







Article

A Semantic Energy-Aware Ontological Framework for Adaptive Task Planning and Allocation in Intelligent Mobile Systems

Jun-Hyeon Choi ¹, Dong-Su Seo ¹, Sang-Hyeon Bae ², Ye-Chan An ¹, Eun-Jin Kim ¹, Jeong-Won Pyo ³
and Tae-Yong Kuc ^{1,*}

¹ Department of Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University, Suwon 16419, Republic of Korea; choijunhyeon@g.skku.edu (J.-H.C.); seeds@skku.edu (D.-S.S.); ycan@g.skku.edu (Y.-C.A.); blisskej@naver.com (E.-J.K.)

² Samsung Display Co., Ltd., R&D Center, Yongin 17113, Republic of Korea; sh1423.bae@samsung.com

³ DXR Co., Ltd., R&D Center, Seoul 01411, Republic of Korea; jungwon900@dxr-ai.com

* Correspondence: tykuc2022@gmail.com

Abstract

Intelligent robotic systems frequently operate under stringent energy limitations, especially in complex and dynamic environments. To enhance both adaptability and reliability, this study introduces a semantic planning framework that integrates ontology-driven reasoning with energy awareness. The framework estimates energy consumption based on the platform-specific behavior of sensing, actuation, and computational modules while continuously updating place-level semantic representations using real-time execution data. These representations encode not only spatial and contextual semantics but also energy characteristics acquired from prior operational history. By embedding historical energy usage profiles into hierarchical semantic maps, this framework enables more efficient route planning and context-aware task assignment. A shared semantic layer facilitates coordinated planning for both single-robot and multi-robot systems, with the decisions informed by energy-centric knowledge. This approach remains hardware-independent and can be applied across diverse platforms, such as indoor service robots and ground-based autonomous vehicles. Experimental validation using a differential-drive mobile platform in a structured indoor setting demonstrates improvements in energy efficiency, the robustness of planning, and the quality of the task distribution. This framework effectively connects high-level symbolic reasoning with low-level energy behavior, providing a unified mechanism for energy-informed semantic decision-making.

Keywords: task allocation; semantic navigation; adaptive planning; multi-robot systems; semantic knowledge



Academic Editor: Andrea Bonci

Received: 1 August 2025

Revised: 9 September 2025

Accepted: 12 September 2025

Published: 15 September 2025

Citation: Choi, J.-H.; Seo, D.-S.; Bae, S.-H.; An, Y.-C.; Kim, E.-J.; Pyo, J.-W.; Kuc, T.-Y. A Semantic Energy-Aware Ontological Framework for Adaptive Task Planning and Allocation in Intelligent Mobile Systems. *Electronics* **2025**, *14*, 3647. <https://doi.org/10.3390/electronics14183647>

Copyright: © 2025 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) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Long-term autonomy (LTA) is increasingly recognized as a fundamental capability for mobile robots and autonomous vehicles operating in unstructured and dynamic environments. Although notable progress has been made in navigation, perception, and high-level decision-making systems [1], existing methods often fail to assess the feasibility of the mission under internal resource constraints. In particular, the integration of real-time energy awareness into planning and execution processes remains underexplored [2]. This limitation becomes especially critical in energy-constrained autonomous driving platforms,

which must handle complex missions while balancing a limited battery capacity, variable terrain, traffic fluctuations, and changing task demands. Under such conditions, the ability to predict mission success based on the current energy availability and to adapt behavior accordingly is essential to achieving robust autonomy. However, obtaining an exact prediction of the energy use in real environments is extremely challenging since factors such as terrain changes, unexpected obstacles, and hardware conditions can hardly be modeled in advance. For long-term autonomy, what is more useful than numerical precision is the capability to interpret energy feasibility in context and to adjust planning and task assignment toward options that are safer and more efficient. In this view, the energy-related properties in our model are not intended as absolute predictors but rather as relative cues that guide robots to choose missions and routes with a lower chance of failure under resource limitations.

To address this problem, we adopt a semantic modeling framework [3,4]. At its core, the Triplet Ontological Semantic Model (TOSM) provides a hierarchical representation that allows mobile platforms to interpret and understand their surrounding environment semantically. The framework integrates heterogeneous sensor and mission data into an ontology-based semantic map, enabling symbolic reasoning. Traditional semantic maps have mainly focused on object categories or static geometric information [5,6]. In contrast, knowledge-based reasoning approaches aim to provide structured knowledge representations that support higher-level interpretation and inference [7,8]. Building upon such knowledge-based approaches, our work extends them by introducing the Triplet Ontological Semantic Model (TOSM), which not only incorporates categorical and geometric attributes but also integrates empirical, execution-driven knowledge. Through this extension, the TOSM advances semantic mapping beyond a technical tool toward a knowledge-oriented framework, thereby enabling richer semantic reasoning and more robust long-term autonomy in resource-constrained environments. Examples include discrepancies between planned and actual travel times, delays caused by dynamic obstacles, variations in energy consumption due to slopes or congestion, and cumulative evaluations of specific routes.

The extended framework is structured within an ontology-based system that supports symbolic reasoning about energy consumption and mission feasibility. By incorporating this semantic knowledge into the decision-making pipeline, a robot or an autonomous vehicle can determine in real time whether it can complete a mission under its current energy conditions. It can also adapt to changing environments by deciding whether to continue, reroute, suspend the task, recharge, or terminate the task accordingly.

Moreover, the knowledge accumulated within the TOSM over time serves as a foundation for cooperative reasoning and efficient task distribution in multi-robot systems. By maintaining shared energy efficiency indicators, the framework enables each robot to autonomously assess whether a mission can be accomplished under its current resource conditions and, if necessary, decide to suspend the task, recharge, or terminate the task. From a planning perspective, the framework does not rely solely on numerical optimization or shortest-path computation. Instead, it supports more flexible and adaptive planning that accounts for diverse situational factors, allowing robots to interpret their environmental context and adjust their execution strategies accordingly. Finally, this semantic knowledge extends beyond individual decision-making to support multi-robot task allocation. Based on shared energy efficiency indicators, robots can autonomously coordinate which agent is most suitable for a given task, thereby enabling efficient and sustainable cooperation across the team. As summarized in Figure 1, the proposed semantic planning framework integrates multiple modules that collectively support real-time inference, adaptive decision-making, and multi-robot coordination.

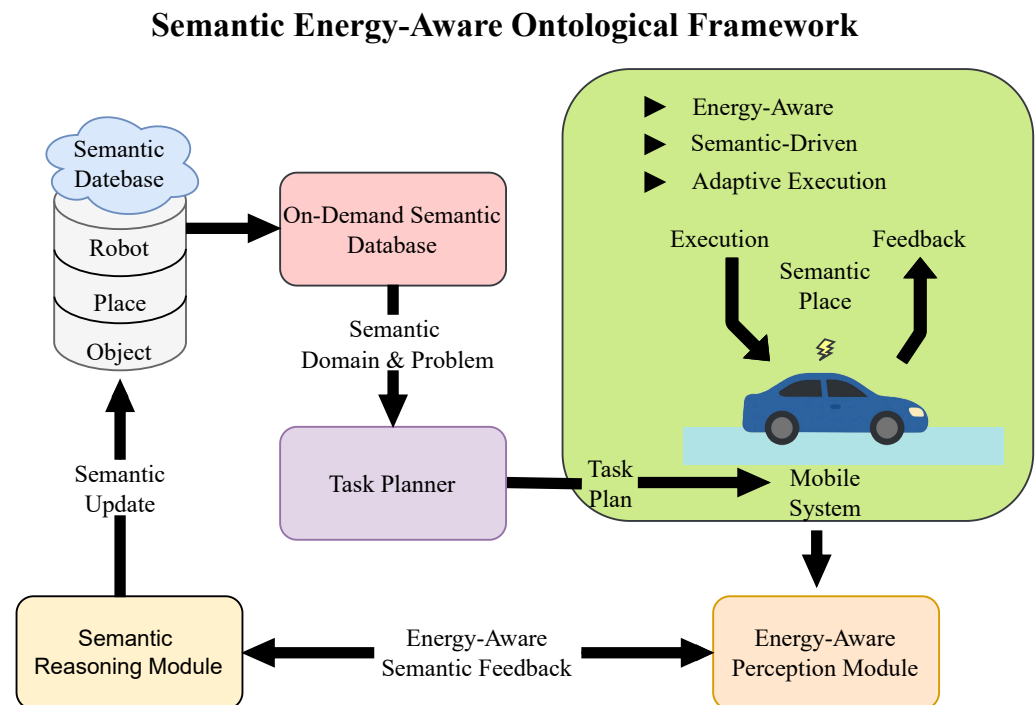


Figure 1. An overview of the semantic energy-aware ontological framework.

The main contributions of this work are summarized as follows:

- We extend the Triplet Ontological Semantic Model (TOSM) by embedding empirical energy-related knowledge into spatial and semantic structures, supporting context-aware reasoning beyond numerical prediction;
- We develop an ontology-based reasoning mechanism that enables autonomous assessment of mission feasibility and adaptive task decisions, highlighting semantic knowledge integration;
- We propose a cooperative semantic framework that exploits accumulated knowledge for real-time inference, adaptive planning, and efficient task allocation in multi-robot systems.

2. Related Work

2.1. Long-Term Autonomy and Resource-Constrained Planning

Long-term autonomy (LTA) refers to the ability of a robotic system to continuously perform tasks over extended durations without human intervention, especially in dynamic or uncertain environments [9,10]. To achieve this, systems must maintain robustness not only in perception and localization but also in long-horizon planning and real-time monitoring of hardware, computation, and energy resources [1,11]. As robots take on missions such as field exploration, planetary operations, and infrastructure inspections, sustaining autonomous behavior over time becomes increasingly complex due to environmental variability and system degradation.

Prior research introduced techniques such as continual map updates, self-repairing knowledge models, and context-aware replanning to address issues like sensor drift and changing environments [1,11,12]. These strategies allow systems to adapt during long-term deployment. However, most of them either ignore energy as a decision-making factor or treat it as an externally managed constraint [13,14]. In reality, internal energy states and battery degradation significantly affect mission feasibility in autonomous operation.

Recent works in energy-aware motion planning have aimed to optimize the routes for power-limited platforms like planetary rovers or underwater robots by incorporating en-

energy consumption models [15,16]. However, these methods often rely on static terrain costs or heuristic profiles, which means they typically do not leverage past execution feedback or symbolic reasoning. While these strategies certainly offer efficiency improvements, they frequently fall short when dealing with unexpected energy drifts or addressing questions of long-horizon feasibility.

In contrast to these approaches, our method treats energy-related experiences as semantic knowledge, embedding them into a symbolic framework that supports dynamic reasoning. By doing so, the robot can assess the feasibility under varying environmental and internal conditions using accumulated experience, beyond mere numerical predictions. This symbolic shift enhances adaptability, especially in scenarios where long-term reliability is mission-critical.

2.2. Energy-Aware Reasoning and Adaptive Behaviors

Recent efforts have focused on enabling robots to adapt their behavior in response to energy-related constraints. For instance, motion planners may assign higher traversal costs to steep terrains or reroute paths to avoid obstacles with uncertain power demands [17,18]. These strategies help reduce unnecessary energy consumption and improve operational efficiency during long-term deployments. At the task level, energy awareness has been incorporated into schedulers and multi-robot coordination frameworks, allowing agents to defer, reassign, or reprioritize tasks based on their real-time battery conditions [19,20].

Despite these advancements, most existing approaches still cannot reason symbolically or semantically about energy usage. Typically, energy metrics are encoded as scalar cost functions or static thresholds, without considering their spatial or temporal context. As a result, robots struggle to anticipate the long-term impact of energy-related actions or to generalize from prior inefficiencies and mission-level failures.

In our approach, we add a semantic reasoning layer that collects energy-related experiences from past missions and organizes them into a structured knowledge base. Unlike conventional planners that respond to energy values without context, our system can reason symbolically about how missions turned out, what the environment was like, and how efficiently energy was used over time. This makes it easier for the robot to make decisions that are both flexible and easier to explain, especially in situations where energy is limited or safety is a key concern.

2.3. Semantic Mapping and Ontology-Based Representations

Semantic mapping plays a key role in robotic perception and task planning, primarily by labeling spaces with meaningful categories such as objects, rooms, and traversability scores [21,22]. These maps are typically derived from 2D/3D sensors and enriched through segmentation or classification, providing functional but mostly static annotations.

To enhance flexibility, ontology-based models have been proposed to represent symbolic knowledge about the world, capturing relations among objects, tasks, and environments [23,24]. Ontologies define semantic concepts and their interrelations, enabling robots to reason about task preconditions, possible actions, or spatial constraints. Furthermore, recent research has focused on developing semantic models tailored to diverse environments for enhanced mapping and task planning [25–27].

Nonetheless, most existing ontology frameworks do not treat energy as a first-class semantic entity [28]. Instead, energy is usually represented as a simple scalar value or threshold, limiting a robot's ability to reason about energy-related implications across time or different operational contexts. Only a few systems have attempted to bridge symbolic reasoning with empirical energy profiles or regional energy efficiency characteristics.

In contrast, our framework embeds energy-related experiences such as terrain-based consumption patterns and mission-level deviations into a semantic ontology. This enables symbolic reasoning that not only accounts for spatial and functional knowledge but also reflects historical energy behavior, providing a richer context for adaptive planning.

2.4. Hierarchical and Experience-Based Semantic Models

In robotic systems, the need for representations that integrate spatial, task-related, and execution-level information is becoming increasingly important. However, hierarchical semantic models that structurally organize such multi-level knowledge remain relatively underexplored [29].

Recent works have explored using scene-graph-based methods to capture relationships between objects or between objects and their environments [30–32]. These methods offer a compact way to represent static semantic relationships, yet they tend to lack temporal awareness or the ability to incorporate experiential knowledge. As a result, it becomes difficult to reason about long-term performance trends or adapt future behaviors based on past failures.

To address this, some approaches have introduced experience-based representations, where robots accumulate traversal history, build failure likelihood maps, or maintain memory modules that track execution outcomes [33]. In addition, several studies have explored hierarchical representations of environments through object-centric abstraction [34,35]. While these models improve reactivity and robustness, they are often developed as stand-alone modules with limited integration into semantic reasoning pipelines.

Parallel efforts have also been made to encode navigation preferences and recurrent failure patterns using experience-driven learning [36]. However, such models are often domain-specific or detached from structured semantic representations, which limits their reusability and generalization across diverse environments and tasks.

To overcome these challenges, our framework proposes a unified approach that combines a structured ontological model with symbolic encoding of accumulated experiences. By embedding energy-related deviations, recovery attempts, region-specific difficulty, and mission-level outcomes into a hierarchical semantic space, our system enables symbolic reasoning that supports both contextual adaptation and long-term generalization.

In contrast to conventional experience-based memory models, our method allows robots to interpret and reuse operational knowledge within a multi-level semantic framework, supporting more informed planning and execution in complex, dynamic environments.

2.5. Task Allocation and Multi-Robot Coordination

Efficient task allocation plays a central role in autonomous systems, particularly in scenarios where multiple mobile agents must collaborate under limited and shared resources. Conventional strategies typically rely on heuristic cost functions, auction-based mechanisms, or centralized planners to assign tasks based on factors such as distance or estimated completion time [37]. While these approaches perform well in structured and static environments, they often fall short in dynamic settings where energy availability and task-level uncertainty become critical [38,39].

In general, multi-robot systems, planning, and coordination are handled through either centralized optimization or distributed negotiation. Centralized approaches can produce globally optimal solutions but are less robust to communication failures or dynamic changes. Distributed frameworks, on the other hand, allow for more scalable and flexible coordination by relying on inter-robot communication and local policies [40,41]. Despite their effectiveness in many scenarios, both approaches tend to overlook the impact of contextual constraints such as energy capacity, terrain complexity, or accumulated task fatigue.

Semantic reasoning also facilitates more transparent and adaptable decision-making [25,42]. For instance, rather than simply assigning tasks based on cost, robots can assess the semantic implications of executing certain tasks in specific places, such as potential exposure to high-risk zones or energy-intensive operations. Such awareness improves planning robustness, supports explainable delegation, and enhances mission-level coordination in dynamic or uncertain environments.

To address this limitation, recent research has explored energy-aware allocation frameworks that explicitly consider robot battery status, task-specific energy consumption, and environmental cost factors [43,44]. For example, robots may avoid tasks with historically high energy demands or delegate such functions to peers with more available power. These strategies contribute to system resilience but are generally reactive and lack a deeper semantic awareness regarding context or past performance.

Furthermore, most existing task allocation methods focus on symbolic task descriptors without integrating knowledge derived from prior experience or semantic relationships. As a result, they often fail to capture the underlying difficulty of tasks in varying contexts or anticipate potential failures. The absence of such high-level reasoning restricts their ability to generate explainable or proactive plans.

To overcome these gaps, our framework introduces semantic energy reasoning into the task allocation process. By encoding historical task outcomes and energy usage patterns into a shared ontology, the system enables robots to infer task feasibility and potential risk within a given operational context. This semantic embedding allows for more informed and adaptive planning decisions, especially in collaborative multi-robot environments where energy-aware task delegation is essential for sustained autonomy.

In the following section, we introduce our proposed ontological framework, which formalizes these semantic constructs and supports context-aware reasoning for adaptive and energy-conscious task execution.

3. Semantic Knowledge Modeling

3.1. Semantic Environment Modeling

To perform complex tasks in dynamic environments, autonomous robots require more than just low-level geometric representations. They must understand the structure, relationships, and semantics of the environment in a way that supports reasoning and adaptive planning.

Building upon our previous works [3,4], we extend the Triplet Ontological Semantic Model (TOSM) to constructing a comprehensive Semantic Environment Modeling Framework tailored to autonomous navigation.

The proposed framework decomposes the environment into three core components, *Object*, *Place*, and *Robot*, each modeled through three semantic layers: explicit, implicit, and symbolic. In particular, Figure 2 illustrates our semantic place database, where stem places (warm yellow) and leaf places (cool blue) are connected by *isInsideOf* (black arrows) and *isConnectedTo* (bold red arrows), with overlaid energy flow highlighting high-risk paths.

Figure 2 illustrates our semantic place database, where stem places (warm yellow) and leaf places (cool blue) are connected by *isInsideOf* (black arrows) and *isConnectedTo* (bold red arrows), with overlaid energy flow high-risk paths.

- **The explicit model:** This includes sensor-derived attributes such as object pose, velocity, size, and color, polygonal boundaries for places, and the dynamic states of robots. These attributes are directly measurable through various sensing modalities.
- **The implicit model:** This encodes relational and contextual knowledge, including spatial relations (e.g., *isNextTo*, *isInsideOf*) and perception states (e.g., *isLookingAt*, *isLocatedAt*). These facilitate cognitive-style reasoning similar to human semantic memory.

- **The symbolic model:** This assigns abstract identifiers (e.g., names or IDs) to objects, places, and robots, enabling symbolic interaction and integration with knowledge-based planning systems.

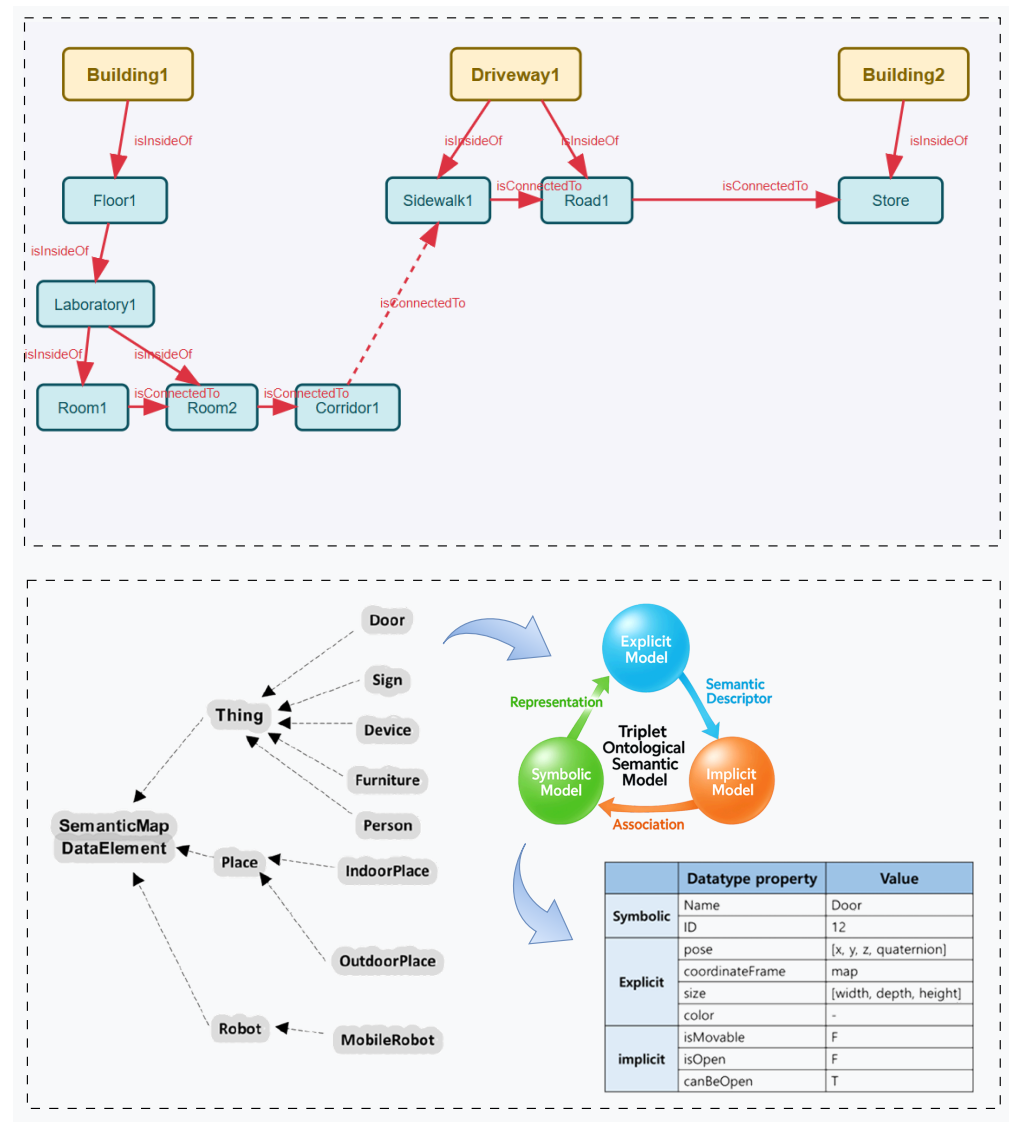


Figure 2. The semantic relationships of places in the semantic database.

All components are defined using OWL (Web Ontology Language) [45], with classes representing abstract concepts (e.g., Door, Room, MobileRobot), object properties modeling inter-entity relations (e.g., isConnectedTo, isInsideOf), and datatype properties specifying concrete attributes (e.g., pose, color, capability).

To support energy-aware decision-making, we extend the semantic model further by introducing additional attributes that represent energetic and task-related characteristics of environmental elements. These properties are mainly associated with *Place* and *Object* classes and provide abstract knowledge for evaluating energetic feasibility and planning risks. Table 1 summarizes these extended semantic properties. To enable fine-grained reasoning, we define energy-related attributes directly at the *unit_place* level. *unit_place* is the minimal spatial unit a robot can occupy (approximately 1 m²), enabling precise modeling of the local energy cost, difficulty, and risk. Each *unit_place* is annotated with properties such as *energyComplexity*, *delayCost*, *difficultyLevel*, and *riskLevel*, which are updated based on past executions and contextual factors. Semantically, *unit_places* are

organized under higher-level places using the `isInsideOf` property (e.g., building \rightarrow room \rightarrow unit_place) and spatially linked via `isConnectedTo`. This layered structure supports localized reasoning and global path planning. Figure 3 illustrates this hierarchy and attribute mapping. Table 1 summarizes the extended semantic properties.

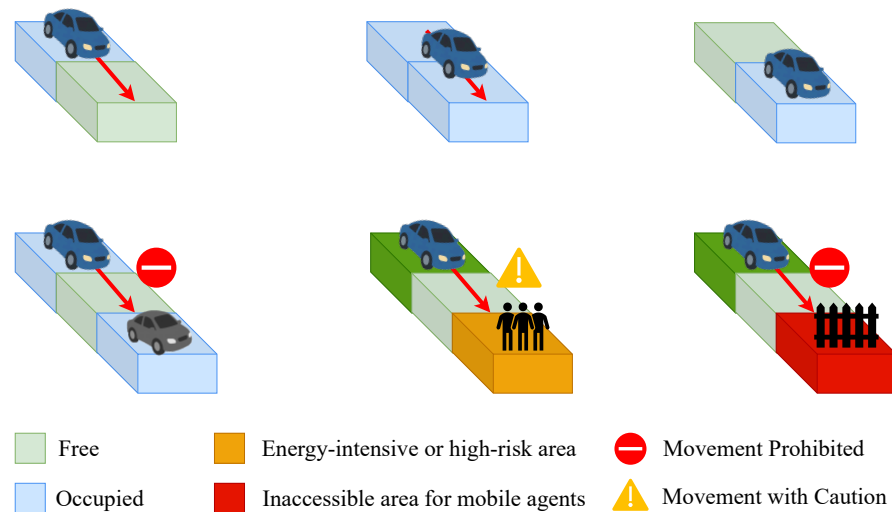


Figure 3. Hierarchical place structure with energy-aware properties.

Table 1. Extended energy-related properties for semantic modeling.

Property Name	Domain	Description
<code>energyComplexity</code>	Place/Object	Estimated energy required for access or interaction
<code>delayCost</code>	Place/Object	Time delay or latency expected in task execution
<code>difficultyLevel</code>	Place	Relative difficulty of traversal (e.g., stairs, slope)
<code>riskLevel</code>	Place/Object	Potential safety or functional risks (e.g., narrowness, crowd)
<code>energyDeviation</code>	Place	Difference between predicted and observed energy usage

Table 1 lists the energy-related semantic properties used in our framework. These properties are updated immediately after each plan execution (see Section 3.1) and then drive the ontology-based semantic reasoning for plan refinement in Section 5.4.

- `energyComplexity`: This indicates how much extra energy was consumed compared to an ideal execution. A higher value flags inefficient or power-hungry interactions.
- `delayCost`: This represents the proportional delay in the actual execution time versus the ideal. Larger values denote tasks that ran significantly slower than expected.
- `difficultyLevel`: A fixed score reflecting physical traversal hindrance at a location (e.g., stairs, narrow passages, uneven terrain).
- `riskLevel`: This accumulates whenever a task failure or perception error occurs, marking locations that repeatedly cause mission issues.
- `energyDeviation`: This records the raw difference between observed and predicted energy usage, providing feedback on model accuracy.

These properties enrich the implicit layers with energetic and situational semantics, allowing the robot to reason beyond geometry and assess the cost and risk of action choices. As a result, the framework supports more adaptive and efficient planning strategies in real-world conditions.

3.2. Semantic-Knowledge-Based Representation of Robot Capabilities and Context

To enable energy-aware planning and adaptive decision-making, we extend the Triplet Ontological Semantic Model (TOSM) to incorporate detailed knowledge about the robot's

internal subsystems and their operational context. This ontology-driven representation allows reasoning not only about spatial and semantic knowledge but also about system-level capabilities and constraints.

The robot entity in the TOSM is modeled using three semantic layers: explicit, implicit, and symbolic. Table 2 summarizes the representative modeling strategies for each layer.

Table 2. Example of TOSM modeling for robot capabilities and context.

TOSM Layer	Example Properties	Description
Explicit Model	batteryCapacity (Wh) nominalPowerDraw (W) scanEnergyUsage (W/scan) cpuLoad (%)	Attributes obtained directly from sensor specifications or system logs, such as capacity, power draw, or CPU load.
Implicit Model	isOverloaded (SensorModule) isUnderpowered (LocomotionUnit) hasThermalRisk (ProcessingUnit) perceivedCost (region)	States inferred from runtime sensor conditions, such as thermal risk, overload status, or underpower alerts.
Symbolic Model	Type (SensorModule) Model (LiDAR-Model-A) ID (L-3D-1001)	Abstract identifiers used to support symbolic interaction and integration with knowledge-based planning.

The robot's internal structure is further decomposed into modular components, each modeled as an OWL class. Table 3 presents example datatype properties for each module.

Table 3. Example ontological properties for robot subsystems.

Module Class	Property	Type	Context Use
PowerSystem	batteryCapacity voltageLevel perModuleUsage	float (Wh) float (V) float (W/module)	Energy feasibility check Safe operation range Consumption estimation
LocomotionUnit	terrainType nominalPowerDraw accelerationCost	float (c_{rr}) float (W) float (W/s)	Motion constraint reasoning Cost of navigation Task transition estimation
SensorModule	scanEnergyUsage thermalLimit sensingRange	float (W/scan) float (°C) float (m)	Sensor scheduling Thermal shutdown risk Environment coverage planning
ProcessingUnit	cpuLoad inferenceLatency thermalThrottleState	float (%) float (ms) boolean	Resource balancing Action delay estimation Performance degradation warning

This multi-layered semantic representation enables robots to not only execute tasks safely and efficiently but also adapt behavior plans based on current status, context, and energetic feasibility.

4. The Energy-Aware Modeling and Feedback Mechanism

To enable adaptive energy-constrained planning, this study proposes a unified framework that estimates the energy consumption across multiple robot subsystems, evaluates plan feasibility, and updates semantic knowledge based on execution feedback. The models presented here are not intended to provide perfectly accurate physical predictions; instead, they offer relative and experience-based indicators that help assess feasibility and determine suitable execution strategies under energy constraints. This section describes the modeling approach, behavior-level energy estimation, and semantic update mechanism.

4.1. The Multi-Source Energy Prediction Strategy

To predict the energy usage before task execution, we decompose the power consumption into four sources: computation modules, sensing modules, locomotion systems, and miscellaneous components. Each source is modeled separately using a combination of specifications, parametric equations, and empirical correction terms.

4.1.1. Computation Module Power

The energy consumed by CPUs and GPUs (e.g., the Jetson Orin AGX) is modeled as a function of task type, load profile, and contextual semantic cost at the corresponding location. The correction term δ_{compute} accounts for empirical deviation.

$$\hat{P}_{\text{compute}} = f_{\text{model}}(\text{TaskType}, \text{LoadProfile}, \text{Cost}_l^{\text{energy}}) + \delta_{\text{compute}} \quad (1)$$

4.1.2. Sensor Module Power

When detailed profiles are unavailable, functionally similar sensors are grouped and averaged to form an approximate estimate.

$$\hat{P}_{\text{sensor}} = \sum_{i=1}^{N_s} w_i \left(P_{\text{nominal}}^{(i)} + \delta_{\text{sensor}}^{(i)} \right) \quad (2)$$

$$\hat{P}_{\text{sensor}}^* = \bar{P}_{\text{group}} = \frac{1}{M} \sum_{j=1}^M P_{\text{empirical}}^{(j)} \quad (3)$$

4.1.3. Drive-Level Energy via Motion Commands

While c_{rr} is treated as a fixed coefficient initially, it can later be adapted using semantic information about floor type, texture, or slope.

$$v_l = v - \frac{L}{2}\omega, \quad v_r = v + \frac{L}{2}\omega \quad (4)$$

$$a_l = \frac{dv_l}{dt}, \quad a_r = \frac{dv_r}{dt} \quad (5)$$

$$\tau = \frac{1}{2}mar + \left(\frac{1}{2}mgc_{rr} \right) r \quad (6)$$

$$P(t) = \tau \cdot \omega_{\text{wheel}} = \tau \cdot \frac{v}{r} \quad (7)$$

$$E_{\text{drive}} = \int_0^T P(t)dt \approx \sum_{i=1}^N P_i \Delta t \quad (8)$$

4.1.4. Miscellaneous Power Consumption

We model all remaining onboard power draw (e.g., fans, communication modules, internal electronics) as a constant offset based on empirical measurements:

$$\hat{P}_{\text{other}} = P_{\text{const}}$$

Here, P_{const} is obtained from system-level power logging under nominal idle conditions and remains fixed across all tasks.

4.1.5. Empirical Deviation Estimation

The empirical deviation in each power source is estimated using an EKF rather than treated as a fixed constant. For each sensor i , an independent EKF refines the model output by estimating its correction term and drift rate. The state vector is defined as

$$x_k^{(i)} = \begin{bmatrix} \delta_k^{(i)} \\ \dot{\delta}_k^{(i)} \end{bmatrix}, \quad (9)$$

where $\delta^{(i)}$ denotes the offset and $\dot{\delta}^{(i)}$ its drift rate. In the prediction step, the nominal power $\hat{P}_{\text{sensor},k}^{(i)}$ is obtained from the models in Section 4.1, while the correction term evolves according to

$$x_{k+1}^{(i)} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_k^{(i)} + w_k^{(i)}, \quad (10)$$

with the process noise $w_k^{(i)}$. The measurement update uses the observed instantaneous power consumption:

$$z_k^{(i)} = P_{\text{real},k}^{(i)} = \hat{P}_{\text{sensor},k}^{(i)} + \delta_k^{(i)} + v_k^{(i)}, \quad (11)$$

where only the offset component is directly observable. Finally, the aggregated sensor power prediction is given by

$$\hat{P}_{\text{sensor}} = \sum_{i=1}^{N_s} w_i \left(\hat{P}_{\text{sensor}}^{(i)} + \delta_k^{(i)} \right). \quad (12)$$

Miscellaneous power consumption is treated in the same manner since it represents a system-level empirical deviation estimated through the EKF rather than a fixed constant.

4.2. Behavior-Level Energy Estimation and Plan Feasibility

For each behavior unit (e.g., *MoveTo*, *ScanArea*), the expected energy is calculated over its execution period:

$$E_{\text{behavior}} = \int_0^T (\hat{P}_{\text{drive}}(t) + \hat{P}_{\text{sensor}} + \hat{P}_{\text{compute}} + \hat{P}_{\text{other}}) dt \quad (13)$$

The plan-level energy is the sum over all behavior units:

$$E_{\text{plan}} = \sum_{k=1}^N E_{\text{behavior}}^{(k)} \quad (14)$$

If $E_{\text{plan}} > E_{\text{available}}$, the plan is infeasible and triggers dynamic replanning.

4.3. Comparative Feedback and Semantic Updates

The semantic properties employed in this framework can be broadly distinguished between those that are explicitly defined and those that are implicitly derived. Explicit properties correspond to absolute values that can be directly obtained from sensors or physical models. They include measurable or specification-based parameters such as robot mass (m), wheel radius (r), rolling resistance coefficient (c_{rr}), pose, velocity, shape, and color, which are primarily utilized in the *Multi-Source Energy Prediction Strategy* to estimate the energy consumption before execution.

In contrast, implicit properties serve as heuristic indicators that cannot be directly measured but are continuously updated through execution experience and feedback. These properties do not aim to provide precise numerical accuracy; instead, they function as

relative signals that help answer questions like “Was this path more demanding than others?” or “Was this place riskier compared to alternatives?”. By doing so, they allow robots to make context-aware judgments about mission feasibility and task allocation, even under conditions where exact prediction is neither possible nor necessary. In what follows, we define and update the five implicit properties that guide this adaptive reasoning process.

4.3.1. Energy Complexity

This represents the proportion of additional energy consumed at place p compared to an ideal execution, normalized to $[0, 1]$.

$$\text{energyComplexity}_p = \max\left(0, \frac{E_p^{\text{real}} - E_p^{\text{ideal}}}{E_p^{\text{ideal}}}\right), \quad \in [0, 1] \quad (15)$$

4.3.2. The Delay Cost

This measures the relative delay in execution at place p . Values closer to 1 indicate significant delays.

$$\text{delayCost}_p = \max\left(0, \frac{T_p^{\text{real}} - T_p^{\text{ideal}}}{T_p^{\text{ideal}}}\right), \quad \in [0, 1] \quad (16)$$

4.3.3. Difficulty Level

This continuous score integrates energy inefficiency, obstacle density, and stop frequency. Higher values correspond to greater traversal difficulty. Each weight is selected according to the environment, depending on whether energy efficiency, obstacle complexity (e.g., in airports), or rapid task execution is prioritized. The weights satisfy $w_e + w_o + w_s = 1$.

$$\text{difficultyLevel}_p = w_e \min\left(1, \frac{E_p^{\text{real}}}{E_p^{\text{ideal}}}\right) + w_o \frac{o_p}{o_{\max}} + w_s \frac{s_p}{s_{\max}}, \quad \in [0, 1] \quad (17)$$

4.3.4. Risk Level

The risk level quantifies the average failure and error rate per visit, capped to $[0, 1]$. Here, Err_p counts excessive-drift events, F_p counts task failures, V_p is the visit count, and $\gamma \in [0, 1]$ balances error versus failure importance.

$$\text{riskLevel}_p = \min\left(1, \frac{Err_p + \gamma F_p}{V_p + 1}\right), \quad \in [0, 1] \quad (18)$$

4.3.5. Energy Deviation

This captures whether execution was more efficient (<0) or more costly (>0) than predicted, scaled to $[-1, 1]$ for stability.

$$\text{energyDeviation}_p = \tanh\left(\frac{E_p^{\text{real}} - \hat{E}_p}{E_p^{\text{ideal}}}\right), \quad \in [-1, 1] \quad (19)$$

All updated properties are stored in the semantic model as implicit knowledge and used in the ontology-based reasoning process for plan refinement (Section 5.4). By combining **explicit constants** such as c_{rr} , m , and r with these **implicit heuristic indicators**, the framework balances physical modeling with experience-driven adaptation. This design reflects the reality that precise energy prediction in dynamic environments is inherently infeasible. It emphasizes that relative, experience-based indicators are more effective for guiding robots toward safer and more efficient long-term autonomy.

5. Semantic Energy-Aware Planning and Execution Reasoning

Our semantic planning framework builds upon structured ontological representations introduced in previous studies [3,4], where environments are abstracted as graphs composed of semantically annotated locations and objects. These representations go beyond geometric mapping by incorporating contextual properties such as spatial semantics, agent roles, and interaction zones.

This framework enables robots to autonomously interpret and execute high-level abstract instructions. Even in the absence of explicitly defined goals, robots can autonomously interpret abstract queries and respond accordingly. For example, instead of issuing a concrete command such as “Robot A, move to location X”, one may pose an abstract query like “Is location X organized?”. In this case, a robot that identifies the unmet condition can autonomously allocate the task to itself and carry it out. Unlike classical approaches that primarily focus on numerical efficiency or path optimality, our contribution emphasizes context-aware semantics and cooperative flexibility.

Consequently, the proposed semantic planning framework allows robots to dynamically infer appropriate tasks and define goals by reasoning over task relevance, object affordances, and spatial constraints. As a result, in complex and unstructured scenarios, robots can perform commands not merely in a numerically optimized way but in a flexible and context-sensitive manner, leading to more robust and cooperative task execution.

5.1. Semantic-Knowledge-Based Task Modeling and Domain Definition

In our semantic planning framework, tasks are hierarchically modeled to reflect the semantic structure of robots, environments, and their interactions. We adopt a multi-level task abstraction, including mission-level, coarse-level, and fine-level tasks, each defined with domain actions in Planning Domain Definition Language (PDDL) [46]. Unlike classical domain modeling, our structure separates cost/duration assignment into the problem file, enabling adaptive and robot-specific planning based on semantic knowledge.

Each behavior refers to semantic preconditions such as connectivity, occupancy, or interaction capability (e.g., `is_connected_to`, `is_occupied_by`) and is grounded in the domain file. However, the numerical cost values, such as energy consumption, duration, and feasibility, are dynamically loaded at planning time from the problem file. This separation ensures modularity, robot independence, and efficient cost-aware reasoning in heterogeneous multi-robot systems.

The domain file defines actions and their corresponding logical preconditions and effects. For instance, actions such as `move`, `visit`, and `disinfect` are specified as durative-action types in PDDL. Their duration and energy requirements are abstracted as external functions evaluated at planning time using runtime parameters from the problem file. As shown in Listing 1, the planner checks both symbolic connectivity and available energy before applying the action and updates the energy state upon completion. This formulation ensures that actions are only scheduled when energetically feasible and that the remaining energy is correctly reflected after execution.

This semantic modeling approach facilitates structured task representation, hierarchical decomposition, and energy-viable filtering of robot–place combinations. By integrating energy-conscious constraints directly into the reasoning process, the planner can effectively eliminate infeasible configurations at an early stage, thereby minimizing the computational overhead and ensuring that only valid, executable plans are produced.

Listing 1. Coarse-level move action referencing external energy and duration functions.

```

(:durative-action move
:parameters (?r - robot ?from ?to - place)
:duration (= ?duration (delay ?from ?to))
:condition (and
  (over all (is_connected_to ?from ?to))
  (at start (is_located_at ?r ?from))
  (at start (< (available-energy) (energy ?r)))
  (at start (is_not_occupied_by ?to ?r))
)
:effect (and
  (at start (not (is_located_at ?r ?from)))
  (at start (not (is_docking ?r)))
  (at end (is_located_at ?r ?to))
  (at end (is_not_occupied_by ?from ?r))
  (at end (increase (available-energy) (energy ?from ?to)))
  (at end (decrease (total-energy ?r) (energy ?from ?to)))
)))
)

```

5.2. Semantic Energy-Aware Goal Refinement and Problem Definition

In our semantic planning framework, the PDDL problem file is generated at runtime to reflect mission-level goals and the robot's semantic environment model. This ensures efficient task planning by including only relevant entities and constraints.

As the number of places, agents, and attributes in PDDL grows, traditional planners suffer from a high computational overhead. To address this, we exploit semantic knowledge such as robot *purpose*, *capability*, and contextual information in Section 3.2 to prune irrelevant objects and actions.

We further apply a Cost-Bounded Semantic Expansion process that explores only locations reachable within a given energy budget. This algorithm does not optimize action sequences but filters out infeasible candidate places. Thus, our method balances two goals: (1) retaining semantically relevant and executable places and (2) excluding unreachable areas that slow down planning or yield invalid plans. The expansion strategy is summarized in Algorithm 1.

Algorithm 1 Cost-bounded semantic expansion.

Require: Semantic graph $G = (V, E)$, start node v_0 , energy limit E_{\max}

Ensure: Set of valid places $\mathcal{P}_{\text{valid}}$

```

1:  $\mathcal{P}_{\text{valid}} \leftarrow \{v_0\}$ 
2:  $Q \leftarrow$  priority queue with  $(v_0, 0)$ 
3:  $C[v_0] \leftarrow 0$ 
4: while  $Q$  is not empty do
5:    $(v, c) \leftarrow \text{pop from } Q$ 
6:   if  $c > E_{\max}$  then
7:     continue
8:   end if
9:   for each neighbor  $u$  of  $v$  do
10:     $w \leftarrow \text{energy\_cost}(v, u)$ 
11:     $c_{\text{new}} \leftarrow c + w$ 
12:    if  $u \notin C$  or  $c_{\text{new}} < C[u]$  then
13:       $C[u] \leftarrow c_{\text{new}}$ 
14:      push  $(u, c_{\text{new}})$  to  $Q$ 
15:       $\mathcal{P}_{\text{valid}} \leftarrow \mathcal{P}_{\text{valid}} \cup \{u\}$ 
16:    end if
17:  end for
18: end while
19: return  $\mathcal{P}_{\text{valid}}$ 

```

The resulting set $\mathcal{P}_{\text{valid}}$ contains only semantically relevant places that are reachable under the current energy constraints and is used to construct the initial state and reachable goals in the PDDL problem file. Our problem generator operates as an on-demand module that dynamically queries the semantic environment database, incorporating attributes such as energy cost, risk level, and task complexity at each location [4], thereby enabling adaptive and context-aware planning. This formulation further allows for early verification of mission feasibility; if no valid expansion satisfies the symbolic goal within the given constraints, the planner can trigger replanning or discard the task altogether, avoiding unnecessary computation over infeasible options.

5.3. The Semantic-Knowledge-Based Task Planner

Once the PDDL domain and the corresponding problem instance are constructed, the task planner generates an action sequence using the Partial Order Planning Forward (POPF) planner [47,48], which supports time-dependent and energy-constrained behaviors. The resulting plan consists of a temporally ordered list of high-level actions such as `move`, `disinfect`, and `dock` that collectively achieve the assigned mission while satisfying spatial connectivity, the resource constraints, and the task requirements. Each action is grounded in the domain logic and is linked to a corresponding executable behavior via the semantic execution interface, which bridges planning and low-level control systems. The system interprets the POPF plan file to extract execution parameters such as temporal order, action duration, and location bindings, which are then dispatched to the task execution module. Throughout execution, the robot continuously monitors the energy consumption, the progress of actions, and environmental feedback. If any failure conditions arise, such as insufficient energy, unreachable destinations, or task timeout, the system either adapts the current plan or initiates a replanning routine. This execution framework promotes energy-efficient operation by ensuring that only feasible and cost-effective actions are selected and executed throughout the task cycle.

5.4. Semantic Energy-Aware Execution Feedback and Replanning

To ensure robust task completion, the system incorporates an execution time feedback mechanism that continuously monitors symbolic preconditions, energy consumption, and task success status. During task execution, discrepancies often arise between planned symbolic transitions and real-world outcomes due to unexpected environmental dynamics, resource fluctuation, or actuator-level errors. If action failure is detected, such as precondition violation or energy depletion, the system initiates semantic reasoning to evaluate the plan's validity and identify the cause of the failure. Based on this reasoning, the system performs semantic replanning by incorporating updated world states and execution history.

Ontology-Based Semantic Reasoning for Plan Refinement

Execution results are not recorded indiscriminately; instead, semantic reasoning is used to update relevant properties in the ontology. For example, suppose a specific location consistently causes high energy consumption or delays during execution. In that case, it may be considered for semantic annotation with properties such as `hasHighEnergyCost` or `hasDelayRisk`. However, such updates are not applied immediately. The system considers whether the observed anomalies are persistent or context-dependent, such as temporary congestion or abnormal task conditions, before modifying the ontology.

To this end, we define five representative categories of semantic rules:

- **Multi-Observation Update Rules:** Integrate repeated observations from the same robot or heterogeneous robots, and update the ontology only when sufficient consensus and reliability are ensured.

- **Integrated Stability Rules:** Combine hysteresis, sensor health gating, and outlier quarantine to prevent erroneous updates caused by transient noise, abnormal sensor readings, or degraded measurements.
- **Context-Aware Rules:** Incorporate situational factors such as time-of-day, congestion level, and environmental conditions so that semantic properties are tagged only under appropriate contexts. Thresholds are applied differentially across contexts (e.g., indoor vs. outdoor, rush-hour vs. off-peak) rather than as fixed global constants.
- **Platform-Specific Normalization Rules:** Normalize the observed values according to the robot platform characteristics (e.g., payload weight, locomotion type, sensor resolution) to ensure fair and consistent representation across different hardware. Thresholds are also adapted per platform to avoid bias toward a specific robot.
- **Aging and Forgetting Rules:** Remove or downgrade semantic annotations that are outdated or not reconfirmed for a long period, thereby maintaining the long-term consistency and adaptability of the semantic map.

It is important to note that a single fixed threshold value (e.g., for energy or delay) is not universally applicable across all environments or platforms. Indoor versus outdoor areas, narrow corridors versus open spaces, or robots with different payload weights and locomotion mechanisms may each require different thresholds θ . To address this, thresholds are applied in a context- and platform-dependent manner, rather than as global constants. Moreover, instead of embedding such constants directly into SWRL rules, we manage them as datatype properties within the ontology. This design allows thresholds to be dynamically adapted, updated, and validated from empirical execution data, ensuring both fairness across heterogeneous platforms and robustness under diverse environmental conditions.

Once validated, these annotations are automatically incorporated into the ontology using Semantic Web Rule Language (SWRL) rules [49,50]. Representative examples are provided in Table 4. This closed-loop feedback between execution and knowledge ensures long-term adaptability and context-aware task optimization.

Table 4. Example of SWRL rules for semantic energy-aware ontology updates.

SWRL Rule	Description
$\text{Place}(\text{?p}) \wedge \{ (\text{observedBy}(\text{?p}, \text{?r}) \wedge \text{obsCountBy}(\text{?p}, \text{?r}, \text{?c}) \wedge \text{thetaRepeatObs}(\text{?ctx}, \text{?n}) \wedge \text{swrlb:greaterThanOrEqual}(\text{?c}, \text{?n})) \vee (\text{distinctRobotCount}(\text{?p}, \text{?dr}) \wedge \text{thetaDistinctRobots}(\text{?ctx}, \text{?m}) \wedge \text{swrlb:greaterThanOrEqual}(\text{?dr}, \text{?m})) \} \rightarrow \text{CommitUpdate}(\text{?p}, \text{true})$	Multi-Observation Update Rule: Commit an ontology update if either (i) the same robot has confirmed the observation at least n times, or (ii) at least m distinct robots have observed it. In practice, this OR condition is implemented as two separate SWRL rules. Both thresholds n and m are stored as ontology properties for adaptive calibration.
$\text{Place}(\text{?p}) \wedge \text{hasEnergyComplexity}(\text{?p}, \text{?e}) \wedge \text{prevEnergyComplexity}(\text{?p}, \text{?eprev}) \wedge \text{isSensorHealthy}(\text{?r}, \text{true}) \wedge \text{notOutlier}(\text{?p}, \text{true}) \wedge \text{swrlb:greaterThan}(\text{?e}, \text{?thon}) \wedge \text{swrlb:greaterThan}(\text{?eprev}, \text{?thoff}) \rightarrow \text{hasStableHighEnergy}(\text{?p}, \text{true})$	Integrated Stability Rule: Combine hysteresis (on/off thresholds), sensor health gating, and outlier quarantine to prevent noisy or faulty updates.

Table 4. Cont.

SWRL Rule	Description
Place(?p) \wedge hasDelayCost(?p, ?d) \wedge timeOfDay(?t) \wedge delayThresholdByTime(?t, ?thd) \wedge swrlb:greaterThan(?d, ?thd) \rightarrow hasDelayRisk(?p, true)	Context-Aware Rule: Risk annotations are applied only under specific contexts, with thresholds adapted to factors such as time, environment, or congestion levels.
Place(?p) \wedge observedBy(?p, ?r) \wedge hasEnergyCost(?p, ?eraw) \wedge platformNormFactor(?r, ?k) \wedge swrlb:multiply(?enorm, ?eraw, ?k) \wedge thetaHighEnergy(?t, ?the) \wedge swrlb:greaterThan(?enorm, ?the) \rightarrow hasHighEnergyCost(?p, true)	Platform-Specific Normalization Rule: Normalize energy costs by platform characteristics (e.g., weight, locomotion type, sensor load). Different thresholds are applied per context to ensure fairness and prevent bias toward specific robots.
Place(?p) \wedge hasHighEnergyCost(?p, true) \wedge lastConfirmedAt(?p, ?t) \wedge now(?now) \wedge swrlb:subtract(?age, ?now, ?t) \wedge swrlb:greaterThan(?age, ?tmax) \wedge visited(?p, true) \wedge hasRecentObservation(?p, false) \rightarrow RemoveHighEnergyTag(?p)	sAging and Forgetting Rule: Annotations are removed not due to the simple absence of an observation but through counter-evidence from visits or gradual confidence decays, ensuring stable long-term map maintenance.

\wedge , \vee Logical AND operator in SWRL rule. \rightarrow Logical implication operator in SWRL rule.

5.5. Implicit Task Allocation via Energy-Aware Semantic Reasoning

In the proposed framework, task-to-robot assignment is not explicitly commanded. Instead, tasks are implicitly taken by robots that are both energetically feasible and currently unoccupied, so that the most suitable agent autonomously steps forward when a task arises. This implicit allocation is achieved by evaluating each robot's capability and contextual suitability through energy-aware semantic reasoning and symbolic planning feasibility, allowing decentralized and scalable decision-making without centralized coordination or negotiation.

5.5.1. Overview

Given a task T and a set of available robots \mathcal{R} , the system performs the following sequence:

- Filter infeasible robots using SWRL rules based on ontology-defined energy constraints and semantic properties;
- Estimate the cost for each robot based on the expected energy usage and travel distance to the task location;
- Sort candidates and attempt symbolic planning using PDDL for the most promising robots first.

5.5.2. Semantic Filtering via SWRL

Each robot $r \in \mathcal{R}$ is evaluated using Semantic Web Rule Language (SWRL) rules over the ontology \mathcal{O} to filter out those that are semantically incompatible. For example, if a robot's current energy level is insufficient for a task marked with high energy complexity or if the task involves inter-floor navigation and the robot lacks stair-climbing capability, it is excluded from consideration. As shown in Equation (20), a robot is considered eligible for a disinfection task if it is explicitly capable of performing the task (canDisinfect) and its current energy level exceeds a predefined threshold.

$$\begin{aligned}
& \text{Task}(?t) \wedge \text{taskType}(?t, \text{disinfect}) \wedge \text{assignedTo}(?t, ?r) \wedge \text{Robot}(?r) \\
& \wedge \text{canDisinfect}(?r, \text{true}) \wedge \text{hasEnergyLevel}(?r, ?e) \\
& \wedge \text{swrlb:greaterThan}(?e, 20) \rightarrow \text{isEligible}(?r, \text{true})
\end{aligned} \tag{20}$$

5.5.3. Optimization-Based Task Allocation

Task allocation is formulated not as a sequential candidate evaluation but as an optimization problem that considers time, energy, and task importance. Each robot–task pair is associated with a cost function that integrates expected execution time, energy consumption for navigation and task execution, and a weight derived from task priority. Within the PDDL framework, this is implemented using the `(:metric minimize A)` construct, where the planner minimizes objectives such as total makespan, cumulative energy cost, and weighted penalties for unfulfilled high-priority tasks. As a result, the planner does not merely check feasibility but derives an allocation that simultaneously satisfies symbolic constraints and optimizes temporal efficiency, energy awareness, and priority-driven fairness. This formulation enables a decentralized yet context-aware task distribution and integrates naturally with symbolic planning frameworks.

5.5.4. Replanning Under Execution Failures

In practical execution, task feasibility may change due to unexpected conditions such as mission failure, energy depletion, or sensor malfunction. When such situations occur, the system triggers a replanning process that updates the ontology with the latest execution feedback and re-evaluates candidate allocations. This ensures that tasks are reassigned in a timely manner, maintaining overall mission continuity despite dynamic uncertainties.

6. Experiments and Results

This section presents the experimental setup for evaluating the proposed semantic planning framework based on an extended Triplet Ontological Semantic Model (TOSM) that incorporates energy-aware reasoning. The experiments were conducted in a real-world indoor office environment with dynamic elements such as human movement, variable lighting, and obstacle placement. The two-wheeled disinfection robot was tested under diverse task scenarios to validate its long-term adaptive planning, energy-based semantic reasoning, and cooperative task capabilities.

6.1. The Experimental Platform and Execution Environment

We employed a two-wheeled mobile disinfection robot equipped with LiDAR, ultrasonic sensors, an RGB camera, an infrared sterilization lamp, cooling fans, and a mechanical blind. The semantic planner and ontology-based reasoning engine were executed on an onboard local computing unit.

In the real-world setup, we tested the system in an actual office environment containing both people and static obstacles, using a single robot configuration, as shown in Figure 4. Additionally, we conducted further experiments in a virtual simulation environment to evaluate multi-robot energy-aware task planning and the scalability of semantic reasoning.

6.2. Energy Prediction Accuracy Evaluation

This section presents a quantitative analysis of the energy prediction accuracy through two experiments, using standard evaluation metrics such as the MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and MAPE (Mean Absolute Percentage Error) to assess the difference between the predicted and measured energy consumption.



Figure 4. The experimental setup in an indoor office environment.

6.2.1. Model-Based Energy Prediction

This experiment aims to evaluate the accuracy of energy prediction using our internal computational model. The robot was placed in a calm and controlled environment (i.e., with minimal dynamic obstacles and stable temperature) and executed a set of predefined tasks. The predicted energy consumption E_{model} was computed from our energy modeling framework, considering movement, task type, and module-level usage. The measured energy E_{real} was collected via onboard sensors. The goal is to assess the fundamental accuracy of our energy abstraction under ideal conditions. The comparison results are summarized in Table 5, and the Figure 5 illustrates the difference between the predicted and actual energy consumption.

Table 5. Energy prediction accuracy in controlled environment (using model-based estimation).

Task ID	Predicted Energy (E_{model})	Measured Energy (E_{real})	Error (%)
Task-01	6.687	15.288	0.562
Task-02	6.778	12.744	0.468
Task-03	7.69	11.537	0.333
MAE:			12.153
RMSE:			19.128
NMAE:			0.794

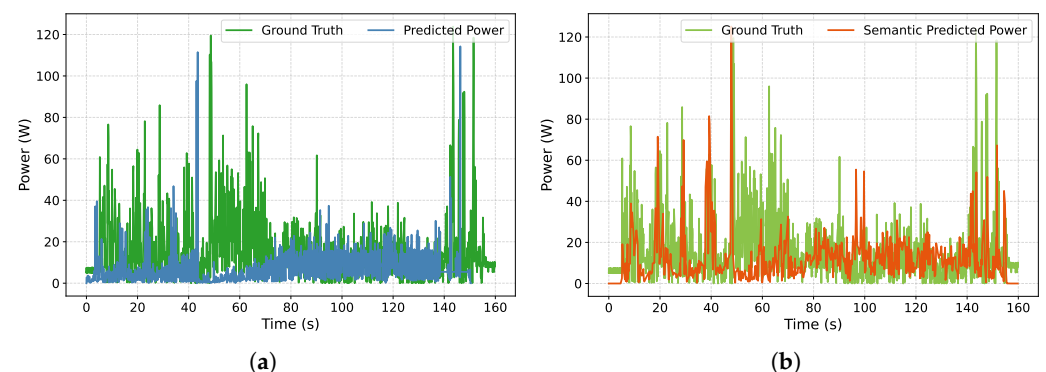


Figure 5. Comparison between model- and semantic-reasoning-based energy prediction methods. (a) Model-based energy prediction. (b) Semantic-reasoning-based energy prediction.

6.2.2. Semantic-Reasoning-Based Energy Prediction

This experiment demonstrates the effectiveness of the proposed semantic reasoning framework. Leveraging high-level contextual features such as energy complexity, place difficulty, and terrain type, the robot inferred the energy predictions E_{semantic} through

ontological reasoning. The evaluation was conducted in a real-world office environment that included human activity and dynamic obstacles. During task execution, the actual energy consumption E_{real} was measured for comparison. The results confirm the practical validity of our semantic-based prediction approach, with a summary provided in Table 6.

Table 6. Energy prediction accuracy based on semantic reasoning in a real environment.

Task ID	Predicted Energy (E_{semantic})	Measured Energy (E_{real})	Error (%)
Task-01	12.027	15.288	0.213
Task-02	8.768	12.744	0.312
Task-03	9.389	11.537	0.186
MAE:			11.7
RMSE:			17.382
NMAE:			0.764

In both experimental settings, the predicted and measured energy values are compared to evaluate the accuracy and reliability of the proposed approach. Performance is assessed using standard metrics, including the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). While model-based predictions under controlled conditions yield high accuracy, the semantic predictions exhibit practical applicability in dynamic environments, maintaining acceptable levels of error.

6.3. Visualization of Semantic Planning Results

To visually demonstrate the effectiveness of our semantic energy-aware planning approach, we present a series of illustrative examples that compare the planning results under different conditions.

We begin by presenting the planning results for both single-robot and multi-robot scenarios. In the single-robot case, four tasks are assigned sequentially, and the robot computes an optimized path considering both energy and time efficiency. In the multi-robot case, four robots are assigned individual tasks and execute them in a coordinated manner without collisions. Figure 6 provides visual examples of the resulting plans for each case.

To demonstrate the impact of semantic information on task planning, we present a comparative analysis of the planning results with and without semantic reasoning. In the absence of semantic knowledge, as shown in Figure 7a, the robot follows an idealized path primarily based on distance or geometric optimality. In contrast, when semantic information is incorporated as illustrated in Figure 7a, the planner can generate more adaptive and accurate plans by leveraging prior experience or context-aware reasoning. This is especially critical in multi-robot scenarios, where accurate durative planning is essential for efficient task coordination and overall mission success.

To investigate the effect of energy awareness on task allocation further, we conducted an experiment simulating a scenario in which one robot has insufficient battery to complete a mission. As shown in Figure 8a, the baseline algorithm, which does not consider energy constraints, assigns the task to a robot that lacks the required energy capacity, leading to an infeasible plan. In contrast, the energy-aware strategy first evaluates the feasibility of task execution based on the robot's available energy and excludes unsuitable candidates during the assignment process. As illustrated in Figure 8b, the planner successfully allocates the task to an alternative robot with sufficient resources, thereby ensuring successful execution and preventing failure.

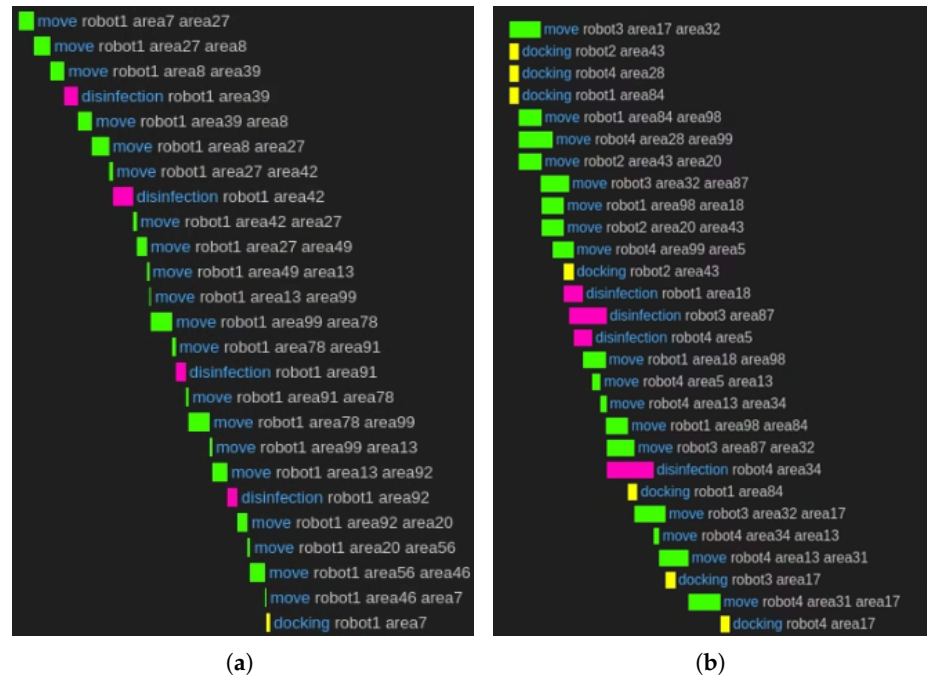


Figure 6. Visualization of task allocation and execution in a robot planning scenario. (a) The task plan for a single robot. (b) The task plan for a multi-robot.

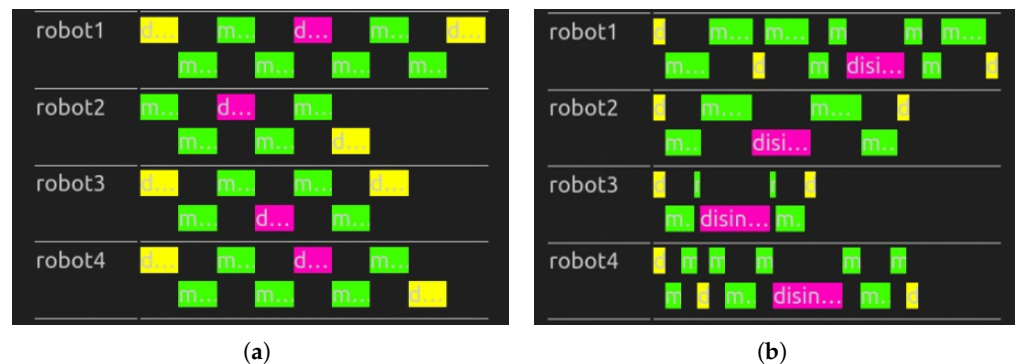


Figure 7. Comparison of planned behavior sequences with and without semantic knowledge. (a) Without semantic knowledge. (b) With semantic knowledge.

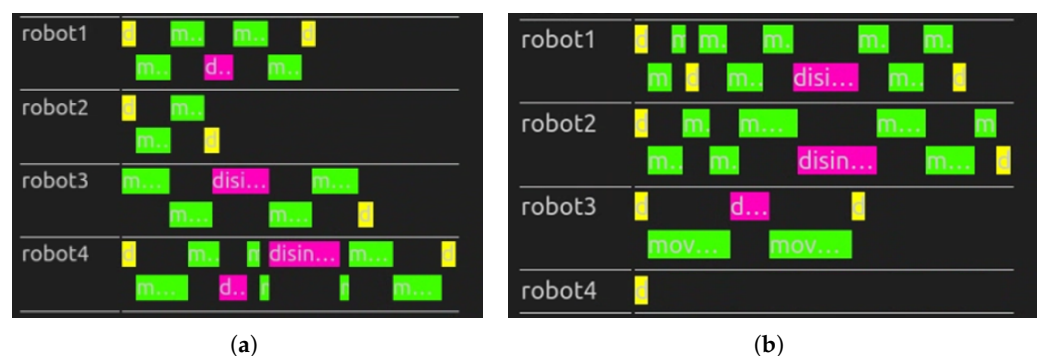


Figure 8. Comparison of allocation task with and without semantic knowledge. (a) Without semantic energy awareness. (b) With semantic energy awareness.

6.4. Comparison with Previous Work

To comprehensively assess the effectiveness of our proposed semantic energy-aware planning framework, we conducted a comparative evaluation with a previous rule-based

task planner [4,42], in which the author participated as a co-developer. The experiments were performed using the same robot platform and task set in a single-robot setting for fair comparison.

6.5. Per-Task Processing Time Analysis

To evaluate the computational efficiency of our semantic planner, we measured the average processing time required to generate a valid multi-task plan across diverse robotic and environmental conditions. Each test involved generating action sequences based on a set of goals distributed over a semantic map. Table 7 summarizes the results across varying map sizes (i.e., number of semantic places). As the environment grew larger, the baseline method exhibited a steady increase in the per-task planning time due to its exhaustive search strategy and lack of semantic filtering. In contrast, our method consistently achieved a lower planning latency by leveraging semantic constraints such as energy complexity and spatial connectivity to prune infeasible actions during planning. To further investigate planner scalability concerning the workload density, we varied the number of tasks relative to the number of available robots. As shown in Table 8, the baseline method showed an increasing latency under higher task-to-robot ratios. Meanwhile, our approach maintained lower and more stable computation times, even under overloaded conditions, such as 4× or 5×. These results demonstrate that the proposed method not only improves the execution performance but also enhances the computational scalability under both spatial and task-based complexity.

Table 7. Average processing time per place count.

Places	1 Robots		2 Robots		4 Robots		8 Robots		16 Robots		32 Robots	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
100	0.191	0.191	0.339	0.327	0.880	0.818	1.246	1.115	20.163	17.340	27.630	22.795
200	0.215	0.207	0.425	0.394	0.980	0.873	1.403	1.201	22.471	18.451	68.347	53.728
300	0.239	0.220	0.511	0.453	1.079	0.920	9.560	7.813	59.313	46.396	109.065	81.496
400	0.262	0.231	0.596	0.506	1.179	0.959	22.776	17.727	68.087	50.611	711.048	503.659
500	0.286	0.242	0.682	0.552	1.278	0.99	27.109	20.046	315.292	222.106	1313.032	879.002
600	4.747	3.824	5.207	3.830	11.963	7.962	37.794	22.508	917.275	482.079	1915.015	872.396
700	9.208	7.059	9.732	6.780	22.648	14.193	521.518	290.312	1519.258	739.372	2516.999	1048.75
800	13.670	9.949	14.256	9.377	32.332	19.004	886.582	459.052	2121.242	949.845	timeout	1178.282
900	18.131	12.490	18.781	11.623	46.017	24.160	898.666	53.242	2604.765	275.699	timeout	2260.994
1000	22.592	14.685	23.306	13.517	54.702	27.898	138.504	60.942	835.481	309.128	timeout	2926.885

Unit: s.

Table 8. Average processing time per task.

Task (Scale Factor)	1 Robots		2 Robots		4 Robots		8 Robots		16 Robots		32 Robots	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
0.5× robot	0.000	0.000	0.176	0.164	4.225	3.938	8.274	7.540	39.986	35.288	71.698	61.128
1× robot	0.266	0.239	0.617	0.542	1.303	1.148	10.631	9.145	56.111	46.079	99.003	79.532
2× robot	5.571	4.879	13.43	11.209	21.289	17.812	118.009	94.407	311.448	235.397	504.888	364.712
3× robot	10.058	8.448	204.475	165.798	398.893	317.933	593.31	438.734	982.145	696.282	1370.98	896.787
4× robot	53.035	42.428	334.309	259.153	615.582	451.252	896.856	612.631	1459.403	887.887	2021.95	1086.211

Unit: s.

6.5.1. Energy Efficiency per Task

To evaluate how efficiently tasks consumed energy, we measured the average energy usage per task while varying both the number of robots and the types of semantic locations. The conventional method, based on a fixed task distribution scheme that lacked responsiveness to environmental changes or power-related constraints, revealed a notable increase in inefficiency as the complexity of the system grew. By comparison, our proposed system incorporates semantic-level reasoning and a modular energy modeling approach to

adaptively determine power-efficient actions, taking into account the terrain properties, expected task workload, and currently available energy resources.

We define the average energy consumed per task $E_{\text{efficiency}}$ as

$$E_{\text{efficiency}} = 1 - \left| \frac{E_{\text{actual}} - E_{\text{theoretical}}}{E_{\text{theoretical}}} \right| \quad (21)$$

where E_{actual} is the total energy consumed during execution, and $E_{\text{theoretical}}$ is the number of tasks performed. The total energy includes both locomotion and operation power, measured in watt-seconds (Ws), accumulated over the full task plan. Lower values of E_{avg} indicate more efficient task scheduling and path execution. It is important to note that this metric represents a relative measure of efficiency. Rather than reflecting the absolute accuracy of energy savings, it highlights the extent to which each robot improves its behavior through semantic reasoning and adaptive decision-making. In addition to analyzing the absolute energy consumption, we also evaluated whether energy usage was effectively distributed across all robots by calculating the normalized entropy of the energy distribution:

$$p_i = \frac{E_i}{\sum_{j=1}^n E_j}, \quad H = - \sum_{i=1}^n p_i \log p_i, \quad H_{\text{norm}} = \frac{H}{\log n} \quad (22)$$

This measure captures how evenly energy is shared among the participating robots, with values closer to 1 indicating more balanced distributions. To evaluate how uniformly the energy was distributed across the robots, we employed entropy-based metrics. First, the energy consumption ratio of each robot was computed as

$$p_i = \frac{E_i}{\sum_{j=1}^n E_j} \quad (23)$$

where E_i denotes the energy consumed by the i -th robot, and n is the total number of robots.

Using these ratios, we computed the Shannon entropy to quantify the dispersion of energy consumption:

$$H = - \sum_{i=1}^n p_i \log p_i \quad (24)$$

A higher entropy value indicates that energy usage is more evenly distributed across robots.

To normalize the entropy into a scale between 0 and 1, regardless of the number of robots, we used

$$H_{\text{norm}} = \frac{H}{\log n} \in [0, 1] \quad (25)$$

Here, $H_{\text{norm}} = 1$ implies a perfect energy balance among robots, whereas values closer to 0 indicate that the energy consumption is concentrated on a few robots. The resulting entropy values suggest that the proposed task assignment strategy promotes a favorable degree of energy distribution among robots. As shown in Tables 9 and 10, our method consistently achieved lower E_{avg} values across all test conditions compared to the baseline. The efficiency gap widened in large-scale scenarios (e.g., 50+ places, 8+ robots), where redundant navigation and static planning in the baseline method caused frequent power spikes. Our approach reduced such inefficiencies by using semantic features such as terrain difficulty and spatial connectivity to prune costly paths and adapt the behavior dynamically based on battery constraints.

6.5.2. Long-Term Execution Stability

To assess the long-term stability of our planner, we conducted a sequential execution of six tasks over a continuous 30-min mission in a dynamic indoor environment. During this period, the time and energy drift were recorded for each task under varying task-to-robot ratios, as summarized in Table 11.

Table 9. Overall energy efficiency in multi-robot tasks.

Task	1 Robots		2 Robots		4 Robots		8 Robots		16 Robots		32 Robots	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
0.5 × robot	0.762	0.953	0.792	0.946	0.764	0.937	0.688	0.901	0.740	0.835	0.736	0.868
1 × robot	0.782	0.953	0.754	0.945	0.760	0.917	0.748	0.899	0.712	0.853	0.673	0.832
2 × robot	0.793	0.952	0.711	0.933	0.752	0.930	0.740	0.875	0.656	0.847	0.692	0.813
3 × robot	0.743	0.932	0.764	0.915	0.682	0.883	0.725	0.856	0.634	0.823	0.664	0.775
4 × robot	0.775	0.922	0.743	0.901	0.693	0.854	0.644	0.842	0.673	0.794	0.612	0.745

Table 10. Energy distribution efficiency among multiple robots.

Task / Robot	1 Robots		2 Robots		4 Robots		8 Robots		16 Robots		32 Robots	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
0.5 × robot	0.751	0.952	0.691	0.923	0.735	0.901	0.674	0.884	0.592	0.833	0.554	0.805
1 × robot	0.643	0.865	0.717	0.934	0.64	0.875	0.656	0.837	0.611	0.843	0.515	0.777
2 × robot	0.725	0.895	0.615	0.913	0.692	0.823	0.564	0.853	0.605	0.814	0.582	0.791
3 × robot	0.663	0.905	0.573	0.822	0.634	0.843	0.594	0.800	0.551	0.783	0.506	0.763
4 × robot	0.685	0.883	0.622	0.853	0.554	0.815	0.583	0.774	0.523	0.754	0.542	0.722

We define time drift and energy drift as the relative deviation between the planned and actual values:

$$\text{Time Drift} = \frac{|t_{\text{real}} - t_{\text{plan}}|}{t_{\text{plan}}}, \quad \text{Energy Drift} = \frac{|E_{\text{real}} - E_{\text{pred}}|}{E_{\text{pred}}}.$$

These metrics quantify the accumulation of planning errors over long horizons, capturing both computational latency and model inaccuracy. As shown in Table 11, the results demonstrate that while our semantic planner maintains robustness at moderate loads, extreme task densities induce cumulative errors that may necessitate mid-mission re-planning or adaptive thresholding.

Table 11. Time and energy drift over repetitive multi-robot task executions.

Runs	1 Robots		2 Robots		4 Robots		8 Robots		16 Robots		32 Robots	
	Time	Energy	Time	Energy	Time	Energy	Time	Energy	Time	Energy	Time	Energy
1	0.123	0.154	0.130	0.171	0.115	0.187	0.124	0.200	0.195	0.210	0.268	0.226
2	0.136	0.163	0.150	0.187	0.109	0.184	0.132	0.202	0.210	0.212	0.276	0.224
3	0.131	0.160	0.110	0.172	0.097	0.188	0.140	0.204	0.304	0.214	0.224	0.225
4	0.125	0.163	0.095	0.175	0.146	0.190	0.162	0.205	0.235	0.207	0.309	0.227
5	0.120	0.168	0.105	0.178	0.134	0.193	0.260	0.215	0.251	0.217	0.342	0.230
6	0.180	0.181	0.115	0.166	0.259	0.196	0.171	0.210	0.190	0.220	0.348	0.232
7	0.097	0.170	0.110	0.185	0.166	0.200	0.229	0.212	0.288	0.215	0.373	0.235
8	0.190	0.172	0.101	0.186	0.177	0.199	0.239	0.224	0.303	0.222	0.386	0.239
9	0.134	0.175	0.150	0.188	0.205	0.203	0.260	0.216	0.319	0.227	0.407	0.241
10	0.157	0.182	0.120	0.208	0.250	0.197	0.279	0.221	0.366	0.232	0.431	0.248

6.5.3. Scenario-Based Evaluation with Diverse Tasks

To evaluate the versatility of the proposed framework, we conducted experiments across a range of tasks and situational contexts. The robot was assigned three representa-

tive tasks with distinct operational characteristics: **disinfection** (periodic area coverage with sterilization devices), **security patrol** (human detection and patrolling with signaling), and **air purification** (continuous movement with purification modules activated). Each task was further tested under two environmental conditions, daytime with a higher presence of humans and nighttime with reduced activity or the lights off, resulting in six experimental scenarios. By combining diverse task types and contextual variations, this evaluation provides a broader perspective on how the proposed approach compares with the baseline method, highlighting its adaptability across heterogeneous task demands and environmental settings.

The results show that our method consistently outperformed the baseline in both the reduction in the planning time and the improvement in the energy efficiency across all tasks and conditions, as summarized in Table 12. In particular, the security patrol task exhibited significant reductions in the planning time while maintaining stable gains in energy efficiency during both daytime and nighttime scenarios. These findings indicate that the proposed semantic energy-aware reasoning framework achieves consistent improvements under varied task and contextual settings, supporting decentralized and context-aware task execution without centralized coordination.

Table 12. Comparison of baseline and proposed methods across tasks and conditions.

Metric/Condition	Disinfection		Security Patrol		Air Purification	
	Baseline	Ours	Baseline	Ours	Baseline	Ours
Planning Time (s)						
Day	5.728	4.422	5.934	3.985	5.801	4.360
Night	6.101	4.863	5.932	3.680	6.215	5.072
Energy Efficiency						
Day	0.653	0.881	0.701	0.905	0.745	0.916
Night	0.688	0.824	0.729	0.891	0.702	0.862

This completes the Section 6. Next, we present the Section 7.

7. Conclusions

This paper introduced a semantic task planning framework tailored to mobile robotic systems operating in dynamic and resource-constrained environments. The framework integrates energy-aware reasoning with an ontological task feasibility assessment, enabling robots to make informed decisions that account for both contextual semantics and physical limitations. At the core of the proposed approach is the extension of the Triplet Ontological Semantic Model (TOSM), which is augmented to incorporate energy-related execution profiles and temporal dynamics. This allows the system to generate task plans that are not only logically valid but also practically executable over extended mission horizons.

Unlike conventional planners that rely solely on geometric information or pre-defined task scripts, our framework leverages high-level semantic attributes such as spatial usage patterns, interaction constraints, and operational affordances to support flexible and adaptive planning. These semantic representations allow robots to infer task suitability, environmental risks, and energy requirements, even in scenarios where the goals are abstract or not explicitly defined. As a result, the system is well suited to real-world applications that demand robustness, scalability, and autonomy under uncertainty.

To evaluate the effectiveness of the proposed system, we conducted five structured experiments across both simulated and physical indoor environments. The results demonstrate that our framework can

- Dynamically regulate region-level semantic costs based on real-time deviations in energy consumption and execution delays;
- Suspend or reroute ongoing missions in response to feasibility violations, such as low remaining battery or obstructed paths;
- Interpret and adapt to human interventions, changes in environmental topology, and task priority updates;
- Enhance the energy efficiency over repeated task executions through contextual feedback and semantic adjustment;
- Enable decentralized task allocation by providing a shared and interpretable semantic cost map among multiple robots.

These findings suggest that the proposed approach represents a promising direction for achieving long-term autonomy (LTA) in multi-robot applications, including innovative indoor service environments, cooperative surveillance and patrol, and energy-constrained task planning in logistics or emergency response domains. In particular, the ability to reason over both semantic abstraction and physical feasibility indicates that the framework can demonstrate high adaptability in contextually defined task spaces or partially specified mission environments, rather than being limited to rigidly pre-structured scenarios.

Future Work

Building on the capabilities demonstrated in this work, several research directions will be pursued to enhance the framework's adaptability and scalability further. First, we will enhance the energy modeling component by incorporating more realistic physical parameters and empirical measurements, moving beyond the current simplified assumptions. In parallel, we plan to conduct extensive validation through long-duration experiments and cross-domain case studies to ensure that the framework is robustly grounded in both theoretical soundness and empirical evidence. Second, we plan to incorporate continual/online learning mechanisms that leverage accumulated execution data to continuously refine semantic attributes and energy models. This will allow the planner to improve progressively over time and adapt proactively to evolving operational contexts. Third, we will validate the applicability of the framework across diverse platforms and heterogeneous environments while also addressing abnormal conditions such as sensor failures, communication disruptions, or highly noisy scenarios. To ensure resilience in such situations, avoidance and recovery strategies will be developed in parallel with strengthened real-time replanning capabilities. Third, we aim to extend the applicability of the framework beyond standard indoor domains to a wide range of environments, including logistics warehouses, smart factories, and hospitals, as well as outdoor scenarios such as urban roads, traffic-congested areas, and large-scale mixed infrastructures. Fourth, perception-driven knowledge extraction and NLP-based semantic grounding will be employed to automate ontology population, reducing reliance on manual engineering. Finally, we will expand the system's deployment to outdoor autonomous vehicles, focusing on unified semantic reasoning for navigation, cooperative multi-agent behaviors, and energy-efficient route planning in shared, large-scale environments. This transition will help bridge the gap between semantic indoor robotics and real-world vehicular autonomy under a standardized planning paradigm.

Author Contributions: Conceptualization, J.-H.C., D.-S.S., S.-H.B., E.-J.K., J.-W.P. and T.-Y.K.; Validation, J.-H.C., D.-S.S., S.-H.B., J.-W.P. and T.-Y.K.; Investigation, J.-H.C., D.-S.S., Y.-C.A. and T.-Y.K.; Resources, J.-H.C., D.-S.S., Y.-C.A. and E.-J.K.; Data curation, J.-H.C., D.-S.S. and E.-J.K.; Writing—original draft, J.-H.C.; Supervision, S.-H.B., J.-W.P. and T.-Y.K.; Project administration, T.-Y.K.; Funding acquisition, J.-H.C. and T.-Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support of the Technology Innovation Program (20018198, Development of Hyper self-vehicle location recognition technology in the driving environment under bad conditions) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Acknowledgments: Thanks for the help of the reviewers and editors.

Conflicts of Interest: Author Sang-Hyeon Bae is employed by the company Samsung Display Co., Ltd. and author Jeong-Won Pyo is employed by the company DXR 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

References

1. Kunze, L.; Hawes, N.; Duckett, T.; Hanheide, M.; Krajník, T. Artificial intelligence for long-term robot autonomy: A survey. *IEEE Robot. Autom. Lett.* **2018**, *3*, 4023–4030. [\[CrossRef\]](#)
2. Stramigioli, S. Energy-aware robotics. In *Mathematical Control Theory I: Nonlinear and Hybrid Control Systems*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 37–50.
3. Joo, S.H.; Manzoor, S.; Rocha, Y.G.; Bae, S.H.; Lee, K.H.; Kuc, T.Y.; Kim, M. Autonomous navigation framework for intelligent robots based on a semantic environment modeling. *Appl. Sci.* **2020**, *10*, 3219. [\[CrossRef\]](#)
4. Joo, S.; Bae, S.; Choi, J.; Park, H.; Lee, S.; You, S.; Uhm, T.; Moon, J.; Kuc, T. A flexible semantic ontological model framework and its application to robotic navigation in large dynamic environments. *Electronics* **2022**, *11*, 2420. [\[CrossRef\]](#)
5. McCormac, J.; Handa, A.; Davison, A.; Leutenegger, S. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 4628–4635.
6. Bowman, S.L.; Atanasov, N.; Daniilidis, K.; Pappas, G.J. Probabilistic data association for semantic slam. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1722–1729.
7. Tenorth, M.; Beetz, M. KnowRob—knowledge processing for autonomous personal robots. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 4261–4266.
8. Riazuelo, L.; Tenorth, M.; Di Marco, D.; Salas, M.; Gálvez-López, D.; Mösenlechner, L.; Kunze, L.; Beetz, M.; Tardós, J.D.; Montano, L.; et al. RoboEarth semantic mapping: A cloud enabled knowledge-based approach. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 432–443. [\[CrossRef\]](#)
9. Wang, Y.; Fan, Y.; Wang, J.; Chen, W. Long-term navigation for autonomous robots based on spatio-temporal map prediction. *Robot. Auton. Syst.* **2024**, *179*, 104724. [\[CrossRef\]](#)
10. Sousa, R.B.; Sobreira, H.M.; Moreira, A.P. A systematic literature review on long-term localization and mapping for mobile robots. *J. Field Robot.* **2023**, *40*, 1245–1322. [\[CrossRef\]](#)
11. Beetz, M.; Beßler, D.; Haidu, A.; Pomarlan, M.; Bozcuoğlu, A.K.; Bartels, G. Know rob 2.0—A 2nd generation knowledge processing framework for cognition-enabled robotic agents. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 512–519.
12. Notomista, G. Resilience and Energy-Awareness in Constraint-Driven-Controlled Multi-Robot Systems. In Proceedings of the 2022 American Control Conference (ACC), Atlanta, GA, USA, 8–10 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 3682–3687.
13. Setter, T.; Egerstedt, M. Energy-constrained coordination of multi-robot teams. *IEEE Trans. Control Syst. Technol.* **2016**, *25*, 1257–1263. [\[CrossRef\]](#)
14. Eiffert, S.; Wallace, N.D.; Kong, H.; Pirmarzashti, N.; Sukkariéh, S. Resource and response aware path planning for long-term autonomy of ground robots in agriculture. *Field Robot.* **2022**, *2*, 1–33. [\[CrossRef\]](#)
15. Zhang, H.; Zhang, Y.; Yang, T. A survey of energy-efficient motion planning for wheeled mobile robots. *Ind. Robot Int. J. Robot. Res. Appl.* **2020**, *47*, 607–621. [\[CrossRef\]](#)
16. Seewald, A.; de Marina, H.G.; Midtby, H.S.; Schultz, U.P. Energy-aware planning-scheduling for autonomous aerial robots. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 2946–2953.
17. Zhang, L.; Zhao, J.; Lamon, E.; Wang, Y.; Hong, X. Energy efficient multi-robot task allocation constrained by time window and precedence. *IEEE Trans. Autom. Sci. Eng.* **2023**, *22*, 18162–18173. [\[CrossRef\]](#)

18. Robbins, J.A.; Thompson, A.F.; Brennan, S.; Pangborn, H.C. Energy-aware predictive motion planning for autonomous vehicles using a hybrid zonotope constraint representation. In Proceedings of the 2025 American Control Conference (ACC), Denver, CO, USA, 8–10 July 2025; IEEE: Piscataway, NJ, USA, 2025; pp. 2546–2553.
19. Sun, Q.; Qi, W.; Qian, H. EeLsT: An Energy-efficient Long-short Term Approach for Sustainable Sailboat Autonomy in Disturbed Marine Environment. *IEEE Trans. Robot.* **2025**, *41*, 4195–4214. [\[CrossRef\]](#)
20. Gasche, S.; Kallies, C.; Himmel, A.; Findeisen, R. Energy aware and safe path planning for unmanned aircraft systems. *arXiv* **2025**, arXiv:2504.03271. [\[CrossRef\]](#)
21. Raychaudhuri, S.; Chang, A.X. Semantic Mapping in Indoor Embodied AI—A Survey on Advances, Challenges, and Future Directions. *arXiv* **2025**, arXiv:2501.05750. [\[CrossRef\]](#)
22. Song, X.; Liang, X.; Huaidong, Z. Semantic mapping techniques for indoor mobile robots: Review and prospect. *Meas. Control* **2025**, *58*, 377–393. [\[CrossRef\]](#)
23. Manzoor, S.; Rocha, Y.G.; Joo, S.H.; Bae, S.H.; Kim, E.J.; Joo, K.J.; Kuc, T.Y. Ontology-based knowledge representation in robotic systems: A survey oriented toward applications. *Appl. Sci.* **2021**, *11*, 4324. [\[CrossRef\]](#)
24. Ge, Y.; Zhang, S.; Cai, Y.; Lu, T.; Wang, H.; Hui, X.; Wang, S. Ontology based autonomous robot task processing framework. *Front. Neurobot.* **2024**, *18*, 1401075. [\[CrossRef\]](#)
25. Choi, J.H.; Bae, S.H.; Gilberto, G.G.; Seo, D.S.; Kwon, S.W.; Kwon, G.H.; Ahn, Y.C.; Joo, K.J.; Kuc, T.Y. A Multi-robot Navigation Framework using Semantic Knowledge for Logistics Environment. In Proceedings of the 2024 24th International Conference on Control, Automation and Systems (ICCAS), Jeju, Republic of Korea, 29 October–1 November 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 927–932.
26. Choi, J.H.; Bae, S.H.; An, Y.C.; Kuc, T.Y. Development of an advanced navigation system for autonomous mobile robots for logistics environments. In Proceedings of the 2023 23rd International Conference on Control, Automation and Systems (ICCAS), Yeosu, Republic of Korea, 17–20 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1286–1291.
27. Pyo, J.W.; Choi, J.H.; Kuc, T.Y. An Object-Centric Hierarchical Pose Estimation Method Using Semantic High-Definition Maps for General Autonomous Driving. *Sensors* **2024**, *24*, 5191. [\[CrossRef\]](#)
28. John, T.; Koopmann, P. Planning with OWL-DL ontologies. In Proceedings of the 27th European Conference on Artificial Intelligence, ECAI 2024, Santiago de Compostela, Spain, 19–24 October 2024; IOS Press BV: Amsterdam, The Netherlands, 2024; pp. 4165–4172.
29. Bernardo, R.; Sousa, J.M.; Gonçalves, P.J. Ontological framework for high-level task replanning for autonomous robotic systems. *Robot. Auton. Syst.* **2025**, *184*, 104861. [\[CrossRef\]](#)
30. Armeni, I.; He, Z.Y.; Gwak, J.; Zamir, A.R.; Fischer, M.; Malik, J.; Savarese, S. 3d scene graph: A structure for unified semantics, 3d space, and camera. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5664–5673.
31. Rosinol, A.; Violette, A.; Abate, M.; Hughes, N.; Chang, Y.; Shi, J.; Gupta, A.; Carlone, L. Kimera: From SLAM to spatial perception with 3D dynamic scene graphs. *Int. J. Robot. Res.* **2021**, *40*, 1510–1546. [\[CrossRef\]](#)
32. Kolve, E.; Mottaghi, R.; Han, W.; VanderBilt, E.; Weihs, L.; Herrasti, A.; Deitke, M.; Ehsani, K.; Gordon, D.; Zhu, Y.; et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv* **2017**, arXiv:1712.05474.
33. Kappler, D.; Bohg, J.; Schaal, S. Leveraging big data for grasp planning. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 4304–4311.
34. Blochliger, F.; Fehr, M.; Dymczyk, M.; Schneider, T.; Siegwart, R. Topomap: Topological mapping and navigation based on visual slam maps. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 3818–3825.
35. Choi, J.H.; Pyo, J.W.; An, Y.C.; Kuc, T.Y. TOSD: A Hierarchical Object-Centric Descriptor Integrating Shape, Color, and Topology. *Sensors* **2025**, *25*, 4614. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Hughes, N.; Chang, Y.; Carlone, L. Hydra: A real-time spatial perception system for 3D scene graph construction and optimization. *arXiv* **2022**, arXiv:2201.13360. [\[CrossRef\]](#)
37. Ito, T.; Funada, R.; Sampei, M.; Notomista, G. Energy-Aware Task Allocation for Teams of Multi-mode Robots. *arXiv* **2025**, arXiv:2503.12787. [\[CrossRef\]](#)
38. Pal, A.; Chauhan, A.; Baranwal, M. Together We Rise: Optimizing Real-Time Multi-Robot Task Allocation using Coordinated Heterogeneous Plays. *arXiv* **2025**, arXiv:2502.16079. [\[CrossRef\]](#)
39. Mokhtari, M.; Mohamadi, P.H.A.; Aernouts, M.; Singh, R.K.; Vanderborght, B.; Weyn, M.; Famaey, J. Energy-aware multi-robot task scheduling using meta-heuristic optimization methods for ambiently-powered robot swarms. *Robot. Auton. Syst.* **2025**, *186*, 104898. [\[CrossRef\]](#)
40. Notomista, G.; Mayya, S.; Emam, Y.; Kroninger, C.; Bohannon, A.; Hutchinson, S.; Egerstedt, M. A Resilient and Energy-Aware Task Allocation Framework for Heterogeneous Multirobot Systems. *IEEE Trans. Robot.* **2022**, *38*, 159–179. [\[CrossRef\]](#)

41. Korsah, G.A.; Stentz, A.; Dias, M.B. A comprehensive taxonomy for multi-robot task allocation. *Int. J. Robot. Res.* **2013**, *32*, 1495–1512. [[CrossRef](#)]
42. Bae, S.; Joo, S.; Choi, J.; Pyo, J.; Park, H.; Kuc, T. Semantic knowledge-based hierarchical planning approach for multi-robot systems. *Electronics* **2023**, *12*, 2131. [[CrossRef](#)]
43. Tomy, M.; Lacerda, B.; Hawes, N.; Wyatt, J.L. Battery charge scheduling in long-life autonomous mobile robots. In Proceedings of the 2019 European Conference on Mobile Robots (ECMR), Prague, Czech Republic, 4–6 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
44. Fouad, H.; Beltrame, G. Energy autonomy for resource-constrained multi robot missions. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2021; IEEE: Piscataway, NJ, USA, 2020; pp. 7006–7013.
45. Bouillet, E.; Febblowitz, M.; Liu, Z.; Ranganathan, A.; Riabov, A. A knowledge engineering and planning framework based on OWL ontologies. *Proc. Second Int. Compet. Knowl. Eng.* **2007**, *191*.
46. Fox, M.; Long, D. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res.* **2003**, *20*, 61–124. [[CrossRef](#)]
47. Coles, A.; Coles, A.; Fox, M.; Long, D. Forward-chaining partial-order planning. In Proceedings of the International Conference on Automated Planning and Scheduling, Toronto, ON, Canada, 12–16 May 2010; Volume 20, pp. 42–49.
48. Chen, Q.; Pan, Y.J. An optimal task planning and agent-aware allocation algorithm in collaborative tasks combining with pddl and popf. *arXiv* **2024**, arXiv:2407.08534. [[CrossRef](#)]
49. Lawan, A.; Rakib, A. The semantic web rule language expressiveness extensions-a survey. *arXiv* **2019**, arXiv:1903.11723. [[CrossRef](#)]
50. Horrocks, I.; Patel-Schneider, P.F.; Boley, H.; Tabet, S.; Grosz, B.; Dean, M. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*; W3C Member Submission; W3C Team: Wakefield, MA, USA, 2004; Volume 21, pp. 1–31.

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.