

Article

Constraint-Guided Reinforcement Learning for Microgrid Energy Storage Scheduling Motivated by Virtual Power Plant Applications

Jiangnan Li ¹, Jiru Jiang ¹, Mingyu Ou ¹, Junfeng Tan ², Bowei Li ^{3,*}  and Tao Yang ³

¹ Shenzhen Power Supply Company, Shenzhen 518001, China; 13355691784@163.com (J.L.); ray924410571@163.com (J.J.); r7q2w9n4@126.com (M.O.)

² Southern Power Grid Artificial Intelligence Technology Co., Ltd., Guangzhou 510663, China; b8k3m6x1@126.com

³ School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, China; tao.yang@hfut.edu.cn

* Correspondence: 2025030032@mail.hfut.edu.cn

Abstract

The increasing integration of renewable energy and energy storage systems presents challenges for microgrid scheduling under uncertain operating conditions and multiple operational constraints. Conventional reinforcement learning methods may generate infeasible actions, while direct safety clipping can create inconsistencies between policy outputs and executed actions, affecting learning efficiency and scheduling performance. To address these issues, this paper proposes a constraint-guided safety-correction soft actor–critic (CGSC-SAC) algorithm for microgrid energy storage scheduling with a view to virtual power plant applications. The proposed method integrates executed-action replay, critic action clipping, and a correction-guided actor loss to incorporate operational constraints into policy learning and reduce the discrepancy between generated and feasible actions. Comparative experiments, ablation studies, and robustness tests are conducted in a grid-connected photovoltaic–battery system. The results show that CGSC-SAC improves the mean reward from -116.358 to -114.091 and reduces cumulative safety correction by 67.9% compared with Safe-Clip-SAC. Under supply shortages and renewable-energy surplus conditions, safety correction is reduced by 64.7–68.4% and 59.3–64.5%, respectively. Furthermore, an extended IEEE 33-bus case is investigated to evaluate the applicability of the proposed method in a multi-node microgrid. The results demonstrate comparable load satisfaction and no battery state-of-charge violations across the tested scenarios, with computational efficiency suitable for hourly scheduling. These findings indicate that CGSC-SAC improves constraint-aware energy storage scheduling and provides a potential solution for local resource management in virtual power plant applications.



Academic Editor: Dorin Petreus

Received: 11 September 2026

Revised: 12 October 2026

Accepted: 14 October 2026

Published: 18 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: virtual power plant; grid-connected microgrid; energy storage scheduling; safe reinforcement learning; soft actor–critic; action correction

1. Introduction

This study examines local energy storage scheduling toward virtual power plant (VPP) applications. We consider a grid-connected microgrid as a prospective resource unit and focus on its internal battery control under changing operating conditions. The objective is to improve local scheduling while respecting storage and grid exchange constraints.

This application perspective motivates the study of resource-level control; aggregation and supervisory coordination are reserved for subsequent development.

Conventional energy storage scheduling in microgrids primarily relies on model-based optimization to balance operating economy and supply adequacy under storage and grid exchange constraints. In photovoltaic (PV)–battery–utility grid systems [1–3], current charging and discharging decisions change the energy available later. Fluctuating PV output and demand introduce temporal coupling and operational uncertainty. Energy management must therefore coordinate equipment constraints, generation and demand forecasts, and energy allocation across time periods [4,5]. Grover-Silva et al. incorporated PV and load uncertainties into stochastic optimal power flow for day-ahead scheduling of flexible microgrid resources [6]. Kou et al. used distributed economic model predictive control to coordinate energy exchanges among microgrids under uncertain generation and demand [7]. For online scheduling, Lu et al. applied deep forecasting models to rolling energy management in industrial microgrids [8]. Casagrande et al. further combined recurrent neural networks with differentiable optimization for online end-to-end predictive control [9]. These approaches explicitly address physical constraints and intertemporal relationships. However, their dependence on operational models, forecast quality, and online optimization creates challenges in model maintenance, forecast error handling, and real-time computation under changing conditions.

Reinforcement learning learns observation-to-action mappings through environmental interaction, reducing dependence on explicit scheduling models and repeated online optimization. It provides a data-driven approach to sequential storage decisions under uncertainty. Reviews of power systems, demand response, and microgrids identify energy management of batteries and other flexible resources as important applications. They also highlight dependence on state representation, pricing mechanisms, and environmental nonstationarity [10]. Zeng et al. combined approximate dynamic programming with deep recurrent neural networks for real-time storage scheduling under operational uncertainty [11]. Guo et al. formulated microgrid energy management as a Markov decision process and applied proximal policy optimization to continuous state and action spaces [12]. To accommodate more complex scheduling structures, Shuai et al. used branching dueling Q-networks to mitigate the combinatorial action problem associated with multiple storage devices. Cui et al. coordinated networked microgrids through real-time pricing and reinforcement learning, learning pricing policies without known response models. Dorokhova et al. compared action space designs for electric vehicle charging with PV generation. Gao et al. reduced the initial exploration burden by imitating scheduling decisions from a mixed-integer linear programming solver. These studies broaden learning-based scheduling through policy learning, action design, and improved training efficiency. Nevertheless, conventional reinforcement learning primarily optimizes cumulative return and does not directly guarantee compliance with physical constraints at every time step. Policies may therefore issue commands that violate storage state-of-charge or power limits during exploration or deployment.

Related studies extend learning-based dispatch to different storage and control settings. Zhu et al. used a deep deterministic policy gradient for a hydrogen-storage microgrid, incorporating electrolyzer efficiency into the operating model [13]. Ioannou et al. compared Q-learning, deep Q-networks, proximal policy optimization, and the advantage actor–critic algorithm for islanded-microgrid energy management across seasonal conditions [14]. Their evaluation considered unmet demand, storage imbalance, and computational runtime alongside reward. More recently, Ioannou et al. combined a federated transformer forecasting layer with a soft actor–critic controller for state-of-charge balancing across battery storage units [15]. These studies broaden storage modeling, policy comparison, and

predictive control. The present work focuses on how execution-time correction information can guide the learning of raw storage actions.

Safe reinforcement learning incorporates operational limits by constraining policy updates or correcting actions before execution. Previous studies have used operational constraint gradients, constrained policy optimization, interior-point policy optimization, and Lyapunov-based policy optimization. These methods balance operational returns and safety requirements in networked microgrids and distribution networks [16–19]. Xia et al. further incorporated safety models into training and online action monitoring for hierarchical coordination [20]. Kang et al. combined a constrained Markov decision process with a D3QN algorithm and a multi-stage reward mechanism for microgrid operation under power-flow constraints [21]. Projection-based safety layers map raw policy actions into the currently admissible set, directly connecting learned control to physical execution. For one-dimensional storage power, Euclidean projection onto a closed interval reduces to action clipping. Wabersich and Zeilinger proposed a predictive safety filter that evaluates learning-based control inputs and adjusts them when necessary [22]. Chen et al. used physical shielding to replace unsafe actions in PV and battery voltage control [23]. Qiu et al. applied safe deep reinforcement learning to real-time multi-energy hub control. Wang et al. developed a physics-informed safety layer using dynamic safety assessment to correct actions under multi-energy microgrid constraints [24]. Paesschesoone et al. combined predictive safety filtering, change-point detection, and policy updates to address environmental changes in grid-connected PV–battery–load systems [25].

These studies provide a foundation for integrating safety layers with learning-based control. For the local resource unit considered here, the remaining question is whether execution-time correction can also improve the policy that produces storage commands. However, execution-time projection does not imply that the raw policy has learned to respect the admissible boundaries. When hard clipping is also used in value queries, different out-of-interval actions map to the same boundary. The clipping derivative with respect to the raw action is zero outside the interval. Consequently, the value gradient transmitted through this path cannot reflect the magnitude of action deviation. A learning mechanism that directly uses safety correction information is therefore needed.

Table 1 compares the five reinforcement learning algorithms evaluated in this study [26–29]. Safe-Clip-SAC, Safe-Clip-PPO, and Safe-Clip-TD3 employ execution-time action clipping. Lagrangian-SAC introduces an adaptive constraint penalty, whereas the proposed CGSC-SAC further integrates executed-action replay, critic action clipping, and an explicit correction loss to guide policy optimization.

Table 1. Qualitative comparison of safety strategies implemented in the evaluated reinforcement learning algorithms.

Algorithm	Ref.	Execution Correction	Executed-Action Rreplay	Critic Action Clipping	Deviation Loss
Safe-Clip-SAC	[26]	Yes	No	No	No
Lagrangian-SAC	[27]	Yes	No	No	Lagrangian
Safe-Clip-PPO	[28]	Yes	N/A	No	No
Safe-Clip-TD3	[29]	Yes	No	No	No
CGSC-SAC (Ours)	–	Yes	Yes	Yes	Squared

To address this issue, this study proposes constraint-guided safety-correction soft actor–critic (CGSC-SAC) for local microgrid storage scheduling motivated by VPP applications. Building on the SAC framework [26], the proposed method integrates three complementary mechanisms. First, executed-action replay aligns stored actions with the physical state transitions. Second, critic action clipping incorporates admissible action boundaries into

value estimation. Third, an explicit correction loss guides raw policy outputs toward admissible intervals during policy optimization.

1. A constraint interface connects learning and execution for local storage scheduling within a microgrid resource unit. Storage energy, charging and discharging power, and grid exchange limits define state-dependent action intervals using forecast and real-time information, respectively. The scheduling model also includes load shedding and a soft terminal energy penalty.
2. CGSC-SAC integrates executed-action replay, critic action clipping, and an action correction loss. Admissible intervals enter value queries after replay actions are aligned with executed power. An explicit correction loss supplies deviation gradients that hard clipping lacks outside the interval, guiding the policy toward lower safety-correction requirements.
3. A progressive evaluation distinguishes policy constraint adaptation from execution safety. Representative-day analysis, component ablations, and supply-stress and surplus-power distribution shifts jointly assess the reward, cost, raw-action violation rate, cumulative correction, load shedding, and final-action violation rate. This separates component contributions from the shared safety layer and defines the method's scope within the single-node storage scheduling model.

2. Microgrid Modeling and Scheduling Problem

2.1. System Configuration and Modeling Scope

For the intended VPP application, the modeled microgrid is considered a local resource unit. The present model optimizes internal storage scheduling using local observations and operating constraints. It includes no supervisory dispatch signal, coordination among multiple resource units, or market bidding mechanism. Grid prices and exchange limits are exogenous inputs rather than outputs of a VPP coordinator.

We consider a grid-connected microgrid comprising PV generation, battery storage, electrical loads, and the utility grid, as illustrated in Figure 1. PV generation and grid imports supply energy to meet demand, while storage shifts energy between time periods through charging and discharging. Surplus power can be exported subject to the grid export limit. The energy manager receives generation and demand forecasts, electricity prices, storage states, and grid exchange limits and outputs a storage power command. A safety layer corrects this command using current operating information, after which grid exchange covers the remaining net demand. Load shedding is allowed when supply is insufficient and is penalized in the operating cost.

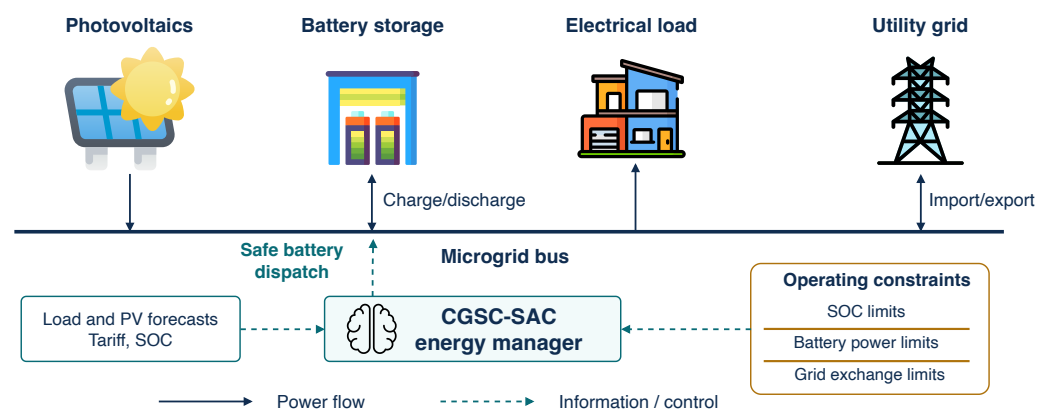


Figure 1. Local PV–battery–utility grid microgrid and energy management architecture considered with the goal of VPP applications. Only the resource unit and its internal control are modeled.

The scheduling horizon contains $T = 24$ periods, indexed by $t = 0, \dots, T - 1$, with a duration of $\Delta t = 1$ h. We use a single-node active-power model without line losses and without explicit voltage, reactive-power, or distribution power-flow constraints. Accordingly, the grid strength, short-circuit ratio, voltage dynamics, and converter-level electromagnetic behavior are not represented. The utility grid is modeled only through exogenous import and export power limits; therefore, the present study does not assign a numerical grid-strength index. The PV and battery interfaces are represented at the energy-management level as active-power injections, without distinguishing grid-forming from grid-following converter control. Such a distinction would require voltage- and frequency-dynamic models beyond the present scheduling scope. PV generation and load are exogenous time series. Storage power is the only continuous variable directly controlled by the policy; grid exchange and load shedding follow from the power deficit and exchange limits. Safety in this study refers only to storage energy, charging and discharging power, and executed-action boundaries within this model. It does not guarantee compliance with all electrical constraints or uninterrupted supply.

2.2. Storage Operation Model

Let E_t denote stored energy at the beginning of period t , and let E_{cap} denote the effective capacity for the current scheduling day. The state of charge (SOC) is defined as $x_t = E_t / E_{\text{cap}}$. The executed storage power draw is denoted by P_t^b , with positive values indicating discharging and negative values indicating charging. Defining $[z]^+ = \max(z, 0)$, the energy update is

$$E_{t+1} = E_t + \eta_c [-P_t^b]^+ \Delta t - \frac{[P_t^b]^+ \Delta t}{\eta_d}, \quad (1)$$

where η_c and η_d are the charging and discharging efficiencies. A single signed power variable excludes simultaneous charging and discharging within a period. Storage operation is constrained by

$$-P_{\text{max}}^c \leq P_t^b \leq P_{\text{max}}^d, \quad E_{\text{min}} \leq E_t \leq E_{\text{max}}, \quad (2)$$

where $E_{\text{min}} = x_{\text{min}} E_{\text{cap}}$ and $E_{\text{max}} = x_{\text{max}} E_{\text{cap}}$. The experiments use fixed SOC limits of $x_{\text{min}} = 0.2$ and $x_{\text{max}} = 0.8$. These fixed limits do not imply a fixed admissible power interval, because available charging and discharging capabilities depend on current stored energy. Equations (1) and (2) give

$$\begin{aligned} \bar{P}_t^c &= \min\left(P_{\text{max}}^c, \frac{E_{\text{max}} - E_t}{\eta_c \Delta t}\right), \\ \bar{P}_t^d &= \min\left(P_{\text{max}}^d, \frac{\eta_d (E_t - E_{\text{min}})}{\Delta t}\right). \end{aligned} \quad (3)$$

Equation (3) assumes $E_t \in [E_{\text{min}}, E_{\text{max}}]$, which is maintained by initialization and the execution safety layer. Rated storage capacity is 300 kWh, while effective training capacity is sampled daily from 285 to 300 kWh. This represents capacity variation. Battery usage cost is approximated as proportional to energy throughput, without modeling electrochemical aging.

2.3. Grid Exchange and Supply–Demand Balance

Let P_t^L and P_t^{PV} denote actual load and PV output, respectively, and define net load as $N_t = P_t^L - P_t^{\text{PV}}$. Grid exchange power P_t^g is positive for imports and negative for exports, with limits

$$-G_{\text{max}}^{\text{exp}} \leq P_t^g \leq G_{\text{max}}^{\text{imp}}, \quad (4)$$

where $G_{\text{max}}^{\text{imp}}$ may vary with time and stress events, while $G_{\text{max}}^{\text{exp}}$ is the scenario-specific export limit. Without a supply–demand residual, power balance requires $P_t^g + P_t^b = N_t$.

Combining Equations (3) and (4) gives the storage power interval satisfying both storage and grid exchange constraints:

$$\ell_t = \max(-\bar{P}_t^c, N_t - G_{t,\max}^{\text{imp}}), \quad u_t = \min(\bar{P}_t^d, N_t + G_{\max}^{\text{exp}}). \quad (5)$$

If $\ell_t \leq u_t$, any $P_t^b \in [\ell_t, u_t]$ satisfies these constraints in the current period. If $\ell_t > u_t$, their intersection is empty. Adjusting storage power alone can then no longer eliminate the supply deficit or surplus while satisfying all constraints. To continue scheduling, the safety layer collapses the operating interval to a singleton:

$$\begin{aligned} b_t^{\text{fb}} &= \text{clip}(N_t - G_{t,\max}^{\text{imp}}, -\bar{P}_t^c, \bar{P}_t^d), \\ \ell_t &= u_t = b_t^{\text{fb}}, \end{aligned} \quad (6)$$

where $\text{clip}(z, l, u) = \min(\max(z, l), u)$. This fallback prioritizes the physical constraints of storage itself. It does not restore the feasibility of the original supply–demand conditions.

For a given executed storage power draw, let the requested grid power be $Q_t = N_t - P_t^b$. The actual grid exchange and shed load are

$$\begin{aligned} P_t^g &= \text{clip}(Q_t, -G_{\max}^{\text{exp}}, G_{t,\max}^{\text{imp}}), \\ P_t^{\text{shed}} &= [Q_t - G_{t,\max}^{\text{imp}}]^+. \end{aligned} \quad (7)$$

To account for surplus when exports are saturated, define the unabsorbed surplus-power residual as $P_t^{\text{sur}} = [-G_{\max}^{\text{exp}} - Q_t]^+$. The complete power accounting identity is

$$P_t^{\text{PV}} + P_t^g + P_t^b + P_t^{\text{shed}} = P_t^L + P_t^{\text{sur}}. \quad (8)$$

In the present single-node scheduling model, active power losses in distribution lines are neglected. Accordingly, the line-loss term is assumed to be zero, i.e., $P_t^{\text{loss}} = 0$, and the power balance is expressed by Equation (8). This simplification allows the scheduling model to focus on energy storage operation and the enforcement of operational constraints without explicitly modeling network power flow. The residual P_t^{sur} accounts for power remaining after grid power is clipped; it is not an additional curtailment command from the policy. The simulation assigns no separate reward penalty to this residual. However, the daily unabsorbed surplus energy is calculated as

$$E_{\text{sur}} = \sum_{t=0}^{T-1} P_t^{\text{sur}} \Delta t, \quad (9)$$

where $\Delta t = 1$ h in the present simulations. The resulting E_{sur} is reported in kWh for each method under the surplus-power scenarios. This metric distinguishes action-correction performance from the utilization of available PV generation.

2.4. Operating Cost and Learning Objective

Operating cost comprises electricity purchases and sales, storage throughput cost, and load-shedding cost:

$$C_t = (c_t^{\text{buy}} [P_t^g]^+ - c^{\text{sell}} [-P_t^g]^+ + c^b |P_t^b| + c^{\text{shed}} P_t^{\text{shed}}) \Delta t, \quad (10)$$

where c_t^{buy} is the time-of-use purchase price and c^{sell} is the export price. The coefficients c^b and c^{shed} represent storage throughput cost and load-shedding penalty, respectively.

Purchase prices are generated from off-peak, shoulder, and peak base prices with small random perturbations. System parameters are given in Table 2.

Let P_t^{raw} denote the storage command before safety processing. The single-step correction magnitude is $\delta_t = |P_t^b - P_t^{\text{raw}}|$. To balance cost, correction requirements, and terminal energy deviation, the environment reward is

$$r_t = -\frac{C_t + \lambda_s \delta_t + \mathbb{I}_{\{t=T-1\}} \lambda_E |E_T - E_0|}{\kappa}, \quad (11)$$

where $\mathbb{I}$ is the indicator function. The correction penalty coefficient is $\lambda_s = 0.10$ per unit of power correction, the terminal energy penalty coefficient is $\lambda_E = 0.10$, and $\kappa = 10$ scales the reward. The terminal term is a soft penalty; the model imposes no hard constraint $E_T = E_0$. The policy can therefore trade daily operating cost against terminal stored energy.

Table 2. Microgrid parameters and algorithm training settings.

Category	Parameter	Setting
Storage	Rated capacity; maximum charge/discharge power	300 kWh; 100 kW each
Storage	SOC range; charge/discharge efficiency	20–80%; 0.95 each
Grid and profiles	Maximum import/export power	260/160 kW
Grid and profiles	Base load; installed PV capacity	115/180 kW
Electricity price	Off-peak/shoulder/peak purchase price	0.12/0.22/0.35 \$/kWh
Electricity price	Export price	0.06 \$/kWh
Cost	Storage throughput/load-shedding coefficient	0.012/3.0 \$/kWh
Training	Episodes; steps per episode	500; 24
Training	Independent training seeds	42, 142, and 242
Training curves	Moving-average window; uncertainty band	20 episodes; mean $\pm$ SD across seeds
SAC-based methods	Hidden layers; learning rate	128×128 ; 3×10^{-4}
SAC-based methods	Replay capacity; learning starts	50,000; 500 steps
SAC-based methods	Batch size; updates per environment step	128; 1
SAC-based methods	Discount factor γ ; target update coefficient τ	0.99; 0.005
CGSC-SAC	Correction loss coefficient λ_{corr}	1.0
Safe-Clip-PPO	Hidden layers; learning rate	128×128 ; 3×10^{-4}
Safe-Clip-PPO	Rollout length; batch size; update epochs	24; 24; 10
Safe-Clip-PPO	Discount factor γ ; GAE coefficient	0.99; 0.95
Safe-Clip-PPO	Clip range; entropy coefficient	0.20; 0.0
Lagrangian-SAC	Initial multiplier; dual learning rate	0.10; 0.05
Lagrangian-SAC	Normalized correction budget; maximum multiplier	0.05; 5.0

The policy learns scheduling behavior by maximizing expected discounted return, $\mathbb{E}[\sum_{t=0}^{T-1} \gamma^t r_t]$, with additional SAC entropy regularization during training. The reported daily cumulative reward is $\sum_t r_t$, the daily operating cost is $\sum_t C_t$, the daily cumulative correction energy is $\sum_t \delta_t \Delta t$, and the daily shed energy is $\sum_t P_t^{\text{shed}} \Delta t$. The cumulative correction energy is obtained by integrating the hourly correction magnitude over time and is reported in kWh. Since $\Delta t = 1$ h in this study, its numerical value equals the sum of the hourly correction magnitudes. Because reward includes correction and terminal penalties in addition to cost, higher reward does not necessarily imply lower operating cost.

For each evaluation episode, the reported daily cumulative reward can equivalently be written as

$$R_{\text{day}} = -\frac{C_{\text{day}} + \lambda_s D_{\text{corr}} + \lambda_E D_E}{\kappa}, \quad (12)$$

where $C_{\text{day}} = \sum_t C_t$ is the daily operating cost, $D_{\text{corr}} = \sum_t |P_t^b - P_t^{\text{raw}}| \Delta t$ is the daily cumulative correction energy, and $D_E = |E_T - E_0|$ is the terminal energy deviation.

2.5. Decision Information and Generation–Demand Uncertainty

We use a finite-horizon, observation-driven decision formulation. Let $\hat{P}_t^L$ and $\hat{P}_t^{PV}$ denote current-period load and PV forecasts, with $\hat{N}_t = \hat{P}_t^L - \hat{P}_t^{PV}$. Generation and demand levels vary daily, and actual profiles are generated using multiplicative random errors. Some scheduling days additionally include joint events with increased demand, reduced PV output, and tighter import limits. The policy receives current-period forecasts and boundary information, rather than the complete future 24 h sequence.

The policy input is the 14-dimensional observation vector

$$s_t = \left[\tilde{x}_t, \frac{\hat{P}_t^L}{250}, \frac{\hat{P}_t^{PV}}{P_{cap}^{PV}}, \frac{\hat{N}_t}{G_0}, \frac{c_t^{buy}}{c_0}, \frac{c_t^{sell}}{c_0}, \sin \frac{2\pi t}{24}, \cos \frac{2\pi t}{24}, x_{min}, x_{max}, \frac{\hat{\ell}_t}{P_0}, \frac{\hat{u}_t}{P_0}, \frac{\delta_{t-1}}{P_0}, \frac{G_{t,max}^{imp}}{G_0} \right], \quad (13)$$

where $P_{cap}^{PV} = 180$ kW, $G_0 = 260$ kW, $P_0 = 100$ kW, and $c_0 = 0.40$ \$/kWh. The load normalization base is 250 kW. The observed SOC, $\tilde{x}_t$, includes noise with standard deviation 0.005 and is clipped to $[0, 1]$; initially, $\delta_{-1} = 0$. The interval $[\hat{\ell}_t, \hat{u}_t]$ follows Equations (5) and (6), but with the actual net load N_t replaced by the forecast net load $\hat{N}_t$. Interval calculations use the simulator's internal stored energy. SOC noise affects only the policy's SOC observation, and so internal energy estimation errors in the safety layer are not modeled.

The policy action $a_t \in [-1, 1]$ maps to $P_t^{raw} = P_0 a_t$. At execution, the safety layer recomputes the interval using actual generation and demand and obtains P_t^b . The environment then updates stored energy and returns the reward. Observations contain forecast errors and noise and do not fully expose the scenario generation parameters. Accordingly, s_t is used as input to the policy and value networks without claiming a strictly fully observable Markov representation.

Forecast errors can cause the observation-based interval used during learning to differ from the real-time feasible interval used during execution. The correction loss may therefore guide the raw policy toward a forecast-based boundary that is inaccurate at execution time. Real-time clipping still enforces the modeled storage and grid-exchange limits, but a larger interval mismatch can increase correction magnitude and reliance on the safety layer. The present results consequently establish robustness only over the tested forecast-error ranges and do not guarantee feasible raw actions under arbitrary prediction errors.

3. Constraint-Guided Safety-Correction Soft Actor–Critic

3.1. Overall Framework

The local scheduling controller, CGSC-SAC, builds on the off-policy maximum-entropy framework of SAC, which combines stochastic policies, value estimation, and entropy regularization to balance return optimization and exploration [26]. For storage scheduling with safety clipping, CGSC-SAC retains execution-time Safe-Clip and introduces constraint information into experience storage, value evaluation, and policy updates. Figure 2 summarizes the framework.

The three components serve distinct purposes. Executed-action replay associates buffer actions with the storage power physically applied to the environment. Critic action clipping uses observation-based admissible actions in target computation and policy value queries. The correction loss directly penalizes the distance between raw policy outputs and the corresponding interval. All methods share the same execution safety layer; their differences primarily concern learning.

3.2. Real-Time Safety Clipping and Executed-Action Replay

For the real-time operating interval from Equations (5) and (6), define the normalized clipping operator

$$\mathcal{P}_t(a) = \text{clip}\left(a, \frac{\ell_t}{P_0}, \frac{u_t}{P_0}\right), \quad (14)$$

$$a_t^{\text{exec}} = \mathcal{P}_t(a_t), \quad P_t^b = P_0 a_t^{\text{exec}}.$$

For a nonempty interval, Equation (14) projects the raw action onto that closed interval, yielding the nearest admissible action by Euclidean distance. If the supply–demand constraint intersection is empty, the singleton fallback in Equation (6) is used. Grid exchange and load shedding then follow Equation (7). The safety layer limits storage power before execution, while load shedding caused by insufficient supply resources is accounted for separately.

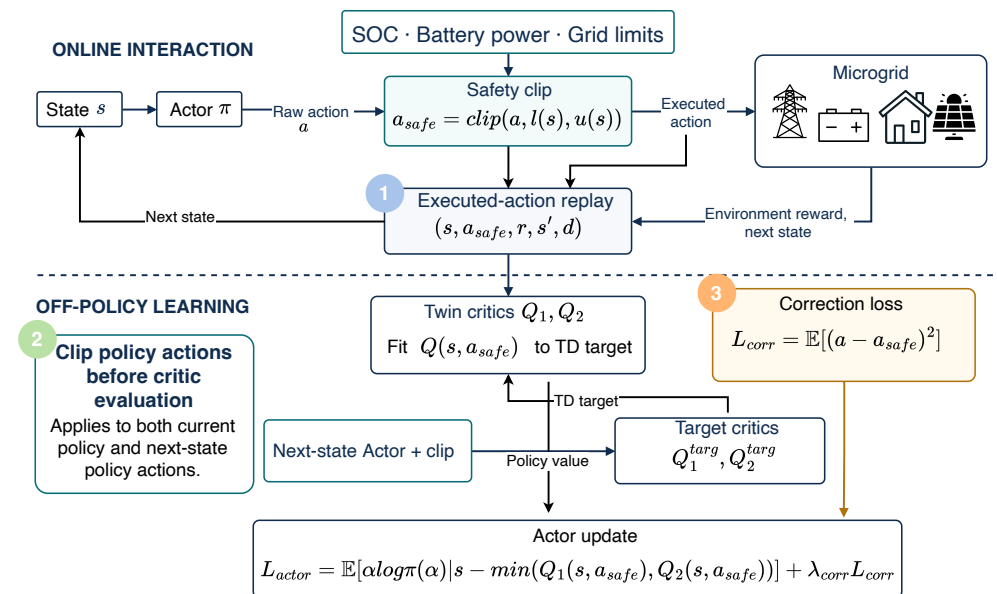


Figure 2. CGSC-SAC framework, including observation input, policy output, real-time safety clipping, executed-action replay, critic updates, and correction-guided policy updates. The schematic uses a_{safe} generically for clipped actions; the text distinguishes real-time execution clipping from forecast-based clipping during learning.

Safe-Clip-SAC stores the raw policy action even though corrected power generates the state transition. CGSC-SAC instead stores the normalized executed action in replay buffer $\mathcal{D}$:

$$(s_t, a_t^{\text{exec}}, r_t, s_{t+1}, d_t) \in \mathcal{D}, \quad (15)$$

where d_t indicates episode termination. This aligns replay actions with executed power but does not eliminate the dependence of the reward on the raw action. The correction penalty in Equation (11) still depends jointly on raw and executed actions. The critic therefore learns a value approximation from sampled interactions. Executed-action replay is not claimed to be a strictly equivalent reconstruction of the original decision process.

3.3. Critic Action Clipping Using Observation-Based Intervals

CGSC-SAC uses two critics, Q_{θ_1} and Q_{θ_2} , and corresponding target networks, $Q_{\bar{\theta}_1}$ and $Q_{\bar{\theta}_2}$. Let π_ϕ denote the stochastic policy. Reparameterization gives the raw action $a = f_\phi(s, \epsilon)$, where ϵ is standard Gaussian noise and a hyperbolic tangent function limits network outputs to $[-1, 1]$. During training, the forecast interval encoded in the observation defines

$$\hat{\mathcal{P}}(s, a) = \text{clip}\left(a, \frac{\hat{\ell}(s)}{P_0}, \frac{\hat{u}(s)}{P_0}\right). \quad (16)$$

For a sampled next observation s' , draw $a' \sim \pi_\phi(\cdot | s')$ and set $\hat{a}' = \hat{\mathcal{P}}(s', a')$. The critic target is

$$y = r + \gamma(1 - d) \left[\min_{j \in \{1, 2\}} Q_{\bar{\theta}_j}(s', \hat{a}') - \alpha \log \pi_\phi(a' | s') \right], \quad (17)$$

where $\alpha > 0$ is the entropy temperature. The critics minimize

$$\mathcal{L}_Q = \frac{1}{2} \sum_{j=1}^2 \mathbb{E}_{\mathcal{D}} \left[\left(Q_{\theta_j}(s, a^{\text{exec}}) - y \right)^2 \right]. \quad (18)$$

The current action in Equation (18) comes from the execution record. The next action in Equation (17) is clipped using the forecast interval in the next observation. Newly sampled actions are similarly clipped by Equation (16) before critic evaluation during policy updates. Thus, “critic action clipping” refers to newly sampled actions in target and policy value queries. It does not overwrite stored executed actions using forecast intervals.

Forecast and actual generation and demand may differ, and so $\hat{\mathcal{P}}$ and $\mathcal{P}_t$ need not coincide. The former provides constraint guidance computable from replay samples; the latter enforces real-time execution constraints. The entropy log-probability remains that of the raw stochastic policy. It is not interpreted as the log-probability of the distribution after hard clipping.

3.4. Action Correction Loss and Policy Update

Clipping actions only during value queries can map different out-of-interval outputs to the same boundary. For a fixed observation, hard clipping has zero derivative outside the interval. The value gradient transmitted through the clipped action therefore cannot directly reflect the distance from the raw output to the boundary. CGSC-SAC supplies this information through a squared correction loss in normalized action space:

$$\mathcal{L}_{\text{corr}} = \mathbb{E}_{s \sim \mathcal{D}, \epsilon} \left[\left(a - \hat{\mathcal{P}}(s, a) \right)^2 \right], \quad a = f_\phi(s, \epsilon). \quad (19)$$

The loss is zero inside the forecast interval and grows quadratically with the distance outside it. Away from nondifferentiable boundary points, its raw-action gradient is $2(a - \hat{\mathcal{P}}(s, a))$, providing an update signal toward the admissible interval. This mechanism penalizes deviation magnitude instead of directly minimizing violation frequency. Reducing cumulative correction and reducing the raw-action violation rate are therefore distinct evaluation objectives.

The complete policy loss is

$$\begin{aligned} \mathcal{L}_\pi = \mathbb{E}_{s \sim \mathcal{D}, \epsilon} \left[\alpha \log \pi_\phi(a | s) - \min_{j \in \{1, 2\}} Q_{\theta_j}(s, \hat{\mathcal{P}}(s, a)) \right] \\ + \lambda_{\text{corr}} \mathcal{L}_{\text{corr}}. \end{aligned} \quad (20)$$

We set $\lambda_{\text{corr}} = 1.0$. Its role differs from that of λ_s in Equation (11). The latter penalizes absolute execution-time correction in kW through the environment reward and thereby influences value learning. In contrast, λ_{corr} directly penalizes normalized squared deviation from the forecast interval in the policy loss. The training correction loss and test cumulative safety correction thus differ in interval source, units, and aggregation.

For a learnable entropy temperature, set $\beta = \log \alpha$ and optimize

$$\mathcal{L}_\beta = -\mathbb{E}[\beta \operatorname{sg}(\log \pi_\phi(a | s) + \mathcal{H}_{\text{tar}})], \quad (21)$$

where sg denotes the stop-gradient and the target entropy is $\mathcal{H}_{\text{tar}} = -1$ for the one-dimensional action. Critic targets are softly updated as

$$\bar{\theta}_j \leftarrow (1 - \tau)\bar{\theta}_j + \tau\theta_j, \quad j = 1, 2. \quad (22)$$

These components jointly address return, exploration, and action correction. The hard safety layer remains the direct source of execution constraint enforcement. A zero final-action violation rate cannot be attributed solely to the correction loss, nor does this objective imply safety under arbitrary perturbations or global optimality.

3.5. Training Procedure and Evaluation Interface

Training generates daily scenarios and interacts with the environment hourly. Each episode samples generation and demand profiles, initial SOC, and effective capacity. At each step, Equation (13) constructs the observation and the policy outputs an action. The execution safety layer then applies Equation (14) using real-time generation and demand. After computing the storage transition, cost, correction, and reward, the environment stores the tuple in Equation (15). The first 500 steps provide initial interaction samples. Thereafter, one minibatch gradient update is performed per environment step, with batch size 128. Both the actor and the critics have two hidden layers with 128 units each and a learning rate of 3×10^{-4} . Other parameters appear in Table 2.

Algorithm 1 summarizes training and testing. The interaction counter n accumulates across training episodes; the first $N_0 = 500$ interactions use random actions, and gradient updates begin with interaction $N_0 + 1$.

Algorithm 1 CGSC-SAC training and execution procedure.

Require: $M = 500$ episodes; $T = 24$ periods; $N_0 = 500$ initial steps; batch size $B = 128$.

Ensure: Trained policy π_ϕ .

- 1: Initialize policy, twin critics, target critics, entropy temperature, replay buffer $\mathcal{D}$, and counter $n = 0$.
 - 2: **for** $e = 1, \dots, M$ **do**
 - 3: Sample a scheduling day; initialize storage and observation s_0 .
 - 4: **for** $t = 0, \dots, T - 1$ **do**
 - 5: Sample a_t randomly if $n < N_0$; otherwise sample $a_t \sim \pi_\phi(\cdot | s_t)$.
 - 6: Compute the real-time interval and execute $a_t^{\text{exec}} = \mathcal{P}_t(a_t)$ using Equation (14).
 - 7: Compute grid exchange, load shedding, storage transition, reward r_t , and observation s_{t+1} .
 - 8: Set $d_t = \mathbb{I}_{\{t=T-1\}}$; store $(s_t, a_t^{\text{exec}}, r_t, s_{t+1}, d_t)$ in $\mathcal{D}$; increment n .
 - 9: **if** $n > N_0$ **then**
 - 10: Sample B transitions from $\mathcal{D}$ and policy actions at their current and next observations.
 - 11: Clip sampled actions to the corresponding forecast intervals using Equation (16).
 - 12: Update entropy temperature using Equation (21) and twin critics using Equations (17) and (18).
 - 13: Compute the correction loss using Equation (19) and update the policy using Equation (20).
 - 14: Softly update target critics using Equation (22).
 - 15: **end if**
 - 16: **end for**
 - 17: **end for**
 - Testing**
 - 18: For each test day, execute deterministic policy actions over T periods with the same real-time safety clipping and state updates.
-

Progressive ablations correspond to the three components. Safe-Clip-SAC retains raw-action replay without learning-time clipping or an additional correction loss. CGSC-A uses executed-action replay only. CGSC-B additionally clips actions in target and policy value queries. Full CGSC-SAC then adds Equation (19). These configurations correspond directly to Table 3.

The raw-action violation rate is the fraction of periods in which raw actions lie outside the real-time operating interval, with numerical tolerance 10^{-6} . The final-action violation rate checks whether executed power exceeds that interval or updated stored energy violates its limits. Executed actions may pass this check under the singleton fallback in Equation (6) even when load shedding or unabsorbed surplus remains. Reward, cost, cumulative correction, and shed energy are therefore reported together to distinguish policy performance, constraint adaptation, and operational consequences of resource insufficiency.

Table 3. Ablation configurations and mean results obtained from the same 50 held-out test episodes for each method. ER: executed-action replay; CC: critic action clipping; CL: correction loss. Bold values mark the best result in each metric.

Method	ER	CC	CL	Mean Reward	Operating Cost	Raw Violation (%)	Correction Energy (kWh)	Shed Energy (kWh)
Safe-Clip-SAC	No	No	No	−116.358	1056.238	68.2	985.424	161.238
CGSC-A	Yes	No	No	−118.790	1053.129	80.2	1259.696	159.371
CGSC-B	Yes	Yes	No	−116.973	1080.244	71.8	806.835	169.709
CGSC-SAC	Yes	Yes	Yes	−114.091	1100.488	73.5	316.198	176.703

4. Simulation Results and Analysis

4.1. Simulation Setup

We evaluated CGSC-SAC in a grid-connected microgrid simulation containing PV generation, storage, and the utility grid. All reported results were obtained from software-based numerical simulations; no physical laboratory experiment, real-time simulator, controller-hardware-in-the-loop platform, or power-hardware-in-the-loop platform was used. Figure 1 summarizes the simulated system architecture and the information exchanged between the energy manager, storage unit, load, PV generation, and utility grid. The scheduling horizon was 24 h with hourly decisions. Table 2 summarizes the system and training parameters. The experiments examined dispatch feasibility, component contributions, and adaptation to distribution shifts. Representative-day analysis assessed charging and discharging, grid exchange, and SOC. Progressive ablations evaluated executed-action replay, critic action clipping, and the correction loss. Supply-stress and surplus-power scenarios then tested policy performance under graded perturbations, as specified in Table 4 in Section 4.4. Safe-Clip-SAC and Safe-Clip-PPO served as baselines with the same safety clipping mechanism. Training behavior was evaluated over 500 episodes using three independent seeds (42, 142, and 242). The curves show the across-seed mean of the 20-episode moving-average reward, with shaded regions indicating one standard deviation across seeds. Each test scenario was independently evaluated over 50 test days. The retained policy for each method was evaluated on the same set of 50 held-out test days. The mean reward, operating cost, raw-action violation rate, cumulative safety correction, shed energy, and final-action violation rate reported in Table 3 were all calculated from these same test episodes. The rewards shown in Figures 3 is training-episode rewards, whereas the mean reward reported in Table 3 is the mean daily cumulative reward obtained during independent policy evaluation. The cumulative reward, operating cost, raw-action violation rate, cumulative safety correction, shed energy, unabsorbed surplus energy, and final-action violation rate jointly assessed overall return, constraint adaptation, energy utilization, and execution safety. These experiments evaluated local microgrid scheduling

under operating-condition shifts; they did not assess VPP aggregation or coordinated dispatch. Lagrangian-SAC was additionally included as an adaptive-penalty baseline. Its correction multiplier was updated between episodes by projected dual ascent using the normalized mean action-correction magnitude. The initial multiplier, dual learning rate, correction budget, and maximum multiplier were 0.10, 0.05, 0.05, and 5.0, respectively.

Lagrangian-SAC was used only as a training-behavior reference in this study. It retained the off-policy SAC backbone and changed primarily the treatment of the correction constraint, which limited confounding from a different policy optimizer. No held-out test-performance claim is made for this baseline.

Figure 3 compares the training rewards of CGSC-SAC, Safe-Clip-SAC, Lagrangian-SAC, Safe-Clip-TD3 and Safe-Clip-PPO over 500 episodes. CGSC-SAC and Safe-Clip-SAC followed similar trajectories, with substantially overlapping variability bands. Lagrangian-SAC and Safe-Clip-PPO produced lower mean training rewards under the tested settings. Three random training seeds were used for computation, and CGSC-SAC showed a certain advantage in the training reward curve comparison.

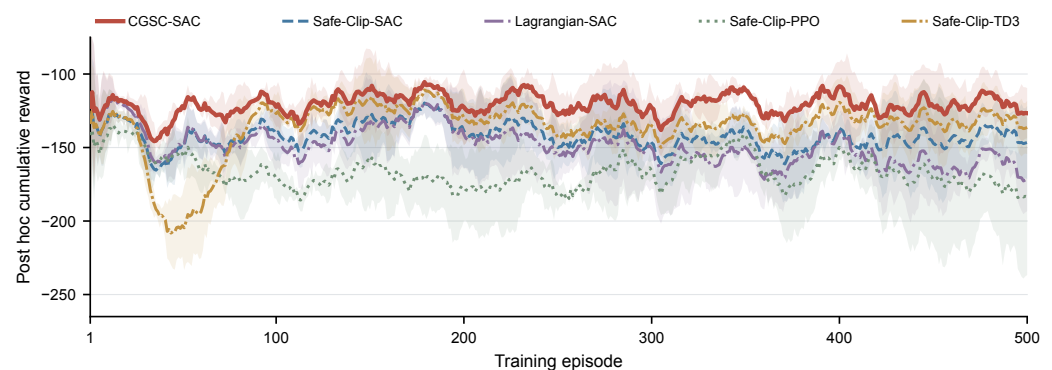


Figure 3. Cumulative training rewards over 500 episodes. Solid lines show the mean 20-episode moving-average reward across three independent training seeds, and shaded regions show one standard deviation across seeds.

4.2. Representative-Day Dispatch

To select a day representative of nominal generation and demand, we generated 365 candidate days with different random seeds. Daily load and PV scale factors ranged from 0.95 to 1.05 and 0.70–0.90, respectively, without additional stress events. Selection used generation and demand statistics. Each candidate's total load and PV energy deviations from their sample medians were normalized by the corresponding standard deviations. The candidate with the smallest two-dimensional Euclidean distance was selected. Policy reward was not used for selection, reducing subjectivity associated with choosing favorable dispatch outcomes. The selected day had total demand of 3218.53 kWh, PV generation of 906.10 kWh, and initial SOC of 65.99%.

Figure 4 presents CGSC-SAC power allocation and SOC throughout that day. Storage discharged during 00:00–02:00, after which the supply from the grid met most of the demand. As daytime PV output increased, storage charged during 07:00–10:00, reaching the 80% SOC ceiling at approximately 11:00. Some midday PV surplus was exported. After 16:00, the policy discharged storage to reduce grid purchases during expensive periods and jointly supply evening demand with the grid. Grid imports nevertheless remained substantial during several high-price hours because evening demand exceeded the combined instantaneous contribution of PV generation and feasible battery discharge. The controller reduced, but could not eliminate, these purchases because battery discharge power, available stored energy, and the 20% lower SOC bound remained active constraints. The 16:00–21:00 pattern therefore represents constrained residual demand rather than a

preference for purchasing electricity at high prices. Daily grid imports and exports were 2290.39 and 85.86 kWh, respectively. The storage charging and discharging energies were 181.54 and 289.44 kWh. SOC remained within 20–80%, and the maximum power-balance residual was 2.84×10^{-14} kW. The selected dispatch thus satisfied the examined power-balance and storage constraints. Terminal SOC reached the 20% lower bound, showing substantial use of initial stored energy. Cross-day applications would require terminal energy coordination.

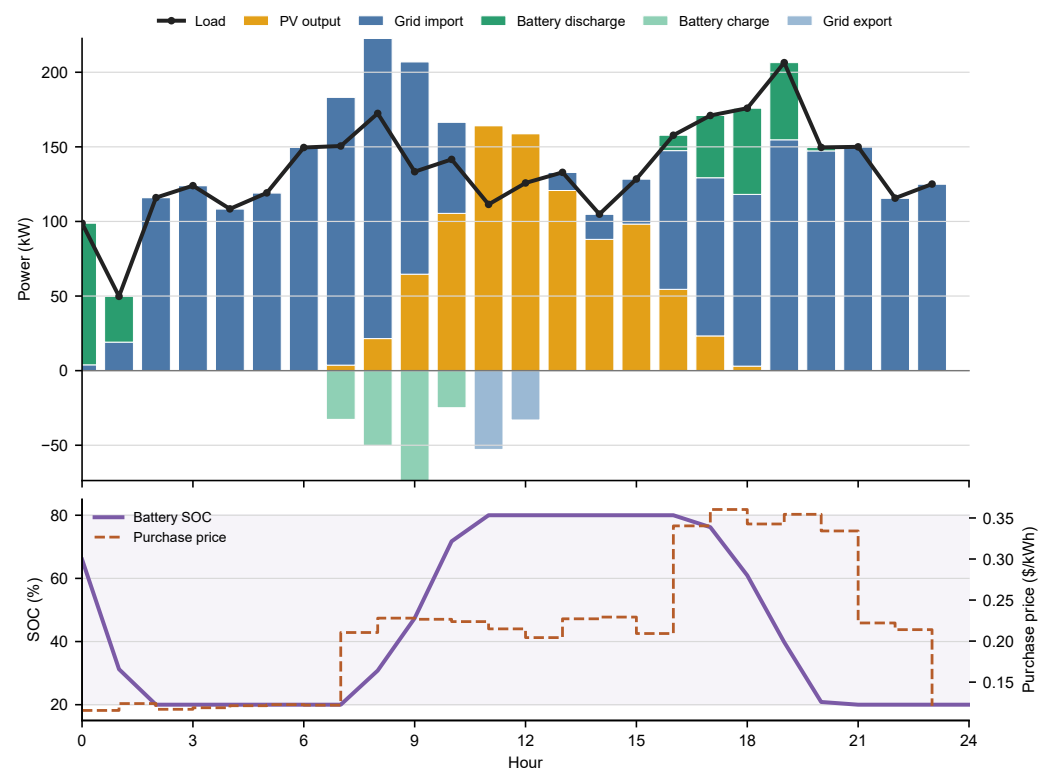


Figure 4. CGSC-SAC power dispatch and SOC on the representative day.

4.3. Ablation Study

We constructed three progressive configurations, CGSC-A, CGSC-B, and full CGSC-SAC, starting from Safe-Clip-SAC. CGSC-A stored the safety-corrected executed action in replay, aligning replay actions with actual power. CGSC-B additionally clipped actions used in critic queries. Full CGSC-SAC added an action correction loss to penalize the deviation between policy outputs and their clipped counterparts. Training rewards and test metrics were compared as components were added. Table 3 summarizes the configurations and results.

Figure 5 shows the mean training reward curves for the four ablation configurations over 500 episodes. The variability bands overlapped substantially, indicating that the small differences between the mean curves were within the observed training-seed variability. In the separate 50-day evaluation reported in Table 3, the retained CGSC-SAC policy achieved the highest mean test reward, -114.091 . This exceeded Safe-Clip-SAC, CGSC-A, and CGSC-B by 2.267, 4.699, and 2.882, respectively. Daily cumulative correction energy decreased to 316.198 kWh, corresponding to reductions of 67.9%, 74.9%, and 60.8%. CGSC-A had lower mean reward than the baseline; thus, executed-action replay alone did not improve overall return in this experiment. Adding the correction loss to CGSC-B improved mean reward and substantially reduced correction, supporting its role in limiting correction magnitude. However, the full method's raw-action violation rate was 73.5%, above the baseline's 68.2%.

Operating cost and shed energy were also not minimal. Improvements therefore concerned overall reward and cumulative safety correction rather than all individual metrics.

Violation frequency and correction magnitude describe different aspects of policy behavior. The higher raw-action violation rate but lower cumulative correction of CGSC-SAC is consistent with more frequent, smaller excursions outside the admissible interval, whereas a lower violation rate does not exclude occasional large deviations. All executed actions remained feasible in the modeled system because clipping was applied before execution. Nevertheless, frequent intervention indicates greater dependence on the safety layer, while rare large deviations may require greater actuator headroom and may be more sensitive to delay or model mismatch. These indicators are therefore interpreted jointly instead of using either one alone to establish practical acceptability.

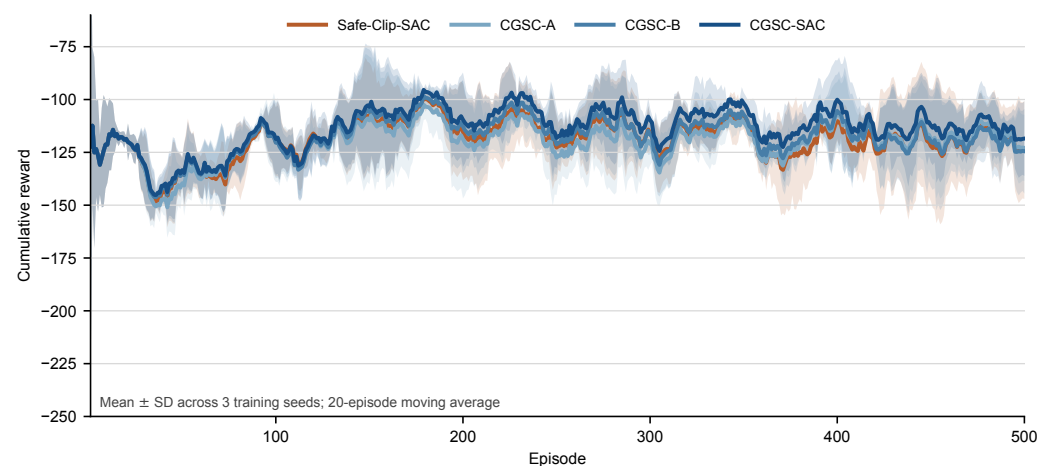


Figure 5. Training reward curves for the ablation configurations.

As a numerical check, the mean terminal-penalty term before reward scaling was 8.798 for each configuration in Table 3. For example, the mean reward of CGSC-B is reconstructed as

$$-\frac{1080.244 + 0.10 \times 806.835 + 8.798}{10} = -116.973. \quad (23)$$

Thus, the reported CGSC-B reward corresponds to a non-negative terminal penalty. The same calculation was applied to all configurations using the operating cost, cumulative correction, and terminal energy deviation obtained from the same 50 test episodes.

4.4. Robustness Experiments

We constructed supply-stress and surplus-power distribution shifts to assess adaptation to changes in generation, demand, and grid exchange capability. Table 4 lists the graded parameters. In Case 1, daily load and PV scale ranges were fixed at 0.95–1.05 and 0.55–0.80, respectively. A four-hour stress event began at 17:00 on every test day. From S0 to S3, initial SOC decreased, event-period demand increased, PV output decreased, and the import limit tightened, reducing supply margins. Case 2 had no additional time-localized stress event. Instead, initial SOC and PV output increased while daily demand decreased, PV forecast error increased, and exports were restricted. This tested policy behavior under changing surplus-power conditions. Table 4 retains parameters that vary by level; common settings are described here, and other parameters follow the base environment.

Table 4. Graded perturbation settings for the robustness scenarios.

Scenario	Parameter	S0	S1	S2	S3
Supply stress	Initial SOC (%)	35–45	30–38	25–33	22–30
Supply stress	Event load multiplier	1.00	1.10–1.20	1.25–1.40	1.40–1.60
Supply stress	Event PV multiplier	0.80–1.00	0.50–0.80	0.25–0.55	0.10–0.35
Supply stress	Event import limit (kW)	180–190	150–170	125–150	105–135
Surplus power	Initial SOC (%)	55–65	62–70	68–75	72–78
Surplus power	Daily PV scale	0.70–0.90	0.90–1.10	1.10–1.30	1.30–1.50
Surplus power	Daily load scale	0.90–1.00	0.82–0.94	0.74–0.88	0.68–0.82
Surplus power	Export limit (kW)	160	100	60	25
Surplus power	PV forecast error standard deviation	0.08	0.14	0.20	0.28

Figure 6 compares the methods across S0–S3 by scenario and metric. Under supply stress (Figure 6a), mean cumulative rewards decreased for all methods as stress increased. CGSC-SAC declined from -55.544 at S0 to -159.911 at S3 while retaining the highest mean at every level. Its advantages over Safe-Clip-SAC and Safe-Clip-PPO were 2.050 – 5.681 and 7.766 – 47.380 , respectively. Reward uncertainty intervals for CGSC-SAC and Safe-Clip-SAC overlapped at S2 and S3; the mean ranking does not establish statistical significance. Under surplus power (Figure 6b), mean rewards increased with level for all three methods. CGSC-SAC increased from -49.489 to -32.002 , exceeding the two baselines by 4.378 – 5.967 and 4.597 – 6.801 , respectively. Higher PV output and lower demand accompanied tighter export limits and greater forecast error. Increasing the perturbation level therefore did not monotonically reduce reward, nor does the reward trend alone demonstrate improved PV utilization.

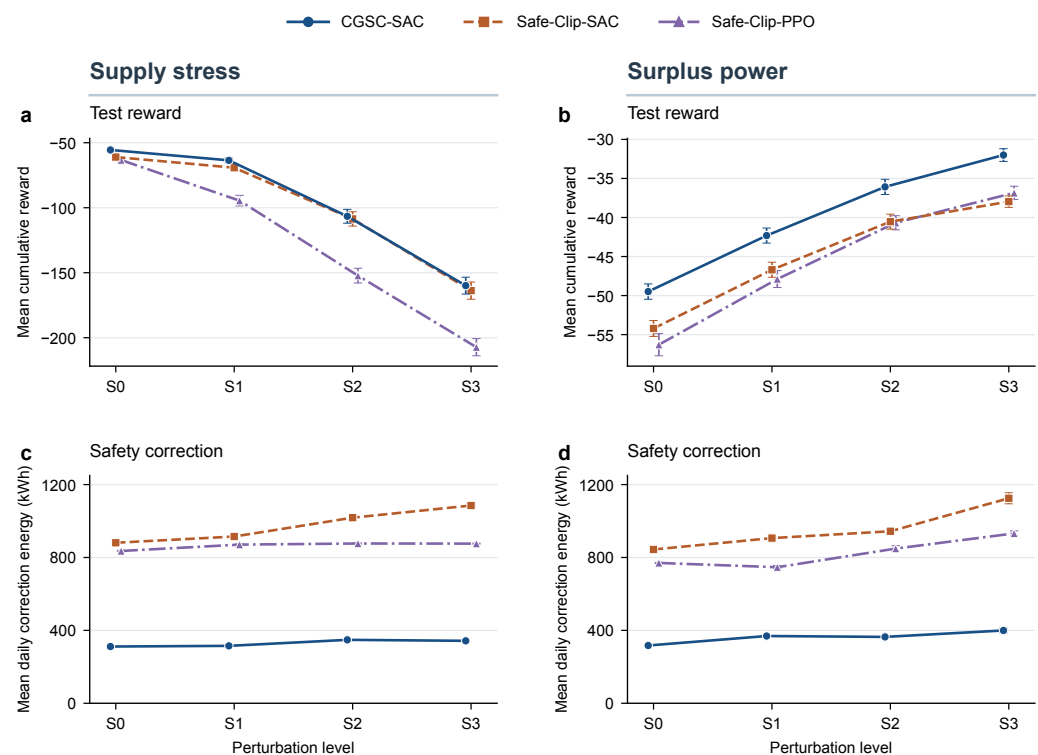


Figure 6. Robustness under supply-stress and surplus-power perturbations. (a) Mean cumulative reward under supply stress; (b) mean cumulative reward under surplus power; (c) mean daily cumulative safety correction under supply stress; (d) mean daily cumulative safety correction under surplus power.

Safety correction followed a different pattern. Under supply stress (Figure 6c), Safe-Clip-SAC's mean daily cumulative correction increased from 880.798 to 1085.149 kW.

CGSC-SAC remained within the range of 311.315–348.027 kW and declined slightly from S2 to S3. Relative reductions were 64.7–68.4% across the four levels. Under surplus power (Figure 6d), CGSC-SAC correction increased overall from 316.983 kW at S0 to 399.446 kW at S3, with intermediate fluctuations. Safe-Clip-SAC increased from 843.948 to 1124.910 kW. CGSC-SAC reductions relative to Safe-Clip-SAC were 59.3–64.5%, and correction remained below Safe-Clip-PPO at every level in both scenarios. CGSC-SAC thus maintained smaller correction magnitudes under the tested distribution shifts. Nevertheless, correction remained necessary and did not vary strictly monotonically with perturbation level. The complete test metrics showed zero final-action violations for all methods, while CGSC-SAC did not always have a lower raw-action violation rate. Its advantages primarily concerned mean overall reward and cumulative safety correction, with insufficient physical resources still limiting performance under supply stress.

To further quantify the utilization of surplus power, the daily unabsorbed surplus energy was calculated as $E_{\text{sur}} = \sum_{t=0}^{T-1} P_t^{\text{sur}} \Delta t$. Averaged over the 50 test days, the unabsorbed surplus energy of CGSC-SAC was 0.000, 9.933, 188.719, and 505.859 kWh under S0–S3, respectively. The corresponding values were 0.000, 7.882, 141.319, and 444.086 kWh for Safe-Clip-SAC, and 0.000, 0.000, 58.660, and 436.067 kWh for Safe-Clip-PPO. These results indicate that CGSC-SAC did not minimize unabsorbed surplus energy under the tested perturbations. Therefore, its lower safety-correction requirement should not be interpreted as improved PV utilization.

To locate the temporal sources of cumulative correction differences, Figure 7 shows hourly and daily correction for one test day in each S3 scenario. Hourly correction is the absolute difference between raw and executed storage power. Under supply stress (Figure 7a,b), Safe-Clip-SAC required larger corrections mainly during the time ranges 00:00–03:00, 12:00–18:00, and 20:00–23:00. Its maximum single-step correction was 106.0 kW at 17:00, compared with CGSC-SAC's maximum of 31.0 kW. Daily correction energy decreased from 1083.1 to 340.5 kWh, a reduction of 68.6%. Under surplus power (Figure 7c,d), larger differences occurred during 02:00–03:00, 13:00–16:00, and 20:00–23:00. Both methods reached their largest single-step corrections at 21:00: 98.1 kW for Safe-Clip-SAC and 32.3 kW for CGSC-SAC. Daily correction decreased from 960.2 to 340.0 kWh, a reduction of 64.6%.

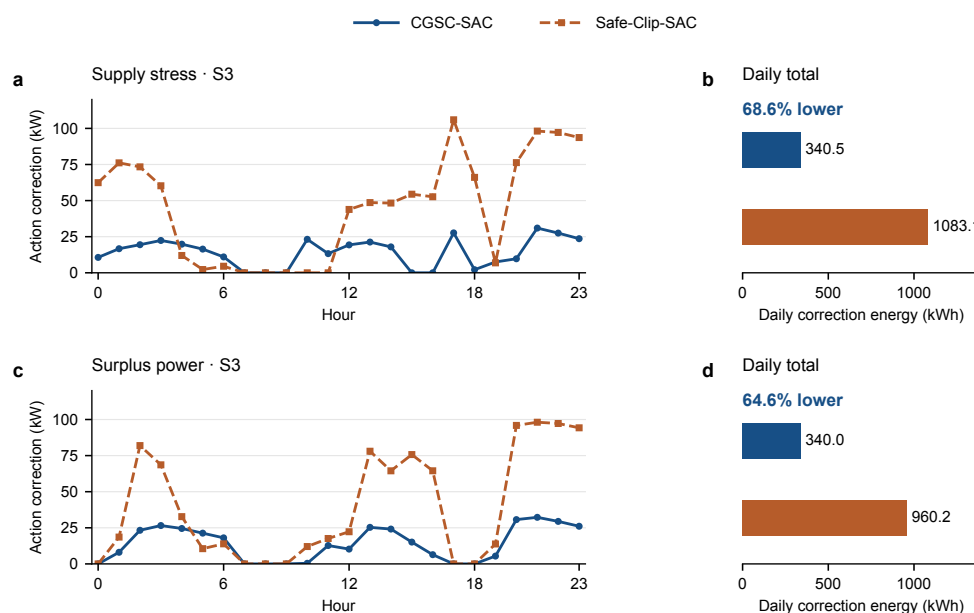


Figure 7. Safety correction in two S3 scenarios. (a) Hourly correction under supply stress; (b) daily cumulative correction under supply stress; (c) hourly correction under surplus power; (d) daily cumulative correction under surplus power.

These single-day results associate lower cumulative correction with smaller large-magnitude corrections across several periods. However, CGSC-SAC still required intervention in multiple periods and did not outperform the baseline at every hour. Examples include 04:00–06:00 and 10:00–11:00 under supply stress. The reductions of 68.6% and 64.6% apply only to the displayed days and do not replace the 50-day statistics in Figure 6. Correction curves alone also do not identify raw charging or discharging directions or positions relative to upper and lower boundaries.

4.5. Network-Level Validation and Scalability Analysis

To evaluate the applicability of CGSC-SAC to larger-scale microgrids, an extended test scenario is constructed based on the IEEE 33-bus system, with three PV units, two load units, and energy storage units integrated at different buses. The original CGSC-SAC algorithm structure is retained. Scheduling performance and computational efficiency in the multi-bus environment are evaluated over 10 independent test days using models trained with five different random seeds. Figure 8 presents the cumulative action correction and load satisfaction rate over 10 test days. Under varying PV generation and load conditions, the action correction varies across test days, while the load satisfaction rate remains above 83%, with no energy storage SOC violations observed. Table 5 further compares the overall performance of the original single-bus system and the extended IEEE 33-bus system.

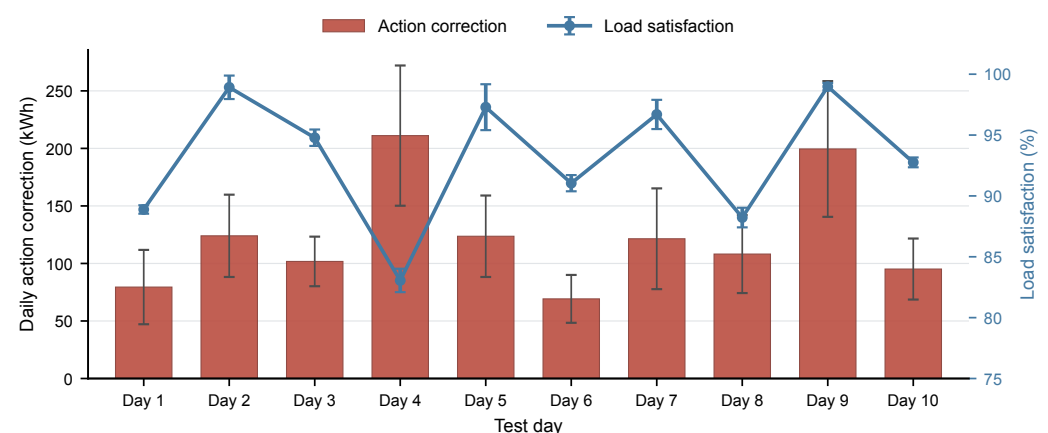


Figure 8. Daily cumulative action correction and load satisfaction rate in the IEEE 33-bus extended microgrid.

Table 5. Performance comparison between different microgrid scales.

Metric	Single-Bus	IEEE 33-Bus
Average load satisfaction rate (%)	93.19	93.06
Average action correction (kWh/day)	92.52	123.35
SOC violation count	0	0
Average decision time (ms/step)	0.1253	5.1813

Compared with the original single-bus system, the extended system exhibits a slight decrease of 0.13 percentage points in the average load satisfaction rate, while the average action correction increases by 33.3%, indicating a greater need for action correction under multi-bus operating conditions. The average decision time increases from 0.1253 ms to 5.1813 ms, which remains suitable for hourly scheduling. These results demonstrate that CGSC-SAC maintains comparable load satisfaction and compliance with energy storage SOC constraints in the tested multi-bus scenario, providing preliminary evidence of its applicability to larger-scale microgrids.

5. Conclusions

This study proposed CGSC-SAC for energy storage scheduling in a grid-connected microgrid with a view to VPP applications. By integrating executed-action replay, critic action clipping, and an action correction loss with real-time safety clipping, the method incorporates execution-time correction information into policy learning. This design allows the policy to learn from interventions made by the safety layer rather than treat them only as external execution constraints. In the tested 24 h simulations, CGSC-SAC improved the mean ablation reward by 2.267 and reduced cumulative correction energy by 67.9% relative to Safe-Clip-SAC. Under supply-stress and surplus-power distribution shifts, it achieved higher mean rewards and reduced cumulative correction energy by 64.7–68.4% and 59.3–64.5%, respectively. These results suggest that feeding executed actions and correction magnitudes back into the learning process can reduce the extent of subsequent safety interventions under the examined operating conditions. However, the improvements mainly concerned overall reward and correction magnitude and did not extend to every individual metric. In particular, a lower cumulative correction energy did not necessarily correspond to a lower raw-action violation rate, because the violation frequency and correction magnitude characterize different aspects of policy behavior. All executed actions remained feasible because every method used the same real-time safety layer, which achieved zero final-action violations. CGSC-SAC should therefore be interpreted as improving policy adaptation within a safeguarded control framework rather than replacing execution-time protection. The conclusions are primarily based on the single-node active-power scheduling model, supplemented by preliminary IEEE 33-bus extended-system validation without explicit network power-flow constraints. Future work should examine network-constrained and multi-resource VPP operation, terminal energy coordination, forecast–real-time interval mismatch, broader statistical evaluation, and hardware-in-the-loop validation.

Author Contributions: Conceptualization, J.L. and T.Y.; methodology, J.L. and J.J.; software, J.L.; validation, J.L., J.J. and M.O.; formal analysis, J.L. and J.T.; investigation, J.J. and M.O.; resources, J.T. and B.L.; data curation, J.J. and M.O.; writing—original draft preparation, J.L.; writing—review and editing, J.J., J.T., B.L. and T.Y.; visualization, J.L.; supervision, B.L. and T.Y.; project administration, T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by China Southern Power Grid under grant number 090000KC23090025.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The processed simulation data supporting the findings of this study are publicly available in the GitHub repository “CGSC-SAC Supporting Data” at <https://github.com/Raylining/cgsc-sac-supporting-data> (accessed on 16 August 2026). The repository provides source data for the reported tables, training curves, representative-day dispatch, robustness analysis, action-correction trajectories, and unabsorbed-surplus results, plus configuration metadata and variable definitions; simulations used Python 3.9 and PyTorch 2.2.2.

Conflicts of Interest: Authors Jiangnan Li, Jiru Jiang and Mingyu Ou were employed by the Shenzhen Power Supply Company, and Junfeng Tan was employed by the Southern Power Grid Artificial Intelligence Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The authors declare that this study received funding from China Southern Power Grid. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

References

1. Al Talaq, M.; Abdelghany, M.B.; Al-Durra, A.; Zeineldin, H.H.; El-Fouly, T.H.M. Optimal Configuration and Coordinated Scheduling of Hybrid Inverter-Based Energy Sources in Large-Scale Green Management Systems. *IEEE Open J. Ind. Electron. Soc.* **2025**, *6*, 1720–1742. <https://doi.org/10.1109/OJIES.2025.3624462>.
2. Al Talaq, M.; Abdelghany, M.B.; Al-Durra, A.; Zeineldin, H.; El-Fouly, T.H.M. A Coordinated Operation Control of Grid-forming and Grid-following to Improve Microgrid Large Signal Stability. In Proceedings of the 2024 IEEE Industrial Electronics and Applications Conference (IEACon), Kuala Lumpur, Malaysia, 4–5 November 2024; pp. 7–12. <https://doi.org/10.1109/IEACon61321.2024.10797417>.
3. Hong, Y.; Sun, W.; Zhao, C.; Lin, F.; Xu, Q. Multi-objective Low-voltage Ride-through Control Strategy for Two-stage Grid-connected Photovoltaic System. *Autom. Electr. Power Syst.* **2026**, *50*, 158–167. (In Chinese) <https://doi.org/10.7500/AEPS20250811004>.
4. Olivares, D.E.; Mehri-Sani, A.; Etemadi, A.H.; Canizares, C.A.; Iravani, R.; Kazerani, M.; Hajimiragha, A.H.; Gomis-Bellmunt, O.; Saeedifard, M.; Palma-Behnke, R.; et al. Trends in Microgrid Control. *IEEE Trans. Smart Grid* **2014**, *5*, 1905–1919. <https://doi.org/10.1109/tsg.2013.2295514>.
5. Zia, M.F.; Elbouchikhi, E.; Benbouzid, M. Microgrids energy management systems: A critical review on methods, solutions, and prospects. *Appl. Energy* **2018**, *222*, 1033–1055. <https://doi.org/10.1016/j.apenergy.2018.04.103>.
6. Grover-Silva, E.; Heleno, M.; Mashayekh, S.; Cardoso, G.; Girard, R.; Kariniotakis, G. A stochastic optimal power flow for scheduling flexible resources in microgrids operation. *Appl. Energy* **2018**, *229*, 201–208. <https://doi.org/10.1016/j.apenergy.2018.07.114>.
7. Kou, P.; Liang, D.; Gao, L. Distributed EMPC of multiple microgrids for coordinated stochastic energy management. *Appl. Energy* **2017**, *185*, 939–952. <https://doi.org/10.1016/j.apenergy.2016.09.092>.
8. Lu, R.; Bai, R.; Ding, Y.; Wei, M.; Jiang, J.; Sun, M.; Xiao, F.; Zhang, H.T. A hybrid deep learning-based online energy management scheme for industrial microgrid. *Appl. Energy* **2021**, *304*, 117857. <https://doi.org/10.1016/j.apenergy.2021.117857>.
9. Casagrande, V.; Ferianc, M.; Rodrigues, M.R.D.; Boem, F. Online End-to-End Learning-Based Predictive Control for Microgrid Energy Management. *IEEE Trans. Control Syst. Technol.* **2025**, *33*, 463–478. <https://doi.org/10.1109/tcst.2024.3494733>.
10. Chen, X.; Qu, G.; Tang, Y.; Low, S.; Li, N. Reinforcement Learning for Selective Key Applications in Power Systems: Recent Advances and Future Challenges. *IEEE Trans. Smart Grid* **2022**, *13*, 2935–2958. <https://doi.org/10.1109/tsg.2022.3154718>.
11. Zeng, P.; Li, H.; He, H.; Li, S. Dynamic Energy Management of a Microgrid Using Approximate Dynamic Programming and Deep Recurrent Neural Network Learning. *IEEE Trans. Smart Grid* **2019**, *10*, 4435–4445. <https://doi.org/10.1109/tsg.2018.2859821>.
12. Guo, C.; Wang, X.; Zheng, Y.; Zhang, F. Real-time optimal energy management of microgrid with uncertainties based on deep reinforcement learning. *Energy* **2022**, *238*, 121873. <https://doi.org/10.1016/j.energy.2021.121873>.
13. Zhu, Z.; Weng, Z.; Zheng, H. Optimal Operation of a Microgrid with Hydrogen Storage Based on Deep Reinforcement Learning. *Electronics* **2022**, *11*, 196. <https://doi.org/10.3390/electronics11020196>.
14. Ioannou, I.; Javaid, S.; Tan, Y.; Vassiliou, V. Autonomous Reinforcement Learning for Intelligent and Sustainable Autonomous Microgrid Energy Management. *Electronics* **2025**, *14*, 2691. <https://doi.org/10.3390/electronics14132691>.
15. Ioannou, I.; Javaid, S.; Tan, Y.; Vassiliou, V. A Hierarchical Predictive-Adaptive Control Framework for State-of-Charge Balancing in Mini-Grids Using Deep Reinforcement Learning. *Electronics* **2026**, *15*, 61. <https://doi.org/10.3390/electronics15010061>.
16. Zhang, Q.; Dehghanpour, K.; Wang, Z.; Qiu, F.; Zhao, D. Multi-Agent Safe Policy Learning for Power Management of Networked Microgrids. *IEEE Trans. Smart Grid* **2021**, *12*, 1048–1062. <https://doi.org/10.1109/tsg.2020.3034827>.
17. Li, H.; He, H. Learning to Operate Distribution Networks With Safe Deep Reinforcement Learning. *IEEE Trans. Smart Grid* **2022**, *13*, 1860–1872. <https://doi.org/10.1109/tsg.2022.3142961>.
18. Ye, Y.; Wang, H.; Chen, P.; Tang, Y.; Strbac, G. Safe Deep Reinforcement Learning for Microgrid Energy Management in Distribution Networks With Leveraged Spatial–Temporal Perception. *IEEE Trans. Smart Grid* **2023**, *14*, 3759–3775. <https://doi.org/10.1109/tsg.2023.3243170>.
19. Hao, G.; Li, Y.; Li, Y.; Jiang, L.; Zeng, Z. Lyapunov-Based Safe Reinforcement Learning for Microgrid Energy Management. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 9985–9999. <https://doi.org/10.1109/tnnls.2024.3496932>.
20. Xia, Y.; Xu, Y.; Feng, X. Hierarchical Coordination of Networked-Microgrids Toward Decentralized Operation: A Safe Deep Reinforcement Learning Method. *IEEE Trans. Sustain. Energy* **2024**, *15*, 1981–1993. <https://doi.org/10.1109/tste.2024.3390808>.
21. Kang, J.; Zeng, Y.; Wei, Q. Microgrid Operation Optimization Strategy Based on CMDP-D3QN-MSRM Algorithm. *Electronics* **2025**, *14*, 3654. <https://doi.org/10.3390/electronics14183654>.
22. Wabersich, K.P.; Zeilinger, M.N. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica* **2021**, *129*, 109597. <https://doi.org/10.1016/j.automatica.2021.109597>.
23. Chen, P.; Liu, S.; Wang, X.; Kamwa, I. Physics-Shielded Multi-Agent Deep Reinforcement Learning for Safe Active Voltage Control With Photovoltaic/Battery Energy Storage Systems. *IEEE Trans. Smart Grid* **2023**, *14*, 2656–2667. <https://doi.org/10.1109/tsg.2022.3228636>.
24. Wang, Y.; Qiu, D.; Sun, M.; Strbac, G.; Gao, Z. Secure energy management of multi-energy microgrid: A physical-informed safe reinforcement learning approach. *Appl. Energy* **2023**, *335*, 120759. <https://doi.org/10.1016/j.apenergy.2023.120759>.

25. Paesschesoone, S.; Kayedpour, N.; Manna, C.; Crevecœur, G. Reinforcement learning for an enhanced energy flexibility controller incorporating predictive safety filter and adaptive policy updates. *Appl. Energy* **2024**, *368*, 123507. <https://doi.org/10.1016/j.apenergy.2024.123507>.
26. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018*; Dy, J.; Krause, A., Eds.; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2018; Volume 80, pp. 1861–1870. Available online: <https://proceedings.mlr.press/v80/haarnoja18b.html> (accessed on 16 August 2026).
27. Tessler, C.; Mankowitz, D.J.; Mannor, S. Reward Constrained Policy Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
28. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. <https://doi.org/10.48550/arXiv.1707.06347>.
29. Fujimoto, S.; van Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. In *Proceedings of the 35th International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2018; Volume 80, pp. 1587–1596.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.