FedRAIN-Lite: Federated Reinforcement Algorithms for Improving Idealised Numerical Weather and Climate Models

Pritthijit Nath[1] **Sebastian Schemm**[1] **Henry Moss**[1,2] **Peter Haynes**[1]
Emily Shuckburgh[3] **Mark Webb**[4]

[1] Department of Applied Mathematics and Theoretical Physics, University of Cambridge
[2] School of Mathematical Sciences, Lancaster University
[3] Department of Computer Science and Technology, University of Cambridge
[4] Met Office Hadley Centre
{pn341,ss3299,hm493,phh1,efs20}@cam.ac.uk; mark.webb@metoffice.gov.uk

Abstract

Sub-grid parameterisations in climate models are traditionally static and tuned offline, limiting adaptability to evolving states. This work introduces **FedRAIN-Lite**, a federated reinforcement learning (FedRL) framework that mirrors the spatial decomposition used in general circulation models (GCMs) by assigning agents to latitude bands, enabling local parameter learning with periodic global aggregation. Using a hierarchy of simplified energy-balance climate models, from a single-agent baseline (`ebm-v1`) to multi-agent ensemble (`ebm-v2`) and GCM-like (`ebm-v3`) setups, we benchmark three RL algorithms under different FedRL configurations. Results show that Deep Deterministic Policy Gradient (DDPG) consistently outperforms both static and single-agent baselines, with faster convergence and lower area-weighted RMSE in tropical and mid-latitude zones across both `ebm-v2` and `ebm-v3` setups. DDPG's ability to transfer across hyperparameters and low computational cost make it well-suited for geographically adaptive parameter learning. This capability offers a scalable pathway towards high-complexity GCMs and provides a prototype for physically aligned, online-learning climate models that can evolve with a changing climate. Code accessible at `https://github.com/p3jitnath/climate-rl-fedrl`.

1 Introduction

Climate models are indispensable for understanding the Earth's many interacting systems, from atmospheric circulation to the hydrological cycle, and play a central role in forecasting weather and projecting future climate impacts. However, their predictive skill is often limited by uncertainties arising from static sub-grid parameterisations of unresolved processes, traditionally tuned offline against observations using expensive, ad-hoc experiments [1, 2]. This tuning bottleneck often inhibits adaptability to state-dependent variability within the system. Emerging online learning methods, such as Ensemble Kalman Inversion (EnKI) [3], offer a principled and computationally efficient alternative by casting it as a Bayesian inverse problem, successfully applied to convection schemes in idealised general circulation models (GCMs) [4]. Reinforcement learning (RL) [5], one of the key drivers behind recent advances in large language models (LLMs) [6], has also shown promise in idealised climate settings, enabling models to iteratively learn parameterisation components by interacting with the climate system itself [7]. These recent approaches represent a shift toward adaptive, data-informed parameterisation strategies that can respond to distributional changes over time such as those rising from natural variability or externally forced trends such as global warming.

Published as a workshop paper at "Tackling Climate Change with Machine Learning", NeurIPS 2025

While Nath et al. [7] demonstrated RL's potential in idealised models, their setups lacked spatial decomposition and treated the system as a whole, limiting scalability and regional adaptivity. In contrast, operational GCMs routinely use spatial decomposition for both physics and computations. Embracing this paradigm, this work introduces a federated reinforcement learning (FedRL) framework [8], which we term **FedRAIN-Lite**, where "federated" refers to the use of multiple agents assigned to distinct latitude bands that learn local policies independently, while synchronising periodically via global aggregation to stabilise training and enable knowledge transfer across bands. This multi-agent setup not only reduces optimisation complexity within each region but also mirrors real-world model architectures, enabling both faster convergence and geographically adaptive skill.

As running full-scale GCMs can be computationally expensive and challenging to interpret, we explore a hierarchy of idealised energy balance models (EBMs) [9–11], where the proposed **FedRAIN-Lite** framework significantly improves training stability and skill, particularly in tropical and mid-latitude zones, relative to a non-federated approach. Among three RL algorithms tested: Deep Deterministic Policy Gradient (DDPG) [12], Twin-delayed DDPG (TD3) [13], and Truncated Quantile Critics (TQC) [14] (summaries in Appendix A.1), DDPG emerges as the most hyperparameter-robust and computationally efficient candidate, achieving consistent performance gains in both single-agent and federated multi-agent settings. Compared to static baselines and single-agent global RL models, DDPG under federated coordination converges faster and generalises better across latitude bands.

The key contributions of this work are:

1. **A novel application of FedRL** to climate model parameterisation, using spatial decomposition schemes that mirror the structure of operational GCMs.
2. **Systematic benchmarking** of three RL algorithms (DDPG, TD3, TQC) across a hierarchy of idealised EBMs, spanning single-agent and multi-agent configurations with increasing physical complexity.
3. **Demonstration of DDPG's robustness, scalability, and skill**, showing that it consistently achieves strong performance across decompositions and coordination strategies, making it a practical and an efficient baseline for geographically adaptive climate parameter learning.

2 Methodology

2.1 Background

In numerical weather and climate models, key unresolved processes such as radiation, convection, and turbulence are represented through parameterisations, which are simplified functional forms with fixed or empirically tuned coefficients. These parameters are typically calibrated offline through expensive trial-and-error simulations or derived from theoretical considerations (often relying on highly idealised assumptions), and thus lack adaptability and can degrade performance under evolving or unseen climate states.

RL offers a compelling alternative by framing parameterisation as a sequential decision process, where an agent learns a control policy that dynamically adjusts parameters based on the evolving model state. Recent work demonstrates RL's growing impact across science, from fusion plasma stabilisation to environmental management and fluid control [15–20]. Conventional ML approaches for climate model calibration often lack this feedback-driven adaptability inherent to RL. Moreover, existing RL frameworks typically treat the model as a single unit, neglecting spatial heterogeneity and regional dynamics. This motivates a decentralised approach that better reflects the modular and geographically decomposed design of real-world GCMs.

2.2 Budyko–Sellers Energy Balance Model

The Budyko–Sellers EBM [9–11] is a latitudinally resolved idealised climate model that simulates the zonal-mean surface temperature $T_s(\phi)$ as a function of latitude ϕ. It represents the balance between absorbed solar radiation, outgoing longwave radiation, and meridional heat transport (using a downgradient diffusion assumption), governed by the following equation:

$$C(\phi)\frac{\partial T_s}{\partial t} = \underbrace{(1-\alpha(\phi))Q(\phi)}_{\text{absorbed shortwave}} - \underbrace{(A+BT_s)}_{\text{longwave cooling}} + \underbrace{\frac{D}{\cos\phi}\frac{\partial}{\partial\phi}\left(\cos\phi\frac{\partial T_s}{\partial\phi}\right)}_{\text{diffusive transport}} \tag{1}$$

where $C(\phi)$ is the effective heat capacity, $\alpha(\phi)$ the surface albedo, $Q(\phi)$ the insolation, A and B the outgoing longwave radiation (OLR) coefficients, and D the meridional heat transport parameter, typically treated as constants chosen to match observations. Mathematical details on equilibria and instabilities are discussed in Appendix A.2.

The model is discretised into 96 latitude bands, enabling numerical simulation of temperature dynamics across the globe. The OLR parameters A and B, are the primary targets for optimisation in this work. In the RL setting, these coefficients are treated as learnable policy outputs, with agents optimising them to reduce the temperature error relative to a prescribed climatological target, allowing for spatially adaptive correction policies.

2.3 climateRL Environments

For training RL agents, we construct three climateRL environments (schematics in Appendix A.3), based on the Budyko–Sellers EBM, increasing in spatial complexity and progressively aligning with real GCM design, as explained below.

`ebm-v1` extends the single-agent RL setup from Nath et al. [7], where a global agent observes the full zonal-mean temperature profile $T_s(\phi)$ and learns to modulate radiative parameters A and B adaptively per latitude to minimise the mean squared error against a target climatology (e.g.. reanalysis). This serves as a single agent centralised baseline without spatial decomposition or regional specialisation.

`ebm-v2` introduces spatial decomposition by assigning latitude bands (grouped into two or six regions) to separate agents. Each agent receives the full temperature profile as input but optimises a region-specific reward. FedRL ensures coordination via periodic global aggregation of local policy networks enabling geographically adaptive learning while preserving global coherence.

`ebm-v3` mirrors the GCM design by restricting each agent's input to a local temperature slice. This creates a decentralised, partially observed setting closer to real models, where local physics modules operate on region-specific state variables. Like `ebm-v2`, FedRL is used here with local rewards for global synchronisation.

All climateRL environments are built using climlab [21] and Gymnasium [22], and trained using off-policy algorithms under episodic evaluation. The FedRL setup is implemented via Flower [23], using synchronous aggregation every K episodes (e.g., `fed05`, `fed10`) to balance trade-offs between local adaptation and global synchronisation.

3 Results

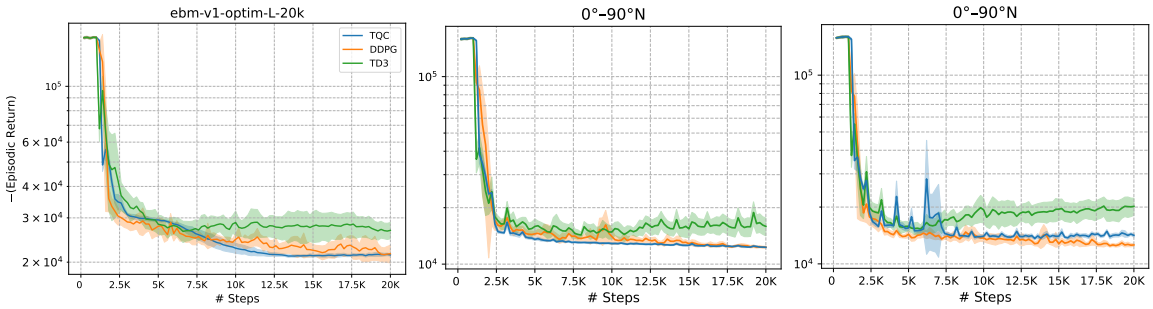
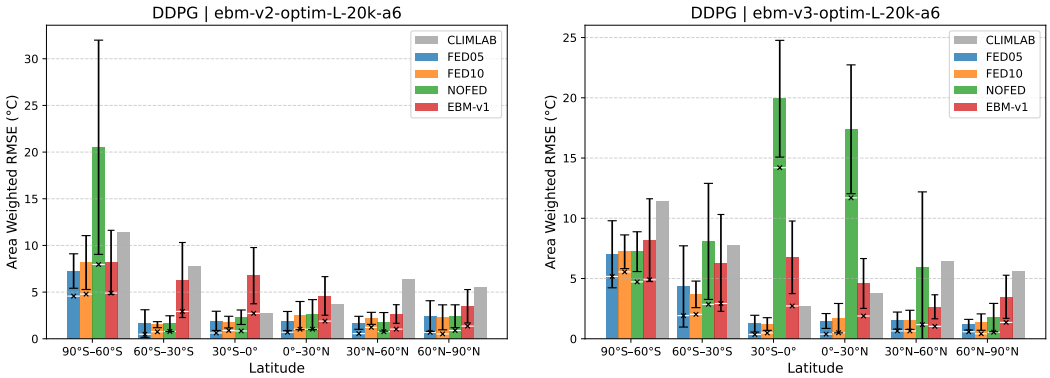


Figure 1: Episodic return curves (log-scaled) with 95% spreads over 10 seeds for three RL algorithms: TQC (blue), DDPG (orange), and TD3 (green), across three climateRL environments. Left: `ebm-v1` (single-agent, global input, global reward and latitude-specific parameters). Middle: `ebm-v2` (multi-agent FedRL setup with shared global profile input and local rewards). Right: `ebm-v3` (multi-agent FedRL setup with sliced inputs and local rewards, mirroring GCM-like spatial decomposition).



(a) Zonal skill (areaWRMSE) of DDPG in `ebm-v2`

(b) Zonal skill (areaWRMSE) of DDPG in `ebm-v3`

Figure 2: Comparison of zonal skill achieved by DDPG under FedRL coordination in `ebm-v2` and `ebm-v3`, both using the 6-agent spatial decomposition (`a6`). Skill is evaluated using areaWRMSE between predicted and reference temperature profiles, averaged with 95% spreads over 10 seeds. Each subplot reports results for three FedRL schemes: `fed05`, `fed10`, `nofed`, along with single-agent `ebm-v1` and the static `climlab` baseline. White horizontal bars with a cross indicate the best-performing seed for each scheme. Both setups adopt the same policy network architecture and hyperparameters as `ebm-v1`. Detailed skill metrics for all experiments presented in Appendix B.1.

Convergence is significantly faster in the FedRL schemes: `ebm-v2` and `ebm-v3` (as can be seen in Figure 1) compared to the single-agent `ebm-v1` setup. Training curves indicate that most policies stabilise by 2.5k–5k steps in `ebm-v2/3`, while `ebm-v1` shows delayed convergence beyond 10k steps. This accelerated training can be attributed to the localised policy learning and reward structures, which reduce the complexity of the optimisation landscape in the decentralised setups.

In Figure 2, across nearly all latitude bands, `fed05` outperforms both the static baseline and non-federated (`nofed`) counterparts, achieving significant skill improvements particularly in the tropics (e.g., over 50% reduction in area-weighted RMSE (areaWRMSE) in 30°S–0° and 0°–30°N for both `ebm-v2` and `ebm-v3`). The gains are more pronounced in `ebm-v3`, where region-specific inputs likely aid specialisation. In contrast, `fed10`, while still outperforming `nofed`, yields higher variance and inconsistent benefits, indicating that frequent aggregation (`fed05`) is essential for stable coordination. In polar regions, all federated schemes perform comparably to or better than `ebm-v1`, showcasing that local specialisation assists in challenging loss landscapes with sharp gradients. Even under coarser decomposition (`a2` in Appendix B.1), DDPG under `ebm-v2/3` maintains monotonic convergence and achieves low final errors, reinforcing its robustness across spatial setups.

Results here highlight the benefits of regional specialisation via FedRL and confirm DDPG's hyper-parameter robustness to changes in reward structure and input resolution (alongside computational efficiency), reinforcing its suitability for GCM-style architectures. Additional results for TD3 and TQC are provided in Appendix B.2. While occasionally competitive, both exhibit higher variance and instability, especially under frequent aggregation or in equatorial and polar regions.

4 Conclusion

This work demonstrates that combining RL with federated learning and spatial decomposition offers a scalable and effective strategy for adaptive climate model parameterisation. By aligning with the spatial decomposition used in GCMs, FedRAIN-Lite allows regional agents to learn locally specialised corrections while preserving global coordination. Among the methods evaluated, DDPG consistently achieves stable convergence, low zonal errors, and strong generalisation across both single- and multi-agent setups. By producing region-specific parameter adjustments, the framework also supports interpretability through physical analysis of learnt policies, offering strong potential for future work. Overall, these results position lightweight RL as a practical bridge from idealised EBMs to operational GCMs, paving the way for more responsive, data-driven parameterisations in future climate change assessments.

Acknowledgements and Disclosure of Funding

P. Nath was supported by the UKRI Centre for Doctoral Training in Application of Artificial Intelligence to the study of Environmental Risks [EP/S022961/1]. Mark Webb was supported by the Met Office Hadley Centre Climate Programme funded by DSIT.

References

[1] Hourdin F, Mauritsen T, Gettelman A, Golaz JC, Balaji V, Duan Q, et al. The Art and Science of Climate Model Tuning. 2017 Mar. Section: Bulletin of the American Meteorological Society. Available from: https://journals.ametsoc.org/view/journals/bams/98/3/bams-d-15-00135.1.xml.

[2] Rasp S, Pritchard MS, Gentine P. Deep learning to represent subgrid processes in climate models. Proceedings of the National Academy of Sciences. 2018 Sep;115(39):9684-9. Publisher: Proceedings of the National Academy of Sciences. Available from: https://www.pnas.org/doi/10.1073/pnas.1810286115.

[3] Iglesias MA, Law KJH, Stuart AM. The Ensemble Kalman Filter for Inverse Problems. Inverse Problems. 2013 Apr;29(4):045001. ArXiv:1209.2736 [math]. Available from: http://arxiv.org/abs/1209.2736.

[4] Dunbar ORA, Garbuno-Inigo A, Schneider T, Stuart AM. Calibration and Uncertainty Quantification of Convective Parameters in an Idealized GCM. Journal of Advances in Modeling Earth Systems. 2021;13(9):e2020MS002454. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1029/2020MS002454.

[5] Sutton RS, Barto AG. Reinforcement learning: An introduction. vol. 1. MIT Press Cambridge; 1998. Available from: http://incompleteideas.net/book/RLbook2020.pdf.

[6] DeepSeek-AI, Guo D, Yang D, Zhang H, Song J, Zhang R, et al.. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv; 2025. ArXiv:2501.12948 [cs]. Available from: http://arxiv.org/abs/2501.12948.

[7] Nath P, Moss H, Shuckburgh E, Webb M. RAIN: Reinforcement Algorithms for Improving Numerical Weather and Climate Models. arXiv; 2024. ArXiv:2408.16118 [cs]. Available from: http://arxiv.org/abs/2408.16118.

[8] Jin H, Peng Y, Yang W, Wang S, Zhang Z. Federated Reinforcement Learning with Environment Heterogeneity. In: Proceedings of The 25th International Conference on Artificial Intelligence and Statistics. PMLR; 2022. p. 18-37. ISSN: 2640-3498. Available from: https://proceedings.mlr.press/v151/jin22a.html.

[9] Budyko MI. The effect of solar radiation variations on the climate of the Earth. Tellus. 1969;21(5):611-9. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2153-3490.1969.tb00466.x.

[10] Sellers WD. A Global Climatic Model Based on the Energy Balance of the Earth-Atmosphere System. 1969 Jun. Section: Journal of Applied Meteorology and Climatology. Available from: https://journals.ametsoc.org/view/journals/apme/8/3/1520-0450_1969_008_0392_agcmbo_2_0_co_2.xml.

[11] North GR. Theory of Energy-Balance Climate Models. 1975 Nov. Section: Journal of the Atmospheric Sciences. Available from: https://journals.ametsoc.org/view/journals/atsc/32/11/1520-0469_1975_032_2033_toebcm_2_0_co_2.xml.

[12] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, et al.. Continuous control with deep reinforcement learning. arXiv; 2019. ArXiv:1509.02971 [cs]. Available from: http://arxiv.org/abs/1509.02971.

[13] Fujimoto S, Hoof Hv, Meger D. Addressing Function Approximation Error in Actor-Critic Methods. arXiv; 2018. ArXiv:1802.09477 [cs]. Available from: http://arxiv.org/abs/1802.09477.

[14] Kuznetsov A, Shvechikov P, Grishin A, Vetrov D. Controlling Overestimation Bias with Truncated Mixture of Continuous Distributional Quantile Critics. In: Proceedings of the 37th International Conference on Machine Learning. PMLR; 2020. p. 5556-66. ISSN: 2640-3498. Available from: https://proceedings.mlr.press/v119/kuznetsov20a.html.

[15] Degrave J, Felici F, Buchli J, Neunert M, Tracey B, Carpanese F, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. Nature. 2022 Feb;602(7897):414-9. Publisher: Nature Publishing Group. Available from: https://www.nature.com/articles/s41586-021-04301-9.

[16] Mole A, Weissenbacher M, Rigas G, Laizet S. Reinforcement Learning Increases Wind Farm Power Production by Enabling Closed-Loop Collaborative Control. arXiv; 2025. ArXiv:2506.20554 [physics]. Available from: http://arxiv.org/abs/2506.20554.

[17] Seo J, Kim S, Jalalvand A, Conlin R, Rothstein A, Abbate J, et al. Avoiding fusion plasma tearing instability with deep reinforcement learning. Nature. 2024 Feb;626(8000):746-51. Publisher: Nature Publishing Group. Available from: https://www.nature.com/articles/s41586-024-07024-9.

[18] Yu J, Schreck JS, Gagne DJ, Oleson KW, Li J, Liang Y, et al.. Reinforcement Learning (RL) Meets Urban Climate Modeling: Investigating the Efficacy and Impacts of RL-Based HVAC Control. arXiv; 2025. ArXiv:2505.07045 [cs]. Available from: http://arxiv.org/abs/2505.07045.

[19] Chapman M, Xu L, Lapeyrolerie M, Boettiger C. Bridging adaptive management and reinforcement learning for more robust decisions. Philosophical Transactions of the Royal Society B: Biological Sciences. 2023 May;378(1881):20220195. Publisher: Royal Society. Available from: https://royalsocietypublishing.org/doi/10.1098/rstb.2022.0195.

[20] Cai W, Wang G, Zhang Y, Qu X, Huang Z. Reinforcement Learning for Active Matter. arXiv; 2025. ArXiv:2503.23308 [cond-mat]. Available from: http://arxiv.org/abs/2503.23308.

[21] Rose BE. CLIMLAB: a Python toolkit for interactive, process-oriented climate modeling. J Open Source Softw. 2018;3(24):659. Available from: https://www.theoj.org/joss-papers/joss.00659/10.21105.joss.00659.pdf.

[22] Towers M, Kwiatkowski A, Terry J, Balis JU, Cola GD, Deleu T, et al.. Gymnasium: A Standard Interface for Reinforcement Learning Environments. arXiv; 2024. ArXiv:2407.17032 [cs]. Available from: http://arxiv.org/abs/2407.17032.

[23] Beutel DJ, Topal T, Mathur A, Qiu X, Fernandez-Marques J, Gao Y, et al.. Flower: A Friendly Federated Learning Research Framework. arXiv; 2022. ArXiv:2007.14390 [cs]. Available from: http://arxiv.org/abs/2007.14390.

[24] Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M. Deterministic Policy Gradient Algorithms. In: Proceedings of the 31st International Conference on Machine Learning. PMLR; 2014. p. 387-95. ISSN: 1938-7228. Available from: https://proceedings.mlr.press/v32/silver14.html.

[25] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. arXiv; 2018. ArXiv:1801.01290 [cs]. Available from: http://arxiv.org/abs/1801.01290.

Appendix A Additional Background

A.1 RL Algorithm Summaries

Table A.1: Four point summaries of the DDPG, TD3 and TQC

Algorithm	Properties
Deep Deterministic Policy Gradient (DDPG) [12]	1. Off-policy actor-critic algorithm that extends DPG [24] using deep function approximators and additional stabilisation mechanisms. 2. Introduces experience replay and target networks with soft target updates to decorrelate samples and improve training stability. 3. Actor and critic networks are both updated similar to DPG. 4. Overestimation bias and sensitivity to exploration noise often limit performance unless mitigated by design changes (e.g. TD3).
Twin Delayed DDPG (TD3) [13]	1. Off-policy actor-critic method designed to reduce the overestimation bias observed in DDPG. 2. Uses a double critic architecture where the minimum of two Q-value estimates is used for critic updates. 3. Actor is updated less frequently than the critics, and target networks are softly updated to reduce update variance. 4. Injects temporally correlated Gaussian noise into the target actions to promote exploration in continuous action spaces.
Truncated Quantile Critics (TQC) [14]	1. Off-policy actor-critic algorithm that builds on SAC [25] with a distributional critic using quantile regression. 2. Models the full distribution of returns $Z(s, a)$ as quantiles $\{\tau_i\}$, capturing uncertainty and reducing bias. 3. Discards the top k quantiles before computing target values to avoid overestimation from outlier returns. 4. Provides a smooth and robust training signal for distributional value estimation by minimising the quantile Huber loss: $$\mathcal{L}_\tau = \frac{1}{N} \sum_{i=1}^{N} \rho_\kappa(\tau_i - y)$$ where τ_i is the predicted i-th quantile, y is the target return, ρ_κ denotes the Huber loss with threshold κ, and N is the number of quantile estimates used in training.

A.2 EBM Equilibria and Instabilities

Multiple Equilibria and Climate Tipping. The Budyko-Sellers EBM can admit multiple steady-state solutions due to the non-linear dependence of albedo on temperature. To understand this, we consider a spatially averaged version of the model, where the diffusion term is neglected ($D = 0$) and all quantities are averaged over latitudes. The energy balance equation in Eq. 1 reduces to an ordinary differential equation:

$$C\frac{dT}{dt} = (1 - \alpha(T))Q - (A + BT) \tag{2}$$

At steady state, $\frac{dT}{dt} = 0$, we solve:

$$(1 - \alpha(T))Q = A + BT \tag{3}$$

If $\alpha(T)$ is a smooth or piecewise function with a sharp transition around the freezing point T_c, this equation may admit more than one root. In particular:

- A warm stable equilibrium: low albedo (e.g., open ocean), high T
- A cold stable equilibrium: high albedo (e.g., ice-covered), low T
- An intermediate unstable solution separating the two

This structure forms an *S-shaped* bifurcation diagram, where gradual changes in solar forcing Q or feedback strength can lead to sudden jumps between climate states, a phenomenon known as climate tipping. The multiplicity of solutions results from the positive ice-albedo feedback that amplifies temperature perturbations.

Linear Stability of Warm and Cold Equilibria. To assess the stability of a given steady-state temperature profile $T^\star(\phi)$, we linearise the full equation about $T^\star$. Let $T(\phi, t) = T^\star(\phi) + \delta T(\phi, t)$, where δT is a small perturbation. Substituting into Eq. 1 and retaining only linear terms gives:

$$C(\phi)\frac{\partial \delta T}{\partial t} = -B\delta T + \frac{D}{\cos\phi}\frac{\partial}{\partial \phi}\left(\cos\phi \frac{\partial \delta T}{\partial \phi}\right) \tag{4}$$

or, in transformed coordinates $x = \sin\phi$:

$$C(x)\frac{\partial \delta T}{\partial t} = -B\delta T + D\frac{d}{dx}\left((1 - x^2)\frac{d\delta T}{dx}\right) \tag{5}$$

This is a linear partial differential equation in $\delta T(x, t)$, which can be interpreted as an eigenvalue problem:

$$\lambda \delta T = -\frac{B}{C(x)}\delta T + \frac{D}{C(x)}\mathcal{L}[\delta T] \quad \text{where} \quad \mathcal{L}[\delta T] := \frac{d}{dx}\left((1 - x^2)\frac{d\delta T}{dx}\right) \tag{6}$$

To solve this, we expand $\delta T(x, t)$ in terms of the eigenfunctions of the Sturm–Liouville operator $\mathcal{L}$. These eigenfunctions are the Legendre polynomials $P_n(x)$, which satisfy:

$$\mathcal{L}[P_n] = -n(n + 1)P_n(x) \tag{7}$$

Substituting into the eigenvalue equation, we obtain:

$$\lambda_n = -\frac{B}{C} - \frac{D}{C}n(n + 1) \tag{8}$$

where $n = 0, 1, 2, \ldots$ and we assume $C(x) = C$ is constant for simplicity.

Each eigenvalue λ_n is strictly negative since both terms are negative, implying that all modes decay exponentially with time. The higher the value of n, the faster the decay, corresponding to the damping of fine-scale spatial features. Hence, the equilibrium is linearly stable. If $C(x)$ varies with latitude, this eigenvalue spectrum will be modified accordingly, but the sign of the real part remains governed by the relative magnitudes of B, D, and the spatial variation in $C(x)$.

The spectrum of eigenvalues $\{\lambda_n\}$ provides insight into the stability of the different equilibria:

- **Warm Equilibrium.** At the warm stable branch, the surface temperature $\overline{T_s}(x)$ remains well above the ice threshold T_c over most latitudes, resulting in a uniformly low albedo $\alpha(\phi) \approx \alpha_{\text{water}}$. The radiative damping coefficient B and diffusivity D dominate the dynamics, ensuring that all eigenvalues λ_n remain strictly negative. This guarantees that all perturbation modes decay exponentially, and the equilibrium is linearly stable.
- **Cold Equilibrium.** In the cold branch, nearly the entire domain is ice-covered, and $\alpha(\phi) \approx \alpha_{\text{ice}}$ is large. The outgoing radiation $A + B\overline{T_s}$ is lower due to lower temperatures, but the structure of the eigenvalue spectrum remains similar. Although the temperature sensitivity of albedo becomes small (flat high-albedo state), the overall damping remains strong, and the eigenvalues λ_n are still negative. Thus, this branch is also linearly stable.

Instability in the Intermediate Branch. Although the linearised eigenvalue spectrum for constant-coefficient EBM in Eq. 8 yields strictly negative eigenvalues, this result assumes that the albedo $\alpha(x)$ is independent of temperature. In the intermediate equilibrium, however, the steady-state temperature $\overline{T}(x)$ lies near the ice–water transition threshold T_c, and hence small perturbations in temperature cause large changes in albedo.

To account for temperature-dependent albedo in the linear stability analysis, we modify the perturbed form of the EBM:

$$C(x)\frac{\partial \delta T}{\partial t} = -B\delta T + D\frac{d}{dx}\left((1-x^2)\frac{d\delta T}{dx}\right) - Q(x)\cdot\frac{d\alpha}{dT}\cdot\delta T \tag{9}$$

Here, the albedo perturbation introduces an additional source term through the chain rule:

$$\delta\alpha(x) = \frac{d\alpha}{dT}\cdot\delta T(x) \tag{10}$$

Substituting into the linearised energy balance yields a modified restoring term:

$$-B\delta T(x) - Q(x)\cdot\frac{d\alpha}{dT}\cdot\delta T(x) = -\left(B + Q(x)\cdot\frac{d\alpha}{dT}\right)\delta T(x) \tag{11}$$

This implies an **effective damping** coefficient:

$$B_{\text{eff}}(x) = B + Q(x)\cdot\frac{d\alpha}{dT} \tag{12}$$

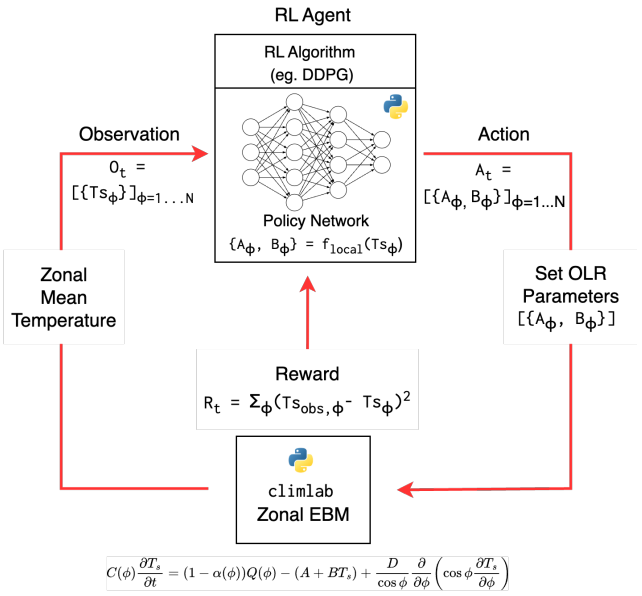
As a result, the eigenvalue equation becomes:

$$\lambda\delta T = -\frac{B_{\text{eff}}(x)}{C(x)}\delta T + \frac{D}{C(x)}\cdot\frac{d}{dx}\left((1-x^2)\frac{d\delta T}{dx}\right) \tag{13}$$

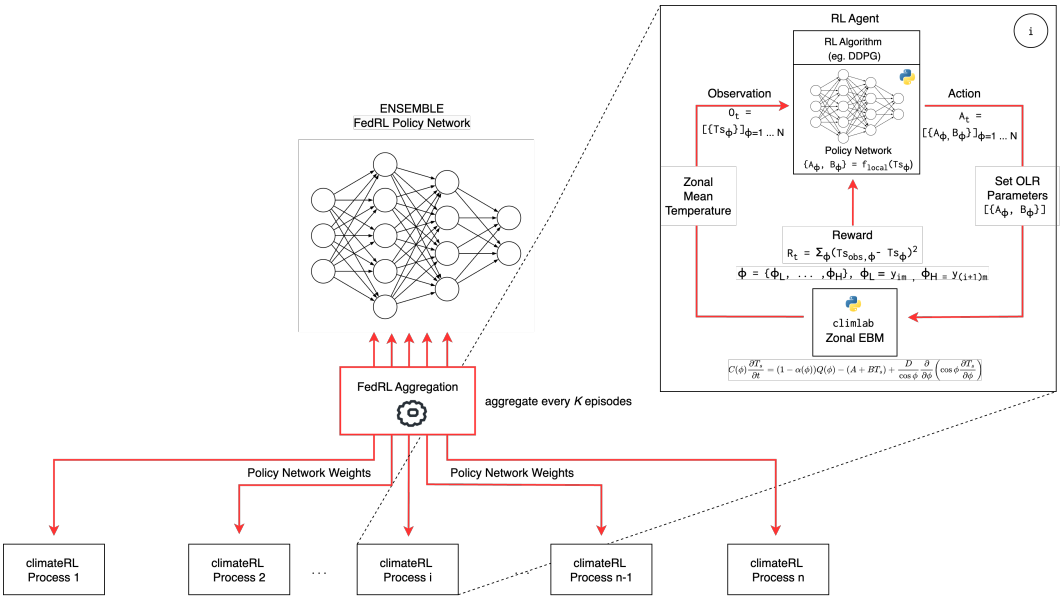
When $\frac{d\alpha}{dT}$ is large and negative, typical near the albedo transition temperature T_c, the effective damping term $B_{\text{eff}}(x)$ can become negative in parts of the domain. If this occurs over a sufficiently wide region in $x \in [-1, 1]$, the dominant eigenvalue λ_0 may cross zero and become positive, indicating the onset of linear instability.

This mechanism explains how the intermediate equilibrium, where the climate state is sensitive to ice–albedo feedback near the freezing threshold, is destabilised. In contrast, both the warm and cold equilibria satisfy $\frac{d\alpha}{dT} \approx 0$, leading to $B_{\text{eff}}(x) \approx B > 0$ and a strictly negative eigenvalue spectrum, ensuring stability.

A.3 climateRL EBM Schematics



(a) `ebm-v1` single-agent setup. The global agent observes the full zonal-mean temperature profile and outputs latitude-dependent radiative parameters $\{A_\phi, B_\phi\}$. Loss from observations are computed over all 96 latitudes.



(b) `ebm-v2` multi-agent ensemble with FedRL agents operate on latitude groups with local rewards while receiving the global profile as input. Periodic aggregation every K episodes synchronises policy weights across n agents.

Figure A.1: Schematics for climateRL EBM environments

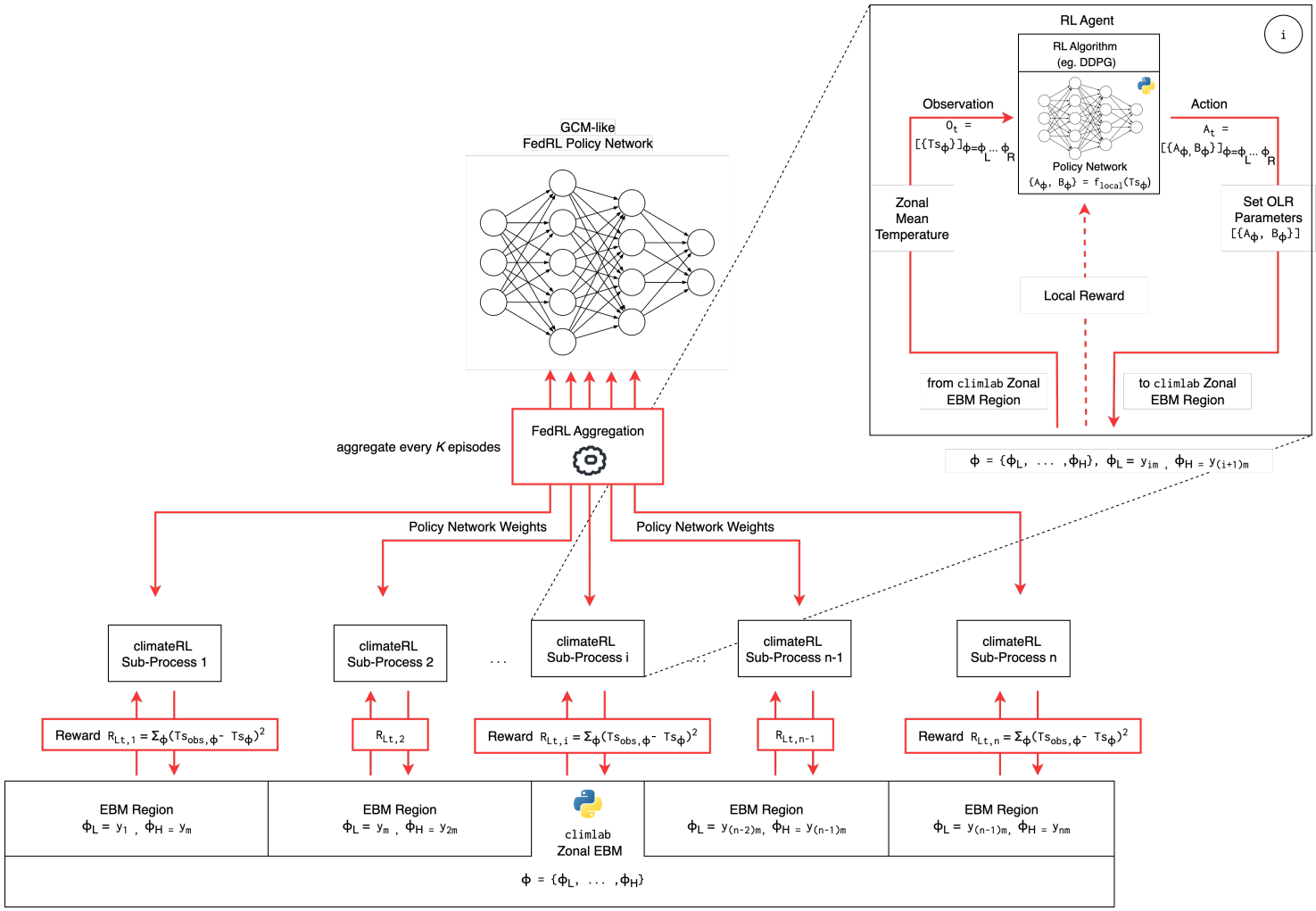


Figure A.1: Schematics for GCM-like `ebm-v3`. FedRL agents observe local temperature slices only (unlike `ebm-v2`) and optimise local rewards. Local rewards are computed over regions for each agent. Policy weights are aggregated every K episodes via Flower for FedRL and also contribute towards a global non-local policy.

A.4 RL Algorithm Hyperparameters

Table A.2: Tabular representation of different RL hyperparameters

Algorithm	Parameter Names	Count
DDPG	`learning_rate, tau, batch_size, exploration_noise, policy_frequency, noise_clip, actor_critic_layer_size`	7
TD3	`learning_rate, tau, batch_size, policy_noise, exploration_noise, policy_frequency, noise_clip, actor_critic_layer_size`	8
TQC	`tau, batch_size, n_quantiles, n_critics, actor_adam_lr, critic_adam_lr, alpha_adam_lr, policy_frequency, target_network_frequency, actor_critic_layer_size`	10

A.5 Experimental Outline

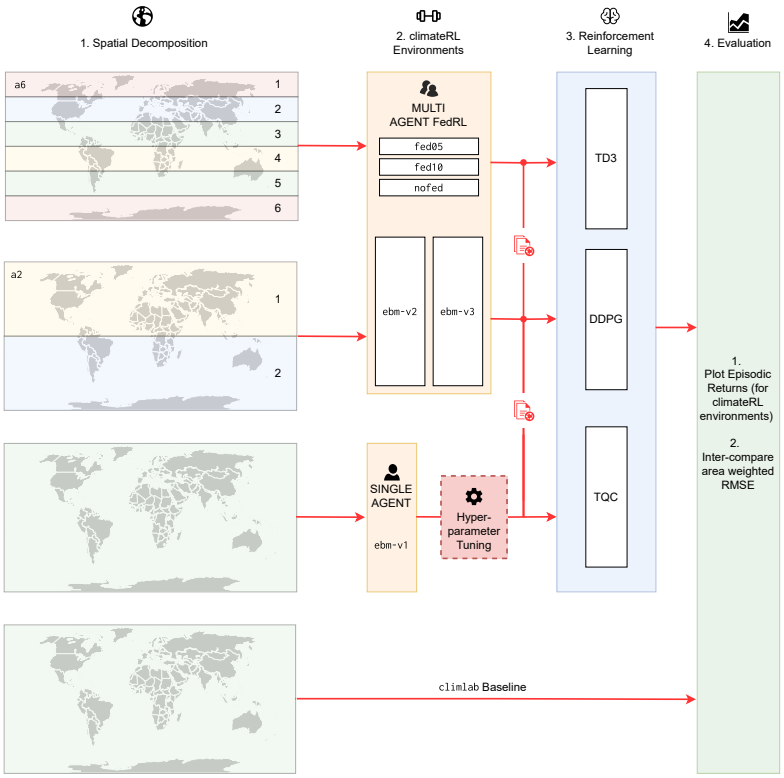


Figure A.2: Pipeline for the `ebm-v1/2/3` experiments. The process begins with configuring the Budyko–Sellers EBM in either single-agent (`ebm-v1`) or spatially decomposed multi-agent forms (`ebm-v2`, `ebm-v3`) using two (`a2`) or six (`a6`) regions. Agents are trained with one of three RL algorithms (DDPG, TD3, TQC) under coordination schemes `fed05`, `fed10`, or `nofed`. In multi-agent settings, policies are periodically aggregated via FedRL every K episodes. Hyperparameters tuned for `ebm-v1` are transferred over to `ebm-v2/v3`. Finally trained models are assessed on their training curves and benchmarked against a static `climlab` baseline, using a skill measure such as areaWRMSE across 30° latitude groups.

A.6 EBM State Evolution

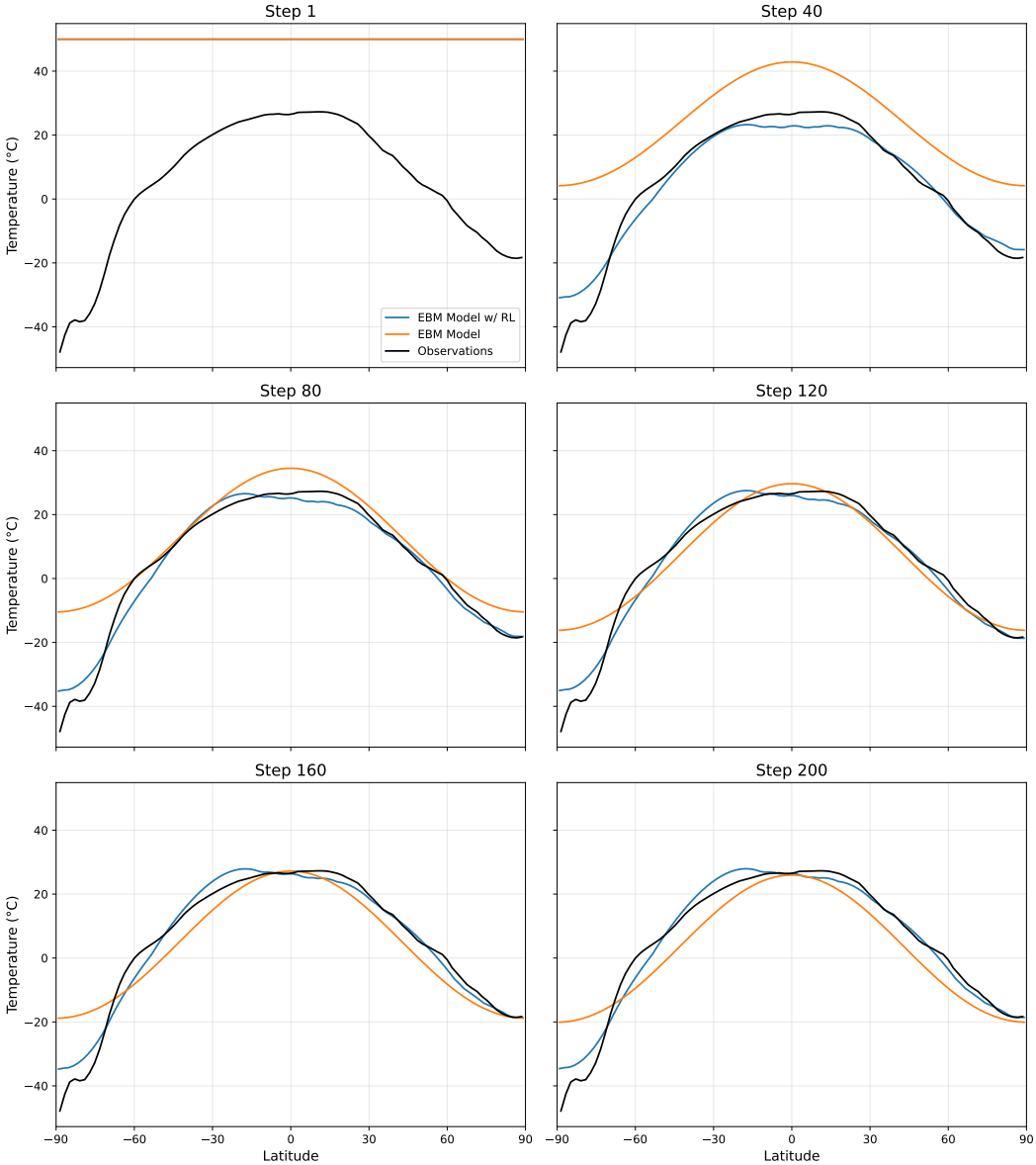


Figure A.3: Evolution of the zonal-mean surface temperature in the `ebm-v1` climateRL environment over 200 integration steps. Each panel shows the latitudinal temperature profile at a selected timestep ($t = 1, 40, 80, 120, 160, 200$), comparing the DDPG-assisted EBM (blue) with the standard `climlab` EBM (orange) and reanalysis observations (black). The RL agent dynamically adjusts the OLR parameters A and B per latitude, improving temperature representation while maintaining physical consistency.

Appendix B Additional Results

B.1 Skill Metrics

Table B.3: Zonal-band errors for `ebm-v2-optim-L-20k-a2`. Each subtable reports mean ± std and relative gain % versus `ebm-v1` for three regimes `fed05`, `fed10` and `nofed`, along with a comparison against the static baseline `climlab`

(a) DDPG

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	5.11 ± 0.533	37.550	6.14 ± 0.964	25.040	16.17 ± 14.718	-97.490	8.19 ± 3.433
60°S–30°S	7.768	3.08 ± 1.155	51.030	3.39 ± 0.775	46.080	10.30 ± 10.647	-63.620	6.30 ± 4.019
30°S–0°	2.730	3.46 ± 1.984	48.890	3.92 ± 1.642	42.080	12.24 ± 13.889	-81.120	6.76 ± 3.013
0°–30°N	3.746	2.80 ± 2.384	39.050	1.89 ± 1.325	58.760	2.23 ± 1.246	51.460	4.59 ± 2.068
30°N–60°N	6.398	2.35 ± 0.866	11.350	2.26 ± 1.307	14.730	2.38 ± 1.253	10.360	2.65 ± 0.997
60°N–90°N	5.566	1.60 ± 0.682	54.090	1.95 ± 0.995	44.060	2.49 ± 0.758	28.400	3.48 ± 1.790

(b) TD3

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	8.00 ± 1.380	17.850	6.87 ± 0.825	29.520	7.72 ± 1.371	20.750	9.74 ± 3.330
60°S–30°S	7.768	7.32 ± 2.327	-2.870	5.51 ± 2.337	22.530	4.98 ± 1.696	29.980	7.12 ± 3.800
30°S–0°	2.730	4.47 ± 1.889	-17.750	4.77 ± 2.313	-25.480	3.49 ± 1.969	8.210	3.80 ± 1.713
0°–30°N	3.746	4.03 ± 2.409	-42.080	5.67 ± 3.152	-100.100	4.06 ± 2.277	-43.360	2.83 ± 1.247
30°N–60°N	6.398	4.30 ± 2.556	14.860	4.79 ± 3.138	5.170	4.39 ± 1.941	13.060	5.06 ± 2.602
60°N–90°N	5.566	3.54 ± 1.867	34.680	3.50 ± 1.518	35.360	5.63 ± 1.948	-4.030	5.42 ± 2.721

(c) TQC

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	8.28 ± 0.896	8.550	8.13 ± 0.765	10.300	8.85 ± 1.124	2.320	9.06 ± 1.549
60°S–30°S	7.768	7.46 ± 1.618	6.440	7.08 ± 1.382	11.120	8.24 ± 1.716	-3.390	7.97 ± 1.459
30°S–0°	2.730	2.83 ± 1.195	-26.510	2.34 ± 1.053	-4.390	3.32 ± 1.269	-48.170	2.24 ± 0.868
0°–30°N	3.746	2.30 ± 0.511	-25.400	2.19 ± 0.604	-19.480	2.49 ± 1.074	-35.470	1.84 ± 0.561
30°N–60°N	6.398	0.93 ± 0.222	61.460	0.92 ± 0.149	61.860	1.23 ± 0.423	49.160	2.42 ± 0.706
60°N–90°N	5.566	1.34 ± 0.219	42.680	1.33 ± 0.352	43.140	1.44 ± 0.297	38.240	2.33 ± 0.767

Table B.4: Zonal-band errors for `ebm-v2-optim-L-20k-a6`. Each subtable reports mean ± std and relative gain % versus `ebm-v1` for three regimes `fed05`, `fed10` and `nofed`, along with a comparison against the static baseline `climlab`

(a) DDPG

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	7.26 ± 1.845	11.340	8.17 ± 2.886	0.260	20.52 ± 11.476	-150.600	8.19 ± 3.433
60°S–30°S	7.768	1.63 ± 1.478	74.100	1.51 ± 0.324	76.020	1.62 ± 0.843	74.240	6.30 ± 4.019
30°S–0°	2.730	1.92 ± 1.029	71.600	1.79 ± 0.624	73.580	2.31 ± 0.776	65.870	6.76 ± 3.013
0°–30°N	3.746	1.89 ± 1.032	58.750	2.49 ± 1.509	45.850	2.59 ± 1.607	43.640	4.59 ± 2.068
30°N–60°N	6.398	1.71 ± 0.697	35.620	2.19 ± 0.643	17.530	1.81 ± 0.998	31.730	2.65 ± 0.997
60°N–90°N	5.566	2.43 ± 1.643	30.190	2.30 ± 1.338	34.050	2.42 ± 1.224	30.670	3.48 ± 1.790

(b) TD3

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	7.01 ± 2.553	28.000	6.04 ± 0.933	38.030	15.90 ± 9.258	-63.150	9.74 ± 3.330
60°S–30°S	7.768	3.16 ± 1.133	55.560	3.52 ± 1.276	50.510	3.52 ± 1.627	50.480	7.12 ± 3.800
30°S–0°	2.730	7.68 ± 2.389	-102.170	8.16 ± 1.813	-114.730	6.10 ± 2.088	-60.670	3.80 ± 1.713
0°–30°N	3.746	7.27 ± 2.023	-156.640	7.63 ± 1.859	-169.470	6.05 ± 2.532	-113.500	2.83 ± 1.247
30°N–60°N	6.398	3.92 ± 1.599	22.380	3.80 ± 0.851	24.770	3.47 ± 1.360	31.460	5.06 ± 2.602
60°N–90°N	5.566	2.97 ± 1.614	45.090	2.48 ± 1.563	54.120	5.55 ± 5.617	-2.380	5.42 ± 2.721

(c) TQC

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	34.96 ± 7.034	-285.900	28.35 ± 7.956	-212.890	33.90 ± 7.319	-274.160	9.06 ± 1.549
60°S–30°S	7.768	1.68 ± 1.087	78.890	1.69 ± 1.215	78.780	1.28 ± 0.393	83.920	7.97 ± 1.459
30°S–0°	2.730	1.97 ± 1.814	11.850	2.25 ± 1.590	-0.480	0.80 ± 0.213	64.350	2.24 ± 0.868
0°–30°N	3.746	1.21 ± 0.670	34.310	1.77 ± 1.363	3.430	0.75 ± 0.190	59.300	1.84 ± 0.561
30°N–60°N	6.398	1.97 ± 1.492	18.750	1.73 ± 0.551	28.600	1.17 ± 0.330	51.920	2.42 ± 0.706
60°N–90°N	5.566	30.70 ± 8.534	-1215.560	32.88 ± 9.606	-1308.920	43.12 ± 14.594	-1747.640	2.33 ± 0.767

Table B.5: Zonal-band errors for `ebm-v3-optim-L-20k-a2`. Each subtable reports mean ± std and relative gain % versus `ebm-v1` for three regimes `fed05`, `fed10` and `nofed`, along with a comparison against the static baseline `climlab`

(a) DDPG

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	6.69 ± 1.766	18.240	7.52 ± 2.718	8.180	7.76 ± 2.193	5.200	8.19 ± 3.433
60°S–30°S	7.768	3.30 ± 1.168	47.540	4.20 ± 1.272	33.290	3.98 ± 2.135	36.810	6.30 ± 4.019
30°S–0°	2.730	4.84 ± 2.151	28.340	3.77 ± 2.190	44.220	3.26 ± 1.873	51.720	6.76 ± 3.013
0°–30°N	3.746	2.42 ± 1.786	47.180	2.96 ± 2.276	35.560	2.49 ± 1.461	45.840	4.59 ± 2.068
30°N–60°N	6.398	2.00 ± 0.885	24.650	2.73 ± 1.056	-2.980	2.67 ± 1.400	-0.700	2.65 ± 0.997
60°N–90°N	5.566	1.96 ± 0.681	43.670	1.69 ± 1.035	51.400	1.63 ± 1.024	53.230	3.48 ± 1.790

(b) TD3

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	7.53 ± 1.444	22.700	7.42 ± 1.159	23.880	7.54 ± 1.390	22.630	9.74 ± 3.330
60°S–30°S	7.768	5.99 ± 2.663	15.800	5.51 ± 2.616	22.590	5.00 ± 2.586	29.790	7.12 ± 3.800
30°S–0°	2.730	3.84 ± 2.378	-1.030	3.65 ± 2.640	3.990	3.32 ± 1.380	12.690	3.80 ± 1.713
0°–30°N	3.746	7.52 ± 2.875	-165.300	7.54 ± 3.263	-166.260	5.94 ± 2.701	-109.610	2.83 ± 1.247
30°N–60°N	6.398	6.55 ± 1.837	-29.630	6.85 ± 2.390	-35.450	7.02 ± 2.581	-38.880	5.06 ± 2.602
60°N–90°N	5.566	3.94 ± 2.302	27.250	4.00 ± 1.708	26.240	4.49 ± 1.536	17.170	5.42 ± 2.721

(c) TQC

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	8.27 ± 0.378	8.700	10.78 ± 4.522	-18.950	8.26 ± 0.303	8.840	9.06 ± 1.549
60°S–30°S	7.768	7.51 ± 0.869	5.780	10.76 ± 4.459	-35.000	7.62 ± 0.729	4.450	7.97 ± 1.459
30°S–0°	2.730	2.60 ± 0.718	-15.960	7.27 ± 5.706	-224.680	3.02 ± 0.490	-34.920	2.24 ± 0.868
0°–30°N	3.746	2.84 ± 0.774	-54.750	9.67 ± 12.637	-426.670	3.90 ± 0.434	-112.690	1.84 ± 0.561
30°N–60°N	6.398	3.59 ± 1.179	-48.070	11.05 ± 17.379	-355.740	4.13 ± 0.461	-70.350	2.42 ± 0.706
60°N–90°N	5.566	3.35 ± 1.003	-43.330	11.17 ± 18.477	-378.590	3.36 ± 0.394	-43.800	2.33 ± 0.767

Table B.6: Zonal-band errors for `ebm-v3-optim-L-20k-a6`. Each subtable reports mean ± std and relative gain % versus `ebm-v1` for three regimes `fed05`, `fed10` and `nofed`, along with a comparison against the static baseline `climlab`

(a) DDPG

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	7.01 ± 2.782	14.340	7.22 ± 1.399	11.800	7.23 ± 1.652	11.690	8.19 ± 3.433
60°S–30°S	7.768	4.34 ± 3.372	31.000	3.69 ± 1.102	41.450	8.08 ± 4.816	-28.370	6.30 ± 4.019
30°S–0°	2.730	1.25 ± 0.695	81.440	1.22 ± 0.526	81.890	19.92 ± 4.845	-194.680	6.76 ± 3.013
0°–30°N	3.746	1.48 ± 0.617	67.830	1.71 ± 1.218	62.710	17.39 ± 5.346	-278.850	4.59 ± 2.068
30°N–60°N	6.398	1.51 ± 0.710	43.220	1.57 ± 0.796	40.880	5.92 ± 6.272	-123.010	2.65 ± 0.997
60°N–90°N	5.566	1.17 ± 0.442	66.490	1.39 ± 0.680	60.240	1.76 ± 1.176	49.480	3.48 ± 1.790

(b) TD3

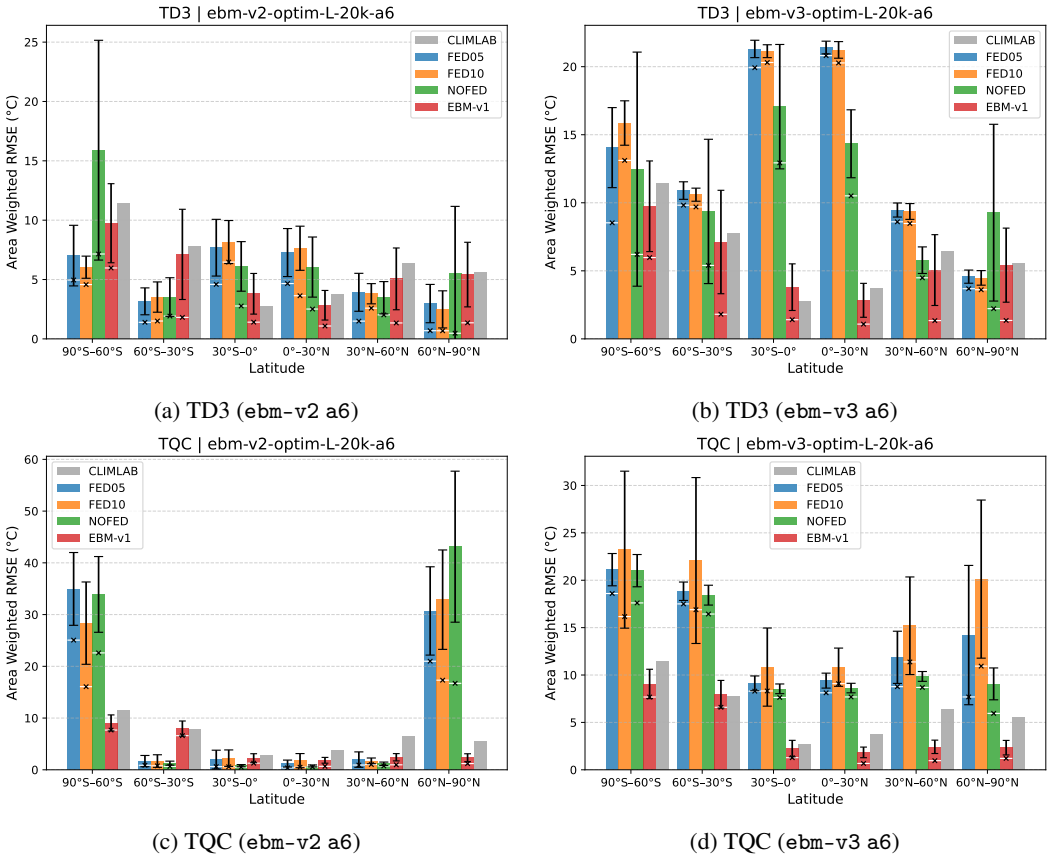
	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	14.05 ± 2.941	-44.230	15.86 ± 1.632	-62.780	12.47 ± 8.601	-28.000	9.74 ± 3.330
60°S–30°S	7.768	10.90 ± 0.644	-53.120	10.59 ± 0.481	-48.860	9.36 ± 5.302	-31.570	7.12 ± 3.800
30°S–0°	2.730	21.30 ± 0.639	-460.730	21.14 ± 0.473	-456.380	17.06 ± 4.570	-349.090	3.80 ± 1.713
0°–30°N	3.746	21.41 ± 0.467	-655.550	21.23 ± 0.611	-649.370	14.34 ± 2.496	-405.970	2.83 ± 1.247
30°N–60°N	6.398	9.47 ± 0.516	-87.290	9.36 ± 0.581	-85.160	5.78 ± 0.977	-14.340	5.06 ± 2.602
60°N–90°N	5.566	4.57 ± 0.484	15.600	4.48 ± 0.536	17.280	9.27 ± 6.494	-71.180	5.42 ± 2.721

(c) TQC

	climlab	fed05		fed10		nofed		ebm-v1
		Mean ± Std	Gain %	Mean ± Std	Gain %	Mean ± Std	Gain %	
90°S–60°S	11.453	21.12 ± 1.703	-133.090	23.23 ± 8.282	-156.370	21.01 ± 1.702	-131.930	9.06 ± 1.549
60°S–30°S	7.768	18.84 ± 0.971	-136.340	22.08 ± 8.752	-177.000	18.43 ± 1.047	-131.180	7.97 ± 1.459
30°S–0°	2.730	9.12 ± 0.779	-307.220	10.84 ± 4.123	-383.720	8.54 ± 0.511	-281.410	2.24 ± 0.868
0°–30°N	3.746	9.41 ± 0.789	-412.530	10.82 ± 2.018	-489.610	8.62 ± 0.503	-369.770	1.84 ± 0.561
30°N–60°N	6.398	11.87 ± 2.754	-389.580	15.20 ± 5.150	-527.000	9.85 ± 0.519	-306.400	2.42 ± 0.706
60°N–90°N	5.566	14.21 ± 7.349	-509.050	20.12 ± 8.341	-762.090	9.07 ± 1.682	-288.430	2.33 ± 0.767

B.2 TD3 and TQC Performance Across FedRL EBM Configurations

For TD3 (in Figures B.4 (a) and (b)), performance in `ebm-v2` shows competitive skill in tropical and mid-latitude zones under `fed05`, often matching or exceeding `ebm-v1`. However, variance increases substantially in the polar bands, particularly in the Southern Hemisphere, where sharp gradients appear harder to capture. In `ebm-v3`, TD3 displays more pronounced instability. While `fed05` remains the most stable regime, episodic collapses in high-latitude bands lead to elevated RMSE compared to `ebm-v2`. This suggests that the reduced and region-specific input state in `ebm-v3`, may cause mismatch between hyperparameters tuned for global state inputs (in `ebm-v1`) and the regional profile inputs used in `ebm-v3`, leading to instability in critic ensemble updates.



(a) TD3 (`ebm-v2 a6`)

(b) TD3 (`ebm-v3 a6`)

(c) TQC (`ebm-v2 a6`)

(d) TQC (`ebm-v3 a6`)

Figure B.4: Comparison of zonal skill (areaWRMSE) for TD3 and TQC across `ebm-v2` and `ebm-v3` in 6-agent federated setups averaged with 95% spreads over 10 seeds. White horizontal bars with a cross indicate the best-performing seed for each algorithm. Both setups adopt the same policy architecture and hyperparameters as `ebm-v1`. Skill metrics are presented in Appendix B.1.

TQC (in Figures B.4 (c) and (d)) performs strongly in `ebm-v2` tropical bands under `fed05`, with clear gains over `ebm-v1`. However, the method shows instability in high-latitude zones and wider error bars under `fed10`, indicating a reliance on more frequent synchronisation for stability. In `ebm-v3`, TQC's performance degrades notably in the mid-latitudes, with polar areaWRMSE exceeding that of `ebm-v1` in several cases. The large critic ensemble, which benefits global contexts, may be less effective when regional input profiles and rewards dominate, leading to overfitting or noisy updates. These patterns reaffirm that while both TD3 and TQC can yield strong results under favourable settings, DDPG's simpler architecture appears more robust to the structural changes between `ebm-v2` and `ebm-v3`.

Appendix C Algorithm Pseudocode

C.1 Deep Deterministic Policy Gradient (DDPG)

Algorithm 1 Deep Deterministic Policy Gradient (DDPG)

Input: Gym environment, Total timesteps T, Replay buffer size N, Discount factor γ, Target smoothing coefficient τ, Batch size B, Learning rate η, Exploration noise σ
Initialise: Policy network parameters θ, Q-function network parameters ϕ, target network parameters θ_{targ}, ϕ_{targ}, empty replay buffer $\mathcal{D}$
Pre-Setup: Configure seed and environment variables, prepare environment and logging

for $t = 1$ **to** T **do**
  Observe state s and select action $a = \pi_\theta(s)$
  Add exploration noise $a \leftarrow a + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma)$ if required
  Execute action a and observe next state s', reward r, and termination signal d
  Store transition (s, a, r, s', d) in $\mathcal{D}$
  if $t \geq$ learning_starts **then**
    Sample a minibatch of B transitions (s, a, r, s', d) from $\mathcal{D}$
    Compute target for Q-function update:

$$y(r, s', d) = r + \gamma(1 - d) Q_{\phi_{\text{targ}}}(s', \pi_{\theta_{\text{targ}}}(s'))$$

    Update Q-function by minimising the loss:

$$\phi \leftarrow \phi - \eta \nabla_\phi \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_\phi(s, a) - y(r, s', d)\right)^2$$

    Update policy by one step of gradient ascent:

$$\theta \leftarrow \theta + \eta \nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_\phi(s, \pi_\theta(s))$$

    Soft-update target networks:

$$\theta_{\text{targ}} \leftarrow \tau\theta + (1 - \tau)\theta_{\text{targ}}, \quad \phi_{\text{targ}} \leftarrow \tau\phi + (1 - \tau)\phi_{\text{targ}}$$

  end if
end for

C.2 Twin Delayed DDPG (TD3)

Algorithm 2 Twin Delayed DDPG (TD3)

Input: Gym environment, Total timesteps T, Learning rate η, Replay buffer size N, Discount factor γ, Target smoothing coefficient τ, Batch size B, Policy noise σ_π, Noise clip σ_{clip}, Exploration noise $\sigma_{\text{exploration}}$, Policy update frequency f_π
Initialise: Actor network θ, Critic networks ϕ_1, ϕ_2, Target networks θ_{targ}, $\phi_{targ,1}$, $\phi_{targ,2}$, Empty replay buffer $\mathcal{D}$
Pre-Setup: Configure seed and environment variables, prepare environment and logging

for $t = 1$ **to** T **do**
  Observe state s and select action $a = \pi_\theta(s)$
  Add exploration noise $a \leftarrow a + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_{\text{exploration}})$ if required
  Execute action a and observe next state s', reward r, and done signal d
  Store transition (s, a, r, s', d) in $\mathcal{D}$
  if $t \geq$ learning_starts **then**
    Sample a minibatch of B transitions (s, a, r, s', d) from $\mathcal{D}$
    Compute target actions:

$$a' \leftarrow \pi_{\theta_{targ}}(s') + \text{clip}(\mathcal{N}(0, \sigma_\pi), -\sigma_{\text{clip}}, \sigma_{\text{clip}})$$

    Compute target Q-values:

$$y(r, s', d) \leftarrow r + \gamma(1 - d) \min_{i=1,2} Q_{\phi_{targ,i}}(s', a')$$

    Update critic networks by minimising the loss:

$$\phi_i \leftarrow \phi_i - \eta \nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_{\phi_i}(s, a) - y(r, s', d)\right)^2, \text{for } i = 1, 2$$

    if $t \mod f_\pi = 0$ **then**
      Update actor network by policy gradient:

$$\theta \leftarrow \theta + \eta \nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_{\phi_1}(s, \pi_\theta(s))$$

      Soft update target networks:

$$\theta_{targ} \leftarrow \tau\theta + (1 - \tau)\theta_{targ}, \quad \phi_{targ,i} \leftarrow \tau\phi_i + (1 - \tau)\phi_{targ,i} \text{ for } i = 1, 2$$

    end if
  end if
end for

C.3 Truncated Quantile Critics (TQC)

Algorithm 3 Truncated Quantile Critics (TQC)

Input: Gym environment, Total timesteps T, Replay buffer size N, Discount factor γ, Smoothing coefficient τ, Batch size B, Learning rate η, Number of quantiles N_q, Number of critics N_c, Drop quantiles N_{drop}, Entropy coefficient α, Target entropy coefficient α_{targ}
Initialise: Actor network θ, Critic network parameters $\phi_1, \ldots, \phi_{N_c}$, Target critic network parameters $\phi_{\text{targ},1}, \ldots, \phi_{\text{targ},N_c}$, Replay buffer $\mathcal{D}$
Pre-Setup: Configure seed and environment variables, prepare environment and logging

for $t = 1$ **to** T **do**
  Select action $a \sim \pi_\theta(s)$ based on current policy and exploration strategy
  Execute action a and observe next state s', reward r, and done signal d
  Store transition tuple (s, a, r, s', d) in $\mathcal{D}$
  if $t \geq$ learning_starts **then**
    for $i = 1$ **to** N_c **do**
      Sample a minibatch of B transitions (s, a, r, s', d) from $\mathcal{D}$
      Compute target quantile values for critic $\phi_{target,i}$:

$$y(r, s', d) = r + \gamma(1 - d)\left(Q_{\phi_{\text{targ},i}}(s', \tilde{a}', N_{\text{drop}}) - \alpha \log \pi_\theta(\tilde{a}'|s')\right)$$

      where $\tilde{a}' \sim \pi_\theta(s')$
      Update critic ϕ_i by minimising the quantile Huber loss:

$$\mathrm{L}^{\phi_i} = \frac{1}{N_q} \sum_{k=1}^{N_q} \text{HuberLoss}(Q_{\phi_i}(s_j, a_j, \tau_k) - y_j)$$

      where τ_k are the quantile fractions
    end for
    Update policy by one step of gradient ascent:

$$\theta \leftarrow \theta + \eta \nabla_\theta \frac{1}{|B|} \sum_{s \in B} \left(-\alpha \log \pi_\theta(a|s) + \frac{1}{N_c} \sum_{i=1}^{N_c} Q_{\phi_i}(s, \pi_\theta(s)) \right)$$

    Soft-update target networks:

$$\phi_{\text{targ},i} \leftarrow \tau \phi_i + (1 - \tau)\phi_{\text{targ},i} \text{ for } i = 1, 2, ..., N_c$$

    Optionally adjust α based on entropy targets:

$$\alpha \leftarrow \alpha + \eta \nabla_\alpha \frac{\alpha}{|B|} \sum_{s \in B} (\log \pi_\theta(a|s) + \alpha_{\text{targ}})$$

  end if
end for