University of Sannio

Department of Engineering

Master's Degree in

Electronics Engineering for Automation and Sensing

Deep Reinforcement Learning for Adaptive Bidirectional Electric Vehicle Charging Management (Vehicle-to-Grid)

Supervisor:
Prof. Carmela Bernardo

Co-Supervisor:
Dr. Antonio Pepiciello

Candidate:
Angelo Caravella
Student ID 389000016

Academic Year 2024–2025

Contents

Abstract in italian

L'adozione crescente dei **Veicoli Elettrici (EV)** in concomitanza con la sempre maggiore penetrazione di **Fonti di Energia Rinnovabile (RES)** intermittenti, presenta sfide significative alla **stabilità** e all'**efficienza della rete elettrica**. La tecnologia **Vehicle-to-Grid (V2G)** emerge come soluzione fondamentale, trasformando gli EV da carichi passivi a **risorse energetiche flessibili** capaci di fornire vari **servizi di rete**. Questa tesi affronta il complesso **problema di ottimizzazione multi-obiettivo** della gestione intelligente di carica e scarica degli EV, che intrinsecamente implica un equilibrio tra **benefici economici**, **esigenze di mobilità dell'utente**, **preservazione della salute della batteria** e **stabilità della rete** in condizioni stocastiche.
Di fronte alla complessa sfida di ottimizzare la ricarica dei veicoli elettrici (EV) in scenari Vehicle-to-Grid (V2G), un approccio che si limita a un singolo modello di controllo, come il Deep Q-Networks (DQN), risulterebbe inadeguato. La natura del problema, caratterizzata da molteplici obiettivi contrastanti (benefici economici, esigenze dell'utente, salute della batteria, stabilità della rete) e da una profonda incertezza; richiede un'analisi comparativa e rigorosa di un'ampia gamma di strategie di controllo. Per questo motivo, la ricerca si concentra sulla valutazione di un portafoglio diversificato di algoritmi, che include numerosi modelli di Deep Reinforcement Learning (DRL), approcci euristici e il Model Predictive Control (MPC). Questo metodo consente di mappare in modo completo il panorama delle soluzioni, identificando i punti di forza e di debolezza di ciascun approccio in relazione alle diverse sfaccettature del problema V2G.
In conclusione questa lavoro di tesi non si focalizza su un singolo modello, ma adotta un approccio comparativo su larga scala perché:
Non esiste una soluzione unica: La complessità del problema V2G rende improbabile che un solo algoritmo sia ottimale in tutte le condizioni.
Si ricercano i compromessi: L'obiettivo è comprendere i trade-off tra l'efficienza dei dati, la stabilità dell'addestramento, la robustezza all'incertezza e la complessità computazionale delle diverse famiglie di algoritmi.
La validazione è più rigorosa: Confrontare i modelli di DRL non solo tra loro ma anche con benchmark consolidati come le euristiche e l'MPC fornisce una misura molto più credibile del loro reale valore aggiunto.

Abstract

The growing adoption of **Electric Vehicles (EVs)**, combined with the increasing penetration of intermittent **Renewable Energy Sources (RES)**, presents significant challenges to the **stability** and **efficiency** of the power grid. **Vehicle-to-Grid (V2G)** technology emerges as a key solution, transforming EVs from passive loads into **flexible energy resources** capable of providing various **grid services**. This thesis addresses the complex **multi-objective optimization problem** of smart EV charging and discharging, which requires balancing **economic benefits**, **user mobility needs**, **battery health preservation**, and **grid stability** under stochastic conditions.
Given the complexity of optimizing EV charging in V2G scenarios, relying on a single control model is insufficient. The nature of the problem, characterized by multiple conflicting objectives (economic benefits, user needs, battery health, grid stability) and profound uncertainty, demands a rigorous comparative analysis of a wide range of control strategies.
For this reason, the research focuses on evaluating a diverse portfolio of algorithms, including numerous Deep Reinforcement Learning (DRL) models, heuristic approaches, and Model Predictive Control (MPC). This method allows for a complete mapping of the solution landscape, identifying the strengths and weaknesses of each approach in relation to the different facets of the V2G problem.
In short, this thesis adopts a **broad-spectrum comparative approach** for several reasons:
No Single Solution: The complexity of the V2G problem makes it unlikely that a single algorithm can be optimal in all conditions.
Understanding Trade-offs: The goal is to understand the trade-offs between data efficiency, training stability, robustness to uncertainty, and the computational complexity of different algorithm families.
Rigorous Validation: Comparing DRL models not only against each other but also against established benchmarks like heuristics and MPC provides a more credible measure of their true value.

List of Acronyms

Acronym	**Description**
Artificial Intelligence & Control	
A2C	Advantage Actor-Critic
AC	Actor-Critic
AI	Artificial Intelligence
AL-SAC	Augmented Lagrangian Soft Actor-Critic
ARS	Augmented Random Search
CL	Curriculum Learning
CMDP	Constrained Markov Decision Process
DDPG	Deep Deterministic Policy Gradient

Acronym	Description
DQN	Deep Q-Networks
DRL	Deep Reinforcement Learning
LQR	Linear Quadratic Regulator
LSTM	Long Short-Term Memory
MARL	Multi-Agent Reinforcement Learning
MDP	Markov Decision Process
MILP	Mixed-Integer Linear Program
MPC	Model Predictive Control
NN	Neural Network
PER	Prioritized Experience Replay
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
SAC	Soft Actor-Critic
TD3	Twin-Delayed Deep Deterministic Policy Gradient
TQC	Truncated Quantile Critics
TRPO	Trust Region Policy Optimization
Electric Vehicles & Charging	
AFAP	As Fast As Possible (Heuristic)
ALAP	As Late As Possible (Heuristic)
CAFA	Charge As Fast As Possible
CALA	Charge As Late As Possible
CPO	Charge Point Operator
EV	Electric Vehicle
G2V	Grid-to-Vehicle
SCP	Scheduled Charging Power
SoC	State of Charge
SoH	State of Health
V2B	Vehicle-to-Building
V2G	Vehicle-to-Grid
V2H	Vehicle-to-Home
V2M	Vehicle-to-Microgrid
V2V	Vehicle-to-Vehicle
VPP	Virtual Power Plant
Power Grid & Energy Markets	
ACE	Area Control Error
ARR	Area Regulation Requirement
DER	Distributed Energy Resources
DR	Demand Response
RES	Renewable Energy Sources
Metrics & Technical Parameters	
DC	Constant Current (charging phase)
CV	Constant Voltage (charging phase)
DoD	Depth of Discharge
MSE	Mean Square Error
OU	Ornstein-Uhlenbeck (stochastic process)

Acronym	**Description**
RMSE	Root Mean Square Error

Chapter 1

Introduction

The shift toward electric mobility is a pivotal element in global strategies for decarbonizing transport. However, the large-scale integration of electric vehicles into existing power grids presents a complex set of challenges and opportunities, which this thesis aims to investigate.

1.0.1 Background and Relevance of Electric Vehicles and Vehicle-to-Grid

The surging Electric Vehicle (EV) market is accelerating a major reconfiguration of modern mobility, promising to lower carbon emissions and foster greater energy efficiency[1]. This evolution is more than a technological trend: it underpins environmental sustainability by reducing dependence on fossil resources, alleviating the impacts of climate change through diminished greenhouse gas emissions, and improving air quality in densely populated areas. However, embedding millions of EVs into existing power systems is a non-trivial task. It can intensify peak demand, place additional stress on transmission and distribution networks, and trigger side effects such as voltage irregularities or higher line losses[2].
It is in this context that the **Vehicle-to-Grid (V2G)** concept emerges as a strategic solution. Through bidirectional power exchange, V2G redefines EVs: no longer passive electrical loads, but mobile and flexible energy assets, able to deliver a spectrum of services to the power system[3]. This potential becomes even more compelling when one considers that, on average, EVs remain parked and unused for nearly 96% of the day, offering an ample time window to actively engage with the grid[4]. A further distinctive benefit lies in the rapid responsiveness of EV batteries, which makes them especially suitable for ancillary services demanding quick interventions, such as frequency regulation[5]. Alongside V2G, other schemes of bidirectional power flow have been proposed, each with its own scope:

1. **Vehicle-to-Home (V2H)**, where an EV sustains household demand during

[1]Orfanoudakis, Diaz-Londono, Yilmaz, et al. 2022.
[2]Orfanoudakis, Diaz-Londono, Yilmaz, et al. 2022; Salvatti et al. 2020.
[3]Alfaverh, Denaï, and Sun 2022.
[4]Evertsson and Nylander 2024.
[5]Alfaverh, Denaï, and Sun 2022.

outages or periods of elevated prices, strengthening domestic energy resilience;

2. **Vehicle-to-Building (V2B)**, extending this logic to commercial or industrial facilities, enabling EVs to support load management and improve consumption efficiency; and

3. **Vehicle-to-Vehicle (V2V)**, which allows direct power transfer among EVs, a valuable feature for emergency charging or shared resources.

Taken together, these modalities highlight the versatility of EV batteries as distributed energy units, reinforcing both energy resilience and the transition toward a more sustainable energy ecosystem.

1.0.2 Challenges in EV Integration into the Electricity Grid and the Role of Artificial Intelligence

Modern electricity systems are increasingly shaped by the penetration of intermittent **Renewable Energy Sources (RESs)** such as wind and solar. Their variability causes pronounced swings in power generation and persistent mismatches between supply and demand, which in turn fuels price volatility and complicates dispatch strategies. This continuously puts the stability and economic efficiency of the grid under strain. Managing these fluctuations, while making rapid operational choices to balance the system and minimize costs, has proven difficult for conventional control frameworks[6].

The parallel rise of EV adoption and RES deployment has produced an environment marked by both uncertainty and complexity. In such conditions, traditional approaches are increasingly inadequate, prompting a growing reliance on methods rooted in artificial intelligence—and particularly in **Reinforcement Learning (RL)**. This shift alters the very nature of the grid: from a relatively predictable and centralized infrastructure to one that is decentralized, stochastic, and highly dynamic. Rule-based or deterministic controllers, designed for a past paradigm, are ill-suited to cope with this degree of volatility. The result is a pressing need for more adaptive and intelligent decision-making mechanisms. This transformation extends beyond the simple challenge of absorbing extra load or integrating new generators: it signals a genuine paradigm change towards a *smart grid*[7], where adaptive, real-time, and autonomous operation is no longer optional but vital to preserve efficiency, resilience, and reliability. From this perspective, RL is not just an optimization tool, but an enabling technology for a more cognitive and robust energy infrastructure—one capable of navigating the uncertainties of a decarbonized, electrified future.

Against this backdrop, **Deep Reinforcement Learning (DRL)** has gained attention as an especially powerful approach. Its capacity to derive near-optimal strategies in dynamic and uncertain environments—without requiring a precise model of

[6] Orfanoudakis, Diaz-Londono, Yilmaz, et al. 2022; Minchala-Ávila, Arévalo, and Ochoa-Correa 2025.

[7] Al-HMOUD and Al-Raweshidy 2024.

the system or flawless forecasts—makes DRL particularly well-suited for EV integration and advanced grid management[8].

1.0.3 Objectives and Contributions of the Thesis

This thesis confronts the complex multi-objective optimization problem at the heart of Vehicle-to-Grid (V2G) systems. The overarching objective is to move beyond a purely theoretical analysis by actively developing, testing, and enhancing a high-fidelity simulation architecture. This platform serves as a digital twin to rigorously evaluate and compare advanced control strategies, balancing economic benefits, user mobility needs, battery health, and grid stability under realistic stochastic conditions.
More than a simple review of existing literature, this work focuses on the practical implementation and validation of a V2G simulation framework in Python. This tool is leveraged to demonstrate and explore novel perspectives for training intelligent agents. The main contributions are:

- **Enhancement of a V2G Simulation Architecture:** A key contribution is the systematic testing, validation, and enhancement of the **EV2Gym** simulation framework. This work solidifies its role as a robust and flexible platform for benchmarking control algorithms, ensuring that the models for battery physics, user behavior, and grid dynamics are coherent and realistic for advanced research.

- **Exploration of Novel Reinforcement Learning Perspectives:** The validated simulation environment is used to investigate and implement advanced training methodologies for RL agents. A key focus is placed on techniques like **adaptive reward shaping**, where the reward function dynamically evolves during training to guide the agent towards a more holistic and robust control policy, overcoming the limitations of static reward definitions.

- **Practical Implementation of Advanced Control Paradigms:** The thesis demonstrates the practical transition from a theoretical, offline optimal controller to a realistic, online controller. Specifically, it details the implementation of an **offline MPC using Gurobi**, which acts as a "judge" with perfect foresight, and contrasts it with an **online MPC formulated in PuLP**, designed to operate as a real-time "controller" with limited future information, highlighting the trade-offs and challenges of real-world deployment.

[8] Orfanoudakis, Diaz-Londono, Yilmaz, et al. 2022; Shibl, Ismail, and Massoud 2023.

1.0.4 Thesis Structure

The remainder of this thesis is organized as follows:

- **Chapter 2: Overview of Optimal Management of EV Charging and Discharging** provides foundational knowledge on V2G technology, the complex multi-objective nature of EV charging optimization, and presents a comprehensive review of state-of-the-art research approaches.

- **Chapter 3: The V2G Simulation Framework: A Digital Twin for V2G Research** details the architecture and core models of the simulation environment. This chapter describes the enhancements made to the framework, establishing it as the central experimental platform for implementing and evaluating the control agents analyzed in this work.

- **Chapter 4: Experimental Campaign and Results Analysis** This chapter presents the results of the comparative analysis between the different control strategies (DRL, MPC, heuristics). It analyzes the performance of novel training techniques and discusses the implications of the findings.

- **Bibliography** lists all cited references.

Chapter 2

State of the Art in Optimal V2G Management

2.1 The V2G Imperative: A Cornerstone of Europe's Green Transition

Society finds itself at a critical juncture, grappling with the twin revolutions of decarbonizing transport and reshaping our energy systems. This is not merely an ambition but a legally binding mandate, enshrined in frameworks like the **European Green Deal** and its ambitious **"Fit for 55"** package[1]. These policies impose a rapid phase-out of internal combustion engines and demand a massive scale-up of renewable energy sources, as detailed in the revised Renewable Energy Directive (RED III). At the very nexus of this challenge lies the proliferation of Electric Vehicles (EVs).

Initially viewed with apprehension—a looming threat of massive, synchronized loads poised to destabilize fragile distribution networks—that perception is now obsolete. Today, EVs must be seen not as a problem, but as a foundational pillar of the solution. This change in perspective is embodied in the concept of **Vehicle-to-Grid (V2G)**. V2G acts as the critical enabling technology that can transform millions of EVs from passive consumers into active, distributed, and intelligent assets for the grid. The key lies hidden in plain sight: private vehicles remain parked and connected for an astonishing 96% of their existence[2], representing a potential of terawatt-hours of mobile storage waiting to be harnessed.

The true power of V2G emerges not from the individual, but from the collective. A single EV's contribution is a whisper, but a coordinated fleet, managed by an aggregator, becomes a roar—a **Virtual Power Plant (VPP)**. This collective entity, with the lightning-fast response of battery inverters, can deliver a spectrum of critical services. This capability is the linchpin for stabilizing a grid increasingly reliant on the fluctuating whims of wind and sun, making the high renewable penetration targets of the EU feasible. The services unlocked by this capability are foundational to building the smart, resilient grid of tomorrow:

[1] *'Fit for 55': delivering the EU's 2030 Climate Target on the way to climate neutrality* 2021.
[2] Evertsson and Nylander 2024.

- **Frequency Regulation:** The grid's heartbeat. V2G fleets can inject or absorb power in seconds, instantly counteracting supply-demand imbalances to maintain the stable 50/60 Hz frequency, preventing cascading failures and blackouts[3].

- **Demand Response and Peak Shaving:** By intelligently shifting charging to off-peak hours and discharging during peak demand, V2G flattens the load curve. This reduces our reliance on expensive and polluting "peaker" plants and can defer trillions in grid infrastructure upgrades[4].

- **Renewable Energy Integration:** Perhaps its most profound impact, V2G fleets can act as a giant, distributed sponge, absorbing surplus solar and wind energy that would otherwise be curtailed, and releasing it when the sun sets or the wind dies down. This directly supports the integration goals of RED III and mitigates intermittency[5].

This vision is no longer a distant prospect; it is actively being codified into European law and technical standards. The landmark **Alternative Fuels Infrastructure Regulation (AFIR, EU 2023/1804)** now mandates that new public charging infrastructure must support smart and bidirectional charging. This legal requirement is given its technical teeth by specific standards, as a delegated regulation specifies that from 2027, charging points must comply with **ISO 15118-20**—a standard that explicitly defines the communication protocols for bidirectional power transfer. This regulatory push is complemented by large-scale pilot projects like **'SCALE'** and **'V2G Balearic Islands'**, which are testing the technology's technical and economic viability on an industrial scale.

However, while the regulatory foundation is being laid, significant barriers to widespread adoption remain, creating a complex landscape that technology and policy must navigate together. Key challenges include:

- **Market and Economic Hurdles:** A clear, pan-European framework for remunerating EV owners for grid services is still absent. Critical issues like the **"double taxation"** of electricity—taxed both on charging and discharging—create significant economic disincentives that must be resolved.

- **Regulatory and Grid Access Rules:** The role of EV fleets as a flexibility resource is not yet uniformly recognized in electricity markets. Standardized procedures for grid connection, aggregator certification, and secure data exchange are still under development, hindering market access.

- **Technical and Consumer Barriers:** On the consumer side, concerns about accelerated **battery degradation** and its impact on vehicle warranties remain a primary obstacle. Furthermore, the reality is that not all EVs or chargers are currently equipped with the necessary hardware and software to support V2G.

[3] Alfaverh, Denaï, and Sun 2022; Sadeghi 2021.
[4] Orfanoudakis, Diaz-Londono, Yilmaz, et al. 2022.
[5] Khan et al. 2024; Xie 2021.

The central challenge, which is the focus of this thesis, is therefore not simply about enabling V2G, but about orchestrating it *intelligently.* This requires a control strategy sophisticated enough to operate within this nascent regulatory framework, navigate its economic uncertainties, and overcome technical constraints to unlock the immense potential of EVs as a cornerstone of a sustainable energy future.

2.2 The Optimizer's Trilemma: Navigating a Stochastic World

While the potential is immense, orchestrating this symphony of distributed assets presents a formidable challenge. An aggregator's primary driver is economic viability, but pursuing profit in isolation is a recipe for failure. Optimal V2G management is a delicate balancing act, a genuine multi-objective optimization problem often framed as the "V2G trilemma": the simultaneous pursuit of **economic profitability**, the preservation of **battery longevity**, and the guarantee of **user convenience**.

This is not a simple trade-off but a dynamic problem steeped in **stochasticity** and **uncertainty** from multiple sources:

- **Market Volatility:** Electricity prices can fluctuate wildly based on unpredictable supply and demand.
- **Renewable Intermittency:** The output of solar and wind generation is inherently variable.
- **Human Behavior:** EV owners' arrival times, departure times, and energy needs are not deterministic; a driver might need to leave unexpectedly, a non-negotiable constraint that any intelligent system must respect.

Such a chaotic environment renders static, rule-based control systems obsolete. We need an approach that can learn, adapt, and make intelligent decisions in real-time under profound uncertainty. This is precisely the domain where Reinforcement Learning excels.

2.3 A New Paradigm for Control: Reinforcement Learning

To tackle the V2G challenge, this work turns to Reinforcement Learning (RL), a domain of machine learning focused on how an intelligent agent can learn to make optimal decisions through trial and error. Unlike traditional methods that require a perfect model of the world, RL learns directly from interaction, making it exceptionally robust.

2.3.1 The Language of Learning: Markov Decision Processes (MDPs)

The mathematical bedrock of RL is the **Markov Decision Process (MDP)**, formally defined by the tuple (S, A, p, R, γ). In the V2G context, these elements represent:

- S: The state (a snapshot of the world: battery levels, electricity price, time).
- A: The action (the decision: the charging/discharging rate for each EV).
- $p(s', r|s, a)$: The environment's response (the probability of transitioning to a new state s' and receiving reward r).
- R: The reward (the feedback signal: profit generated, penalty for user dissatisfaction).
- γ: The discount factor, which balances immediate versus future rewards.

This framework rests on the **Markov Property**, which posits that the future is independent of the past given the present, allowing the agent to make decisions based solely on the current state.

2.3.2 Judging the Future: Value Functions and Actor-Critic Architectures

The agent's goal is to learn a **policy**, $\pi(a|s)$, which is a strategy for choosing actions. To achieve this, it learns **value functions** that estimate the long-term value of being in a certain state ($v_\pi(s)$) or taking a specific action in a state ($q_\pi(s, a)$).
The **Actor-Critic** architecture offers an elegant way to learn this policy. It maintains two distinct neural networks:

- **The Critic**: It learns the value function. Its job is to judge the actor's decisions.
- **The Actor**: It represents the policy itself. Its job is to select actions, using the critic's feedback to refine its strategy over time.

This architecture is particularly potent for V2G because it can directly learn a policy over a continuous action space, allowing for fine-grained control of power. The agent's entire behavior, however, is shaped by the reward signal it receives. The complex art of designing this signal to align the agent's goals with our multifaceted objectives is a critical discipline in itself, known as reward engineering.

2.4 Reward Engineering: Shaping Agent Behavior

The reward function is arguably the most critical component in a Reinforcement Learning system. It is the sole mechanism through which a designer can communicate the desired behavior to the agent. A poorly designed reward can lead to unintended and suboptimal behaviors, even if the agent successfully maximizes it. The

V2G problem, with its multiple competing objectives—maximizing profit, ensuring user satisfaction, preserving battery health, and maintaining grid stability—makes reward design particularly challenging. This work explores several sophisticated techniques to guide the learning process effectively.

2.4.1 Potential-Based Reward Shaping (PBRS)

Potential-Based Reward Shaping (PBRS) is perhaps the most theoretically sound method for reward augmentation[6]. It involves adding a shaping term, $F(s, s')$, to the environment's intrinsic reward, $R(s, a, s')$. This shaping term is defined as the difference between a potential function, Φ, evaluated at the new state and the old state:

$$F(s, s') = \gamma\Phi(s') - \Phi(s)$$

where γ is the discount factor. The key theoretical guarantee of PBRS is **policy invariance**: adding a potential-based shaping reward does not change the optimal policy of the underlying MDP. While the agent learns the same optimal behavior, the shaping term can provide dense, intermediate rewards that significantly accelerate the learning process by guiding the agent's exploration.

2.4.2 Dynamic and Adaptive Rewards

Unlike PBRS, which provides a static reward bonus, a dynamic or adaptive reward function can evolve during training. This approach is particularly useful for complex problems where the relative importance of different objectives may change as the agent becomes more competent. For example, an agent could initially be rewarded simply for keeping an EV charged, but as training progresses, the reward function could adapt to introduce penalties for grid overloads or incentives for V2G services[7]. This allows the agent to master different facets of the problem sequentially.

2.4.3 Curriculum Learning

While technically a training paradigm rather than a reward modification technique, Curriculum Learning (CL) is highly relevant to reward engineering. CL involves training the agent on a sequence of tasks that gradually increase in difficulty. In the V2G context, an agent might first be trained in a simple scenario with only a few EVs and stable prices. Once it masters this, it is moved to a more complex environment with more EVs, volatile prices, and hard constraints. This structured learning process prevents the agent from being overwhelmed by the full complexity of the problem from the start and can lead to more robust and generalizable policies.

[6] Ng, Harada, and Russell 1999.
[7] Wan et al. 2022.

2.5 The Rise of Deep Reinforcement Learning for V2G Control

Fusing RL with the representational power of deep neural networks gives rise to **Deep Reinforcement Learning (DRL)**, which currently represents the forefront of V2G control. The evolution of DRL algorithms for V2G has produced a sophisticated and robust toolkit, primarily branching into two main families: off-policy and on-policy methods, each with its own philosophy and trade-offs.

2.5.1 Off-Policy Methods: Data-Efficient Learning from Experience

Off-policy algorithms are distinguished by their ability to learn the optimal policy from data generated by a different, often more exploratory, policy. This decoupling allows them to reuse past experiences stored in a *replay buffer*, making them highly sample-efficient and well-suited for complex problems where real-world interaction is costly.

Deep Deterministic Policy Gradient (DDPG) A seminal algorithm that extended the success of Deep Q-Networks (DQN) to continuous action spaces, DDPG was a foundational breakthrough for control problems like V2G[8]. As an Actor-Critic method, it learns a deterministic policy (the Actor) that maps states to specific actions, guided by a Q-value function (the Critic). However, its practical application is often hampered by training instability and a crippling vulnerability to **overestimation bias**, where the Critic systematically overestimates Q-values. This error propagates through the learning process, leading the Actor to converge on suboptimal policies[9].

Twin Delayed DDPG (TD3) TD3 was engineered specifically to counteract the instabilities of DDPG[10]. It introduces three crucial innovations:

1. **Clipped Double Q-Learning:** It learns two independent Critic networks and uses the minimum of their Q-value predictions to calculate the target value. This conservative approach effectively mitigates overestimation bias.
2. **Delayed Policy Updates:** The Actor and target networks are updated less frequently than the Critic. This allows the Critic's value estimate to stabilize before the policy is modified, leading to smoother and more reliable training.
3. **Target Policy Smoothing:** A small amount of clipped noise is added to the target action, which helps to regularize the learning process and prevent the policy from exploiting narrow peaks in the value function.

These additions make TD3 a much more robust and reliable baseline for V2G tasks[11].

[8] Lillicrap et al. 2015.
[9] Orfanoudakis, Diaz-Londono, Yilmaz, et al. 2022; Alfaverh, Denaï, and Sun 2022.
[10] Fujimoto, Hoof, and Meger 2018.
[11] Z. Liu et al. 2023; Siyuan Wang, Shuo Wang, and B. Liu 2022.

Soft Actor-Critic (SAC) SAC stands at the cutting edge of continuous control, offering superior sample efficiency and stability[12]. Its core innovation is the **maximum entropy framework**. Here, the agent's objective is not just to maximize cumulative reward, but to do so while acting as randomly (stochastically) as possible. This entropy bonus encourages broad exploration, preventing premature convergence to a narrow, suboptimal policy, and improves robustness by teaching the agent to "keep its options open"[13].

Truncated Quantile Critics (TQC) TQC tackles overestimation bias from a distributional perspective, presenting a more fundamental solution than TD3[14]. Instead of learning a single expected return (a Q-value), it learns the entire *distribution of returns* using quantile regression with multiple Critic networks. Its key mechanism is to "truncate" (discard) the top-k most optimistic quantile estimates when forming the target distribution, thereby systematically removing the primary source of overestimation bias.

Enhancement: Prioritized Experience Replay (PER) This is not a standalone algorithm but a crucial modification for off-policy methods. Rather than sampling uniformly from the replay buffer, PER samples transitions with a probability proportional to their "importance," typically measured by the magnitude of their TD error. This focuses the learning process on "surprising" or informative experiences, significantly accelerating convergence[15].

2.5.2 On-Policy Methods: Stability through Cautious Updates

On-policy methods learn from data generated exclusively by the current policy. This means that after each policy update, all previously collected data must be discarded. While this makes them inherently less sample-efficient, their updates are often more stable and less prone to divergence.

Advantage Actor-Critic (A2C/A3C) A2C is a foundational on-policy Actor-Critic algorithm. Its practical and powerful extension, **Asynchronous Advantage Actor-Critic (A3C)**, employs parallel workers to interact with multiple copies of the environment. These workers update a global set of parameters asynchronously, which decorrelates the data stream and provides a powerful stabilizing effect on the learning process[16].

Trust Region Policy Optimization (TRPO) TRPO was the first algorithm to rigorously formalize the idea of controlling the policy update size to guarantee sta-

[12] Haarnoja et al. 2018.
[13] Logeshwaran, Fan, and Naung 2022.
[14] Kuznetsov et al. 2020.
[15] Schaul et al. 2015.
[16] Mnih et al. 2016.

ble, monotonic improvements[17]. It maximizes a surrogate objective function subject to a constraint on the "behavioral change" of the policy, measured by the Kullback-Leibler (KL) divergence. This creates a "trust region" around the old policy, preventing catastrophic updates that could destroy performance. Its implementation, however, is complex as it requires second-order optimization.

Proximal Policy Optimization (PPO) PPO achieves the stability benefits of TRPO using only first-order optimization, making it far simpler to implement and more broadly applicable[18]. Instead of a hard constraint, PPO modifies the objective function with a **clipping** mechanism that disincentivizes policy updates resulting in a large probability ratio between the new and old policies. This creates a "soft" trust region and has become a default choice for many on-policy applications due to its robustness and ease of use.

2.5.3 Gradient-Free Methods: An Alternative Path

Augmented Random Search (ARS) ARS is an on-policy, gradient-free method that optimizes the policy by operating directly in the parameter space[19]. Instead of calculating gradients, it explores random directions around the current policy parameters and updates them based on observed performance. While often much less sample-efficient than gradient-based methods for complex V2G problems, its simplicity can make it competitive in certain scenarios.

2.6 The Model-Based Benchmark: Model Predictive Control (MPC)

While DRL offers a powerful model-free approach, it must be benchmarked against its most robust model-based counterpart: **Model Predictive Control (MPC)**[20]. MPC is an advanced control method that uses an explicit model of the system to predict its future evolution and compute an optimal control sequence over a finite prediction horizon, N. Its primary strength is the ability to proactively handle complex dynamics and operational constraints.

2.6.1 Implicit MPC: Online Optimization

The most common formulation of MPC is **Implicit MPC**, where a detailed optimization problem is solved online at each control step. For a linear time-invariant system, this problem is typically a Quadratic Program (QP).
The controller's objective is to find a sequence of future control inputs $U = [u_{t|t}, ..., u_{t+N-1|t}]$ that minimizes a cost function J, which penalizes deviations from a desired state and the control effort itself.

[17]Schulman, Levine, et al. 2015.
[18]Schulman, Wolski, et al. 2017.
[19]Mania, Guy, and Recht 2018.
[20]Minchala-Ávila, Arévalo, and Ochoa-Correa 2025.

$$\min_U J(x_t, U) = \sum_{k=0}^{N-1} (x_{t+k|t}^T Q x_{t+k|t} + u_{t+k|t}^T R u_{t+k|t}) + x_{t+N|t}^T P x_{t+N|t} \tag{2.1}$$

where $x_{t+k|t}$ is the predicted state at future time k based on information at current time t, and Q, R, and P are weighting matrices.
This optimization is subject to critical constraints that define the system's valid operating envelope:

- **System Dynamics:** The model that predicts the next state.

$$x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t} \tag{2.2}$$

- **State and Input Constraints:** Physical or operational limits.

$$x_{min} \leq x_{t+k|t} \leq x_{max} \tag{2.3}$$
$$u_{min} \leq u_{t+k|t} \leq u_{max} \tag{2.4}$$

At each time step t, this problem is solved to find the optimal control sequence U^*. Only the first action, $u^*_{t|t}$, is applied to the system. The entire process is then repeated at the next time step, $t+1$, using new state measurements—a principle known as a *receding horizon*. This constant re-evaluation gives MPC its feedback mechanism and robustness to disturbances.

2.6.2 Explicit MPC: Offline Pre-computation

For systems with fast dynamics or limited online computational power, **Explicit MPC** offers an alternative. In this paradigm, the optimization problem is solved offline for all possible states within the operating range using multi-parametric programming.
The result is a pre-computed, explicit control law, $\pi(x_t)$, which is a **piecewise affine function** of the state vector x_t. The state space is partitioned into a set of distinct polyhedral regions, $\mathcal{X}_i$, each with its own corresponding optimal control law.

$$u^*(x_t) = F_i x_t + g_i \quad \text{if } x_t \in \mathcal{X}_i \tag{2.5}$$

Here, F_i is the gain matrix and g_i is the offset for region i. The online operation is reduced to two simple steps:

1. Identify which region $\mathcal{X}_i$ the current state x_t belongs to (a fast lookup procedure).

2. Apply the corresponding pre-computed affine control law.

This eliminates the need for a powerful online solver but comes at the cost of a potentially very high offline computation burden and significant memory requirements to store the lookup table of control laws.

2.7 A Comparative Perspective on Control Methodologies

While DRL represents the cutting edge, it is crucial to contextualize it within the broader landscape.

Table 2.1: Comparative Analysis: DRL vs. Model Predictive Control (MPC) for V2G

Aspect	**Deep Reinforcement Learning (DRL)**	**Model Predictive Control (MPC)**
Paradigm	Model-Free, learning-based. Learns optimal policy via trial-and-error.	Model-Based, optimization-based. Solves an optimization problem at each step.
Strengths	• Highly robust to uncertainty and stochasticity. • No need for an explicit system model. • Can learn complex, non-linear control policies. • Fast inference time once trained.	• Explicitly handles hard constraints (safety guarantees). • Proactive and anticipatory if forecasts are accurate. • Well-established and understood.
Weaknesses	• Can be sample-inefficient during training. • Lacks hard safety guarantees (an active research area). • "Black box" nature can make policies hard to interpret.	• Performance is fundamentally tied to model and forecast accuracy. • Computationally expensive at each time step (curse of dimensionality). • Brittle to forecast errors and unmodeled dynamics.
V2G Suitability	Excellent for dynamic, uncertain environments with complex trade-offs.	Good for problems with simple dynamics and reliable forecasts, but struggles with real-world V2G complexity.

Model Predictive Control (MPC) is the most powerful model-based alternative[21]. Its primary strength is its ability to handle constraints. However, its performance is fundamentally shackled to the accuracy of its internal model and forecasts[22]. In the V2G domain, creating an accurate model is nearly impossible due to non-linear battery dynamics, market volatility, and human unpredictability. Furthermore, solving the large-scale Mixed-Integer Linear Program (MILP) required at each time step becomes computationally intractable for large fleets[23].

Other methods, such as **meta-heuristic algorithms** (e.g., genetic algorithms), are typically used for offline scheduling and lack the real-time responsiveness required for dynamic V2G control[24].

Ultimately, the singular advantage of DRL lies in its native ability to learn and internalize the complex, non-linear trade-offs of the multi-objective V2G problem directly from data. This makes it uniquely suited to navigating the uncertainties of the real world. While other methods have their place, DRL stands out as the most promising technology for deploying the truly intelligent, autonomous, and robust V2G management systems required to achieve the ambitious energy and climate goals of the European Union.

2.8 A Primer on Lithium-Ion Battery Chemistries and Degradation

The effectiveness of any V2G strategy is, at its core, constrained by the physical characteristics of the vehicle's battery. The choice of battery chemistry is not a minor detail; it dictates the operational envelope of the EV, influencing its energy density, power capabilities, lifespan, and, critically, its safety. A clear understanding of these trade-offs is essential to the development of robust and realistic control algorithms. This section provides an overview of the primary degradation mechanisms, the most prevalent lithium-ion chemistries, and the core concepts governing their performance.

2.8.1 Fundamental Concepts and Degradation Mechanisms

Battery degradation is an irreversible process that reduces a battery's capacity (energy fade) and increases its internal resistance (power fade). It can be broadly categorized into two types: calendar aging and cyclic aging[25].

- **Calendar Aging:** This refers to degradation that occurs whenever the battery is at rest, even when not in use. The primary mechanism is the slow, continuous growth of the **Solid Electrolyte Interphase (SEI)** layer on the anode surface. The SEI is a necessary passivation layer that forms during the first few cycles, but its continued growth consumes active lithium ions and

[21] Alsabbagh and Siu 2022.
[22] Faggio 2023.
[23] Schwenk et al. 2022.
[24] V. Kumar, Singh, and D. Kumar 2024.
[25] Birkl et al. 2017.

electrolyte, leading to irreversible capacity loss and increased impedance. The rate of SEI growth is strongly accelerated by two factors:

  - **High Temperature:** Higher temperatures increase the rate of chemical reactions, causing the SEI to grow faster.
  - **High State of Charge (SoC):** A high SoC corresponds to a low anode potential, which makes the anode more reactive with the electrolyte, thus promoting SEI growth[26]. Storing a battery at 100% SoC, especially in a hot environment, is one of the most significant contributors to calendar aging.

- **Cyclic Aging:** This degradation occurs as a direct result of charging and discharging the battery. Key mechanisms include:
  - **Mechanical Stress:** During intercalation and de-intercalation, the active materials in the electrodes expand and contract. Over many cycles, this repeated mechanical stress can cause micro-cracks in the electrode particles, leading to a loss of electrical contact and capacity. This effect is more pronounced with larger **Depths of Discharge (DoD)**.
  - **SEI Layer Instability:** The volume changes during cycling can also crack the protective SEI layer, exposing fresh anode material to the electrolyte. This triggers the formation of new SEI, consuming more lithium in the process.
  - **Lithium Plating:** Under conditions of high charging rates (high C-rate) and/or low temperatures, lithium ions may not have sufficient time to properly intercalate into the graphite anode. Instead, they deposit on the anode surface as metallic lithium. This is highly detrimental as it causes rapid, irreversible capacity loss and can form needle-like structures called dendrites, which can pierce the separator and cause an internal short circuit, posing a severe safety risk[27].

For V2G applications, which inherently involve frequent charge/discharge cycles, understanding and mitigating cyclic aging is paramount.

2.8.2 Key Automotive Chemistries

The EV market is dominated by a few key families of lithium-ion batteries, primarily distinguished by their cathode materials.

- **Lithium Nickel Manganese Cobalt Oxide (NMC):** A highly popular choice due to its balanced performance. By adjusting the ratio of Nickel, Manganese, and Cobalt, manufacturers can tailor the battery to prioritize either energy density (higher Nickel content, e.g., NMC811) or safety and longevity (higher Manganese/Cobalt content, e.g., NMC532).

[26] Vetter et al. 2005.
[27] Birkl et al. 2017.

- **Lithium Nickel Cobalt Aluminum Oxide (NCA):** Similar to NMC but uses Aluminum instead of Manganese. This chemistry, famously used by Tesla for many years, offers very high energy density, enabling longer ranges, but at the cost of slightly lower cycle life and safety margins compared to NMC.

- **Lithium Iron Phosphate (LFP):** This chemistry is rapidly gaining market share. It contains no cobalt, making it cheaper and more ethically sourced. LFP batteries offer exceptional cycle life and are considered the safest among common Li-ion types. Their main drawbacks are lower nominal voltage and lower energy density.

- **Lithium Titanate Oxide (LTO):** LTO batteries use a titanate anode. They are exceptional in terms of safety, cycle life (>10,000 cycles), and low-temperature performance. However, their very low energy density and high cost make them a niche solution.

2.8.3 Voltage Profiles and the Challenge of SoC Estimation

The relationship between a battery's voltage and its SoC is a critical, non-linear function. The derivative of the cell voltage with respect to the DoD, $\frac{dV_{cell}}{d(DoD)}$, is a crucial parameter for the Battery Management System (BMS). A steep, consistent slope allows the BMS to accurately infer the SoC from a voltage measurement. Conversely, a flat slope ($\frac{dV_{cell}}{d(DoD)} \approx 0$) makes this estimation extremely difficult, as a small voltage measurement error can translate into a massive SoC error.

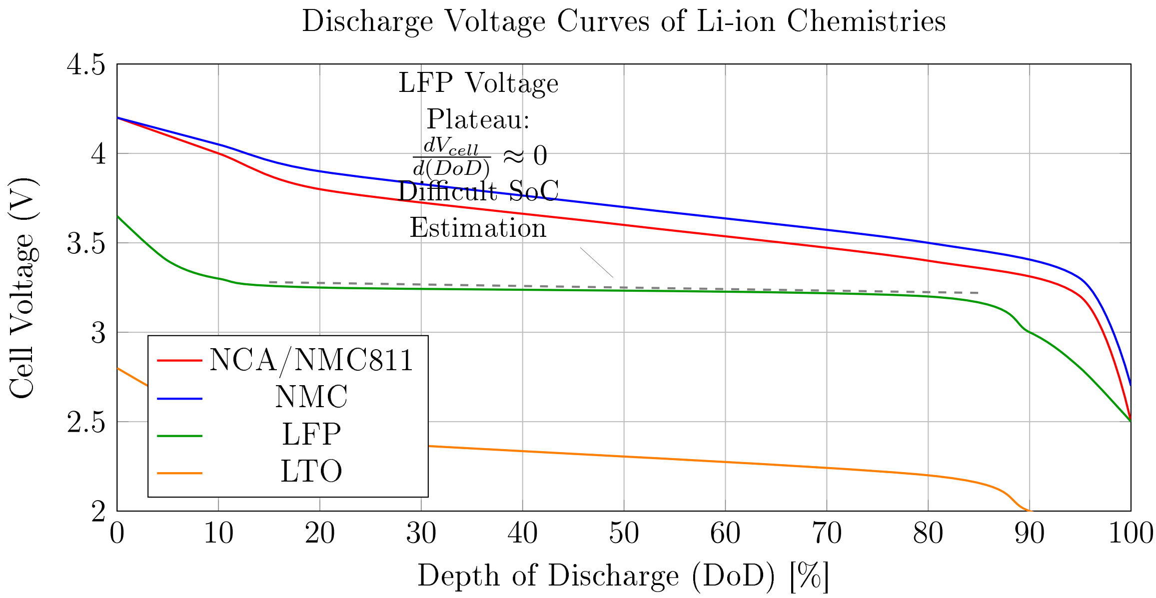

Figure 2.1: Typical discharge voltage curves for various lithium-ion chemistries. The flat profile of LFP makes accurate SoC estimation challenging based on voltage alone[29].

As shown in Figure 2.1, LFP's flat voltage plateau makes it difficult for the BMS to determine the precise SoC in the central part of its operating range. This necessitates periodic full charges to 100% to recalibrate the system at a point where the voltage curve is steep again, an important operational constraint for V2G control strategies.

2.8.4 Comparative Analysis and Safety Considerations

The trade-offs between chemistries are summarized in Table 2.2. Safety is paramount, and the primary risk is thermal runaway. The risk is directly related to the stored energy density ($\Delta E/\Delta m$). A higher energy density means more energy is packed into a smaller mass, which can be released violently if the cell's structure is compromised. Consequently, the critical temperature for initiating thermal runaway is generally lower for higher energy density chemistries. As energy density decreases, the thermal stability increases.

Table 2.2: Comparative analysis of key automotive battery chemistries[30].

Metric	**NCA**	**NMC**	**LFP**	
Energy Density (Wh/kg)	200 - 260 (Highest)	150 - 220 (High)	90 - 160 (Moderate)	6
Cycle Life	1000 - 2000	1000 - 2500	2000 - 5000+	
Safety	Good	Very Good	Excellent	
Thermal Runaway Temp (°C)	∼150 - 180	∼180 - 210	∼220 - 270	

2.8.5 Battery Pack Architecture

Individual cells are assembled into modules and packs using a series-parallel configuration, denoted as **XsYp**. 'X' cells in series determine the pack voltage ($V_{pack} = X \cdot V_{cell}$), which is typically 350-400V for modern EVs. 'Y' cells in parallel determine the pack capacity ($C_{pack} = Y \cdot C_{cell}$). While the physical form factor (cylindrical, prismatic, pouch) and BMS design are critical for engineering, our focus remains on the electrochemical degradation influenced by V2G control strategies.

Chapter 3

An Enhanced V2G Simulation Framework for Robust Control

Developing, validating, and benchmarking advanced control algorithms for Vehicle-to-Grid (V2G) systems is a complex endeavor. Real-world experimentation is often impractical due to prohibitive costs, logistical challenges, and risks to grid stability and vehicle hardware. To bridge the gap between theory and practice, a realistic, flexible, and standardized simulation environment is a scientific necessity. This thesis builds upon the foundation of **EV2Gym**, a state-of-the-art, open-source simulator designed for V2G smart charging research[1]. This work, however, extends the original framework significantly, transforming it into a high-fidelity **digital twin** engineered not just for single-scenario optimization, but for the development and rigorous evaluation of **robust, generalist control agents**.

This enhanced framework offers a two-pronged approach to experimentation: it allows for deep-dive analysis of agents specialized for a single environment, while also introducing a novel methodology for training and testing agents designed to generalize across a multitude of diverse, unpredictable scenarios. This chapter provides an in-depth tour of this extended architecture, its data-driven models, and its unique evaluation capabilities, establishing the methodological bedrock for the rest of this work.

3.1 Core Simulator Architecture

The framework is built on the modular architecture of EV2Gym, which mirrors the key entities of a real-world V2G system. Its foundation on the OpenAI Gym (now Gymnasium) API is a cornerstone, providing a standardized agent-environment interface defined by the familiar language of states, actions, and rewards[2].

The architecture consists of several interacting components:

- **Charge Point Operator (CPO):** The central intelligence of the simulation, managing the charging infrastructure and serving as the primary interface for

[1] Orfanoudakis, Diaz-Londono, Yïlmaz, et al. 2024.
[2] Brockman et al. 2016.

the control algorithm (the DRL agent). The CPO aggregates system state information and dispatches control actions to individual chargers.

- **Chargers:** Digital representations of physical charging stations, configurable by type (AC/DC), maximum power, and efficiency. This allows for the simulation of heterogeneous charging infrastructures.
- **Power Transformers:** These components model the physical connection points to the grid, aggregating the electrical load from multiple chargers. Crucially, they enforce the physical power limits of the local distribution network and can model inflexible base loads (e.g., buildings) and local renewable generation (e.g., solar panels).
- **Electric Vehicles (EVs):** Dynamic and autonomous agents, each defined by its unique battery capacity, power limits, current and desired energy levels, and specific arrival and departure times.

The simulation process follows a reproducible three-phase structure: (1) **Initialization** from a comprehensive YAML configuration file, (2) a discrete-time **Simulation Loop** where the agent interacts with the environment, and (3) a final **Evaluation and Visualization** phase that generates standardized performance metrics.

3.2 Core Physical Models

The simulation's fidelity is anchored in its detailed, empirically validated models, which are essential for developing control strategies robust enough for real-world application.

3.2.1 EV Model and Charging/Discharging Dynamics

The framework implements a realistic two-stage charging/discharging model that captures the non-linear behavior of lithium-ion batteries, simulating both the **constant current (CC)** and **constant voltage (CV)** phases. Each EV is defined by a rich parameter set: maximum capacity (E_{max}), a minimum safety capacity (E_{min}), separate power limits for charging and discharging ($P_{ch}^{max}, P_{dis}^{max}$), and distinct efficiencies for each process (η_{ch}, η_{dis}).

3.2.2 Battery Degradation Model

To address the critical issue of battery health in V2G operations, the simulator incorporates a semi-empirical battery degradation model. It quantifies capacity loss (Q_{lost}) as the sum of two primary aging mechanisms[3]:

- **Calendar Aging (d_{cal}):** Time-dependent capacity loss, influenced by the battery's average State of Charge (SoC) and temperature.

[3] Orfanoudakis, Diaz-Londono, Yïlmaz, et al. 2024.

- **Cyclic Aging (d_{cyc}):** Wear resulting from charge/discharge cycles, dependent on energy throughput, depth-of-cycle, and C-rate.

This integrated model allows for the direct quantification of how different control strategies impact the battery's long-term State of Health (SoH), enabling the training of agents that balance profitability with battery preservation.

3.2.3 EV Behavior and Grid Models

To ensure realism, the simulation is driven by authentic, open-source datasets. EV arrival/departure patterns and energy requirements are modeled using probability distributions derived from a large real-world dataset from **ElaadNL**. Grid conditions are similarly grounded in reality, using inflexible load data from the **Pecan Street** project and solar generation profiles from the **Renewables.ninja** platform[4].

3.3 A Dual-Pronged Evaluation Architecture

A key contribution of this thesis is the development of a sophisticated, two-mode evaluation pipeline that distinguishes between specialized and generalized agent performance. This is implemented through two primary execution scripts: `Single_Domain_Env.py` and `MultiScenarioEnv.py`.

3.3.1 Single-Domain Specialization

The `Single_Domain_Env.py` script is designed to train and evaluate "specialist" agents. In this workflow, a Reinforcement Learning agent is trained from scratch on a single, fixed configuration file. This approach is used to answer the question: "What is the optimal performance achievable for this specific, known environment?" It allows for a deep-dive analysis of an agent's ability to master one particular scenario, serving as a crucial baseline for performance.

3.3.2 Multi-Scenario Generalization

The `MultiScenarioEnv.py` script introduces a more challenging and realistic paradigm: training a single, "generalist" agent that must perform well across a diverse set of scenarios. This is achieved through two key innovations:

- **MultiScenarioEnv:** A custom Gymnasium environment that acts as a wrapper around multiple underlying `EV2Gym` instances. At the beginning of each training episode (i.e., on `reset()`), this environment randomly selects one of the provided configuration files. This forces the agent to learn a robust policy that is not overfitted to any single scenario's characteristics (e.g., number of chargers, grid capacity, or price volatility).

[4] Orfanoudakis, Diaz-Londono, Yïlmaz, et al. 2024.

- **CompatibilityWrapper:** A critical technical solution to handle the varying observation and action space sizes across different scenarios. Since a neural network policy has a fixed input and output size, this wrapper **pads** observations from smaller environments to a maximum size and **slices** action vectors from the agent to match the specific needs of the currently active environment. This enables a single agent to seamlessly control infrastructures of varying scales.

This multi-scenario training methodology is fundamental for developing agents that are truly robust and ready for deployment in the real world, where conditions are never static.

3.4 Software and Experimentation Workflow

The project's functionality is organized into a modular structure to facilitate clear and reproducible experimentation.

- `ev2gym/`: The core directory containing the simulator's heart.
  - `models/`: Defines the main environment (`ev2gym_env.py`) and the physical components (`ev.py`, `ev_charger.py`, `transformer.py`).
  - `baselines/`: Contains the classical control algorithms used for benchmarking, including heuristics (`heuristics.py`) and Model Predictive Control (`pulp_mpc.py`).
  - `rl_agent/`: Houses DRL-specific components, such as state space definitions (`state.py`) and reward functions (`reward.py`).
  - `data/`: Contains the input time-series data for EV arrivals, energy prices, and loads.
- `Compare.py`: A powerful utility script for pre-analysis and scenario comparison. It reads multiple YAML configuration files and generates summary tables and legends as images, allowing for a quick, visual comparison of experimental setups.
- `Single_Domain_Env.py`: The primary script for training and evaluating specialist agents on a single, user-selected scenario. It orchestrates the entire benchmark for one environment.
- `MultiScenarioEnv.py`: The script for training and evaluating robust, generalist agents. It utilizes the `MultiScenarioEnv` to train a single agent on a collection of scenarios and then evaluates its performance across each of them.

3.5 Evaluation Metrics

To ensure a fair and comprehensive comparison, all algorithms are evaluated against an identical set of pre-generated scenarios through a "replay" mechanism. The **mean** and **standard deviation** of performance are calculated across multiple simulation runs. The key metrics include:

- **Total Profit ($):** The net economic outcome, calculated as revenue from energy sales minus the cost of energy purchases.

$$\Pi_{\text{total}} = \sum_{t=0}^{T_{\text{sim}}} \sum_{i=1}^{N} \left(C_{\text{sell}}(t) P_{\text{dis},i}(t) - C_{\text{buy}}(t) P_{\text{ch},i}(t) \right) \Delta t$$

- **Tracking Error (RMSE, kW):** For grid-balancing scenarios, this measures the root-mean-square error between the fleet's aggregated power and a target setpoint.

$$E_{\text{track}} = \sqrt{\frac{1}{T_{\text{sim}}} \sum_{t=0}^{T_{\text{sim}}-1} \left(P_{\text{setpoint}}(t) - P_{\text{total}}(t) \right)^2}$$

- **User Satisfaction (Average):** The fraction of energy delivered compared to what was requested by the user, averaged across all EV sessions. A score of 1 indicates perfect service.

$$US_{\text{avg}} = \frac{1}{N_{\text{EVs}}} \sum_{k=1}^{N_{\text{EVs}}} \min\left(1, \frac{E_k(t_k^{\text{dep}})}{E_k^{\text{des}}}\right)$$

- **Transformer Overload (kWh):** The total energy that exceeded the transformer's rated power limit. An ideal controller should achieve a value of 0.

$$O_{\text{tr}} = \sum_{t=0}^{T_{\text{sim}}} \sum_{j=1}^{N_T} \max(0, P_j^{\text{tr}}(t) - P_j^{\text{tr,max}}) \cdot \Delta t$$

- **Battery Degradation ($):** The estimated monetary cost of battery aging due to both cyclic and calendar effects.

$$D_{\text{batt}} = \sum_{k=1}^{N_{\text{EVs}}} (\text{CyclicCost}_k + \text{CalendarCost}_k)$$

3.6 Reinforcement Learning Formulation

The control problem is formalized as a Markov Decision Process (MDP), defined by the tuple (S, A, P, R, γ).

3.6.1 State Space (S)

The state $s_t \in S$ is a feature vector providing a snapshot of the environment at time t. A representative state, as defined in modules like `V2G_profit_max_loads.py`, includes:

$$s_t = [t, P_{\text{total}}(t-1), \mathbf{c}(t, H), \mathbf{L}_1(t, H), \mathbf{PV}_1(t, H), \ldots, \mathbf{s}_1^{\text{EV}}(t), \ldots, \mathbf{s}_N^{\text{EV}}(t)]^T$$

where the components are:

- t: The current time step.
- $P_{\text{total}}(t-1)$: The aggregated power from the previous time step.
- $\mathbf{c}(t, H)$: A vector of **predicted future** electricity prices over a horizon H.
- $\mathbf{L}_j(t, H), \mathbf{PV}_j(t, H)$: Forecasts for inflexible loads and solar generation.
- $\mathbf{s}_i^{\text{EV}}(t) = [\text{SoC}_i(t), t_i^{\text{dep}} - t]$: Key information for each EV i, including its State of Charge and remaining time until departure.

3.6.2 Action Space (A)

The action $a_t \in A$ is a continuous vector in $\mathbb{R}^N$, where N is the number of chargers. For each charger i, the command $a_i(t) \in [-1, 1]$ is a normalized value that is translated into a power command:

- If $a_i(t) > 0$, the EV is charging: $P_i(t) = a_i(t) \cdot P_{\text{charge},i}^{\max}$.
- If $a_i(t) < 0$, the EV is discharging (V2G): $P_i(t) = a_i(t) \cdot P_{\text{discharge},i}^{\max}$.

3.6.3 Reward Function

The reward function $R(t)$ encodes the objectives of the control agent. The framework allows for the selection of different reward functions from the `reward.py` module to suit various goals. Key examples include:

- **Profit Maximization with Penalties** (`ProfitMax_TrPenalty_UserIncentives`): This function creates a balance between economic gain and physical constraints.

  $$R(t) = \underbrace{\text{Profit}(t)}_{\text{Economic Gain}} - \underbrace{\lambda_1 \cdot \text{Overload}(t)}_{\text{Grid Penalty}} - \underbrace{\lambda_2 \cdot \text{Unsatisfaction}(t)}_{\text{User Penalty}}$$

  The agent is rewarded for profit but penalized for overloading transformers and for failing to meet the charging needs of departing drivers.

- **Squared Tracking Error** (`SquaredTrackingErrorReward`): Used for grid service applications where precision is paramount.

  $$R(t) = -\left(P_{\text{setpoint}}(t) - \sum_{i=1}^{N} P_i(t)\right)^2$$

  The reward is the negative squared error from the power setpoint, incentivizing the agent to minimize this error at all times.

By using this enhanced framework, this thesis moves beyond single-scenario optimization to develop and validate an intelligent V2G control agent that is not only high-performing but also robust, adaptable, and ready for the complexities of real-world deployment.

3.6.4 A History-Based Adaptive Reward for Profit Maximization

To effectively steer the learning agent towards a policy that is both highly profitable and reliable, we have designed and implemented a novel, history-based adaptive reward function, named `FastProfitAdaptiveReward`. This function departs from traditional static-weight penalties and instead introduces a dynamic feedback mechanism where the severity of penalties is directly influenced by the agent's recent performance. The core philosophy is to aggressively prioritize economic profit while using adaptive penalties as guardrails that become stricter only when the agent begins to consistently violate operational constraints.

The total reward at each timestep t, R_t, is calculated as the net economic profit minus any active penalties for user dissatisfaction or transformer overload.

$$R_t = \Pi_t - P_t^{\text{sat}} - P_t^{\text{tr}} \tag{3.1}$$

Economic Profit

The foundation of the reward signal is the direct, instantaneous economic profit, Π_t. This component provides a clear and strong incentive for the agent to learn market dynamics, encouraging it to charge during low-price periods and discharge (V2G) during high-price periods.

$$\Pi_t = \sum_{i=1}^{N} \left(C_t^{\text{sell}} \cdot P_{i,t}^{\text{dis}} - C_t^{\text{buy}} \cdot P_{i,t}^{\text{ch}} \right) \Delta t \tag{3.2}$$

where N is the number of connected EVs, C_t^{sell} and C_t^{buy} are the electricity prices, and $P_{i,t}^{\text{dis}}$ and $P_{i,t}^{\text{ch}}$ are the discharging and charging powers for EV i.

Adaptive User Satisfaction Penalty

The penalty for failing to meet user charging demands, P_t^{sat}, is not a fixed value. Instead, it adapts based on the system's recent history of performance. The environment maintains a short-term memory of the average user satisfaction over the last 100 timesteps. From this history, we calculate an average satisfaction score, $\bar{S}_{hist}$.

A *satisfaction severity multiplier*, λ_t^{sat}, is then calculated. This multiplier grows quadratically as the historical average satisfaction drops, meaning that if the system has been performing poorly, the consequences for a new failure become much more severe.

$$\lambda_t^{\text{sat}} = \lambda_{\text{base}}^{\text{sat}} \cdot (1 - \bar{S}_{hist})^2 \tag{3.3}$$

where $\lambda_{\text{base}}^{\text{sat}}$ is a base scaling factor (e.g., 20.0). A penalty is only applied if any departing EV's satisfaction, S_k, is below a critical threshold (e.g., 95%). The magnitude of the penalty is the product of the adaptive multiplier and the current satisfaction deficit.

$$P_t^{\text{sat}} = \lambda_t^{\text{sat}} \cdot (1 - \min(S_k)) \quad \forall k \in \text{EVs departing at } t \tag{3.4}$$

This creates a powerful feedback loop: a single, isolated failure in an otherwise well-performing system results in a mild penalty. However, persistent failures lead to a rapidly escalating penalty, forcing the agent to correct its behavior.

Adaptive Transformer Overload Penalty

Similarly, the transformer overload penalty, P_t^{tr}, adapts based on the recent frequency of overloads. The environment tracks how often an overload has occurred in the last 100 timesteps, yielding an overload frequency, F_{hist}^{tr}.

This frequency is used to compute a linear *overload severity multiplier*, λ_t^{tr}. The more frequently overloads have happened, the higher the penalty for a new one.

$$\lambda_t^{\text{tr}} = \lambda_{\text{base}}^{\text{tr}} \cdot F_{hist}^{\text{tr}} \tag{3.5}$$

where $\lambda_{\text{base}}^{\text{tr}}$ is a base scaler (e.g., 50.0). If the total power drawn, $P_j^{\text{total}}(t)$, exceeds the transformer's limit, $P_j^{\max}$, a penalty is applied. This penalty consists of a small, fixed base amount plus the adaptive component, which scales with the magnitude of the current overload.

$$P_t^{\text{tr}} = P_{\text{base}} + \lambda_t^{\text{tr}} \cdot \sum_{j=1}^{N_T} \max(0, P_j^{\text{total}}(t) - P_j^{\max}) \tag{3.6}$$

This mechanism teaches the agent that while a rare, minor overload might be acceptable in pursuit of high profit, habitual overloading is an unsustainable and heavily penalized strategy.

Rationale and Significance

This history-based adaptive reward function represents a significant advancement over static or purely state-based approaches. By making the penalty weights a function of the system's recent performance history, we provide a more nuanced and stable learning signal. The agent is not punished excessively for isolated, exploratory actions that might lead to a minor constraint violation. Instead, it is strongly discouraged from developing policies that lead to chronic system failures.

The intuition is to mimic a more realistic management objective: maintain high performance on average, and react strongly only when performance trends begin to degrade. This method is also computationally efficient, avoiding complex state-dependent calculations in favor of simple updates to historical data queues. Ultimately, this reward structure guides the agent to discover policies that are not only profitable but also robust and reliable over time, striking a more intelligent balance between economic ambition and operational safety.

3.7 Model Predictive Control (MPC)

The MPC, implemented in `mpc.py` and `eMPC.py`, solves an optimization problem at every time step over a prediction horizon H.

3.7.1 System Model

The system is modeled in linear state-space form. The state $\mathbf{x}_k \in \mathbb{R}^N$ is the vector of SoCs of all EVs at time k. The input $\mathbf{u}_k \in \mathbb{R}^{2N}$ is the vector of charging and discharging powers.

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k$$

The matrices A_k (`Amono`) and B_k (`Bmono`) are time-varying because they depend on which EVs are connected. A_k is typically a diagonal identity-like matrix modeling the persistence of EVs. B_k maps power to SoC change, including efficiencies and Δt.

3.7.2 Optimization Problem

At time t, the MPC solves:

$$\min_{\{\mathbf{u}_k\}_{k=t}^{t+H-1}} \sum_{k=t}^{t+H-1} \mathbf{f}_k^T \mathbf{u}_k$$

subject to:

$$\begin{aligned}
&\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k, \quad \forall k \in [t, t+H-1] && \text{(Dynamics)}\\
&\mathbf{x}_k^{\min} \leq \mathbf{x}_k \leq \mathbf{x}_k^{\max} && \text{(SoC limits)}\\
&\mathbf{0} \leq \mathbf{u}_k^{\text{ch}} \leq \mathbf{u}_k^{\text{ch,max}} \cdot \mathbf{z}_k && \text{(Charge limits)}\\
&\mathbf{0} \leq \mathbf{u}_k^{\text{dis}} \leq \mathbf{u}_k^{\text{dis,max}} \cdot (1 - \mathbf{z}_k) && \text{(Discharge limits)}\\
&\sum_{i \in \text{CS}_j} (u_i^{\text{ch}} - u_i^{\text{dis}}) + L_j(k) - PV_j(k) \leq P_j^{\text{tr,max}}(k) && \text{(Transformer limits)}
\end{aligned}$$

where $\mathbf{z}_k$ is a vector of binary variables to prevent simultaneous charge and discharge. The cost vector $\mathbf{f}_k$ contains the energy prices. The code formulates this problem compactly as $\mathbf{AU} \leq \mathbf{bU}$, where $\mathbf{U}$ is the vector of all actions over the horizon.

3.8 Offline Optimization with Gurobi

Gurobi is used to find the optimal offline (a posteriori) solution, providing a performance benchmark. The files `profit_max.py` and `tracking_error.py` define the optimization problem over the entire simulation horizon T_{sim}.

3.8.1 Decision Variables

- $E_{p,i,t}$: Energy in the EV at port p of station i at time t.
- $I_{p,i,t}^{\text{ch}}, I_{p,i,t}^{\text{dis}}$: Charging/discharging currents.

- $\omega^{\text{ch}}_{p,i,t}, \omega^{\text{dis}}_{p,i,t}$: Binary variables for operating modes.

3.8.2 Objective Function (Example: Profit Maximization)

$$\max \sum_{t=0}^{T_{\text{sim}}} \sum_{i=1}^{N_{CS}} \sum_{p=1}^{N_p} \left(C_{\text{sell}}(t) P^{\text{dis}}_{p,i,t} - C_{\text{buy}}(t) P^{\text{ch}}_{p,i,t}\right) \Delta t - \lambda \sum_{k \in \text{EVs departed}} (E^{\text{des}}_k - E_k(t^{\text{dep}}_k))^2$$

where $P = V \cdot I \cdot \eta$.

3.8.3 Main Constraints

- **Energy Balance:**

$$E_{p,i,t} = E_{p,i,t-1} + (\eta_{\text{ch}} V_i I^{\text{ch}}_{p,i,t} - \frac{1}{\eta_{\text{dis}}} V_i I^{\text{dis}}_{p,i,t}) \Delta t$$

- **Activation of Current:**

$$I^{\text{ch}}_{p,i,t} \le M \cdot \omega^{\text{ch}}_{p,i,t} \quad , \quad I^{\text{dis}}_{p,i,t} \le M \cdot \omega^{\text{dis}}_{p,i,t}$$

- **Mutual Exclusion:**

$$\omega^{\text{ch}}_{p,i,t} + \omega^{\text{dis}}_{p,i,t} \le 1$$

- **Current and SoC Limits:**

$$I^{\min} \le I_{p,i,t} \le I^{\max} \quad , \quad E^{\min} \le E_{p,i,t} \le E^{\max}$$

- **SoC at Departure:**

$$E_{p,i}(t^{\text{dep}}) \ge E^{\text{des}}_{p,i}$$

3.9 Online MPC Formulation (PuLP Implementation)

The Model Predictive Control (MPC) implemented with PuLP solves a profit maximization problem at each time step t over a finite prediction horizon H. This formulation is designed for online, real-time control, where decisions are made based on the current system state and future predictions.

3.9.1 Mathematical Formulation

At each time step t, the MPC controller solves the following optimization problem.

Objective Function: Net Operational Profit

The objective is to maximize the total net operational profit over the control horizon H. This provides a comprehensive economic model that goes beyond simple energy arbitrage.

$$\max_{P^{\mathrm{ch}},P^{\mathrm{dis}},z} \sum_{k=t}^{t+H-1} \sum_{i\in\mathrm{CS}} (\mathrm{Revenues}_{i,k} - \mathrm{Costs}_{i,k}) \tag{3.7}$$

The revenue and cost components are defined for each station i at time step k as:

- **Revenues** consist of:
  - Grid Sales Revenue (V2G): $c_k^{\mathrm{sell}} \cdot P_{i,k}^{\mathrm{dis}} \cdot \Delta t$
  - User Charging Revenue: $c^{\mathrm{user}} \cdot P_{i,k}^{\mathrm{ch}} \cdot \Delta t$
- **Costs** consist of:
  - Grid Purchase Cost: $c_k^{\mathrm{buy}} \cdot P_{i,k}^{\mathrm{ch}} \cdot \Delta t$
  - Battery Degradation Cost: $c^{\mathrm{deg}} \cdot (P_{i,k}^{\mathrm{ch}} + P_{i,k}^{\mathrm{dis}}) \cdot \Delta t$

where c_k^{sell} and c_k^{buy} are the time-varying electricity prices, c^{user} is the fixed price for the end-user, c^{deg} is the estimated cost of battery degradation per kWh cycled, and Δt is the time step duration.

System Constraints

The optimization is subject to the following constraints for each station i and time step $k \in [t, t+H-1]$.

Energy Balance Dynamics. The state of energy of the EV battery evolves according to:

$$E_{i,k} = E_{i,k-1} + \left(\eta^{\mathrm{ch}} P_{i,k}^{\mathrm{ch}} - \frac{1}{\eta^{\mathrm{dis}}} P_{i,k}^{\mathrm{dis}}\right) \cdot \Delta t \tag{3.8}$$

where the initial state $E_{i,t-1}$ is the currently measured energy level of the EV.

Power Limits and Mutual Exclusion. Charging and discharging powers are bounded by the EV's capabilities and controlled by a binary variable $z_{i,k}$ to prevent simultaneous operation.

$$0 \leq P_{i,k}^{\mathrm{ch}} \leq P_i^{\mathrm{ch,max}} \cdot z_{i,k} \tag{3.9}$$

$$0 \leq P_{i,k}^{\mathrm{dis}} \leq P_i^{\mathrm{dis,max}} \cdot (1 - z_{i,k}) \tag{3.10}$$

State of Energy (SoE) Limits. The battery energy level must remain within its physical operational window.

$$E_i^{\mathrm{min}} \leq E_{i,k} \leq E_i^{\mathrm{max}} \tag{3.11}$$

User Satisfaction (Hard Constraint). The desired energy level must be met at the time of departure. This is modeled as a hard constraint, reflecting a non-negotiable service requirement.

$$E_{i,k_{\text{dep}}} \geq E_i^{\text{des}} \tag{3.12}$$

where k_{dep} is the predicted departure step of the EV within the horizon.

Transformer Power Limit. The total net power drawn from (or injected into) the grid by all charging stations must not exceed the transformer's maximum capacity.

$$\sum_{i \in \text{CS}} (P_{i,k}^{\text{ch}} - P_{i,k}^{\text{dis}}) \leq P^{\text{tr,max}} \tag{3.13}$$

3.10 Conceptual Comparison: PuLP MPC vs. Gurobi Offline Optimizer

While both the PuLP MPC and the Gurobi offline optimizer are used to solve the EV charging problem, they operate on fundamentally different principles and serve distinct purposes. This section provides a discursive comparison of their core concepts.

3.10.1 Core Philosophy: Controller vs. Judge

The most significant difference lies in their philosophy. The **PuLP MPC** is designed as a **controller**. It operates online, making decisions in real-time with incomplete information about the future (e.g., EV arrivals, price fluctuations beyond the prediction horizon). Its goal is to find a practical and robust strategy for the immediate future.

Conversely, the **Gurobi formulation** acts as a **judge**. It is an offline tool that solves the problem over the entire simulation period with perfect hindsight (a-posteriori). Its purpose is not to control the system in real-time, but to establish a theoretical performance benchmark—the "perfect score"—against which the performance of a practical controller like the MPC can be measured.

3.10.2 Objective Function: Operational Profit vs. Energy Arbitrage

The objectives, while both related to profit, reflect their different roles. The PuLP MPC maximizes a detailed **Net Operational Profit**, incorporating a realistic business model that includes revenue from end-users and operational costs like battery degradation. This makes its decisions economically grounded from a business perspective.

The Gurobi optimizer, on the other hand, typically maximizes profit from a simpler **energy arbitrage** model, focusing on the difference between buying and selling electricity. While it includes a penalty for not meeting user demand, it does not explicitly account for the same level of operational economic detail as the MPC.

3.10.3 Handling of User Satisfaction: Hard vs. Soft Constraints

This distinction is critical from an operational standpoint. The PuLP MPC treats user satisfaction as a **Hard Constraint**. The EV *must* reach its desired energy level by its departure time. If the model determines this is impossible, the optimization problem becomes infeasible, signaling a failure to meet a mandatory service level agreement.

The Gurobi formulation treats user satisfaction primarily as a **Soft Constraint** via a penalty term in its objective function. This allows the optimizer to make a trade-off: it can choose to not fully charge a vehicle if the economic benefit of doing so (e.g., selling a large amount of energy to the grid at a high price) outweighs the penalty for customer dissatisfaction. This is useful for theoretical analysis but less practical for guaranteeing service.

Bibliography

'Fit for 55': delivering the EU's 2030 Climate Target on the way to climate neutrality (2021). Tech. rep. COM(2021) 550 final. Brussels: European Commission.

Alfaverh, Farag, Mouloud Denaï, and Yifei Sun (2022). "Optimal vehicle-to-grid control for supplementary frequency regulation using deep reinforcement learning". In: *Applied Energy* 325, p. 119881. DOI: 10.1016/j.apenergy.2022.119881.

Alsabbagh, Mohamad and Wan-Chi Siu (2022). "Reinforcement learning for vehicle-to-grid: a review". In: *Journal of Energy Storage* 53, p. 105149. DOI: 10.1016/j.est.2022.105149.

Birkl, Christoph R et al. (2017). "Degradation diagnostics for lithium ion cells". In: *Journal of Power Sources* 341, pp. 373–386.

Brockman, Greg et al. (2016). "OpenAI Gym". In: *arXiv preprint arXiv:1606.01540*. DOI: 10.48550/arXiv.1606.01540.

Evertsson, Albin and Anton Nylander (2024). "Investigating Vehicle-to-grid from a User-centric Perspective". MA thesis. Chalmers University of Technology.

Faggio, Gabriele (2023). "Design and Testing of Online and Offline Optimization Algorithms for Vehicle-to-Grid (V2G) Industrial Applications". MA thesis. Politecnico di Milano.

Fujimoto, Scott, Herke van Hoof, and David Meger (2018). "Addressing function approximation error in actor-critic methods". In: *International conference on machine learning*. PMLR, pp. 1587–1596.

Haarnoja, Tuomas et al. (2018). "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International conference on machine learning*. PMLR, pp. 1861–1870.

Al-HMOUD, Ghaith and Hamed Al-Raweshidy (2024). "A Review of Smart Grid Evolution and Reinforcement Learning: Applications, Challenges and Future Directions". In: *Energies* 18.7, p. 1837. DOI: 10.3390/en18071837.

Khan, Sheroz Ullah et al. (2024). "A Review of Bidirectional Charging Grid Support Applications and Control". In: *Energies* 17.6, p. 1320. DOI: 10.3390/en17061320.

Kumar, Vinay, Surender Singh, and Dinesh Kumar (2024). "Integration of electric vehicle into smart grid: a meta heuristic algorithm for energy management between V2G and G2V". In: *Frontiers in Energy Research* 12, p. 1357863. DOI: 10.3389/fenrg.2024.1357863.

Kuznetsov, Anton et al. (2020). "Controlling overestimation bias with truncated quantile critics". In: *International Conference on Machine Learning*. PMLR, pp. 5528–5538.

Lillicrap, Timothy P et al. (2015). "Continuous control with deep reinforcement learning". In: *arXiv preprint arXiv:1509.02971*.

Liu, Zhaomiao et al. (2023). "Optimal scheduling for charging and discharging of electric vehicles based on deep reinforcement learning". In: *Frontiers in Energy Research* 11, p. 1273820. DOI: 10.3389/fenrg.2023.1273820.

Logeshwaran, J, Chao Fan, and Swan Htet Naung (2022). "A comparative study of deep reinforcement learning algorithms for energy management in commercial buildings". In: *Energy and Buildings* 254, p. 111589.

Mania, Horia, Aurelia Guy, and Benjamin Recht (2018). "Simple random search of static linear policies is competitive for reinforcement learning". In: *Advances in neural information processing systems*. Vol. 31.

Minchala-Ávila, Carlos A, Paúl Arévalo, and Diego Ochoa-Correa (2025). "A Systematic Review of Model Predictive Control for Robust and Efficient Energy Management in Electric Vehicle Integration and V2G Applications". In: *Modelling* 6.1, p. 20. DOI: 10.3390/modelling6010020.

Mnih, Volodymyr et al. (2016). "Asynchronous methods for deep reinforcement learning". In: *International conference on machine learning*. PMLR, pp. 1928–1937.

Ng, Andrew Y, Daishi Harada, and Stuart Russell (1999). "Policy invariance under reward transformations: Theory and application to reward shaping". In: *ICML* 99, pp. 278–287.

Orfanoudakis, Stylianos, Christian Diaz-Londono, Yasin Emir Yilmaz, et al. (2022). "A Deep Reinforcement Learning-Based Charging Strategy for Electric Vehicles with V2G Services in a Volatile Market". In: *arXiv preprint arXiv:2209.09772*. DOI: 10.48550/arXiv.2209.09772.

Orfanoudakis, Stylianos, Christian Diaz-Londono, Yasin Emir Yïlmaz, et al. (2024). "EV2Gym: A Flexible V2G Simulator for EV Smart Charging Research and Benchmarking". In: *arXiv preprint arXiv:2404.01849*. DOI: 10.48550/arXiv.2404.01849.

Sadeghi (2021). "Cost and Power Loss Aware Coalitions under Uncertainty in Transactive Energy Systems". In: *Université d'Ottawa / University of Ottawa*.

Salvatti, Gabriel Antonio et al. (2020). "Electric Vehicles Energy Management with V2G/G2V Multifactor Optimization of Smart Grids". In: *Energies* 13.5, p. 1191. DOI: 10.3390/en13051191.

Schaul, Tom et al. (2015). "Prioritized experience replay". In: *arXiv preprint arXiv:1511.05952*. DOI: 10.48550/arXiv.1511.05952.

Schulman, John, Sergey Levine, et al. (2015). "Trust region policy optimization". In: *International conference on machine learning*. PMLR, pp. 1889–1897.

Schulman, John, Filip Wolski, et al. (2017). "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347*.

Schwenk, Johannes et al. (2022). "A computationally efficient MPC for residential EV charging with V2G and 3-phase power flow". In: *Energies* 15.16, p. 5923. DOI: 10.3390/en15165923.

Shibl, Mohamed M, Loay S Ismail, and Ahmed M Massoud (2023). "Electric vehicles charging management using deep reinforcement learning considering vehicle-to-grid operation and battery degradation". In: *Energy Reports* 10, pp. 494–509. DOI: 10.1016/j.egyr.2023.07.039.

Vetter, Joachim et al. (2005). “Ageing mechanisms in lithium-ion batteries”. In: *Journal of Power Sources* 147.1–2, pp. 269–281.

Wan, Zhong et al. (2022). “A dynamic reward-based deep reinforcement learning for V2G control with EV battery degradation”. In: *Applied Energy* 309, p. 118462. DOI: 10.1016/j.apenergy.2021.118462.

Wang, Siyuan, Shuo Wang, and Bo Liu (2022). “Multi-objective optimal scheduling of charging stations with solar-storage-diesel generator system”. In: *Frontiers in Energy Research* 10, p. 1042882. DOI: 10.3389/fenrg.2022.1042882.

Xie, Hongbin (2021). “Deep reinforcement learning-based control strategies for electric vehicle charging and V2G services”. In: *Advances in Applied Energy*. DOI: https://doi.org/10.1016/j.adapen.2025.100214.