EU-CRD: RESPONSIBILITY-GUIDED CREDIT ASSIGNMENT FOR FORECAST-AWARE CARBON SCHEDULING

Anonymous authors
Paper under double-blind review

ABSTRACT

Geo-distributed datacentres can reduce carbon emissions by shifting delay-tolerant jobs across time and space to better match renewable supply. However, such spatial-temporal scheduling depends on weather-dependent energy forecasts, where inevitable forecasting errors yield suboptimal decisions or even mislead the scheduler. Conventional reinforcement learning learns from returns that reflect both routing quality and forecast quality, without an explicit mechanism to isolate their contributions. EU-CRD (Epistemic Uncertainty Counterfactual Responsibility Decomposition) addresses this credit-assignment problem by constructing a hindsight forecast regret proxy and a policy-action value gap, modulating the latter with critic ensemble disagreement, and using their relative magnitudes to reweight the routing advantage during training. No additional responsibility computation is required at deployment. In a five-datacentre simulation driven by measured wind, paired continuation runs from three shared checkpoints compare EU-CRD with a matched control retaining the same backbone, critic ensemble, responsibility computation and auxiliary training. Under deterministic decoding on six development windows, EU-CRD reduces carbon emissions by 14.5% with a degraded forecast and by 10.1% with a perfect forecast. The same models also reduce carbon by 12.2% and 10.5%, respectively, on six confirmation windows from a separate year. All three paired seeds favour EU-CRD under both forecast settings in both evaluations, with all service requirements met. On the development windows, alternative objectives and component ablations also yield higher carbon emissions under degraded forecasts for every seed. These results support responsibility-guided reweighting as a training-only mechanism for forecast-aware carbon scheduling.

1 INTRODUCTION

Datacentres already account for a substantial share of electricity demand (Jones, 2018), and the growth of large-scale AI training and inference has further increased attention to their energy and carbon footprint. The largest cloud providers report rising emissions even as they pledge carbon neutrality (Google, 2025; Microsoft, 2024). Renewable energy is a major lever for reducing operational emissions and its output is intermittent, so deferrable workloads provide an important opportunity for carbon reduction when they can be moved towards generation that has not yet arrived. Scheduling such work is a decision about future supply and therefore a decision taken on a forecast. Schedulers delay deferrable batch workloads into predicted green windows (Wiesner et al., 2021) or route them towards datacentres about to receive green supply (Xu et al., 2018; Zhang et al., 2023). Deep reinforcement learning (DRL) is widely used to learn these policies (Mao et al., 2016; Zhang et al., 2022), and carbon-aware benchmarks write the forecast straight into their reference agents' observations (Naug et al., 2024; Venkataswamy et al., 2022). Forecast features are therefore a standing part of the observation, and deployment carbon depends both on their accuracy and on how the policy has learned to use them.

Forecast quality is not stable over the lifetime of a deployed system. It degrades silently as sensors malfunction, models go stale and data pipelines fail (Polasek & Čadík, 2023; Meenal et al., 2022), and forecasting pipelines can be actively attacked (Chen et al., 2019; Tang et al., 2021). Such failures can leave a numerically plausible but less reliable forecast, weakening its relationship to realised supply. As illustrated in Figure 1, measured against an uninformative forecast, flat at each site's long-run mean, a perfect forecast saves 10.8% of carbon and one whose future is shuffled

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.

in time costs 2.8% more. Relative to the perfect-forecast condition, time shuffling raises pooled carbon by 15.2%, so temporal corruption reverses the benefit of the informative forecast. Standard PPO updates the policy from an advantage reflecting both action quality and the forecast-conditioned trajectory, without separating forecast-related from action-related variation (Dietterich et al., 2018), and it contains no mechanism for reducing policy credit when forecast reliability degrades.

Existing approaches to robustness differ in where they intervene. Risk-sensitive objectives change how outcomes are valued (Tamar et al., 2015; Di Castro et al., 2012) and domain randomisation changes the distribution the policy trains on (Tobin et al., 2017), both acting before any input is implicated. Uncertainty-aware methods make caution conditional, but they estimate uncertainty over the overall state (Koren et al., 2025) rather than isolating the reliability of the forecast features. Counterfactual credit assignment intervenes on the learning signal itself, separating what an action influenced from what it did not (Mesnard et al., 2020; Foerster et al., 2018), while conditioning on the exogenous signal as given, so a corrupted forecast enters the credit it assigns unchanged. The cited formulations do not construct an explicit per-transition forecast-specific signal for reweighting policy credit, which is the intervention studied here.

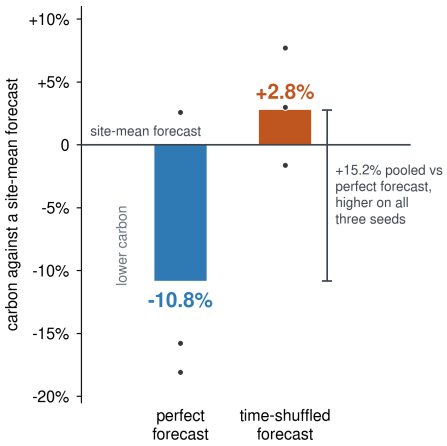

Figure 1: The forecast as asset and liability. One conventionally trained policy, three forecasts, carbon measured against the same policy given an uninformative forecast, flat at each site's long-run mean. A perfect forecast saves 10.8%. A forecast whose future is shuffled in time within each site raises pooled carbon 15.2% above the perfect one, with an increase on each seed, and pooled it lands above the uninformative one. Dots are seeds.

Does such a weight lower deployment carbon under a degraded forecast, and does it do so beyond what the shared architecture and the auxiliary critic training already deliver? This work proposes EU-CRD for a hierarchical scheduler whose global agent routes or defers each arriving job and whose local agent paces each datacentre's queue, so the forecast enters only through the global agent. During training EU-CRD prices, for each transition, the reference candidate the forecast selects against the reference candidate hindsight selects, both under the realised supply, and pairs that hindsight regret with a value gap between the action taken and a counterfactual action drawn from the policy itself. Critic-ensemble disagreement modulates the value gap before the two proxies are combined, and their relative magnitudes become weights on the routing advantage. All additional responsibility machinery is confined to training, and at inference the policy uses the same backbone as a conventionally trained one. The contributions of this work are summarised as follows:

- **Forecast-aware credit formulation.** This work formulates forecast-based scheduling as a credit-assignment problem in which policy returns reflect the quality of both routing and exogenous forecast, and constructs separate counterfactual proxies for forecast-related and routing-related credit rather than treating all observation channels uniformly.
- **Responsibility-guided policy optimisation.** EU-CRD combines a hindsight forecast regret, a policy-action counterfactual and critic-ensemble disagreement into a training-only weight on the routing advantage, requiring no responsibility computation at deployment.
- **Matched-control evaluation with cross-year confirmation.** The incremental effect of the weighting rule is isolated by paired continuations against a matched control that keeps the same backbone, critic ensemble, responsibility computation, auxiliary training and budget. Under deterministic decoding on the development windows, EU-CRD lowers pooled carbon by 14.5% under attenuation and 10.1% under perfect forecasts. Without further training, the same frozen checkpoints meet the preregistered cross-year confirmation criterion, with reductions of 12.2% and 10.5%, respectively. Alternative objectives and component removals provide additional development evidence (Section 4).

2 Related Work

Carbon-aware and renewable-driven scheduling. Carbon-aware scheduling matches demand to low-carbon supply, spatially by routing to cleaner sites and temporally by deferring tolerant jobs to greener windows. One line is heuristic or combinatorial. Wiesner et al. (2021) shift load by a forecast-driven threshold-deferral rule, Schweisgut et al. (2025) solve carbon-aware workflow scheduling as a combinatorial optimisation under a deadline constraint, and Yu et al. (2025) feed conformal prediction intervals on renewable supply and load into a risk-aware task-deferral optimiser. Recent work learns these decisions with reinforcement learning (RL). Jayanetti et al. (2024) use a multi-agent actor-critic across datacentres, Alqithami (2025) adds a primal-dual emission-budget layer over PPO, and Gao et al. (2025) schedule HPC jobs with a gated-transformer PPO close to the backbone used here. These systems differ in how well they exploit a forecast rather than in how credit for a decision depends on it. Where forecast error is modelled, it enters as an interval or distribution over the prediction rather than as a per-transition responsibility.

Credit assignment and exogenous-signal decomposition. A counterfactual line marginalises out a single agent action via a centralised critic (Foerster et al., 2018), strips exogenous variance with future-conditional hindsight baselines (Mesnard et al., 2020; Harutyunyan et al., 2019), or assigns Shapley-value credit (Li et al., 2021). Closest to this work, Dietterich et al. (2018) remove the exogenous state variables and rewards an action cannot influence and Efroni et al. (2022) formalise the exogenous-information MDP, which takes the signal out of the credit together with whatever value it carried. Decision-focused learning instead trains the predictor under the downstream decision loss (Elmachtoub & Grigas, 2022; Wilder et al., 2019). EU-CRD keeps the predictor frozen and changes how the policy assigns credit to its output, using the forecast-regret proxy of Section 3 to construct routing-advantage weights modulated by ensemble disagreement.

Risk-sensitive, uncertainty-aware and input-robust RL. The risk-sensitive family reshapes the objective, optimising conditional value-at-risk (Tamar et al., 2015) or penalising return variance (Di Castro et al., 2012), with distributional and utility-based variants (Dabney et al., 2018), and domain randomisation (Tobin et al., 2017) perturbs training inputs instead. These formulations do not construct an explicit per-transition forecast-specific signal for reweighting policy credit. A second family estimates confidence over the state, reading epistemic uncertainty from an ensemble to stay conservative on unfamiliar states (An et al., 2023) or running risk-constrained RL on a probabilistic forecast (Koren et al., 2025), but these methods estimate uncertainty over the overall state rather than isolating the reliability of the forecast features. A third family defends the observation itself through smoothness regularisation (Zhang et al., 2020), a learned adversary (Zhang et al., 2021), or denoising (Yang & Xu, 2024), acting on the observation as a whole rather than on the credit assigned through one channel.

3 Methodology

3.1 Problem Formulation

An online stream of deferrable jobs, each with a compute size and a deadline d_j, is scheduled across N geo-distributed datacentres. A DC i may draw on on-site wind supplying green power $g_i(t) \geq 0$, zero at sites without it, with any shortfall met by the grid. Since $\kappa_i^{\text{green}} \ll \kappa_i^{\text{brown}}$, emissions come almost entirely from that shortfall.

Control is hierarchical and learned with RL. At a fixed control interval a global router π_G routes or defers the batch of jobs that arrived since the last interval, conditioned on per-DC state and a green-power forecast $\hat{\mathbf{y}}_i = (\hat{y}_i^s, \hat{y}_i^l)$ over a short and a long horizon. A local scheduler π_L then paces each DC's queue from its own partial view (Section 3.2). The joint policy $\pi_\theta = (\pi_G, \pi_L)$ is trained with PPO to maximise $J(\theta) = \mathbb{E}_{\pi_\theta}[\sum_t \gamma^t r_t]$, serving the operational goal of minimising carbon subject to an SLA on timely completion,

$$\min_\theta \ \mathbb{E}\left[\sum_{t,i}\Big(\kappa_i^{\text{brown}} b_i(t) + \kappa_i^{\text{green}}\big(u_i(t) - b_i(t)\big)\Big)\,\Delta t\right] \quad \text{s.t.} \quad \frac{1}{|\mathcal{J}|}\sum_j \mathbb{1}[f_j \leq d_j] \geq \tau, \qquad (1)$$

where $u_i(t)$ is the total power demand at DC i, $b_i(t) = \max\{0, u_i(t) - g_i(t)\}$ the brown power drawn once on-site green is exhausted, κ_i^{green} the small embodied intensity of on-site generation (values in Section 4), $\mathcal{J}$ the job set, f_j the completion time of job j, and τ the required completion rate, with Δt the fixed length of a control step. The reward scalarises this objective as a per-step carbon penalty and a credit for completed compute (Appendix K). Deadline discipline is handled separately. A mask removes candidates under which a job could no longer finish on time, and an environment-level backstop forces execution once its slack is exhausted. The aggregate rate τ is not enforced during optimisation, so completion and on-time compute share are reported alongside carbon in Section 4.

The difficulty this work targets is that $g_i(t)$ is exogenous and the router acts on its forecast rather than its realisation. The observed return therefore reflects both routing quality and forecast quality (Figure 2), while the standard policy-gradient update contains no explicit signal separating them. EU-CRD assigns lower routing-update weight to transitions with low routing responsibility, and the effect on carbon is measured rather than assumed.

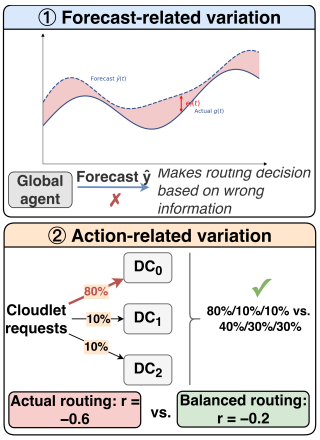

Figure 2: Standard PPO updates from an advantage that reflects both forecast-related variation and action-related variation, without explicitly separating their contributions.

This work proposes EU-CRD, illustrated in Figure 3. During training, EU-CRD constructs forecast-related and routing-related counterfactual proxies for each forecast-exposed transition and converts their relative magnitudes into a weight on the routing advantage.

3.2 Hierarchical MDP Design

The global router acts once per control interval on the waiting batch, observing per-DC green power, queue length and utilisation, the forecast features $\hat{\mathbf{y}}_i$ (short-horizon mean and trend, long-horizon mean and time to peak), per-job size and deadline, and, for each candidate placement, the green cover that the forecast promises the job over a 121-step horizon. Its action assigns up to B jobs in the batch a candidate (d, δ), a datacentre d and a wait offset $\delta \in \{0, \ldots, W\}$ in control steps, treated as B per-job decisions rather than one joint distribution (B=128 and W=72 in Section 4, B exceeding any batch the workload produces). A job with δ>0 is released at $t+\delta$ by an executor, so deferral is a chosen delay. A deadline mask shared by all methods removes candidates under which the job could not finish on time. Each DC's local scheduler shares parameters across sites, observes its own hosts, VMs and queue, and releases a count $k \in \{0, \ldots, K_L\}$ per step. The local agent never observes the forecast. Full observation and action specifications are in Appendix A.

The two reward channels mirror this division. The global reward r_t^G carries the carbon objective as the total carbon of the step normalised by a fixed constant, a credit proportional to the increase in the share of compute finished, and a per-candidate credit that decays with the capacity overflow a placement would cause. The implementation also supports marginal-carbon, green-waste, throughput, deferral-cost and urgency terms, and every one of them is set to zero in every arm of this paper. The full expression with the coefficients in force is in Appendix K. The local reward r_t^L is a forecast-independent quality-of-service signal. The carbon objective is carried by the global reward and forecast features are observed only by the global router.

3.3 Counterfactual Responsibility Decomposition

EU-CRD constructs two heuristic responsibility proxies for forecast-exposed transitions, one associated with forecast quality and one with routing. The local scheduler does not observe the forecast and trains with PPO. These proxies are converted into relative responsibility scores once the routing proxy is formed in Section 3.4 (Fig. 3).

For the forecaster, the reference is the realisation. Every job in the batch has a legal set of candidates, a site paired with a wait offset, and the forecast reaches the router as the green cover each candidate would receive. With the batch, the legal set and the cost model held fixed, a reference candidate

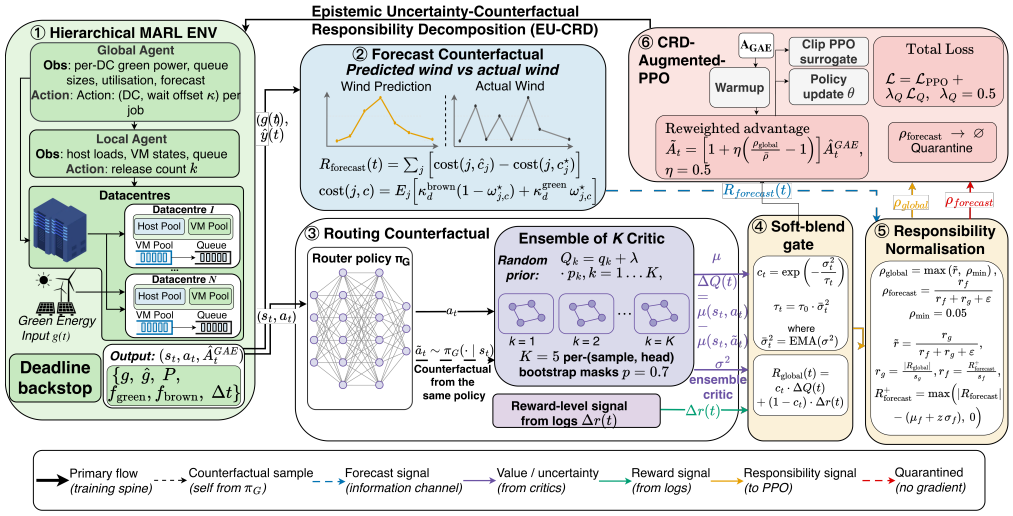

Figure 3: EU-CRD training loop. Counterfactual forecast and routing proxies form responsibility weights that rescale the routing advantage, with ensemble disagreement σ^2 modulating the routing signal. Responsibility computation is used only during training.

is selected twice for each job, once on the cover the forecast promises and once on the cover the hidden future would have given, and both are priced at their true cost,

$$\mathrm{cost}(j,c) = E_j\left[\kappa_d^{\mathrm{brown}}(1-\omega^\star_{j,c}) + \kappa_d^{\mathrm{green}}\,\omega^\star_{j,c}\right], \qquad R_{\mathrm{forecast}}(t) = \sum_j \left[\mathrm{cost}(j,\hat{c}_j) - \mathrm{cost}(j,c^\star_j)\right], \tag{2}$$

with E_j the energy of the job, $\omega^\star_{j,c}$ the cover candidate c truly receives, and $\hat{c}_j$, $c^\star_j$ the candidates selected on the forecast and on the truth. These are chosen by cover alone and need not coincide with the action the learned router took, so R_{forecast} is a local, model-based regret proxy in kilograms of CO_2, not the realised action's causal loss and not a proven bound on it.

For the router, the reference action $\tilde{a}_t$ is drawn from its own policy at the same state, an on-policy counterfactual reference, in the spirit of COMA-style credit assignment, that holds the information fixed and varies only the action, giving

$$\Delta Q^\pi(t) = Q^\pi(s_t, a_t) - Q^\pi(s_t, \tilde{a}_t). \tag{3}$$

Q^π has no closed form and is estimated from learned value functions, which the next subsection estimates and modulates by uncertainty.

3.4 Epistemic Uncertainty and Soft Blending

ΔQ^π of Eq. 3 needs Q^π, and the estimate is unreliable early in training, on rarely visited states, and on the states a corrupted forecast drives the agent into. An ensemble of K_Q critics trained on bootstrapped targets with randomised priors (Osband et al., 2016; 2018) supplies the value gap $\Delta Q(t) = \mu(s_t, a_t) - \mu(s_t, \tilde{a}_t)$ from the member mean μ, and the disagreement σ_t^2 as the sum of the member variances at the two actions. The summed form upper-bounds the variance of the difference under positive correlation, so an overestimate pushes the gate towards the fallback rather than towards the learned estimate.

ΔQ is paired with a reward-level signal computed analytically from the decision-time state. Each routed job is scored by the green fit of the datacentre it is sent to and the reference action is scored the same way,

$$\Delta r(t) = \alpha\ \mathrm{mean}_j\left[h_t(a_{t,j}) - h_t(\tilde{a}_{t,j})\right], \qquad h_t(\textsc{defer}) = 0, \tag{4}$$

with $h_t(i)$ the current green-power ratio at datacentre i that the router already observes, distinct from the green power $g_i(t)$ of Section 3.1. A soft blend gives the routing-responsibility proxy,

$$R_{\mathrm{global}}(t) = c_t\,\Delta Q(t) + (1-c_t)\,\Delta r(t), \qquad c_t = \exp\!\big(-\sigma_t^2/\tau_t\big), \qquad \tau_t = \tau_0\,\bar{\sigma}_t^2, \tag{5}$$

where $\bar{\sigma}_t^2$ is a running mean of the disagreement, so τ_0 calibrates it throughout training. The gate acts on the value-based signal ΔQ before it is combined with Δr, and not on the responsibility scores formed below. The ablation of Section 4 fixes $c_t \equiv 1$, which removes the conditional blend and the model-free term together and so measures the two jointly.

The two terms of Eq. 5 are combined in their own units. The scaling below is applied afterwards, to $|R_{\text{global}}|$ as a whole, so Δr is not placed on the scale of ΔQ before the blend and is not an equally scaled substitute for it. Reweighting begins only after the warmup specified in Appendix B. Over the iterations in which the weights are applied, the logged dispersion of Δr stays below 1% of that of the gated value gap on the arms whose gate is free to move, where c_t averages near 0.55. The $c_t \equiv 1$ ablation is excluded because its gate is fixed. That is an aggregate reading of logged moments rather than a per-row accounting, and no experiment here isolates the contribution of the model-free term.

With R_{global} formed, the two proxies are converted into relative responsibility scores in three steps. An anomaly gate keeps only the excess of the forecast signal over its own recent behaviour, $R^{+}_{\text{forecast}} = \max(|R_{\text{forecast}}| - (\mu_f + z\sigma_f), 0)$ with $z = 1$. Here μ_f and σ_f are exponential moving averages, with decay 0.99, of the batch mean and the batch standard deviation of $|R_{\text{forecast}}|$ over the valid rows, updated once per call. Because gating precedes scale normalisation, transient increases in forecast regret are assessed against the raw forecast-regret statistics rather than being immediately suppressed by an enlarged scale. Both the anomaly threshold and the scale remain adaptive, so sufficiently persistent shifts can eventually be absorbed (Appendix D). The intent is to leave ordinary residual error uncharged, which the construction does not guarantee. Each signal is then divided by a running estimate of its own typical magnitude, $x_f = R^{+}_{\text{forecast}}/s_f$ and $x_r = |R_{\text{global}}|/s_r$, because the two are incommensurate. Each s_k is an exponential moving average, with the same decay 0.99, of the batch mean of that signal's own magnitude over the valid rows, and a channel whose average is still numerically zero is passed through unscaled rather than divided by an epsilon. The shares are

$$\rho_{\text{forecast}} = \frac{x_f}{x_f + x_r + \varepsilon}, \qquad \rho_{\text{global}} = \max\left(\frac{x_r}{x_f + x_r + \varepsilon},\ \rho_{\min}\right), \tag{6}$$

with $\varepsilon{=}10^{-8}$ and a floor on the routing share only, so the two do not sum exactly to one. Only ρ_{global} enters the update, renormalised by its mean over the valid rows of the batch. Both signals are heuristic proxies and the mechanism is judged by its effect on carbon in Section 4. Appendix D records the scaling, the horizon mismatch between the two signals and the sense in which the gate is relative rather than absolute.

3.5 CRD-Augmented Policy Optimisation

The responsibility weights act on learning through the advantage. The router optimises the clipped PPO surrogate on a reweighted advantage,

$$\tilde{A}_t = \left[1 + \eta\Big(\frac{\rho_{\text{global}}(t)}{\bar{\rho}} - 1\Big)\right]\hat{A}_t, \qquad \eta = 0.5, \tag{7}$$

where $\hat{A}_t$ is the GAE advantage, ρ_{global} is the routing share from Eq. 6, $\bar{\rho}$ is its batch mean, and η is a guard that halves each weight's departure from one while keeping the batch mean at one and the ordering of the weights. The local scheduler carries no responsibility term and is optimised by standard PPO. The weights are detached, but state- and action-dependent reweighting still changes the optimisation target. Unit-mean normalisation preserves neither the advantage mean, the gradient norm nor the optimiser step.

The ensemble critic producing ΔQ and σ^2 trains jointly, its K_Q heads following a bootstrapped temporal-difference loss added to the surrogate as $\mathcal{L} = \mathcal{L}_{\text{PPO}} + \lambda_Q \mathcal{L}_Q$. The separate value head trains on the ordinary return without reweighting, so its bootstrap targets are those of standard PPO.

The implementation uses detached responsibility weights, a fixed warmup, a routing-share floor and the guard in Eq. 7. Reweighting is disabled during warmup, and the floor retains a positive routing share. Settings are in Appendix B.

The weights produced by the rule are concentrated rather than diffuse. On the arms of Section 4.2, the tenth and ninetieth percentiles of ρ_{global} are both exactly one on every logged iteration, while its mean ranges from 0.980 to 0.988 and its standard deviation from 0.10 to 0.14. A share of one is

Table 1: Carbon (kg CO_2) pooled over six windows and three seeds for the 2021 development and 2020 preregistered confirmation sets. Deterministic decoding. All continued-arm cells satisfy service requirements. Percentages use unrounded totals.

Arm	Perfect forecast	Attenuated	Increase from perfect (%)
2021 development windows			
Base policy, no continuation	0.0757	0.0792	+4.68
Matched control	0.0766	0.0813	+6.13
EU-CRD	**0.0689**	**0.0695**	**+0.90**
EU-CRD vs control (%)	−10.11	−14.53	—
Seeds favouring EU-CRD	3/3	3/3	3/3
2020 confirmation windows			
Matched control	0.1006	0.1063	+5.62
EU-CRD	**0.0900**	**0.0933**	**+3.67**
EU-CRD vs control (%)	−10.54	−12.19	—
Seeds favouring EU-CRD	3/3	3/3	—

returned by Eq. 6 when the anomaly gate passes nothing. The fraction of valid rows on which the forecast channel claims any share ranges from 2.1% to 3.5% across the three seeds. The weighting is therefore selective. Most transitions receive near-unit weights. The normalised share $\rho_{\text{global}}/\bar{\rho}$ has a logged maximum of 1.012 to 1.020, while some gated rows reach the floor $\rho_{\min} = 0.05$. At the logged $\bar{\rho}$, such a row receives an effective advantage weight of approximately 0.53. These are properties of the weights, not evidence that the selected rows are those the forecast actually misled.

Algorithm 1 in Appendix C summarises the complete EU-CRD training procedure.

4 Experiments

4.1 Experiment Setup

The testbed is a five-datacentre CloudSim Plus environment (Silva Filho et al., 2017), with three sites using measured SDWPF wind traces (Zhou et al., 2024) and two grid-only sites. The workload contains 35 wide deferrable jobs over a 7 200-second episode. Only the global router observes the forecast, and both policies use PPO (Schulman et al., 2017) with a GTrXL backbone (Parisotto et al., 2020) (Appendix A).

All arms continue from the same three base checkpoints for 56 000 steps. The primary matched control computes the same responsibility scores as EU-CRD but disables the final reweighting. Component ablations change one registered key, while alternative objectives replace the routing objective (Appendix B). Development evaluation uses deterministic decoding on six 2021 windows. The matched pair is also evaluated on six preregistered 2020 confirmation windows using the same frozen checkpoints. All reported cells satisfy the service requirements.

Two forecast tiers are used, perfect and attenuated. The perfect tier supplies the realised future green power. The attenuated tier applies $m + \lambda(y - m)$ with $\lambda = 0.75$ and site mean m, preserving timing and shape while reducing amplitude. Base checkpoints are trained under the perfect forecast, all continuation arms under the attenuated forecast, and all continued arms are evaluated at both tiers.

Figure 1 applies two further conditions to these base checkpoints. Retained amplitude zero replaces the future view with each site's long-run mean, forming the uninformative reference. The other shuffles each site's future profile in time at full amplitude. Holding the policy fixed isolates the value of forecast information from that of different training weights (Appendix H).

4.2 Isolating the Reweighting

The primary comparison isolates the effect of applying the responsibility weights while retaining the same backbone, critic ensemble, responsibility computation and auxiliary training. On the 2021 development windows, EU-CRD lowers pooled carbon relative to the matched control by 14.53%

under attenuation and 10.11% under perfect forecasts, with lower carbon on every paired seed at both tiers (Table 1).

The same frozen checkpoints meet the preregistered confirmation criterion on six 2020 windows without further training or checkpoint selection. Relative to the matched control, pooled carbon falls by 12.19% under attenuation and 10.54% under perfect forecasts. The attenuated reductions are 0.19%, 22.53% and 11.18% across seeds, so the direction is consistent but the magnitude varies substantially. All 72 confirmation cells satisfy the service requirements. This confirmation covers the matched reweighting comparison, not the alternative objectives or ablations.

On the development windows, attenuation raises the control's pooled carbon by 6.13% and EU-CRD's by 0.90%, with a smaller relative increase for EU-CRD on each paired seed. Each percentage uses that arm's own perfect forecast carbon as denominator, giving a gap of 5.23 percentage points. The corresponding confirmation increases are 5.62% and 3.67%, respectively, although EU-CRD does not have the smaller increase on every seed. Both arms continue training under attenuation from checkpoints trained under perfect forecasts. Reweighting lowers carbon at both tiers without an observed clean performance penalty, but these comparisons do not isolate how much of the benefit is specific to forecast degradation.

The base policy in Table 1 provides an unmatched development reference. EU-CRD lowers carbon emission from this starting point by 8.99% under perfect forecasts and 12.27% under attenuation, whereas the control raises it by 1.24% and 2.65%. This contrasts $56\,000$ additional training steps with none. The reweighting effect is therefore measured against the matched control at equal budget.

The pooled development reduction retains its sign when any one window is withheld, but improvements occur in $11/18$ cells under perfect forecasts and $12/18$ under attenuation rather than uniformly across windows (Appendix F). Agreement across three seeds is not a significance claim. The 36 cells per arm in each year form a seed-window-tier grid rather than independent repetitions. Table 1 uses the matched control as the percentage denominator, while Table 2 uses EU-CRD.

Table 2: Development carbon (kg CO_2) pooled over six 2021 windows and three seeds under deterministic decoding. Percentages are relative to EU-CRD, with positive values indicating higher carbon. ✓ denotes agreement across all three seeds. All 180 alternative-objective cells satisfy the service requirements.

	Perfect forecast		Attenuated forecast	
Training objective	kg CO_2	vs EU-CRD	kg CO_2	vs EU-CRD
EU-CRD	**0.0689**	—	**0.0695**	—
Matched control	0.0766	$+11.2^{\checkmark}$	0.0813	$+17.0^{\checkmark}$
RCV, CVaR-style tail upweighting	0.0709	$+3.0$	0.0774	$+11.4^{\checkmark}$
RDC, distributional-CVaR-style	0.0766	$+11.2^{\checkmark}$	0.0791	$+13.9^{\checkmark}$
RMV, mean-variance-style	0.0732	$+6.3$	0.0782	$+12.5^{\checkmark}$
RRS, exponential-utility-style	0.0735	$+6.8$	0.0805	$+15.8^{\checkmark}$
CCA, hindsight-baseline-style	0.0765	$+11.1$	0.0797	$+14.7^{\checkmark}$

4.3 Comparison with Alternative Objectives

Five alternative objectives were trained from the same three start states under the same budget, protocol and module scope. Four risk-inspired alternatives differ from the matched-control recipe in the routing-objective weighting, in the styles of CVaR, distributional CVaR, mean-variance and exponential utility. The fifth, a hindsight-baseline-style variant, additionally introduces an auxiliary model, its own optimiser and a training-time future-information input (Appendix B). These are local implementations sharing this backbone. The results compare objective variants within that shared backbone and should not be read as evaluations of the original published algorithms.

All continued-training arms, including the matched control, are trained under the same fixed attenuated forecast. The matched control in Table 1 is therefore a policy already trained on degraded information, which computes the same responsibility scores with the final reweighting disabled. That

is exposure to one fixed corruption rather than randomisation over a family of them. Training-time exposure is therefore tested, but domain randomisation is not.

Under attenuation all five alternatives have higher carbon than EU-CRD on every seed, the pooled excess ranging from 11.4 to 15.8% and the per-seed median from 10.7 to 20.0%. Under the perfect forecast only the distributional-CVaR-style line is above EU-CRD on every seed, the other four disagreeing in sign. This work does not separate why the tiers differ. The mean-variance-style margin on one seed is 0.04%, which is a direction rather than a substantive advantage.

4.4 Component Ablation

Three further arms each remove one named part of the mechanism, continuing from the same start states under the same budget and protocol with every other part in place. Fixing the blend coefficient at one, switching off the forecast-anomaly handling, and switching off the per-channel scale standardisation each raise carbon under attenuation on all three seeds, with median increases of 5.83%, 8.92% and 13.68%. Under the perfect forecast only the scale standardisation shows a consistent direction, the other two disagreeing in sign across seeds.

These are conditional effects with the other components retained. Single-component removal separates neither additive contributions nor interactions, so the table orders no component by importance. Appendix E gives the per-arm and per-seed values.

4.5 Cost and Boundaries

Responsibility weights are applied only during training, leaving the deployed policy backbone unchanged. Across the 360 development cells of the ten arms, routing latency averages 1.30 to 1.33 ms, with a 99th percentile of 2.04 to 2.10 ms. The queue-pacing decision takes $3.5\ \mu$s, and peak resident memory is 1.35 GB. The spread of per-arm medians is below 5% on every statistic. This is measured closeness rather than evidence of equal cost, and confirmation cells are not included in the overhead summary (Appendix J).

No consistent benefit is established under sampled decoding, where the effect changes sign with the decoding seed (Appendix I). Deterministic decoding is the deployment form chosen here. The preregistered cross-year confirmation supports the matched reweighting comparison, while the alternative objectives and component ablations remain development evidence. Confirmation uses the same trained checkpoints, sites and workload on wind traces from another year, so it does not establish generalisation to new workload distributions or forecast-error processes. The study uses one fixed workload trace and one primary synthetic attenuation operator, and the perfect forecast tier supplies realised future generation rather than predictions from a deployed forecaster. The matched control isolates the incremental effect of applying the responsibility weights rather than the benefit of the complete system over conventional training. Neither the development results nor their confirmation establish the accuracy of the shares as causal responsibility estimates.

5 Conclusion

A forecast-conditioned scheduler is trained on returns reflecting both the quality of a decision and the quality of the information behind it, and standard PPO provides no signal separating the two. EU-CRD introduces heuristic proxies for forecast-related and routing-related credit, modulates the value-based proxy by critic-ensemble disagreement, and reweights the routing advantage using factors normalised to unit batch mean. Under deterministic decoding, reweighting lowers pooled carbon against the matched control by 14.5% under attenuation and 10.1% under perfect forecasts on the development windows. The same frozen checkpoints meet the preregistered cross-year confirmation criterion without further training, with reductions of 12.2% and 10.5%, respectively. These results support the incremental benefit of the weighting rule under the studied continuation recipe. Generalisation to other workloads and forecast-error processes remains open.

AI USE STATEMENT

Generative AI tools were used to aid and polish the writing of the manuscript, assist with literature retrieval and discovery, and support research ideation and execution, including debugging experimental code. All AI-assisted work was reviewed and verified by the authors. Relevant literature and citations were independently checked, AI-assisted code was tested by the authors, and all reported results were produced from the authors' own experimental runs. The authors take full responsibility for the final text, claims, code, references, and results presented in this work.

REPRODUCIBILITY STATEMENT

All experiments are driven by generated configuration blocks that fix the testbed (Appendix A), the training recipe (Appendix B) and the forecast tier, and each reported cell names its configuration key, seed, window and checkpoint. The control and the three ablation arms are derived from the full arm's block by changing exactly one key each, asserted on the effective configuration before training starts. The five alternative objectives share the recipe of the control. Four risk-inspired alternatives modify the routing-objective weighting, while the CCA variant additionally introduces an auxiliary model and optimiser. The workload trace and the SDWPF turbine traces are fixed artefacts shared by every arm, tier, seed and window. The code version is part of the record rather than an incidental detail, because a checkpoint restores only under a learner whose state contract matches the one that wrote it, and the training and evaluation revisions are recorded separately by hash. The main deterministic result tables comprise 360 development cells and 72 confirmation cells, with service requirements checked mechanically on every cell. The confirmation uses the same six frozen final checkpoints without further training or checkpoint selection. The code, configuration blocks, workload trace, 360 development cells, 72 confirmation cells and figure scripts are provided as supplementary material, and the same release is mirrored anonymously at `https://anonymous.4open.science/r/eucrd-F7CF`. A script there recomputes the reported experimental aggregates and percentages from the archived cells without rerunning a simulation. The wind series are not redistributed, since SDWPF carries its own licence, and a script builds them from the official release.

REFERENCES

Saad Alqithami. EcoFair-CH-MARL: Scalable Constrained Hierarchical Multi-Agent RL with Real-Time Emission Budgets and Fairness Guarantees. In Inês Lynce, Nello Murano, Mauro Vallati, Serena Villata, Federico Chesani, Michela Milano, Andrea Omicini, and Mehdi Dastani (eds.), *Frontiers in Artificial Intelligence and Applications*. IOS Press, October 2025. ISBN 978-1-64368-631-8. doi: 10.3233/FAIA251248. URL https://ebooks.iospress.nl/doi/10.3233/FAIA251248.

Zhiyu An, Xianzhong Ding, Arya Rathee, and Wan Du. CLUE: Safe Model-Based RL HVAC Control Using Epistemic Uncertainty Estimation. In *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 149–158, Istanbul Turkey, November 2023. ACM. ISBN 979-8-4007-0230-3. doi: 10.1145/3600100.3623742. URL https://dl.acm.org/doi/10.1145/3600100.3623742.

Yize Chen, Yushi Tan, and Baosen Zhang. Exploiting Vulnerabilities of Load Forecasting Through Adversarial Attacks. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, pp. 1–11, Phoenix AZ USA, June 2019. ACM. ISBN 978-1-4503-6671-7. doi: 10.1145/3307772.3328314. URL https://dl.acm.org/doi/10.1145/3307772.3328314.

Will Dabney, Georg Ostrovski, David Silver, and Remi Munos. Implicit quantile networks for distributional reinforcement learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th international conference on machine learning*, volume 80 of *Proceedings of machine learning research*, pp. 1096–1105. PMLR, July 2018. URL https://proceedings.mlr.press/v80/dabney18a.html.

Dotan Di Castro, Aviv Tamar, and Shie Mannor. Policy gradients with variance related risk criteria. *arXiv preprint arXiv:1206.6404*, 2012.

Thomas Dietterich, George Trimponias, and Zhitang Chen. Discovering and removing exogenous state variables and rewards for reinforcement learning. In *International conference on machine learning*, pp. 1262–1270. PMLR, 2018.

Yonathan Efroni, Dylan J Foster, Dipendra Misra, Akshay Krishnamurthy, and John Langford. Sample-efficient reinforcement learning in the presence of exogenous information. In *Conference on learning theory*, pp. 5062–5127. PMLR, 2022.

Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management science*, 68(1): 9–26, 2022.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual Multi-Agent Policy Gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v32i1.11794. URL https://ojs.aaai.org/index.php/AAAI/article/view/11794.

Xu Gao, Hang Dong, Lianji Zhang, Yibo Wang, Xianliang Yang, and Zhenyu Li. Self-Attention Mechanisms in HPC Job Scheduling: A Novel Framework Combining Gated Transformers and Enhanced PPO. *Applied Sciences*, 15(16):8928, August 2025. ISSN 2076-3417. doi: 10.3390/app15168928. URL https://www.mdpi.com/2076-3417/15/16/8928.

Google. 2025 Environmental Report, 2025. URL https://sustainability.google/google-2025-environmental-report/.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Azar, Bilal Piot, Nicolas Heess, Hado van Hasselt, Greg Wayne, Satinder Singh, Doina Precup, and Remi Munos. Hindsight Credit Assignment, December 2019. URL http://arxiv.org/abs/1912.02503. arXiv:1912.02503 [cs].

Amanda Jayanetti, Saman Halgamuge, and Rajkumar Buyya. Multi-Agent Deep Reinforcement Learning Framework for Renewable Energy-Aware Workflow Scheduling on Distributed Cloud Data Centers. *IEEE Transactions on Parallel and Distributed Systems*, 35(4):604–615, April 2024. ISSN 1045-9219, 1558-2183, 2161-9883. doi: 10.1109/TPDS.2024.3360448. URL https://ieeexplore.ieee.org/document/10417832/.

Nicola Jones. How to stop data centres from gobbling up the world's electricity. *nature*, 561(7722): 163–166, 2018.

Michal Koren, Or Peretz, Tai Dinh, and Philip S. Yu. UAMDP: Uncertainty-Aware Markov Decision Process for Risk-Constrained Reinforcement Learning from Probabilistic Forecasts, December 2025. URL http://arxiv.org/abs/2510.08226. arXiv:2510.08226 [cs].

Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley Counterfactual Credits for Multi-Agent Reinforcement Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 934–942, Virtual Event Singapore, August 2021. ACM. ISBN 978-1-4503-8332-5. doi: 10.1145/3447548.3467420. URL https://dl.acm.org/doi/10.1145/3447548.3467420.

Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. Resource Management with Deep Reinforcement Learning. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, HotNets '16, pp. 50–56, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 978-1-4503-4661-0. doi: 10.1145/3005745.3005750. URL https://doi.org/10.1145/3005745.3005750.

R. Meenal, D. Binu, K. C. Ramya, Prawin Angel Michael, K. Vinoth Kumar, E. Rajasekaran, and B. Sangeetha. Weather Forecasting for Renewable Energy System: A Review. *Archives of Computational Methods in Engineering*, 29(5):2875–2891, August 2022. ISSN 1134-3060, 1886-1784. doi: 10.1007/s11831-021-09695-3. URL https://link.springer.com/10.1007/s11831-021-09695-3.

Thomas Mesnard, Theophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Tom Stepleton, Nicolas Heess, Arthur Guez, and others. Counterfactual credit assignment in model-free reinforcement learning. *arXiv preprint arXiv:2011.09464*, 2020.

Microsoft. 2024 Environmental Sustainability Report, 2024. URL https://www.microsoft.com/en-us/corporate-responsibility/sustainability/report/.

Avisek Naug, Antonio Guillen, Ricardo Luna, Vineet Gundecha, Cullen Bash, Sahand Ghorbanpour, Sajad Mousavi, Ashwin R Babu, Dejan Markovikj, Lekhapriya D Kashyap, and others. Sustaindc: Benchmarking for sustainable data center control. *Advances in Neural Information Processing Systems*, 37:100630–100669, 2024.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. *Advances in neural information processing systems*, 29, 2016.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, and others. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, pp. 7487–7498. PMLR, 2020.

Tomas Polasek and Martin Čadík. Predicting photovoltaic power production using high-uncertainty weather forecasts. *Applied Energy*, 339:120989, June 2023. ISSN 03062619. doi: 10.1016/j.apenergy.2023.120989. URL https://linkinghub.elsevier.com/retrieve/pii/S0306261923003537.

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

Dominik Schweisgut, Anne Benoit, Yves Robert, and Henning Meyerhenke. Carbon-Aware Workflow Scheduling with Fixed Mapping and Deadline Constraint. In *Proceedings of the 54th International Conference on Parallel Processing*, pp. 627–637, San Diego CA USA, September 2025. ACM. ISBN 979-8-4007-2074-1. doi: 10.1145/3754598.3754673. URL https://dl.acm.org/doi/10.1145/3754598.3754673.

Manoel C. Silva Filho, Raysa L. Oliveira, Claudio C. Monteiro, Pedro R.M. Inacio, and Mario M. Freire. CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 400–406, Lisbon, May 2017. IEEE. ISBN 978-3-901882-89-0. doi: 10.23919/INM.2017.7987304. URL https://ieeexplore.ieee.org/document/7987304/.

Aviv Tamar, Yonatan Glassner, and Shie Mannor. Optimizing the CVaR via sampling. In *Proceedings of the twenty-ninth AAAI conference on artificial intelligence*, AAAI'15, pp. 2993–2999, Austin, Texas, 2015. AAAI Press. ISBN 0-262-51129-0. Number of pages: 7.

Ningkai Tang, Shiwen Mao, and R. Mark Nelms. Adversarial Attacks to Solar Power Forecast. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Madrid, Spain, December 2021. IEEE. ISBN 978-1-7281-8104-2. doi: 10.1109/GLOBECOM46510.2021.9685910. URL https://ieeexplore.ieee.org/document/9685910/.

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 *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, Vancouver, BC, September 2017. IEEE. ISBN 978-1-5386-2682-5. doi: 10.1109/IROS.2017.8202133. URL http://ieeexplore.ieee.org/document/8202133/.

Vanamala Venkataswamy, Jake Grigsby, Andrew Grimshaw, and Yanjun Qi. RARE: Renewable Energy Aware Resource Management in Datacenters. In *Job Scheduling Strategies for Parallel Processing: 25th International Workshop, JSSPP 2022, Virtual Event, June 3, 2022, Revised Selected Papers*, pp. 108–130, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-22697-7. doi: 10.1007/978-3-031-22698-4-6. URL https://doi.org/10.1007/978-3-031-22698-4-6.

Philipp Wiesner, Ilja Behnke, Dominik Scheinert, Kordian Gontarska, and Lauritz Thamsen. Let's wait awhile: how temporal workload shifting can reduce carbon emissions in the cloud. In *Proceedings of the 22nd International Middleware Conference*, pp. 260–272, Québec city Canada, December 2021. ACM. ISBN 978-1-4503-8534-3. doi: 10.1145/3464298.3493399. URL https://dl.acm.org/doi/10.1145/3464298.3493399.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1658–1665, 2019. Number: 01.

Chenhan Xu, Kun Wang, Peng Li, Rui Xia, Song Guo, and Minyi Guo. Renewable energy-aware big data analytics in geo-distributed data centers with reinforcement learning. *IEEE Transactions on Network Science and Engineering*, 7(1):205–215, 2018.

Zhihe Yang and Yunjian Xu. DMBP: Diffusion model-based predictor for robust offline reinforcement learning against state observation perturbations. In *International conference on learning representations*, volume 2024, pp. 38549–38578, 2024.

Qingdi Yu, Min Hu, Zhen Yang, and Xin Zhou. Carbon-Efficient Data Center Scheduling under Uncertainty: A Conformal Prediction Approach. In *2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, pp. 232–236, Shanghai, China, March 2025. IEEE. ISBN 979-8-3315-3508-7. doi: 10.1109/ICAACE65325.2025.11019847. URL https://ieeexplore.ieee.org/document/11019847/.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in neural information processing systems*, 33:21024–21037, 2020.

Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. *arXiv preprint arXiv:2101.08452*, 2021.

Kuo Zhang, Peijian Wang, Ning Gu, and Thu D. Nguyen. GreenDRL: managing green datacenters using deep reinforcement learning. In *Proceedings of the 13th Symposium on Cloud Computing*, pp. 445–460, San Francisco California, November 2022. ACM. ISBN 978-1-4503-9414-7. doi: 10.1145/3542929.3563501. URL https://dl.acm.org/doi/10.1145/3542929.3563501.

Siyue Zhang, Minrui Xu, Wei Yang Bryan Lim, and Dusit Tao Niyato. Sustainable AIGC Workload Scheduling of Geo-Distributed Data Centers: A Multi-Agent Reinforcement Learning Approach. *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pp. 3500–3505, 2023. URL https://api.semanticscholar.org/CorpusID:258180402.

Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jian Tang, Jiantao Su, Yu Li, Ji Liu, Junfu Lyu, Yanjun Ma, and Dejing Dou. SDWPF: A Dataset for Spatial Dynamic Wind Power Forecasting over a Large Turbine Array. *Scientific Data*, 11(1):649, June 2024. ISSN 2052-4463. doi: 10.1038/s41597-024-03427-5. URL https://www.nature.com/articles/s41597-024-03427-5.

A Testbed Details

Datacentres. The testbed simulates five datacentres in CloudSim Plus. Three carry on-site wind and two draw only from the grid, and every site is charged the same brown intensity of 0.5 kg CO_2/kWh and the same embodied green intensity of 0.01 kg/kWh, so carbon differences come from when and where work runs rather than from a site-specific emission factor. Fleet sizes are 10, 8, 5, 4 and 3 hosts, giving an aggregate of 30. The two brown-only sites make spatial routing consequential, since work placed there is charged the grid intensity whatever its timing.

Green supply. On-site wind uses measured 15-minute SCADA traces from the SDWPF dataset. The three green sites draw on turbines #133 and #78, #22 and #81, and #94 respectively. Because the simulation compresses wall-clock time, turbine output is divided by a fixed constant of 3 000 so that aggregate green supply and aggregate demand are of comparable magnitude. The constant was

calibrated once, before any method comparison, and is identical across every method, tier, seed and window, so it was not adjusted per arm.

Workload. Each episode replays a fixed trace of 35 deferrable jobs over a 7 200-second horizon. Every job requests 32 cores and carries a per-job deadline. Jobs are deliberately wide and few rather than numerous and small, so that a single routing decision commits a substantial block of demand to one site and the choice of site is visible in the carbon account. Job splitting is disabled, so a job occupies the cores it asks for. A deadline backstop forces execution of any job whose slack is exhausted, so completion discipline belongs to the environment rather than to any method. Completion means finished within the episode, and on-time share is the fraction of compute finished before its deadline. Both are 1.0 on every cell reported in this paper.

Forecast channel and tiers. The router observes, per datacentre, current green power, a green-cover feature over a 121-step horizon, queue and utilisation state, and per-job size and remaining slack. The perfect tier supplies the realised green profile itself, a perfect forecast rather than a predictor. The attenuated tier replaces the future profile by $m + \lambda(y - m)$ with $\lambda = 0.75$ and m the long-run mean of the site, which keeps the shape and timing of the profile and removes a quarter of its amplitude. The present observation is never altered, because the current row is a measurement and not a prediction. The base checkpoints were trained under the perfect forecast, every continued-training arm trains under the attenuated one, and evaluation covers both. Attenuation acts on the forecast transport rather than on any forecaster's parameters.

Windows. A window is an offset into the year-long turbine series at which the episode starts, so different windows present different green-supply regimes to the same workload. Six 2021 windows are used for the development evaluation. Six preregistered 2020 windows were held out during development and opened only for the cross-year confirmation of the matched reweighting comparison reported in Appendix G.

Agents and actions. A global routing agent assigns each arriving job to a datacentre or defers it, scoring job and datacentre embeddings pairwise on a GTrXL backbone. Per-datacentre local agents pace their own queues and never observe the forecast. A job given a wait offset is handed to an executor that releases it at the chosen step, so the commitment to a site and to a start time is made at routing time.

B Training Details

Recipe. Every arm in this paper continues from one of three base checkpoints and adds 56 000 environment steps in seven iterations of 8 000, with batch size 8 000, a two-layer GTrXL backbone ($d_{\text{model}} = 96$, 4 heads, feed-forward width 192, memory length 16, dropout 0.1, separate critic trunk, learning rate 3×10^{-4}). The base checkpoints were trained from scratch on this testbed for 56 000 steps under the same recipe, so an arm's total training is 112 000 steps. The three seeds, 20260915, 20260916 and 20260917, are paired by start state and labelled A, B and C in the appendix tables. The router uses discount 0.999 and GAE $\lambda = 0.98$, the local scheduler discount 0.99 and $\lambda = 0.95$. The advantage estimator bootstraps at truncation boundaries, does not zero the value of the final real state and does not bootstrap past a true termination, and standardises over valid rows only.

Single-key discipline. The control and each ablation arm are generated from the full arm's block by changing exactly one key, and the difference is asserted on the effective configuration, after defaults are merged, before training starts. The alternative-objective lines share the recipe of the control and add their objective term, and the CCA line additionally adds an auxiliary model and its optimiser. Each job verifies its resolved start checkpoint, the transplanted weights, the optimiser state, the freshly initialised ensemble heads, the effective configuration and the start-state hashes shared with the other arms of that seed.

EU-CRD hyperparameters. The critic ensemble uses $K_Q = 5$ heads of width 128 with frozen randomised priors (prior scale 3.0) and per-sample bootstrap masks ($p = 0.7$). The ensemble loss is weighted $\lambda_Q = 0.5$ and trained every iteration. The gate temperature is scale-adaptive, $\tau_t = \tau_0 \bar{\sigma}^2$ with $\tau_0 = 1.5$ and EMA rate 0.05. Responsibility shares are scale-normalised by a running EMA of decay 0.99 and floored at $\rho_{\min} = 0.05$. Reweighting activates after a warmup of 450 learner calls, which covers the first four of the seven iterations, and the run logs of every reweighting arm record weights applied in iterations five to seven only. The counterfactual reference action is drawn from

the policy itself. A single setting produces every EU-CRD number in this paper and nothing was tuned per tier, seed or window.

Alternative objectives. The four risk-inspired lines are advantage-level surrogates sharing this backbone rather than implementations of CVaR policy gradient, distributional reinforcement learning, exponential-utility control or mean-variance policy gradient, and no result in this paper speaks to those published methods. The fifth line is a hindsight baseline in the style of counterfactual credit assignment, which adds an auxiliary model, its own optimiser and a training-time future-information channel, so it is not the same learner with one changed objective. All five act on the routing module only, the same scope the mechanism uses, and all five carry responsibility reweighting switched off by registration.

Each risk-inspired line multiplies the routing advantage by a weight built from the batch, exactly as the mechanism does, and differs only in how that weight is formed. Write A for the flattened advantage over the valid rows of the batch, μ_A and σ_A for its mean and standard deviation over those rows, and G for the value target. Every weight below is normalised to unit mean over the valid rows before it multiplies the advantage, masked rows passing through unchanged, so the four differ from one another and from EU-CRD in the shape of the weight rather than in whether one is applied.

$$\text{RCV} \quad w = (1-\lambda) + \frac{\lambda}{\alpha}\,\mathbb{1}[A \le Q_\alpha(A)]\,, \qquad \alpha = 0.2,\ \lambda = 0.5, \tag{8}$$

$$\text{RDC} \quad w = (1-\lambda) + \frac{\lambda}{\alpha}\,\mathbb{1}[G \le Q_\alpha(G)]\,, \qquad \alpha = 0.1,\ \lambda = 0.7, \tag{9}$$

$$\text{RMV} \quad w = \max\!\left(1 - 2\lambda\,\frac{A-\mu_A}{\sigma_A},\ 0\right), \qquad \lambda = 1.0, \tag{10}$$

$$\text{RRS} \quad w = \min\!\left(\exp\!\left(-\beta\,\frac{A-\mu_A}{\sigma_A}\right),\ w_{\max}\right), \qquad \beta = 1.0,\ w_{\max} = 5. \tag{11}$$

Q_α is the empirical α-quantile over the valid rows. RCV upweights the lower tail of the advantage and RDC the lower tail of the value target, which stands in for the return distribution rather than estimating it, and this is one of the reasons these lines are named by style. RMV tilts against advantages above the batch mean and RRS does the same through an exponential utility with a cap. The fifth line, CCA, keeps the unmodified PPO advantage and subtracts a hindsight baseline predicted by an auxiliary network of one hidden layer of width 64, trained for one inner iteration per update at learning rate 10^{-3} on a window of 12 steps of future information. The constants above are the ones in force for every seed and were not tuned per arm.

These are the implementations that ran on this backbone. They are not the published methods, and the numbers in Table 2 compare objective terms inside one learner rather than method families.

Build provenance. Training and evaluation builds are recorded separately by source and gateway JAR hashes. The confirmation evaluator adds data provenance checks, and development rehearsals reproduced archived actions and outcome metrics exactly before confirmation. Each completed run is linked to its recorded build and configuration. The alternative-objective arms satisfy their service contract on all 180 cells. The revision of every run is recorded by hash in the released artefact.

Hardware and runtime. Continuation training runs on a GH200 cluster, one arm per GPU, taking between three and four hours per arm. The deterministic cells and the ladder are evaluated on a single workstation and the sampled-decoding cells on the cluster. Each evaluated cell is one episode, and a reported figure pools six windows.

C EU-CRD Algorithm

Algorithm 1 lists one full EU-CRD training iteration, and the loop it implements is Figure 3. Notation follows Section 3.

Algorithm 1 EU-CRD training (one PPO iteration)

Require: hierarchical policy $\pi_\theta = (\pi_G, \pi_L)$; critic ensemble $\{Q_{\theta_k}\}_{k=1}^{K_Q}$ with frozen random priors; running scales s_k; gate temperature τ_0; floor $\rho_{\min}$; warmup N_w; ensemble weight λ_Q

1: Collect rollouts $\{(s_t, a_t, r_t, \hat{\mathbf{y}}_t, g_t)\}$ with π_θ; compute GAE advantages $\hat{A}_t$
2: **for** each forecast-exposed (global) transition t **do**
3: **Forecast responsibility:** choose each job's candidate on the forecast cover and on the true cover; $R_{\text{forecast}}(t)$ is the regret of the forecast choice (Eq. 2)
4: **Routing-responsibility proxy:** draw policy-self reference $\tilde{a}_t \sim \pi_G(\cdot \mid s_t)$; $\Delta Q(t) = \mu(s_t, a_t) - \mu(s_t, \tilde{a}_t)$, $\sigma_t^2 = \mathrm{Var}_k Q_{\theta_k}(s_t, a_t) + \mathrm{Var}_k Q_{\theta_k}(s_t, \tilde{a}_t)$
5: Epistemic gate $c_t = \exp(-\sigma_t^2/\tau_t)$, $\tau_t = \tau_0 \bar{\sigma}_t^2$; $R_{\text{global}}(t) = c_t \Delta Q(t) + (1 - c_t)\Delta r(t)$ (Eq. 5)
6: Anomaly gate R^+_{forecast}; scale by s_k; shares $\rho_k(t)$ (Eq. 6), routing share floored at $\rho_{\min}$; update s_k (EMA) (Eq. 6)
7: **end for**
8: **Unit-mean weight normalisation (router only):** if call $> N_w$ then $\tilde{A}_t \leftarrow [1 + \eta(\rho_{\text{global}}(t)/\bar{\rho} - 1)]\, \hat{A}_t$ with ρ detached, else $\tilde{A}_t \leftarrow \hat{A}_t$ (Eq. 7)
9: No gradient is propagated through the responsibility computation
10: Update π_G on the clipped PPO surrogate with $\tilde{A}_t$; update π_L on standard $\hat{A}_t$ (forecast-independent)
11: Update the critic ensemble on the bootstrapped TD loss; total objective $\mathcal{L} = \mathcal{L}_{\text{PPO}} + \lambda_Q \mathcal{L}_Q$

D Mechanism Details

Three properties of the mechanism of Section 3.3 are recorded here rather than in the body.

The two terms of Eq. 2 are priced under the realised supply, and a forecast that is numerically wrong without changing the reference selection contributes nothing. The proxy therefore responds to forecast error that changes a cover-based choice rather than to error as such.

The scaling by s_k in Eq. 6 is necessary because the two signals are incommensurate. The routing signal grows with the scale of the critic while R_{forecast} is a carbon quantity in kilograms, so raw ratios would pin the routing share near one. R_{forecast} is also per-step while the routing signal carries downstream return, so the scaling aligns magnitudes but not horizons, and the two shares are a weighting heuristic rather than an additive decomposition of responsibility.

The anomaly threshold of R^+_{forecast} in Section 3.4 is itself a running statistic. A change in forecast quality that persists long enough is absorbed into it, so the gate responds to forecast error that is large relative to recent history rather than in absolute terms. The reference action of Eq. 3 is drawn on-policy, but the absolute value, the scaling and the floor applied afterwards preserve no zero-charge property for typical routing.

E Component Ablation, Per Arm and Seed

Table 3 reports three single-key removals from the full mechanism. Every arm continues from the same three start states, trains for the same 56 000 further steps under the same attenuated training condition, is evaluated at both forecast tiers, and differs from the full arm in exactly one configuration key, asserted on the effective configuration before training. The reference rows are the two arms of Table 1 and were not re-run for this table. Evaluation follows Section 4.1, deterministic decoding on six development windows at both tiers, 108 new cells with none missing, invalid or duplicated, and every pooled group satisfying its completion and on-time requirements. The interval printed for each arm is the minimum-to-maximum range over the three seeds and is not a confidence interval.

Under the registered perturbation all three removals raise carbon on all three seeds. Under the perfect forecast only the removal of the scale standardisation does, the other two disagreeing in sign. Each figure is the conditional effect of one removal with the other components retained, and single-component removal decomposes neither an additive contribution nor an interaction, so the table orders no component by importance and establishes no component as necessary in general.

Table 3: Component ablation, pooled six-window carbon relative to the full mechanism and to the arm with the reweighting removed, per cent. Median over three seeds, seed range in brackets. Positive is worse. ✓ marks rows where all three seeds agree in sign.

Removed	Key	vs full	vs no-reweight
Perfect forecast			
Uncertainty blend	$c_t \equiv 1$	$+0.44$ $[-1.3, +12.6]$	-8.6
Anomaly handling	gate off	$+2.05$ $[-1.0, +13.4]$	-5.5
Scale standardisation	EMA off	$+9.10^{\checkmark}$ $[+3.2, +14.0]$	$+1.0$
Reweighting	applied off	$+7.99^{\checkmark}$ $[+5.8, +18.8]$	—
Attenuated forecast			
Uncertainty blend	$c_t \equiv 1$	$+5.83^{\checkmark}$ $[+3.5, +13.6]$	-6.1
Anomaly handling	gate off	$+8.92^{\checkmark}$ $[+0.7, +10.1]$	-8.6
Scale standardisation	EMA off	$+13.68^{\checkmark}$ $[+10.2, +22.9]$	$+0.2$
Reweighting	applied off	$+10.15^{\checkmark}$ $[+10.0, +29.9]$	—

The table does not support two further readings. Fixing the blend coefficient stays below the no-reweight arm on all three seeds under perturbation, while switching off the anomaly handling has a negative median against that arm but disagrees in sign across seeds, the range reaching $+0.07\%$. Removing the scale standardisation lands within -5.41% to $+3.20\%$ of the no-reweight arm, median $+0.20\%$. No equivalence interval was registered, no equivalence test was run, and the proximity does not show that the standardisation carries the benefit of the reweighting.

Table 4 gives the per-seed values behind Table 3, in kg CO_2 pooled over the six windows, with the percentage difference against the full mechanism on the same seed.

Table 4: Per-seed carbon of the three ablation arms and the difference against the full mechanism, six windows pooled.

		Carbon (kg CO_2)			vs full (%)		
Tier	Removed	A	B	C	A	B	C
Perfect	$c_t \equiv 1$	0.02206	0.02525	0.02407	-1.3	$+0.4$	$+12.6$
Perfect	Anomaly off	0.02280	0.02489	0.02423	$+2.1$	-1.0	$+13.4$
Perfect	EMA off	0.02437	0.02595	0.02437	$+9.1$	$+3.2$	$+14.0$
Attenuated	$c_t \equiv 1$	0.02338	0.02748	0.02401	$+3.5$	$+13.5$	$+5.8$
Attenuated	Anomaly off	0.02276	0.02636	0.02498	$+0.7$	$+8.9$	$+10.1$
Attenuated	EMA off	0.02569	0.02975	0.02501	$+13.7$	$+22.9$	$+10.2$

F Per-Seed Results

Table 5 gives the per-seed values behind the development block of Table 1. Each entry is carbon in kg CO_2 pooled over six 2021 development windows for that seed, arm and tier under deterministic decoding. Completion and on-time share are 1.0 on all 72 development cells. The corresponding confirmation results appear in Appendix G.

Table 5: Per-seed carbon for the two main arms and the difference the reweighting makes. Each row compares two continuations of the same base checkpoint.

	Perfect		Attenuated		
Seed	control	EU-CRD	control	EU-CRD	Δ_{atten} (ctrl / EU-CRD)
A	0.02412	0.02234 (-7.4%)	0.02489	0.02259 (-9.2%)	$+3.2\%$ / $+1.1\%$
B	0.02987	0.02514 (-15.8%)	0.03145	0.02420 (-23.0%)	$+5.3\%$ / -3.8%
C	0.02261	0.02138 (-5.5%)	0.02496	0.02269 (-9.1%)	$+10.4\%$ / $+6.1\%$
pooled	0.07660	0.06886 (-10.1%)	0.08130	0.06948 (-14.5%)	$+6.1\%$ / $+0.9\%$

The reweighted arm is lower on every seed at both tiers, and the carbon that attenuation adds is smaller for it on every seed. Seed B is lower under attenuation than under the perfect forecast, so the arm's own increment is negative there while the paired difference against the control stays negative at both tiers.

Table 6 takes the same cells one window at a time. The pooled reduction keeps its sign whichever single window is withheld, and its size stays between 8.0 and 13.2% under the perfect forecast and between 12.6 and 17.4% under attenuation, so no single window is necessary for the pooled sign. At the level of the individual cell the picture is coarser. The reweighted arm is lower in 11 of the 18 seed-by-window cells under the perfect forecast and in 12 of 18 under attenuation, so the pooled reduction is a sum over cells of uneven size rather than a uniform per-cell improvement. This is a descriptive sensitivity check on six windows that were used while the campaign was designed, not a test against an independent sample.

Table 6: Pooled reduction of EU-CRD against the matched control with one window withheld, per cent. Negative favours EU-CRD.

Tier	all six	$-k_0$	$-k_1$	$-k_2$	$-k_3$	$-k_4$	$-k_5$
Perfect	-10.11	-9.18	-8.82	-13.15	-12.94	-8.02	-9.13
Attenuated	-14.53	-12.62	-12.74	-17.44	-16.27	-12.76	-15.93

G Cross-Year Confirmation

The confirmation evaluates the same six frozen final checkpoints on six preregistered windows from the 2020 wind traces, without further training or checkpoint selection. The window offsets are 24398, 10829, 7479, 20843, 523 and 14997, selected by the registered deterministic hashing and nonoverlap rule. The workload, forecast interventions and deterministic decoding protocol remain unchanged. The 72 cells cover two arms, three training seeds, six windows and two forecast tiers.

Confirmation requires lower pooled carbon under attenuation on every paired seed, a pooled reduction of at least 5%, no increase exceeding 3% under perfect forecasts on any seed, and satisfaction of the service requirements in every cell. Table 7 reports the results. The evaluation meets all four conditions, with a pooled reduction of 12.19% under attenuation. The three attenuated reductions range from 0.19% to 22.53%, so agreement in direction does not imply a uniform effect size. Completion and on-time share are 1.0 and the forced-deadline count is zero in all 72 cells.

Table 7: Confirmation carbon in kg CO_2 pooled over six 2020 windows per seed. Percentages compare EU-CRD with the matched control using unrounded values.

	Perfect		Attenuated	
Seed	Control	EU-CRD	Control	EU-CRD
A	0.030126	0.030026 (-0.33%)	0.031456	0.031396 (-0.19%)
B	0.038322	0.030987 (-19.14%)	0.039937	0.030941 (-22.53%)
C	0.032169	0.028996 (-9.86%)	0.034875	0.030975 (-11.18%)
Pooled	0.100618	0.090009 (-10.54%)	0.106269	0.093312 (-12.19%)

Two technical attempts preceded the completed evaluation. The first was stopped after a forecast provider was found to load the wrong year. The second was an unintended launch during a guard test and was stopped by a filename scan that also detected an unused environment reconstructed from the training checkpoint. Five cells from the first attempt and one from the second were quarantined and excluded. The execution records report that their performance values were not inspected or used to change the protocol. Both attempts remain in the audit history.

Before the completed attempt, a source gate was placed after environment reset and before the first policy decision. It checked the actual simulator and forecast data sources, underlying power sequences, turbine assignments, row mappings, window offset and forecast tier. Synthetic tests verified that source mismatches stopped execution before any decision, and development rehearsals

reproduced archived actions and outcome metrics exactly. All 72 confirmation cells passed the source checks, and the frozen code, configuration and checkpoint hashes remained unchanged.

This confirmation tests the incremental effect of reweighting on another year's wind traces using the same trained policies, sites and workload. It does not repeat the alternative-objective comparisons or component ablations, and the base policy reference was not evaluated on these windows. Registered sampled-decoding confirmation evaluations remain incomplete. The 72 cells form an evaluation grid rather than independent repetitions.

H The Forecast Amplitude Ladder

Figure 1 shows three of the seven forecast conditions under which the conventionally trained base checkpoints of Section 4.1 were evaluated. Table 8 reports all seven. The five amplitude rungs replace the future view by $m + \lambda(y - m)$ with m the long-run mean of the site and the present observation kept as measured, so $\lambda = 1$ is the perfect forecast and $\lambda = 0$ a flat one. The two operators below the rule keep the full amplitude and change its timing, one shuffling the site's future profile under a frozen permutation and the other reversing it in time, the present observation kept as measured in both. These are descriptive evaluations of a fixed policy, under the perfect condition it trained under and under input interventions it never saw, made after the main campaign was designed. Pooled carbon rises monotonically along the amplitude rungs, but the three seeds do not agree in sign on every step, so the pooled direction is reported and no ordering is claimed except the one stated in the figure caption.

Table 8: Carbon of the base policy under seven forecast conditions, six windows pooled per seed. The last column is the pooled total relative to the flat forecast.

Forecast condition	seed A	seed B	seed C	pooled	vs flat (%)
1.00 (perfect)	0.02246	0.02702	0.02618	0.07566	−10.8
0.75	0.02441	0.02606	0.02873	0.07920	−6.7
0.50	0.02613	0.02595	0.02863	0.08070	−4.9
0.25	0.02701	0.02495	0.03093	0.08289	−2.3
0.00 (flat mean)	0.02743	0.02634	0.03108	0.08486	+0.0
time-shuffled	0.02825	0.02837	0.03058	0.08720	+2.8
time-reversed	0.02640	0.02529	0.03143	0.08313	−2.0

I Sampled Decoding

The main reading of Section 4.2 is registered for deterministic decoding. The 72 development cells of the two main arms were also evaluated with sampled actions under three decoding seeds, giving 216 cells with every completion and on-time requirement met. The same frozen reading rules were applied to each decoding seed separately. Table 9 reports these development results. Sampled-decoding evaluations on the confirmation windows remain incomplete.

Table 9: The two main arms under sampled decoding, pooled six-window carbon at the attenuated tier. The effect is the reweighted arm minus the control, so negative favours the reweighting. Per-seed differences are in kilograms.

Decoding seed	Pooled effect	Seed A	Seed B	Seed C	Reading
42	+5.07%	+0.01248	−0.00813	+0.00236	AGAINST
43	−4.02%	+0.00185	−0.00221	−0.00560	MIXED
44	−17.10%	−0.00762	−0.01153	−0.00559	SUPPORTS

The sign of the pooled effect changes with the decoding seed, and the per-seed differences change sign as well. Training seed B favours the reweighting under all three decoding seeds and seed A opposes it under two. No consistent benefit is therefore established under sampled deployment, which is why every claim in this paper is stated for deterministic decoding. The deterministic reading was

registered as primary and read before these cells existed, and these sampled-decoding results were not used to modify it. Sampled decoding is also a different and worse operating point rather than a perturbation of the deterministic one, deployment carbon being roughly twice the deterministic level in both arms. The comparison lines of Section 4.3 were registered with a sampled secondary that was withdrawn after the deterministic result was seen, and the ablation of Section 4.4 was registered for deterministic decoding only, so neither carries a sampled reading.

J Deployment Overhead

EU-CRD applies its weights inside the learner, and the deployed policy is the same backbone every arm carries. Decision latency is instrumented inside the evaluator on the 360 deterministic development cells of the ten arms, and Table 10 reports the median over each arm's 36 cells. The forecast ladder, sampled-decoding and confirmation evaluations are excluded from this overhead summary.

Table 10: Deployment overhead, median over each arm's 36 evaluated cells. Global latency is the time to route one batch, local latency the time to pace one datacentre's queue. Peak resident memory is the evaluator process.

Arm	Global mean/p99 (μs)	Local mean/p99 (μs)	Peak RSS (MB)	Wall (s)
Matched control	1307 / 2049	3.5 / 10.6	1348	22.0
EU-CRD	1310 / 2069	3.5 / 10.6	1348	22.6
Alternatives	1304–1309 / 2038–2089	3.5 / 10.6	1349–1350	21.9–22.6
Ablation arms	1320–1327 / 2077–2095	3.5 / 11.0	1349	22.2–22.5

The measured statistics are close across the ten arms, the spread of per-arm medians ranging from 0.2% for peak memory to 4.7% for the local 99th percentile, which is the direction the design predicts, since the responsibility weights are computed in the learner and not at deployment. No equivalence test was run, so closeness is reported and equality is not claimed. Responsibility weights are not computed at deployment, and the figures reported are the measured end-to-end latencies.

K Reward Design

The two reward channels mirror the hierarchical division of Section 3.2. Both are stated here with the coefficients in force for every arm in this paper, and terms the implementation supports but that were set to zero are listed as such.

The global reward at control step t is

$$r_t^G = -\,\hat{C}_t + w_m\,\Delta F_t + \sum_{i\in\text{placed}} w_p\,\hat{P}_i, \qquad \hat{C}_t = \min\Big(\frac{C_t}{C_{\text{fix}}},\,3\Big), \tag{12}$$

where C_t is the total carbon emitted in the step in kilograms, $C_{\text{fix}} = 5\times10^{-5}$ a fixed normaliser whose ratio cap was never reached in these runs, ΔF_t the increase in the fraction of the compute of the episode finished during the step with $w_m = 0.3$, and $\hat{P}_i = \exp(-6\,\phi_i)$ a credit for each job placed in the step, with ϕ_i the fractional overflow of the chosen site's capacity that the placement would cause and $w_p = 1$. The carbon term is therefore the total carbon of the step at unit weight. Five further terms exist in the implementation and are zero here, a weighted mean of the local rewards (weight 0), a green-waste penalty (weight 0), a throughput term in the log of completed compute (coefficient 0), a marginal-carbon charge per placed job (weight 0), and a deferral cost with an urgency ramp (base cost 0, urgency weight 0). Because the carbon term is a fixed multiple of the step's carbon and the completion term telescopes over the episode, the undiscounted sum of the global reward differs from a fixed multiple of the episode's total carbon only by the completion credits and the placement credits. The router is trained on the discounted return with $\gamma = 0.999$, which is not the undiscounted total.

The local reward of each datacentre is

$$\begin{aligned} r_t^L = {}& \lambda_1 \log(1{+}n_t) - \lambda_2 \log(1{+}\bar{W}_t) \\ & - \lambda_3\,\mathrm{std}_v\,\rho_{v,t} - \lambda_4 \frac{Q_t}{R_t} - \lambda_5\,\mathbb{1}[\text{invalid}] - \lambda_6\,o_t, \end{aligned} \tag{13}$$

which rewards the n_t completions in the step and penalises the mean wait $\bar{W}_t$ of finished jobs, the imbalance of utilisation $\rho_{v,t}$ across the site's virtual machines, the backlog Q_t relative to the release capacity R_t, invalid actions, and a small operating-cost term o_t, with $\lambda_1 = 0.3$, $\lambda_2 = 0.05$, $\lambda_3 = 0.02$, $\lambda_4 = 0.05$, $\lambda_5 = 0.2$ and $\lambda_6 = 0.02$. Carbon does not enter the local reward, and the local scheduler never observes the forecast.

The fallback of Eq. 5 is the green-fit contrast of Eq. 4 with $\alpha = 1$, evaluated slot by slot on the routing batch from the observation the router already saw, with deferring slots contributing zero. Its recorded dispersion relative to the gated value gap is 12 to 16% at the first training iteration and below 1% by the seventh.