Department of Informatics
Chair of Robotics, Artificial Intelligence and Real-time Systems

A Multi-objective Approach to Safe Reinforcement Learning for Smart Grids

David Fischer, Jonas Fischer, Nikolas Kirschstein

Practical Course *Machine Learning for Smart Grids*, Summer Term 2025

Advisors:	Hannah Markgraf, Michael Eichelbeck
Supervisor:	Prof. Dr.-Ing. Matthias Althoff
Submission:	19 August 2025

A Multi-objective Approach to Safe Reinforcement Learning for Smart Grids

David Fischer
Student no. 03797814
Technical University of Munich
da.fischer@tum.de

Jonas Fischer
Student no. 03798824
Technical University of Munich
jonasetienne.fischer@tum.de

Nikolas Kirschstein
Student no. 03782928
Technical University of Munich
nikolas.kirschstein@tum.de

***Abstract*—Building energy management systems commonly employ Reinforcement Learning (RL) controllers to overcome the dependence on accurate forecasts of key quantities by optimal controllers like Model Predictive Control (MPC). The challenge for RL approaches lies in informing the controller about real-world safety constraints. Existing work simplistically treats safety violations as a penalty to be added to the scalar reward term. In this work, we instead cast the problem as the true Multi-Objective Reinforcement Learning (MORL) task that it actually is by making the reward vector-valued and explore the performance of different MORL algorithms on a smart household scenario. Our results suggest that MORL algorithms can effectively learn the Pareto front of cost and safety constraints and outperform model-based MPC controllers, while allowing flexible selection of operating points without the need for retraining.**

I. Introduction

Renewable energy production in power grids requires building energy management systems to employ optimal strategies for buying and selling electricity. If perfect forecasts of decision factors like weather or electricity prices are available, Model Predictive Control (MPC) solves the problem optimally. However, in reality, the inaccuracy and uncertainty of forecasts reduces the performance of MPC below the theoretical optimum. To deal with the uncertainty and adapt to the highly dynamic environment, Reinforcement Learning (RL) controllers have been shown to outperform MPC in terms of overall electricity cost.

Yet, RL controllers trained on electricity cost as negative reward are intrinsically unable to satisfy hard safety constraints like keeping battery charge levels within a certain range. Hence, most approaches employ a safety shield that checks the actions proposed by the RL controller for safety violations and, if necessary, performs interventions by replacing the proposed unsafe action by a similar but safe one, thus guaranteeing safety. However, the surrogate action is most likely not optimal within the subspace of safe actions. Thus, the RL controller needs to be informed about safety interventions to learn to avoid safety violations while still minimising energy cost.

To this end, existing work straightforwardly adds a penalty term to the reward function that encapsulates the magnitude of the necessary safety intervention, if one occurred. While practicable, this approach has two major drawbacks:

1) The minimisation of safety interventions is not necessarily aligned with the minimisation of cost, e.g., being able to fully (dis)charge a battery would allow lower overall energy cost but degrade battery life.
2) The relative weighting of cost and safety must be specified a priori via the penalty factor and cannot be changed later. This necessitates expensive hyperparameter tuning of this penalty factor and makes it impossible to dynamically choose the weighting on inference time.

To address these issues, we propose to model the building energy management problem as a Multi-Objective Reinforcement Learning (MORL) task, with electricity cost and safety interventions being two separate explicit minimisation objectives, leading to a multi-dimensional reward signal. Note that this formulation conceptually models the actual problem more directly than setting than optimising a safety-penalised cost.

As established MORL algorithms are readily available through the library MORL-Baselines (Felten et al., 2023), we compare their performance on a smart household example scenario with the well-established single-objective PPO algorithm, as well as with MPC. The key performance indicators are electricity cost as well as magnitude and number of safety interventions. We observe that MORL algorithms can successfully learn the Pareto front for cost-safety trade-offs and outperform model-based MPC controllers. Concave-Augmented Pareto Q-learning particularly demonstrates strong safety performance with consistent results. While MORL methods do not surpass the best naive fixed-trade-off PPO baseline, they enable flexible selection of operating points before deployment without the need for retraining.

The remainder of this report is structured as follows: Section II briefly surveys related fields and relevant prior work, Section III explains our methodology in detail, Section IV discusses our experimental results and insights, and Section V concludes the report with a summary and an outlook on future research directions.

II. Related Work

A. Economic Dispatch in Smart Grids

Our problem setting of building energy management can be embedded into the broader scope of economic dispatch in smart grids. Classical optimisation approaches to economic dispatch include model predictive control (Santillán-Lemus et al., 2019), mixed-integer programming (Malysz et al., 2014), and distributed primal-dual optimisation (Zhang et al., 2013). All of these rely on strong assumptions like precise model knowledge and perfect forecast availability, which are not fulfilled in practice, resulting in suboptimal dispatch strategies.

B. RL for Smart Grids

Current state-of-the-art approaches to economic dispatch in smart grids successfully employ reinforcement learning (RL) controllers, such as the safety-aware dispatch control architecture by Eichelbeck, Markgraf, and Althoff (2022). They provide the highly flexible software framework *CommonPower* (Eichelbeck et al., 2024) which we base our work on. The most widely adopted single-objective RL algorithm for smart grid control problems appears to be Proximal Policy Optimisation (PPO; Schulman et al., 2017), which consequently constitutes our single-objective baseline.

C. MORL Algorithms

MORL is a very recent research field aiming at extending RL to the multi-objective case (Hayes et al., 2022). Felten et al. (2023) recently collected the most established MORL algorithms in a unified library called MORL-Baselines. The applicable algorithms for our setting with continuous observation and action spaces are Prediction-Guided Multi-Objective RL (PGMORL; Xu et al., 2020), Pareto Conditioned Networks (PCN; Reymond et al., 2022), Concave-Augmented Pareto Q-Learning (CAPQL; Lu et al., 2023), Generalised Policy Improvement with Prioritised Dynamics (GPI-PD; Alegre et al., 2023), and MORL based on Decomposition (MORL/D; Felten et al., 2024). Due to computational considerations, we restrict our study to PCN and CAPQL and encourage future work to evaluate the other three algorithms as well.

III. Methodology

A. Optimisation Problem

As mentioned in Section I, MORL uses vector-valued rewards to capture different objectives separately. In this work, we consider two optimisation objectives: *electricity costs* cost and *safety adjustments* safety, which together define the reward signal $\boldsymbol{r}_t$ at time step t:

$$\boldsymbol{r}_t := -\begin{bmatrix}\mathrm{cost}_t\\ \mathrm{safecty}_t\end{bmatrix}. \qquad (1)$$

A trivial MORL approach is to scalarise the vector reward (1) back into a single dimension and then apply standard single-objective RL (SORL) algorithms, such as PPO:

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix}^{\top} \boldsymbol{r}_t = -\big(\alpha \cdot \mathrm{cost}_t + \beta \cdot \mathrm{safety}_t\big), \qquad (2)$$

where α and β are fixed scalarisation weights that balance the importance of each objective. Note how this is equivalent to the conventional cost reward with a safety penalty when dividing Equation (2) by α. Hence, we employ PPO on thus scalarised reward as a baseline to compare to.

Our main focus lies on algorithms that use the vectorised reward directly (Eq. 1) and learn the *Pareto front*, the set of non-Pareto-dominated policies. Reymond et al. (2022) define that a policy π *Pareto-dominates* policy π' if the value function $\mathbf{V}^{\pi}$ is weakly larger than $\mathbf{V}^{\pi'}$ across all objectives and strictly larger for at least one objective. A policy π is *Pareto-optimal* if it is not Pareto-dominated by another policy π'. The *Pareto front* includes all policies that are *Pareto-optimal*. Figure 1 visualises the Pareto front approximation in a two-objective setting.

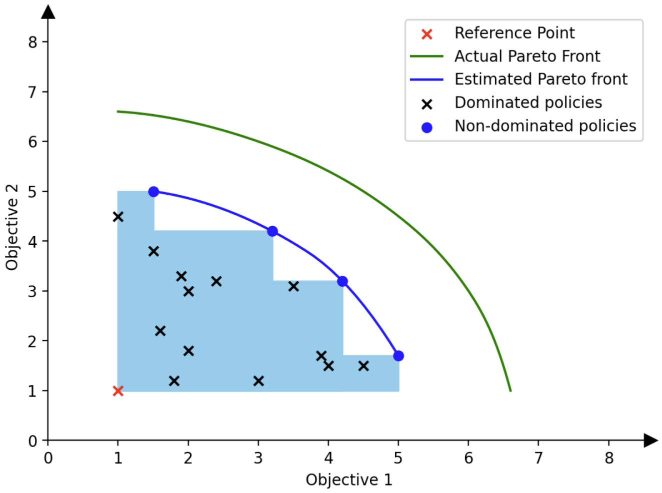

Fig. 1. Illustration of Pareto-front in a two objective setting. The green curve represents the actual Pareto front, while the blue curve shows the estimated Pareto front obtained from the non-dominated policies (blue points). Dominated policies are shown as black crosses. The red cross marks the reference point used to compute the *hypervolume*, the blue-shaded area dominated by the Pareto front and bounded by the reference point.
Source: https://epublications.vu.lt/object/elaba:192058320/192058320.pdf

B. Employed Algorithms

1) Pareto Conditioned Network (PCN): Reymond et al. propose PCN, a single neural network designed to learn all policies on the Pareto front. Given the current state s, PCN is conditioned on the desired return $\hat{\mathbf{R}}$ to be achieved by the end of the episode, as well as the remaining number of time steps $\hat{h}$ until this reward should be reached.

Training: PCN learns to output an action a_t at time step t given the tuple $(s_t, \hat{h}_t, \hat{\mathbf{R}}_t)$ in a supervised learning fashion. The network requires a dataset that is collected and updated throughout training. Initially, random policies are executed for the first few episodes to collect the initial trajectories for the training set, with desired horizon $h_t = T - t$ and desired return $\hat{\mathbf{R}}_t = \sum_{i=1}^{T} \gamma^i r_i$, where T is the length of the trajectory. During training, PCN gradually improves its dataset by starting from existing non-dominated returns, slightly increasing one objective within a realistic range, and generating new trajectories based on these adjusted targets. This incremental conditioning keeps the inputs close to known data while pushing performance boundaries, enabling the network to generalise to higher returns. Exploration during training is encouraged by sampling actions stochastically from the network's output distribution. PCN maintains a fixed-size dataset that balances relevance to the current coverage set with diversity across the objective space. It scores trajectories by combining their proximity to non-dominated solutions with a crowding distance measure, pruning those that are less relevant or overly clustered. The network is trained using stochastic gradient descent with a mean squared error loss for discrete action spaces.

Deployment: During deployment, we predict the action a_t using the cumulative reward up to the current time step t as the desired return $\hat{\mathbf{R}}_t$, and the desired horizon $\hat{h}_t$ as the remaining simulation time:

$$\hat{\mathbf{R}}_t = \sum_{i=0}^{t} \mathbf{r}_i, \tag{3}$$

$$\hat{h}_t = T - t, \tag{4}$$

where T is the total number of simulation steps.

2) Concave-Augmented Pareto Q-Learning (CAPQL): CAPQL can be considered a multi-objective extension of the Soft Actor-Critic (SAC) algorithm. Lu et al. (2023) show that value functions $\mathbf{V}$ are not arbitrarily complex but rather convex, which implies that linear scalarisation methods can potentially learn all *Pareto-optimal* policies. CAPQL learns an extended Q-network $Q(s, a, \boldsymbol{w})$ and trains it to approximate all optimal policies corresponding to the scalarised multi-objective reinforcement learning (MORL) problem for all $\boldsymbol{w} \in \Phi \subseteq W^+$.

To encourage *Pareto-optimal* policies, CAPQL augments the scalarised reward with an entropy term $\mathcal{H}(\pi)$, yielding a modified reward:

$$\begin{aligned} \pi(\cdot; \boldsymbol{w}) &= \arg \max_{\pi'(\cdot;\boldsymbol{w})} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R'(s_t, a_t; \boldsymbol{w})\right] \\ \text{with } R'(s_t, a_t; \boldsymbol{w}) &= \boldsymbol{w}^\top R(a_t, s_t) + \alpha \mathcal{H}(\pi'(s_t; \boldsymbol{w})). \end{aligned} \tag{5}$$

where the first term $\boldsymbol{w}^\top R(s_t, a_t)$ scalarises the multi-objective reward according to $\boldsymbol{w}$, and the second term $\alpha\mathcal{H}(\pi')$ encourages exploration.

Training: During training, $\boldsymbol{w}$ is randomly sampled and normalised to ensure $||\boldsymbol{w}||_1 = 1$ for each environment step. The Q-network Q_θ and the target network $\hat{Q}_{\hat{\theta}}$ are conditioned on the sampled $\boldsymbol{w}$, and updates are performed similar to SAC. The Q-loss is computed using the scalarised reward R' (Eq 5), which ensures that the network learns the scalarised Q-values corresponding to the given $\boldsymbol{w}$.

Deployment: During deployment, we try to predict the next action while dynamically adapting the safety weight based on safety adjustments, which indicate an intervention was required at a given step. The exponential moving average (EMA) of these adjustments is computed as

$$\mathrm{EMA}_t = 0.9 \cdot \mathrm{EMA}_{t-1} + 0.1 \cdot \mathbb{1}\{\mathrm{safetyAdj}_t > 0\}, \tag{6}$$

with threshold $C = 0.1 \cdot 0.9^{10}$, representing the EMA after one intervention followed by ten non-interventions. The lower-bound safety weight is updated as

$$\underline{w}_t = \begin{cases} 0 & t = 0, \\ \underline{w}_{t-1} + 0.01 & \mathrm{safetyAdj}_t > 0, \\ \underline{w}_{t-1} & \text{otherwise}, \end{cases} \tag{7}$$

and the effective safety weight is

$$w_t = \begin{cases} 0.5 & \mathrm{EMA}_t > C, \\ \underline{w}_t + \frac{\mathrm{EMA}_t}{C}(0.5 - \underline{w}_t) & \mathrm{EMA}_t \leq C. \end{cases} \tag{8}$$

The deployed safety weight vector is

$$\boldsymbol{w}_t = \begin{bmatrix} 1 - w_t \\ w_t \end{bmatrix}. \tag{9}$$

This dynamic adjustment ensures that the agent balances exploration with safety, using the safety weight to modulate actions while minimising the risk of interventions during deployment.

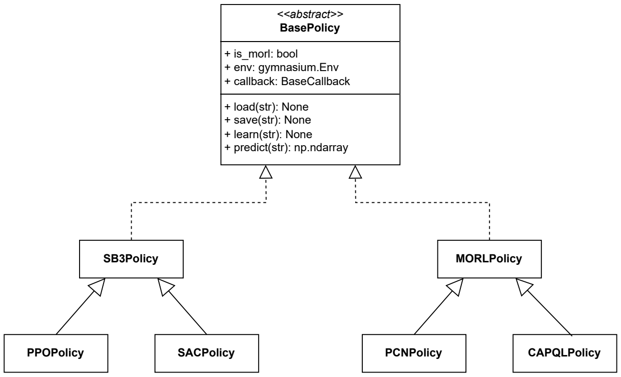

Fig. 2. UML class diagram of our policy abstraction.

C. CommonPower Integration

To integrate the MORL-Baselines library into the CommonPower framework, we introduced a policy abstraction `BasePolicy`. This unified interface enables algorithms from both the Stables Baselines 3 (`SB3Policy`) and MORL-Baselines (`MORLPolicy`) libraries to be seamlessly incorporated into the existing training and deployment pipelines with minimal redundancy (cp. Figure 2). This design allows any algorithm from the MORL-Baselines library to be wrapped by subclassing `MORLPolicy`, enabling direct use within CommonPower without changes to the surrounding infrastructure..

The reward function was modified to output a vector, which is scalarised in the case of SORL algorithms and used in vector form for MORL algorithms. A dedicated MORL environment

MORLEnv was implemented, and the training and deployment procedures were adapted to automatically handle both SORL and MORL settings.

During development, we identified and resolved multiple bugs in MORL-Baselines via pull requests[1], which were subsequently merged into the official repository, contributing to the release of version v1.2.0[2].

D. Household Energy Management Scenario

We evaluate the MORL algorithms in a household energy management scenario comprising the following components:

- grid connection,
- photovoltaic generation,
- electric vehicle, and
- battery storage system.

The environment includes imperfect forecasts of load, photovoltaic generation, and electricity prices. The primary objective is to minimise electricity costs. In this setup, only the battery storage system is controllable, and forecasts span a 6-hour horizon based on data from the previous day.

At each time step t, the MORL algorithm proposes a battery action $\boldsymbol{a}$, which is evaluated using a safety shield (Eichelbeck et al., 2024). If the proposed action was unsafe, the safety shield applies *action projection*: the unsafe action is replaced by the closest safe action $\tilde{\boldsymbol{a}}$, and the 2-norm of the projection vector constitutes the safety penalty $\text{safety}_t := \|\tilde{\boldsymbol{a}} - \boldsymbol{a}\|_2$ for that timestep (cp. Equation (1)).

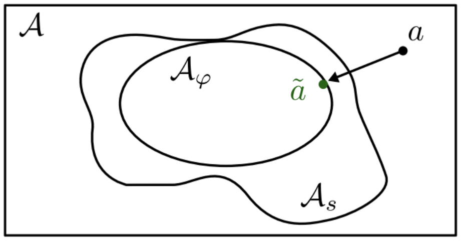

Fig. 3. Projection of unsafe action $\boldsymbol{a} \in \mathcal{A}$ onto closest safe action $\tilde{\boldsymbol{a}} \in \mathcal{A}_\varphi$. Source: Krasowski et al. (2023)

IV. Results & Discussion

All the evaluation was conducted on the household energy management scenario described in Section III-D, using given electricity consumption and PV generation data from 2016. Models were trained on the July 2016 data (Section III) and deployed for a full year to assess generalisation. We state the cumulative annual cost, accumulated penalty, and number of interventions as evaluation metrics. An optimal controller obtained via MPC serves as a baseline.

A. Overall Comparison

Table I summarises the deployment outcomes. CAPQL achieved a competitive cumulative cost (644.4€), outperforming the optimal controller (662.3€). PPO-based agents achieved even lower costs (around 625–629€), but this came at the expense of substantially higher penalties and interventions, especially for the 80/20 scalarisation. In contrast, CAPQL maintained low penalties and interventions, which suggests a better trade-off between cost minimisation and policy robustness. PCN achieved costs close to the baseline but suffered from extremely high penalties and interventions.

B. PPO

The PPO agents achieved the lowest costs overall, ranging between 624.9€ and 629.4€. However, the extent of penalties and interventions varied strongly with the preference vector. The 20/80 weighted agent had the lowest penalties among PPO cases but still averaged 318 interventions. The 50/50 variant achieved the lowest cost but had more than 1,700 interventions. The 80/20 agent was the least stable, with more than 5,000 interventions and penalties being nearest to 100€. These results highlight PPO's sensitivity to scalarisation weights and that additional regularisation or safety layers are needed to deploy.

C. CAPQL

CAPQL showed the best tradeoff between objectives. While its mean cost advantage over the baseline was moderate (approximately 18€), safety interventions were essentially negligible in size (0.05 ± 0.09) and minimal in number (4.8 ± 5.3). This indicates CAPQL properly got hold of the intervention signal during training, leading to a stable policy for deployment.

D. PCN

Despite being designed for multi-objective optimisation, PCN could not handle penalties in an effective manner. Deployment led to an average of more than 7,400 penalties and interventions in the thousands range. The total cost was about similar to the optimal controller. But the instability due to overly violated constraints shows that the PCN configuration used was not suitable for this problem. One contributing factor could be the choice of desired return and horizon (set to the full year).

E. Training and Deployment Analysis

Training curves throughout the July 2016 window for a representative sample of seeds are shown in Figure 4. CAPQL and PPO converged gradually, with PCN varying more. The evaluation for deployment over a whole year (Figure 6) shows the divergence between methods: policies learned by PPO lowered costs but resulted in regular interventions, while CAPQL achieved a more balanced cost trajectory closer to the baseline.

The final deployment cost–penalty trade-off is shown as a 2D scatter plot of total cost against penalties in Figure 8. The CAPQL is almost on the Pareto front in this case. The PPO

[1] https://github.com/LucasAlegre/morl-baselines/pulls?q=is%3Apr+author%3Ankirschi

[2] https://github.com/LucasAlegre/morl-baselines/issues/161#issuecomment-3160178582

TABLE I

Deployment results, reported as mean ± std over five RNG seeds.

Method	Cumulative Energy Cost	Cumulative Intervention Size	Cumulative #Interventions
Optimal Controller (MPC)	662.32	0.00	0.0
PCN	661.17 ± 0.36	3434.02 ± 6283.0	7436.4 ± 1156.2
CAPQL	644.41 ± 21.61	0.05 ± 0.09	4.8 ± 5.3
PPO 20/80	629.39 ± 9.11	0.76 ± 0.93	318.4 ± 413.7
PPO 50/50	624.99 ± 8.95	11.77 ± 10.18	1730.2 ± 1430.2
PPO 80/20	627.94 ± 13.37	99.96 ± 50.70	5011.2 ± 1464.6

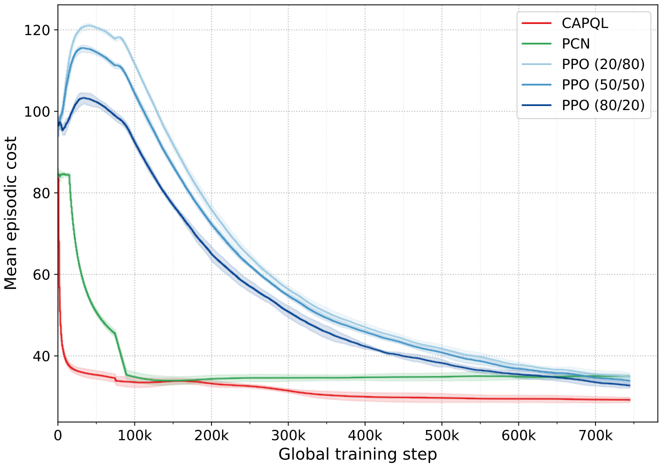

Fig. 4. **Training performance over July 2016.** We plot the *mean episodic cost* (ep_cost_mean) versus training progress (global_step on the x-axis; ticks every 100k). For each method (CAPQL, PCN, and PPO with weights 20/80, 50/50, 80/20), the solid curve is the *seed mean* and the shaded band denotes ±1 standard deviation across five seeds (1–5); a 15-step moving average is applied for readability. Runs are the finished, 1,000-episode configurations (one per seed). *Lower is better:* CAPQL and PCN reduce cost rapidly and then plateau. PPO variants decrease steadily; at convergence, PPO 80/20 attains the lowest cost, PPO 50/50 is intermediate.

runs are distributed along the cost–penalty trade-off. They are cheaper but less secure, some more secure but slightly more costly, and PCN is dominated quite clearly (on both axes).

F. Discussion

Overall, the results confirm that MORL controllers can outperform the Optimal Controller in terms of annual cost under imperfect forecasts, but achieving this robustly requires careful consideration of safety mechanisms and objective scalarisation. CAPQL was the most encouraging of the alternatives in this scenario, trading off between cost effectiveness and low interventions and penalties. PPO saw the greatest raw cost savings but was very sensitive to the weights applied, which is a concern for stability. PCN as it stands the form was not competitive. The results highlight the importance of cost minimisation balance with operational limitation in reinforcement learning for smart-grids.

V. Conclusion & Future Work

In this work, we have demonstrated the effectiveness of multi-objective reinforcement learning (MORL) algorithms in discovering the Pareto front for a cost-safety trade-off in smart grid control. Our results indicate that MORL algorithms can outperform the MPC controller, yet they still fall short of the naive approach that employs a fixed trade-off with PPO.

CAPQL achieves strong performance with respect to both safety score and the frequency of safety interventions. Across both metrics, it records the lowest values and the smallest standard deviation over five random seeds, highlighting its consistency and reliability.

Moreover, both PCN and CAPQL can effectively learn the Pareto front, enabling practitioners to select an appropriate trade-off between cost and safety prior to deployment, without the need for retraining.

Overall, our findings underscore the potential of MORL algorithms to provide flexible and robust solutions for multi-objective decision-making in smart grid environments.

Due to limited computational resources, we were unable to perform hyperparameter optimization for the MORL algorithms. Future research could explore various hyperparameter settings, which may lead to improved performance of the MORL methods.

Additionally, further studies could investigate other MORL algorithms available in the MORL-Baselines library, taking advantage of the existing integration to broaden the scope of comparison.

This work is based solely on data from the year 2016. To enhance model robustness, future work should incorporate data from multiple years, as weather conditions exhibit high variability.

Finally, the experiments presented in this study were limited to a single household scenario. Future research should extend this work by evaluating the proposed approaches across a wider range of household and non-household scenarios to better demonstrate their applicability in diverse, real-world environments.

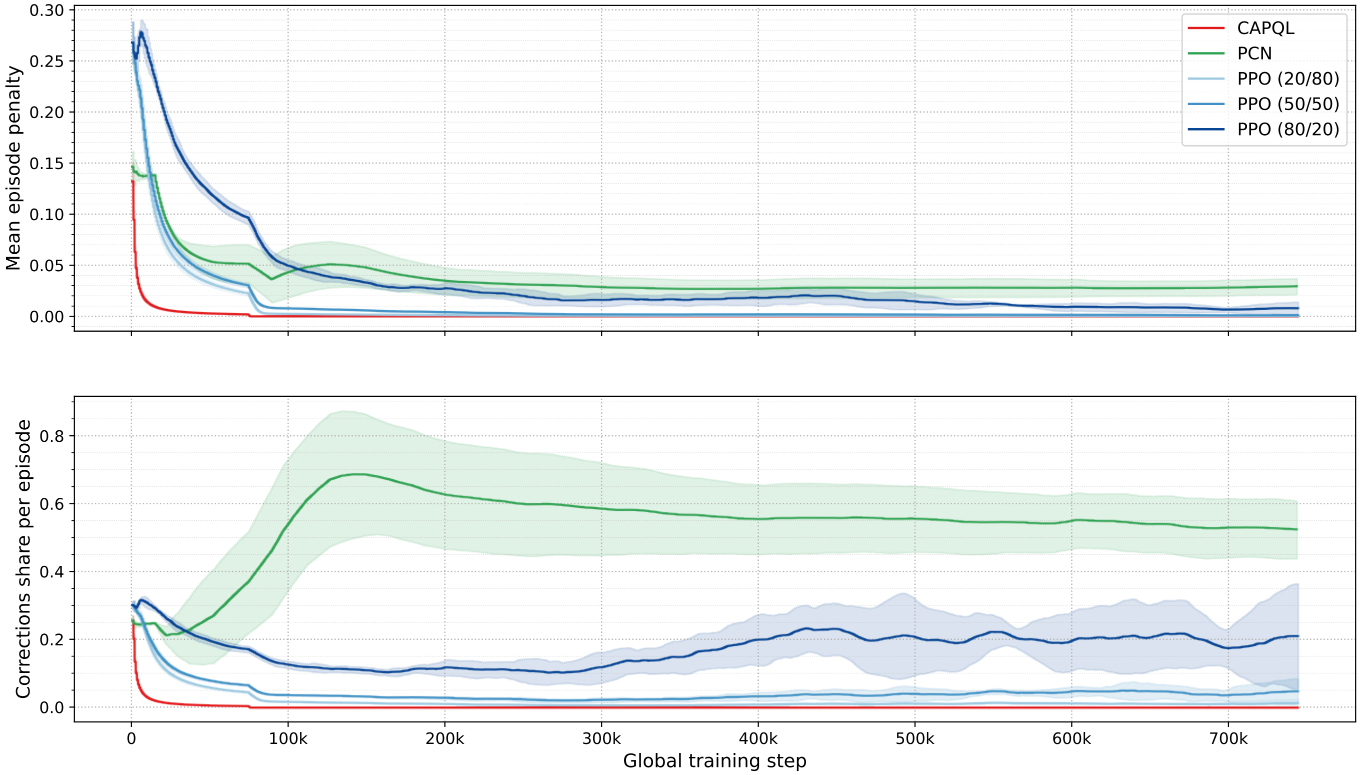

Fig. 5. **Training safety metrics.** *Top Plot - Mean episode penalty:* The y-axis shows the average penalty per episode (`safety/ep_penalty_mean`); lower is better. *Bottom Plot - Corrections share:* The y-axis shows the fraction of steps per episode that triggered the safety shield (`safety/ep_corrections_share_mean`); lower is better. For each method, the solid curve is the *seed mean* and the shaded band is ±1 standard deviation across five seeds (1–5); a 15-step moving average is applied for readability. The x-axis is `global_step`. CAPQL rapidly drives both measures near zero; PPO variants also reduce penalties and interventions but settle at different steady-state levels (20/80 lowest, 80/20 highest); PCN shows the highest and most persistent intervention rate and penalty throughout training.

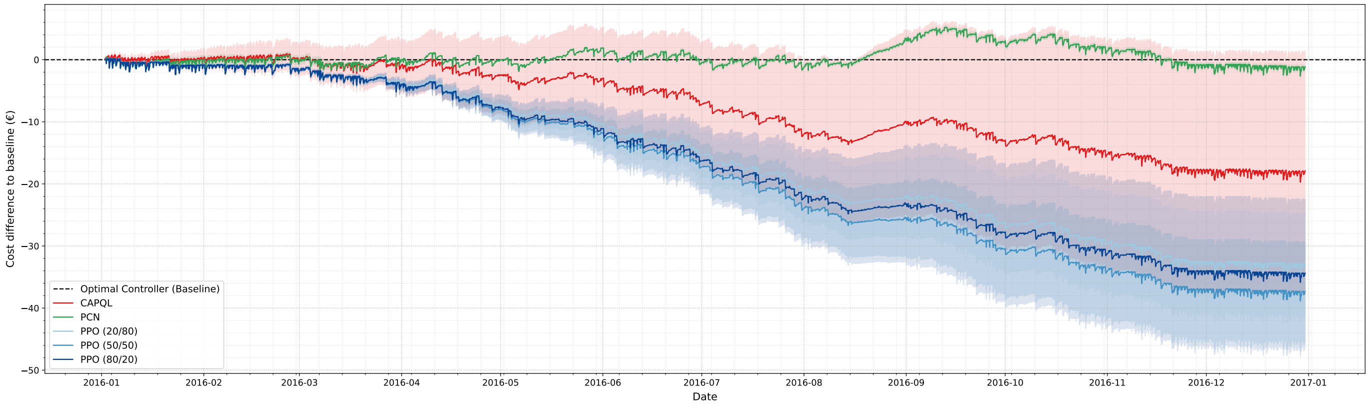

Fig. 6. **Deployment performance over 2016.** The y–axis shows the *cost difference to the Optimal Controller (baseline)* in euro, computed at each deployment hour (lower is better; the dashed line marks the baseline at $\Delta\text{cost} = 0$). For each method (CAPQL, PCN, and PPO with weights 20/80, 50/50, 80/20), the solid curve is the *seed mean* and the shaded band is ±1 standard deviation across five seeds (1–5). At every time step we form "method mean cost minus baseline mean cost" and propagate uncertainty via quadrature: $\sigma_\Delta^2 = \sigma_{\text{method}}^2 + \sigma_{\text{baseline}}^2$. The x–axis shows calendar time for 2016 (monthly ticks). Persistent forecasts are used with a 6 h horizon and a 24 h look-back.

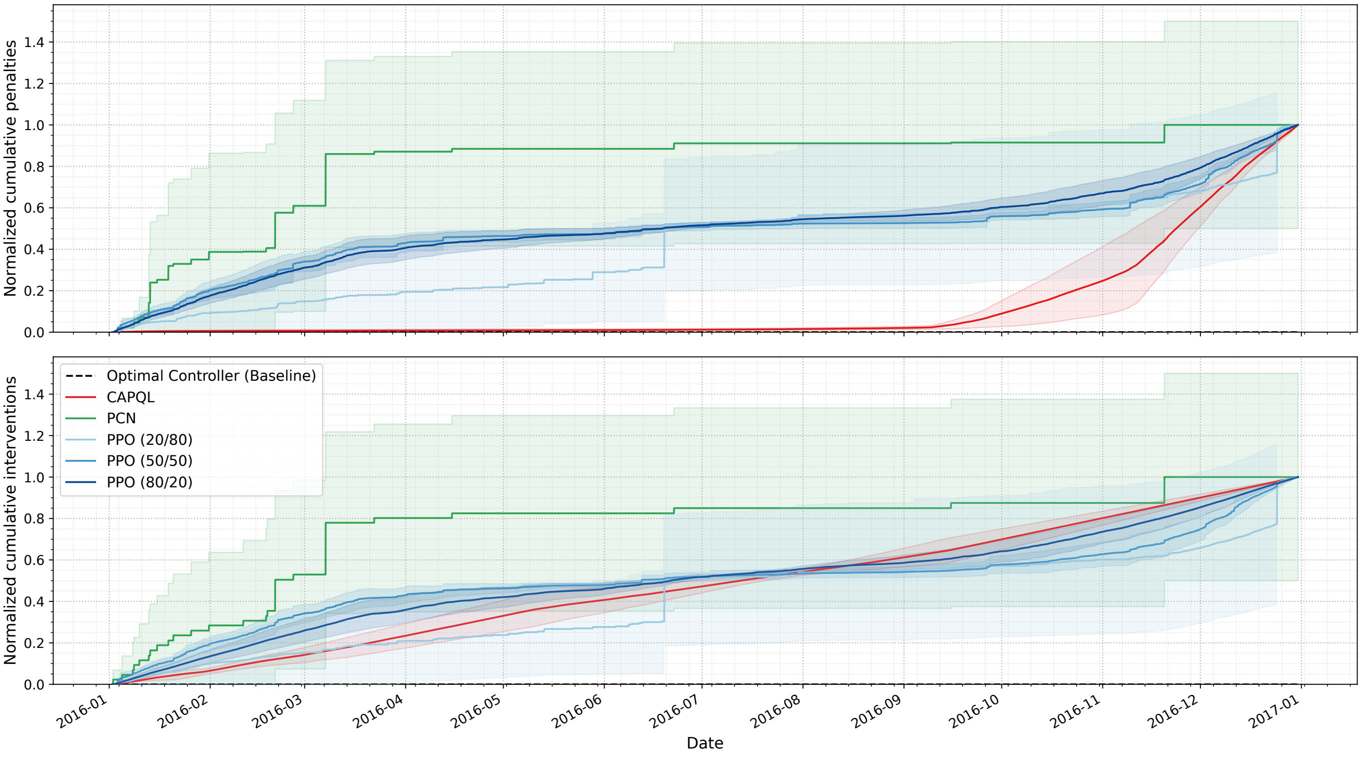

Fig. 7. **Deployment safety over 2016.** *Top Plot - Normalized cumulative penalties:* for each seed, we take the running sum of penalties and divide by that seed's final total so that trajectories lie in $[0, 1]$; seeds with zero total penalties remain at 0. We then average the normalized curves across seeds and display the seed mean (solid) with ± 1 standard deviation (shaded). *Bottom Plot - Normalized cumulative interventions:* defined analogously for Projection-Safeguard interventions (baseline remains at 0 if it never intervenes). This normalization highlights *how* penalties/interventions accumulate over the year, independent of absolute magnitude. The x–axis shows calendar time.

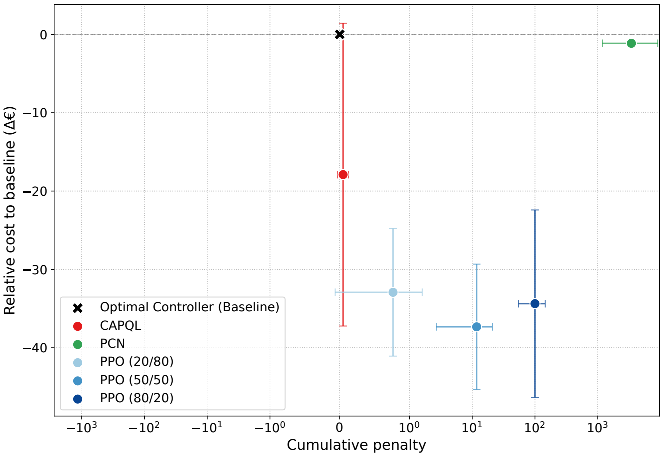

Fig. 8. **Deployment outcomes as a 2D trade-off between penalty and cost.** Each marker summarises a method over all seeds at the final deployment hour. The x-axis is the *cumulative penalty* (symlog scale with a linear core so 0 is shown; larger is worse). The y-axis is the *relative cost to the baseline* in euro, $\Delta\,\text{cost} = \text{cost}_{\text{method}} - \text{cost}_{\text{baseline}}$ (lower is better; 0 means equal to baseline). Points are seed means; horizontal and vertical error bars are standard deviations across seeds. The vertical uncertainty for Δ cost uses the same quadrature propagation as in Fig. 6.

References

Lucas N. Alegre, Ana L. C. Bazzan, Diederik M. Roijers, Ann Nowé, and Bruno C. da Silva. Sample-Efficient Multi-Objective Learning via Generalized Policy Improvement Prioritization. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, pages 2003–2012, Richland, SC, May 2023. International Foundation for Autonomous Agents and Multiagent Systems.

Michael Eichelbeck, Hannah Markgraf, and Matthias Althoff. Contingency-constrained economic dispatch with safe reinforcement learning. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 597–602, December 2022.

Michael Eichelbeck, Hannah Markgraf, and Matthias Althoff. CommonPower: A Framework for Safe Data-Driven Smart Grid Control, 2024.

Florian Felten, Lucas N. Alegre, Ann Nowé, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno Castro da Silva. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023.

Florian Felten, El-Ghazali Talbi, and Grégoire Danoy. Multi-Objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework. *Journal of Artificial Intelligence Research*, 79:679–723, February 2024.

Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, April 2022.

Hanna Krasowski, Jakob Thumm, Marlon Müller, Lukas Schäfer, Xiao Wang, and Matthias Althoff. Provably Safe Reinforcement Learning: Conceptual Analysis, Survey, and Benchmarking, November 2023.

Haoye Lu, Daniel Herman, and Yaoliang Yu. Multi-objective reinforcement learning: Convexity, stationarity and pareto optimality. In *The Eleventh International Conference on Learning Representations*, 2023.

Pawel Malysz, Shahin Sirouspour, and Ali Emadi. An Optimal Energy Storage Control Strategy for Grid-connected Microgrids. *IEEE Transactions on Smart Grid*, 5(4):1785–1796, July 2014.

Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. Pareto conditioned networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, Aamas '22, pages 1110–1118, Virtual Event, New Zealand and Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems.

F. Daniel Santillán-Lemus, Hertwin Minor-Popocatl, Omar Aguilar-Mejía, and Ruben Tapia-Olvera. Optimal Economic Dispatch in Microgrids with Renewable Energy Sources. *Energies*, 12(1):181, January 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017.

Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Yu Zhang, Nikolaos Gatsis, and Georgios B. Giannakis. Robust Energy Management for Microgrids With High-Penetration Renewables. *IEEE Transactions on Sustainable Energy*, 4(4):944–953, October 2013.