbf{-20.34} \pm 0.13$	$\mathbf{-11.44} \pm 0.03$	$\mathbf{-10.86} \pm 0.04$	-16.87 ± 0.06	$\mathbf{-15.50} \pm 0.12$	$\mathbf{-19.18} \pm 0.12$	**1.3**

Table 10: **Preference dial.** One weight is swept over $\{0.40, 0.55, 0.70, 0.85, 1.00\}$ with the remainder split equally over the other three objectives; entries are the three-seed mean of the axis's named metric (\$, kg, or L). "Min" is the smallest value each method reaches anywhere on its dial (the one-hot end for every method except Pref-GNN on water, which bottoms out at 0.85). Ranks are mean ranks over the five dial points by scalar reward and by named metric; W/T/L counts practical wins/ties/losses of FULCRUM against the row over the 5×3 point–seed pairs, on reward and on the named metric.

		weight on the swept objective						rank		W/T/L	
Axis	Method	0.40	0.55	0.70	0.85	1.00	Min	reward	metric	reward	metric
Cost ($)	Pref-MLP-PPO	1406	1127	1052	1036	1031	1031	3.4	3.0	13/2/0	11/3/1
	Pref-GNN-PPO	1359	1141	1070	1051	1048	1048	3.7	4.1	14/1/0	13/1/1
	Pref-MLP-PPO (vector critic)	1435	1155	1098	1082	1076	1076	4.9	4.5	15/0/0	14/0/1
	FULCRUM (no migration)	**1316**	**1087**	**1028**	**1017**	**1013**	**1013**	**1.1**	**1.1**	0/15/0	0/12/3
	FULCRUM	1381	1094	1035	1022	1018	1018	1.9	2.3	—	—
Carbon (kg)	Pref-MLP-PPO	6686	5642	**5236**	5134	5096	5096	2.9	3.1	12/1/2	6/5/4
	Pref-GNN-PPO	**6521**	5690	5271	5140	5114	5114	3.5	3.1	11/2/2	7/2/6
	Pref-MLP-PPO (vector critic)	6784	5558	5247	5152	5114	5114	4.2	3.1	12/1/2	6/4/5
	FULCRUM (no migration)	6827	**5551**	5250	5128	5071	5071	2.5	**2.6**	5/10/0	1/9/5
	FULCRUM	6681	5669	5288	**5118**	**5060**	**5060**	**1.9**	3.0	—	—
Water (L)	Pref-MLP-PPO	**2064**	1865	1832	1742	1642	1642	2.7	2.9	13/1/1	6/5/4
	Pref-GNN-PPO	2418	**1821**	**1790**	1736	1738	1728	3.6	2.3	15/0/0	4/3/8
	Pref-MLP-PPO (vector critic)	2449	1873	1858	1818	1762	1762	4.4	4.5	15/0/0	10/4/1
	FULCRUM (no migration)	2466	1968	1815	**1693**	**1578**	**1578**	3.0	**2.1**	10/3/2	3/4/8
	FULCRUM	2205	1856	1841	1763	1592	1592	**1.3**	3.1	—	—

Two features of Table 10 deserve comment. First, on the scalar objective the three-seed mean of FULCRUM is ahead of every conditioned baseline at each of the 15 dial points, and only its own

no-migration variant is ever ahead of it, chiefly on the cost axis, where migration is pure overhead; the seven per-seed practical losses all fall at carbon weights ≥ 0.85 on seed 42 and at the water corner on seed 44. Second, the named-metric ranks on the carbon and water axes are close to uniform among the conditioned methods (2.1–3.1, except the vector-critic MLP on water), and they vary more across training seeds than across methods: on carbon, seed 42 of FULCRUM is 2–6% behind every conditioned baseline at weights ≥ 0.55 while seeds 43 and 44 are 3–4% and 2% ahead at weights ≥ 0.85; on water at weight 0.40, seed 42 uses 13–44% more water than every conditioned baseline while seed 44 uses 15–37% less and seed 43 7–13% less than two of the three. A claim that one architecture reaches lower water or carbon than another at an interior preference is therefore not supported by three seeds; the claims we make (rank, held-out reward, one-hot minima, size transfer) are the ones that are. As a coverage summary, the normalized dominated hypervolume of the five canonical operating points (App. G procedure; objectives scaled to $[0, 1]$ over the learned methods) is 0.22, 0.14, 0.20 for FULCRUM on seeds 42–44 against 0.14, 0.15, 0.19 for Pref-GNN and 0.12, 0.14, 0.17 for Pref-MLP: larger for FULCRUM on two of three seeds.

F PER-PREFERENCE ABLATION DETAIL

Table 11: Scalar reward of each ablation variant at each canonical preference (seed 42, 50 episodes, 10^6 training steps per variant). Best per column in **bold**.

Variant	Cost-min	Carbon-min	Water-min	QoS-max	Balanced
FULCRUM (full)	−21.20	−20.50	−8.23	−7.48	−17.88
no preference cross-attention	−21.37	−20.74	−8.32	−7.77	−17.85
no FiLM	−21.41	−20.59	−8.20	−7.55	−17.79
extras-off GPM floor	−21.47	−20.75	−8.26	−7.51	−17.96
standard attention GNN	−21.37	−20.40	**−8.19**	−7.54	−17.72
Pref-GNN-PPO	−21.67	−20.86	−8.37	−7.77	−17.94
Pref-MLP-PPO	−21.41	**−20.37**	−8.30	−7.41	−17.73
migration disabled	**−21.12**	−20.49	−8.30	−7.79	−18.00
+ Boltzmann–Tchebycheff auxiliary critic	−21.86	−21.33	−12.29	−8.21	−21.51
+ preference-sensitivity margin	−21.29	−20.53	−8.29	−7.44	−17.73
+ per-preference advantage normalization	−21.36	−20.39	−8.26	**−7.40**	**−17.65**

The full model is within 1.3% of the best variant at every preference but is the best at none: disabling migration is marginally ahead at COST-MIN, the Pref-MLP at CARBON-MIN, the plain attention GNN at WATER-MIN, and per-preference advantage normalization at QOS-MAX and BALANCED. On the named metrics of the two convex-hull corners, disabling migration is marginally better than the full model ($1029 vs. $1038 at COST-MIN; 5244 vs. 5321 kg at CARBON-MIN), confirming Sec. 6.2: migration is a QoS and water lever at $N = 3$. The Tchebycheff add-on collapses the WATER-MIN and BALANCED corners; we retain it only as a documented negative result.

Paired per-preference tests. The pooled test of Table 5 mixes five reward scales, which dilutes small per-preference effects. Table 12 re-rolls every stored variant on the 50 multi-preference episodes and applies the paired procedure of App. G separately at each preference, on the scalar reward, and on the preference's named metric. Over the 50 scalar comparisons the full model has 16 practical wins, 33 ties and one loss (per-preference advantage normalization at BALANCED, 1.3%); the per-preference view is more sensitive than the pooled one and shows that removing cross-attention costs a practical 1.1–3.5% at three of five preferences, and the extras-off floor 1.1–1.4% at the two cost-like corners, even though neither reaches the pooled threshold. On the named metrics the picture is mixed: 20 wins, 17 ties and 13 losses, the losses concentrated at CARBON-MIN (six variants reach 1.1–3.3% lower carbon) and WATER-MIN (five variants reach 1.8–4.0% lower water), the same two corners at which training seed 42 is behind the conditioned baselines in Table 10. These are single-seed comparisons (the ablation checkpoints exist for seed 42 only), so they establish that FiLM and cross-attention are not required for the headline behavior and that no variant is uniformly better, not that any variant should replace the full model.

Table 12: Paired verdict of the full model against each variant at each canonical preference (seed 42, 50 paired episodes, Holm-corrected within each preference). W/T/L = practical win/tie/loss *for the full model*; the subscript is the signed relative margin in % (positive: full model better). Left block: scalar reward; right block: the preference's named metric (cost, carbon, water, QoS penalty; scalar reward for BALANCED). The value-clipping variant is numerically identical to the full model at evaluation and is omitted.

	Scalar reward					Named metric				
Variant	Cost	Carbon	Water	QoS	Bal.	Cost	Carbon	Water	QoS	Bal.
no preference cross-attention	$\mathrm{T}_{+0.9}$	$\mathrm{W}_{+1.1}$	$\mathrm{W}_{+1.1}$	$\mathrm{W}_{+3.5}$	$\mathrm{T}_{-0.2}$	$\mathrm{W}_{+1.4}$	$\mathrm{T}_{+0.9}$	$\mathrm{L}_{-4.0}$	W_{+29}	$\mathrm{T}_{-0.2}$
no FiLM	$\mathrm{T}_{+1.0}$	$\mathrm{T}_{+0.5}$	$\mathrm{T}_{-0.3}$	$\mathrm{W}_{+1.1}$	$\mathrm{T}_{-0.5}$	$\mathrm{W}_{+2.3}$	$\mathrm{L}_{-1.2}$	$\mathrm{T}_{+1.0}$	$\mathrm{T}_{-5.6}$	$\mathrm{T}_{-0.5}$
extras-off GPM floor	$\mathrm{W}_{+1.4}$	$\mathrm{W}_{+1.1}$	$\mathrm{T}_{+0.2}$	$\mathrm{T}_{+0.3}$	$\mathrm{T}_{+0.3}$	$\mathrm{W}_{+3.0}$	$\mathrm{L}_{-1.2}$	$\mathrm{W}_{+2.1}$	$\mathrm{T}_{-1.0}$	$\mathrm{T}_{+0.3}$
standard attention GNN	$\mathrm{T}_{+0.8}$	$\mathrm{T}_{-0.4}$	$\mathrm{T}_{-0.3}$	$\mathrm{T}_{+0.6}$	$\mathrm{T}_{-0.9}$	$\mathrm{W}_{+3.4}$	$\mathrm{T}_{+0.8}$	$\mathrm{W}_{+1.6}$	W_{+17}	$\mathrm{T}_{-0.9}$
Pref-GNN-PPO	$\mathrm{W}_{+2.2}$	$\mathrm{W}_{+1.8}$	$\mathrm{W}_{+1.9}$	$\mathrm{W}_{+3.9}$	$\mathrm{T}_{+0.4}$	$\mathrm{W}_{+3.6}$	$\mathrm{W}_{+1.8}$	$\mathrm{L}_{-1.8}$	W_{+18}	$\mathrm{T}_{+0.4}$
Pref-MLP-PPO	$\mathrm{T}_{+1.0}$	$\mathrm{T}_{-0.7}$	$\mathrm{T}_{+1.0}$	$\mathrm{T}_{-0.8}$	$\mathrm{T}_{-0.8}$	$\mathrm{W}_{+3.6}$	$\mathrm{L}_{-3.3}$	$\mathrm{L}_{-2.5}$	L_{-40}	$\mathrm{T}_{-0.8}$
migration disabled	$\mathrm{T}_{-0.3}$	$\mathrm{T}_{-0.2}$	$\mathrm{T}_{+1.0}$	$\mathrm{W}_{+4.7}$	$\mathrm{T}_{+0.7}$	$\mathrm{T}_{-0.7}$	$\mathrm{L}_{-1.6}$	$\mathrm{T}_{-1.0}$	W_{+44}	$\mathrm{T}_{+0.7}$
+ Boltzmann–Tchebycheff critic	$\mathrm{W}_{+2.9}$	$\mathrm{W}_{+3.7}$	W_{+33}	$\mathrm{W}_{+8.9}$	W_{+17}	$\mathrm{W}_{+4.3}$	$\mathrm{W}_{+2.2}$	W_{+80}	W_{+37}	W_{+17}
+ preference-sensitivity margin	$\mathrm{T}_{+0.4}$	$\mathrm{T}_{+0.0}$	$\mathrm{T}_{+0.9}$	$\mathrm{T}_{-0.5}$	$\mathrm{T}_{-0.9}$	$\mathrm{W}_{+1.7}$	$\mathrm{L}_{-1.7}$	$\mathrm{L}_{-3.3}$	T_{-14}	$\mathrm{T}_{-0.9}$
+ per-pref. advantage norm.	$\mathrm{T}_{+0.7}$	$\mathrm{T}_{-0.4}$	$\mathrm{T}_{+0.5}$	$\mathrm{T}_{-0.9}$	$\mathrm{L}_{-1.3}$	$\mathrm{W}_{+3.3}$	$\mathrm{L}_{-1.1}$	$\mathrm{L}_{-2.1}$	T_{-12}	$\mathrm{L}_{-1.3}$
Total (W/T/L)	16 / 33 / 1					20 / 17 / 13				

G STATISTICAL PROCEDURE

For each family of comparisons (one table column or one preference) we compute, for every baseline b, the paired differences $d_e = x_e^{FULCRUM} - x_e^b$ over episodes e sharing the same seed, apply the two-sided Wilcoxon signed-rank test, and correct the family's p-values with Holm's step-down procedure at $\alpha = 0.05$ (Holm, 1979). We report the signed relative margin $\bar{d}/\overline{|x^b|}$ and Cliff's δ (Cliff, 1993). A difference is *practical* only if Holm-rejected, $|\text{margin}| \geq 1\%$, and $|\delta| \geq 0.147$; Holm-rejected differences below either threshold are reported as ties. With 50 paired episodes the Wilcoxon test rejects at margins far below 1%, which is why the practical floor is needed. Std across training seeds (Tables 1–2) measures training variability and is reported separately from episode variability.

H IMPLEMENTATION DETAILS

Architecture. Node encoder $\mathrm{Linear}(34, 256) \to \mathrm{ReLU} \to \mathrm{Linear}(256, 256)$; two GPM layers with $K = 4$ edge-aware attention heads, residual connection and LayerNorm; preference encoder $\mathrm{Linear}(4, \cdot) \to \mathrm{ReLU} \to \mathrm{Linear}(\cdot, 2K)$ producing $(\boldsymbol{\gamma}, \boldsymbol{\beta})$; preference cross-attention with $d = 256$; edge migration head and node storage/deferral/cooling heads (two-layer MLPs on the concatenations given in Sec. 4.1); sum-pooled value head with scalar and K-vector outputs. Parameter counts: node encoder 75k, preference encoder 3k, GPM layers 398k, cross-attention 264k, migration head 200k, node heads 3×34k, value head 68k; total 1,108,576. Pref-MLP-PPO: 107,555; Pref-GNN-PPO: 91,171; Green-DCC: ≈105k; MAPPO: 23–35k depending on observation size. Dropout is 0.

PPO. 10^6 environment steps; 96 episodes per update; 10 epochs; minibatch 256; $\gamma = 0.99$; GAE $\lambda = 0.95$; clip 0.2; no value clipping; value coefficient 0.5; vector-value coefficient 0.7; entropy $0.01 \to 0$; gradient-norm clip 0.5; Adam, learning rate 3×10^{-4} with 5% warm-up then linear decay to $0.1\times$; initial log-std -0.5, clamped to $[-3, 0.5]$. Tchebycheff, preference-sensitivity, objective-diversity, and per-preference advantage-normalization coefficients are 0 in the deployed model. Pref-MLP-PPO and Pref-GNN-PPO copy this configuration; MAPPO and Green-DCC use their published batch of 16 episodes (2604 updates for 10^6 steps against 434 for FULCRUM).

Preference curriculum. Each episode's $\boldsymbol{\omega}$ is drawn from a mixture of Dirichlet(α) with α annealed to 0.5 over training, the five canonical preferences (corner weight 0.7), and three copies of the balanced preference per batch.

Environment. Horizon 24 hourly steps; reward scales $\mathbf{s} = [50, 130, 2200, 120]$; QoS penalty weight 12; storage 1 MWh per site; maximum migration fraction 0.5 with decode bias 3; transfer volume 500 GB per workload unit priced per link; workload scale 1.18, base 0.5, amplitude 0.36, noise 0.08.

The direct-comparison benchmark uses homogeneous site capacity with exogenous heterogeneity (price, carbon intensity, weather, water stress, time zone); scalability fleets add capacity heterogeneity.

Compute methodology. Wallclock is measured on a single GPU; energy = wallclock $\times 250$ W; CO_2 = energy $\times 387$ gCO_2/kWh (U.S. Environmental Protection Agency, 2024); cooling water = energy $\times 11.2$ L/kWh (Shehabi et al., 2016). These are model-based conversions, not wall-socket measurements.

I BASELINES, DATA PROTOCOL, AND PROPERTY COMPARISON

I.1 PROPERTY COMPARISON

Table 13: Properties of the closest learned baselines. ✓ = part of the method; ✗ = requires retraining or architectural change.

Property	MAPPO	Green-DCC	Pref-MLP/GNN-PPO	FULCRUM
Inference-time preference control	✗	✗	✓	✓
Same weights for a new fleet size	✗	✗	✗	✓
Inter-site migration	✗	✓	✓	✓
Lifecycle retraining cost reported	✗	✗	(this paper)	✓

I.2 DATA AND SPLITS

Training uses synthetic hourly traces with the exogenous heterogeneity above. Synthetic evaluation uses 50 episode seeds (500–549 for the direct comparison, 700–749 for the multi-preference study) never used for training or checkpoint selection. Real-trace evaluation assigns a Google cluster trace (Reiss et al., 2011) to the California site and Alibaba traces (Liu and Yu, 2018) to the Texas and Virginia sites, each on an independent evaluation day, joined by local hour with real carbon-intensity and weather series; electricity prices remain synthetic. Scalability fleets are prefixes of a ten-region portfolio (CA, TX, VA, WA, GA, IL, NY, AZ, CO, NE) with capacity heterogeneity and a sparse WAN for $N \geq 5$; Green-DCC is retrained at each N for 41,666 episodes (10^6 steps), the same budget as its $N = 3$ run.

I.3 BASELINE FIDELITY

MAPPO follows the HARL implementation width (MLP $[64, 64]$, $\tanh$, feature normalization, orthogonal initialization) with a shared critic; it is preference-blind and acts every step with migration off. Green-DCC follows the SustainCluster architecture (MLP $[256, 256]$, ReLU) trained with PPO in our simulator (the original uses SAC), acting every step and decoding actions through the same decoder as FULCRUM. Both are trained inside our simulator with the same preference curriculum as FULCRUM but do not observe $\boldsymbol{\omega}$. Pref-MLP-PPO and Pref-GNN-PPO consume the same 34-d node and 6-d edge tensors as FULCRUM and share its PPO configuration; their action heads are dimensioned by $3N + N(N-1)$, which is why they cannot be loaded at $N \neq 3$. The per-site PPO specialists are Stable-Baselines3 PPO agents trained for 10^6 steps each in a single-site simulator on the balanced reward; they receive $3\times$ the environment steps of the other methods and are labelled as specialists for that reason.

J EVAL-ONLY EXPERIMENTS AND REPRODUCIBILITY

- **Code and provenance.** Anonymized code is submitted with the paper. Every result JSON carries an algorithm fingerprint of the training and environment modules; all numbers in this paper come from files with the same fingerprint.
- **Seeds.** Training seeds 42, 43, 44 for Tables 1–2; 42 for Tables 4–6. Evaluation episode seeds as in App. I.2.
- **Statistics.** App. G.

- **Eval-only experiments.** A script (`tools/eval_only_experiments.py`) re-evaluates the stored checkpoints, without training, at one-hot preferences, along preference sweeps, at held-out non-corner preferences, and per preference for every ablation variant, with the same paired statistics and the same 50 episode seeds as the multi-preference study. Its outputs are Table 3 (Sec. 6.3), the steering rows of Table 8, and Tables 9–12 (App. E–F). The one-hot, sweep and held-out stages were run for all three training seeds; the ablation stage for seed 42, the only seed with ablation checkpoints.
- **Figures.** Fig. 2 and the figures below are regenerated from the same JSONs by `visualize.py`.

K ADDITIONAL FIGURES

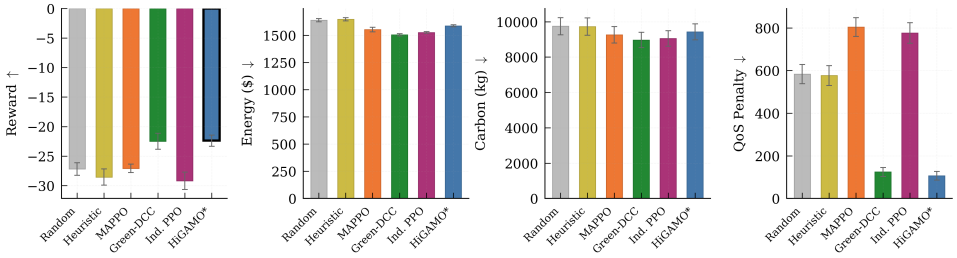


Figure 3: Direct comparison at the balanced preference (Table 1), all named metrics.

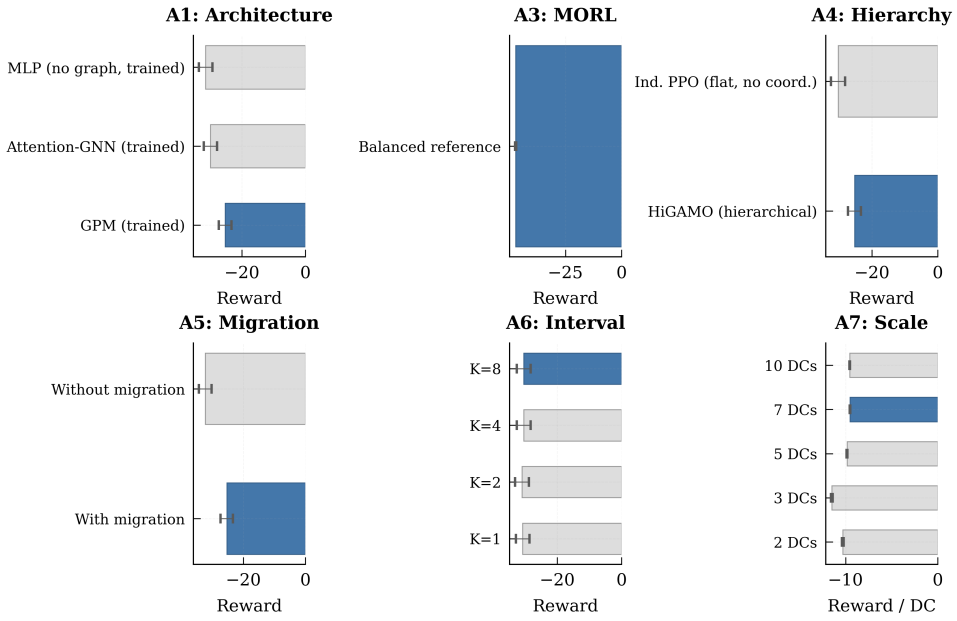


Figure 4: Ablation variants (Table 5) with per-episode spread.

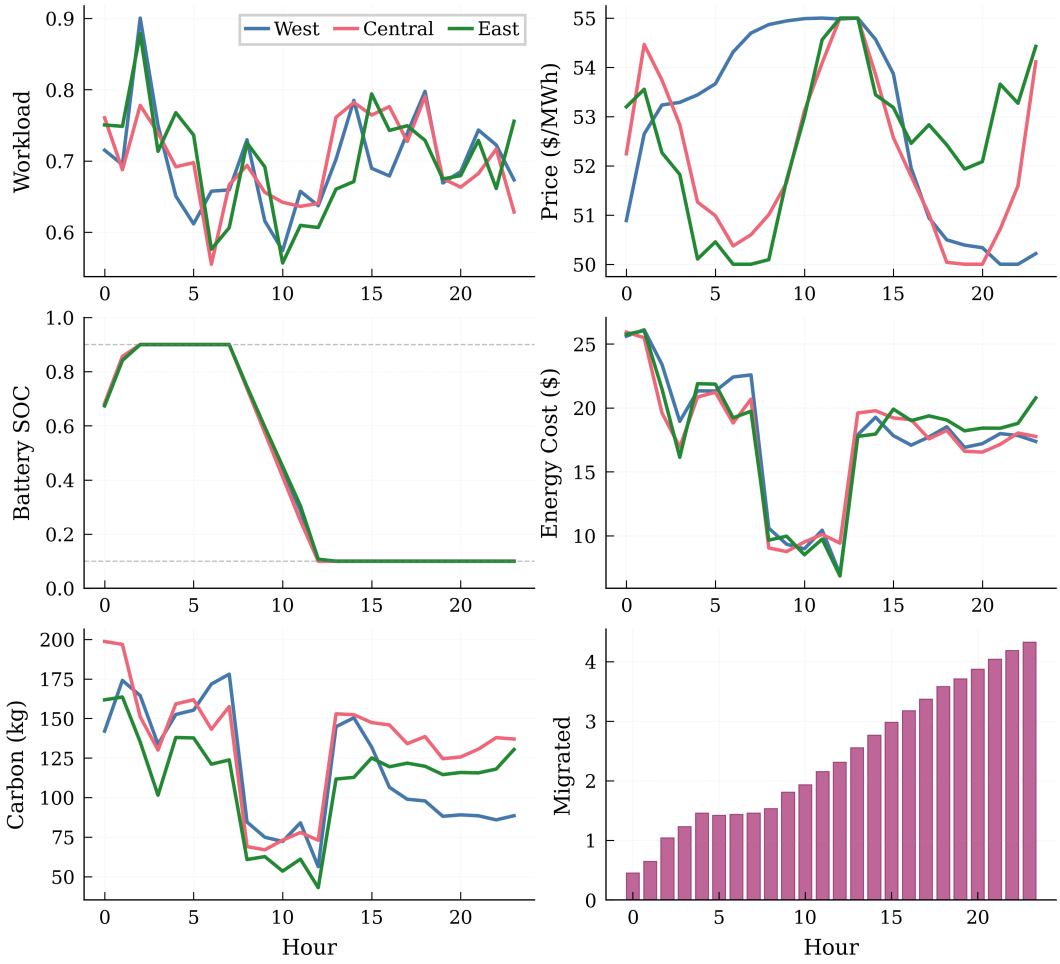
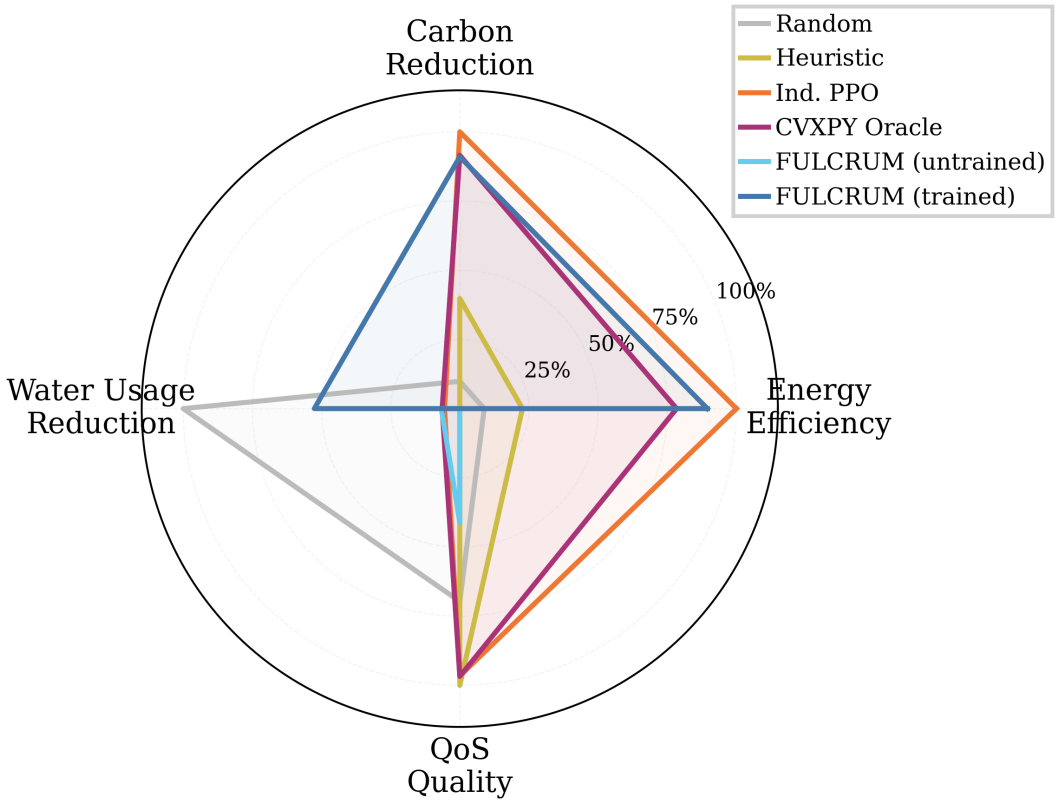


Figure 5: One held-out episode at the balanced preference: per-site load, cooling mode, storage, and migration decisions of FULCRUM.


Carbon
Reduction
Energy
Efficiency
QoS
Quality
Water Usage
Reduction
25%
50%
75%
100%
Random
Heuristic
Ind. PPO
CVXPY Oracle
FULCRUM (untrained)
FULCRUM (trained)