# When Can Planning Benefit from Common-Sense Knowledge?

**Ma'ayan Armony, Albert Meroño-Peñuela, Gerard Canal**
Department of Informatics, King's College London
{maayan.armony, albert.merono, gerard.canal}@kcl.ac.uk

## Abstract

To improve the quality of plans and ensure their alignment with the real-world environment, existing works have explored the integration of external knowledge with AI planning. Most of these approaches are motivated with examples from real-world scenarios and evaluated on a subset of domains, with a focus on household environments. The collective choice of certain tasks for evaluation suggests that some tasks can further benefit from external knowledge than others. However, the choice of environment in integration methods is rarely rationalised or justified, and the generalisability of approaches across domains is not often discussed. In this paper, we examine the characteristics that influence when planning tasks benefit from integrating external knowledge, such as semantic and common-sense knowledge, into decision-making within AI planning. We provide a systematic classification of these characteristics according to their scope (spanning the task, domain, problem, and plan) and relate them to the specific stages of the planning process in which they play a role – the task description, the search process, post-planning analysis, and plan execution.

## Introduction

The information required to solve a planning task is typically expressed using the Planning Domain Definition Language (PDDL) (McDermott et al. 1998), which provides a structured description of the environment and the problem. The knowledge that is available to symbolic planners during the planning process is restricted to what is encoded in the PDDL domain and problem files. The limited presence of background knowledge in standard planning specifications means that planners are constrained in their common-sense reasoning to the knowledge explicitly represented in the input files. Consequently, planners may lack semantic and contextual understanding of the actions they consider and the broader implications these actions may have for the state of the world. Some efforts have been made to represent additional information in the planning description, and provide richer expressivity of planning tasks through extension of classical planning, such as modelling temporal constraints as introduced in PDDL 2.1 (Fox and Long 2003). Nonetheless, these representations still require an explicit and rigid representation that often cannot be implicitly derived from context or dynamics of the environment. Probabilistic planning aims to plan under uncertainty, with possible transitions being modelled with probability distributions. However, Meli, Nakawala, and Fiorini (2023) highlight limitations in probabilistic planning approaches, noting that their reliance on data-driven mechanisms makes it difficult to capture temporal and semantic relations, which are often critical factors for effective task execution.

To address these shortcomings, various approaches have integrated external knowledge into planning (Tenorth and Beetz 2009; Kaiser et al. 2014; Conti, Varde, and Wang 2020). This frequently includes semantic knowledge and common-sense knowledge (CSK), that is, knowledge about the world that is implicitly understood and assumed by humans, and is consequently often omitted from textual data (Ilievski et al. 2021). These integration methods demonstrate that incorporating knowledge about the environment into the planning process, based on semantics and context, could influence the planner's decision-making by informing it on the feasibility, preferred ordering, or hidden constraints of certain actions. Integrating CSK, especially in the forms of spatial, temporal, or causal knowledge, into the planning system allows for more real-world constraints to be considered, with the potential to increase the likelihood that a planner finds a successful plan solution, and that a plan provided by the planner can be successfully executed in the real world.

Current works on knowledge integration evaluate their approach on a limited set of planning tasks; however, they do not typically motivate or reason over the choice of domains or problems, nor analyse the effect the choice of task has on the effectiveness of their approach. Nevertheless, the commonalities in the choice of tasks across methods suggest that some tasks are more appropriate for knowledge integration. The evaluation tasks used in current literature pose further limitations. In terms of abstraction, many PDDL task descriptions differ from the tasks used to assess integrating knowledge into planning, with the latter typically more closely based on real-world scenarios. For example, objects in PDDL descriptions are often denoted as symbols (e.g., $o_1, o_2 \dots o_n$), whereas existing methods generally assume objects of different sizes, shapes, or materials. It is therefore crucial to analyse the characteristics of task planning descriptions that help them benefit from external knowledge.

In this position paper, we identify and classify the characteristics that make tasks benefit from integration of CSK and semantic knowledge. We provide a detailed analysis of task, domain, problem, and plan characteristics that indicate when a planning task could benefit from knowledge integration, expanding on factors related to the representation of each of these scopes as well as their structure. Additionally, to aid in choosing the most appropriate integration stage for a task based on its semantics and structure, we detail the applicable stage of modification for each characteristic. We classify the stages of knowledge integration into four: (1) modification of the task description, (2) integration during the planning process, (3) post-planning analysis of a plan solution, and (4) its execution. This work aims to provide the community with a taxonomy to guide the selection of integration methods for planning tasks according to their characteristics and the methods' stage of modification. Furthermore, this classification can support authors in selecting appropriate tasks to further evaluate existing integration methods.

## Related Work

External knowledge has long been used to support planning processes. Current work varies in the types of knowledge incorporated, with a focus on semantic, spatial, and common-sense knowledge, in the assumptions made about the environment, and in the planning components it aims to improve. Several works incorporate semantic knowledge to support robust task execution in household environments. Bouguerra, Karlsson, and Saffiotti (2008) utilise semantic domain knowledge to monitor and interpret the execution of tasks, proposing two variants: one that assumes deterministic and observable environments, and another that explicitly models uncertainty. Similarly, Tenorth and Beetz (2009) and its later extension Beetz et al. (2018) introduce KnowRob, a knowledge-processing framework that enables robots to reason about the current environment state and select appropriate actions, with a focus on tasks requiring complex manipulation. These works evaluate their models in indoor kitchen environments, where rich spatial semantics and object relations are essential. More recently, Liu et al. (2024) use vision-language models (VLMs) to support pick-and-place tasks without relying on a strict domain definition, also in a domestic environment.

Another line of research focuses on injecting commonsense knowledge (CSK) into symbolic planning. Kaiser et al. (2014) extract CSK from text to augment the domain description with plausible object-location relations. Their approach is evaluated in a kitchen environment, focusing primarily on concrete concepts (e.g., containers, appliances) while filtering out abstract information such as temperature. They assume domains contain objects and locations and emphasise spatial relations such as "at" and "on". They also highlight the role of type checking in ensuring that extracted CSK predicates are semantically valid. Al-Moadhen et al. (2013) enrich planning domains with CSK together with a semantic map of the environment. Their method expands available action descriptions in domestic scenarios by assuming predefined sets of objects, locations, and actions and linking them through semantic relations. External knowledge can also improve planning quality by modelling physical properties of agents and objects. Conti, Varde, and Wang (2020) demonstrate that providing robots with knowledge about the human characteristics and object attributes enhances the quality of human-robot interaction and collaboration in a vehicle assembly task. They assume heterogeneous physical properties among parts, which differ in their weights, materials, and structures. External knowledge can also be incorporated to affect the search process. Zhao, Lee, and Hsu (2023) investigate object rearrangement tasks in partially-observable household settings (VirtualHome (Puig et al. 2018)), which involve long-term planning and many object-interaction actions. Because the number of objects increases the branching factor substantially, injecting CSK into the world model further expands the search space. However, they show that combining this enriched model with LLM-based guidance can make planning more effective despite the larger space. Their findings highlight that external knowledge can both help and hinder planning, depending on the structure of the task, raising the question of when CSK is beneficial and when it creates unnecessary complexity.

## Background

### Classical Planning

We adopt the definition of classical planning (Ghallab, Nau, and Traverso 2004, 2016) as $P = \langle S, A, \gamma, s_0, G\rangle$, where:

- $S$ is a finite set of possible states,
- $A$ is a finite set of available actions,
- $\gamma : S \times A \rightarrow S$ is a state transition function,
- $s_0 \in S$ is the initial state,
- $G \subseteq S$ is the set of goal states.

Each state $s$ is composed of the propositions which are set to be true in that state. We define $\mathcal{P}(s)$ as the set of propositions true in state $s$, and $|\mathcal{P}(s)|$ as the state's cardinality. Every action $a \in A$ has preconditions $\mathrm{pre}(a)$, and effects $\mathrm{eff}(a) = \langle \mathrm{add}(a), \mathrm{del}(a)\rangle$, such that $\mathrm{add}(a)$ are propositions which are introduced to the state, and $\mathrm{del}(a)$ are those that are removed. We assume that these two sets are disjoint, i.e., $\mathrm{add}(a) \cap \mathrm{del}(a) = \emptyset$.

**Plan.** A plan $\pi = \langle a_1, a_2, ..., a_n\rangle$, $a_i \in A$ is a sequence of actions that solves $P$, which, if applied to the initial state, will bring the system to the goal state, i.e., $\gamma(s_0, \pi) = s_n$ such that $s_n \in G$. Let $C(\pi)$ denote the cost of a plan $\pi$. A plan $\pi_{\mathrm{opt}} = \langle a_1, \dots, a_m\rangle$, is optimal if $C(\pi_{\mathrm{opt}}) = \min_{\pi\in\Pi} C(\pi)$, where $\Pi$ is the set of all plans that solve $P$.

**Temporal Planning.** A temporal planning task extends classical planning by allowing durative actions. For a planning task $P = \langle S, A^d, \gamma, s_0, G\rangle$, each durative action $a \in A^d$ is a tuple

$$a = (\mathrm{pre}_{\mathrm{start}}(a),\ \mathrm{pre}_{\mathrm{over}}(a), \mathrm{pre}_{\mathrm{end}}(a),\\ \mathrm{eff}_{\mathrm{start}}(a),\ \mathrm{eff}_{\mathrm{end}}(a), \mathrm{dur}(a))$$

where $\mathrm{pre}_{\{\mathrm{start,over,end}\}}(a)$ and $\mathrm{eff}_{\{\mathrm{start,end}\}}(a)$ are the preconditions and effects that must hold at the start of $a$, throughout its execution, or at its end, respectively, and $\mathrm{dur}(a) > 0$ is the fixed duration of $a$.

### PDDL

PDDL is a formal language used to describe the information required by the planner to solve a task, namely, the actions, their preconditions and effects, the available objects, and the initial and goal states. This representation is defined in a domain file, which represents the known information about the environment, and a problem file, which defines the initial and goal states. A symbolic planner is then used to find a plan, often through heuristic search such as A*, Greedy Best-First Search, or other domain-independent heuristics.

### Common-sense Knowledge

Common-sense knowledge (CSK) is information that is intuitive to humans and is thus typically not mentioned explicitly in textual data, including spatial, temporal, and causal knowledge. CSK is often encoded in common-sense knowledge bases, which may be specific ontologies and common-sense knowledge graphs (CSKGs), such as Cyc (Lenat 1995), ConceptNet (Speer, Chin, and Havasi 2017), and CSKG (Ilievski, Szekely, and Zhang 2021). These structured knowledge bases describe objects, concepts, images, or other data types as entities, with relations that represent the entities' real-world characteristics. Large Language Models (LLMs) have also been increasingly utilised as common-sense knowledge bases, due to the large amount of data they are trained on (Petroni et al. 2019; Zhao, Lee, and Hsu 2023). However, their reasoning abilities are limited due to their architecture (Li et al. 2022; Bian et al. 2024). In this paper, we therefore do not consider the use of LLMs for planning as integration of external knowledge into planning, and only consider LLMs as a knowledge source when this is the prescribed role of the LLM agent.

## Integrating Knowledge into Planning

Using pure symbolic planning, the planner and the agent executing a plan do not have access to the semantic meaning of symbols and to the relevant CSK, and their contextual knowledge is limited to what is represented in the task description.

While large domains and expressive action descriptions allow for modelling substantial common-sense information, they may lead to inefficiencies such as state explosion during search. For this reason, we suggest that some CSK may be better used during or after planning rather than being explicitly encoded in PDDL. The knowledge that should be modelled explicitly in the task description would be the minimal set required to obtain valid plans in an acceptable time frame for the concrete application. While this may be domain-dependent, most planning domains rely on abstractions to make the problem solvable. External knowledge would then be that which improves the obtained solutions, either by bridging the abstractions, or improving the description with respect to a specified metric. For instance, making them more semantically correct or more understandable to a human.

We specify four distinct stages in which external knowledge may be integrated into the planning process:

1. Pre-planning: Modification of the PDDL domain and problem, i.e., editing the PDDL domain before the planner is called.
2. Planning: Restricting the search space, i.e., during plan search for a specific problem.
3. Post-planning: Analysis of the found plan solution based on external knowledge before execution, to improve the plan or choose between plans.
4. Execution: Monitoring the execution stage of a found solution, or repairing a failed plan.

This distinction is crucial, as the environment is represented at different stages with varying levels of abstraction, allowing for different types of knowledge to be considered in various ways with different semantics. For example, the level of abstraction in the high-level plan created by the planner could elicit a different type of CSK than during action execution. This also applies during planning, when the level of information about the current state and the contextual information about the taken actions differs from the task description and the resulting plan. It is therefore the case that the external knowledge has multiple use cases, depending on the stage of integration. These include using knowledge as an additional heuristic, for finding a valid solution to a problem in less time, or with fewer resources (e.g., reducing the search space); directing towards optimal solutions, i.e., bridging the gap between difficulty of satistficing and optimal solutions; or for finding plans that are more appropriate for real-world execution.

In the following sections, we detail the characteristics which make tasks benefit from integrating semantics and CSK, by their scope: task, domain, problem, or plan characteristics. The characteristics were determined based on an analysis conducted on common PDDL tasks (e.g. IPC domains (Hoffmann et al. 2006)). The analysis included extracting applied effects from VAL's output (Howey, Long, and Fox 2004) and planners' search output, with plan steps and the time taken to find a plan used as heuristics for complexity. Information from the PDDL files was extracted, such as numbers of agents and objects in problem, and object types and action properties in domains. In each section, we begin with the features related to semantic representation, followed by those that depend on structural representation. Finally, in Table 1, we consolidate all characteristics with the possible stages of integration for each one.

## Planning Paradigms

Distinct planning paradigms can benefit from the integration of external knowledge into the planning process in different ways. Below, we detail the features of temporal planning, planning with preferences, and planning under uncertainty that affect the inclusion of CSK into planning. We regard classical planning as the standard case, for which CSK is applicable under certain terms, dependent on the characteristics of the task.

**Temporal planning.** In temporal planning, where action durations are modelled explicitly in the domain definition, integrating additional temporal knowledge can be used to create a more realistic and nuanced representation of action durations in the task description. As a result, the actions executed in the found plans would be more reflective of the real world, which can in turn increase the success and optimality of plan execution by reducing the gap between the expected and real duration of actions. It may also be the case that actions have different durations in different contexts, which may be inefficient to model in temporal planning, and the information which influences the action's duration may not be modelled in the task description. Temporal knowledge can be used during search or at the post-planning analysis stage to estimate action duration based on the state of the environment, making the expected duration of the plan better match its actual duration.

**Planning with preferences.** Planning with preferences considers soft goals, that is, goals that are desired to be achieved but are not necessary for a valid plan. The increased and more granular expressivity of tasks when planning with preferences provides additional semantics that can be used to improve plan quality. Soft goals can have a preference hierarchy, with some being preferred over others. CSK and semantic knowledge from the PDDL description can be used to determine goal preferences, for example, by assessing risk and prioritising goals which have been set as safety-critical. CSK for goal preferences can also be utilised when there is a time constraint for execution, and some soft goals may be more important than others, even if they take longer to achieve. Using external knowledge during search can provide further granularity in prioritising goals, by considering the actions previously taken as part of the decision. In this case, preferences based on common sense don't have to be modelled explicitly in the PDDL description but still influence the planner's decisions, improving the alignment of the search process with human decision-making; thus enhancing the user's satisfaction with the plan and its execution.

**Planning under uncertainty.** Probabilistic planning aims to handle uncertainty in stochastic environments, requiring explicit modelling of the known contingencies. External knowledge can be utilised to automatically consider semantics and context for adjusting probabilities and finding new possible transitions, by modifying the task description or by augmenting the search space with contextual transitions. The same approach could be taken for execution monitoring or plan-repair during plan execution. For example, querying a knowledge base to acquire relevant information about the current state of the environment can help determine whether the plan is likely to fail. The knowledge can also be used for replanning or plan-repair, to find the actions taken that are more likely to have caused failure.

Similar approaches could be considered for contingent and conformant planning, where the planner can better consider possible changes with augmented task description, or during the search process and execution by predicting changes in the environment using causal and temporal knowledge. In partially-observable planning, some information is inherently unknown to the agent. The gap in information between the real-world environment and the known environment could be bridged by using CSK to update the belief state. CSK could also be used for domain repair; if a domain includes multiple locations, but some predicates representing the physical relations between them are missing from the problem definition, these could be automatically filled according to CSK. For instance, in the Elevator domain (Koehler and Schuster 2000), this may be `(above f0 f1)`, which can be implicitly inferred using CSK. In a household domain, which may contain locations such as `kitchen` and `living room`, external knowledge can be useful to determine the most probable location of each room based on the objects it contains and the layout of the house. This also applies to locations such as cities, whose geographical location is known. CSK could also be used during plan execution, by modelling augmented knowledge as observations.

## Domains

In this section, we detail the domain characteristics that may impact the benefit of integrating external knowledge into planning, focusing on the language and semantics used to represent the domain, the relations between actions and their characteristics, and the scale and soundness of the domain representation.

**Expressivity.** ADL and STRIPS are both description formalisms for classical planning. ADL allows a richer expression of contextual information in predicates, which may contain quantifiers, implications and conditional effects. This aids with the integration of contextual knowledge directly into the domain description by allowing additional representations of explicit information. In particular, external knowledge can be incorporated into the task description to represent context-dependent consequences of actions by adding universally or existentially quantified conditions or by using conditional effects. In STRIPS, under the closed-world assumption, propositions that are not explicitly stated are inferred as false. CSK could be used to correct or replace facts that should be true based on causal reasoning, but are not explicitly modelled. This could be implemented explicitly in the task description, or rather, for a given state during search, where the context of previous actions can also be considered.

**Semantic alignment.** The semantics of the domain description must not contradict physical laws (unless the setting does not abide by conventional physics, which should be explicitly mentioned), and they should reflect the intended use of the domain. When there is a close or exact alignment between the symbolic representation and its meaning, the integrated knowledge is more likely to be relevant and useful for improving the planning process and the execution of plans, at all stages of integration.

A planner does not parse the semantic meaning of words, and therefore, any symbol can be used to represent an action, type, predicate, or object name. These names should not be meaningless strings and should align with the meaning behind the symbol; for an action, it should also align with its preconditions and effects. For example, if an action is called `pick-up`, but its preconditions include `(holding ?object)` and effects include `(not` `(holding ?object))`, then the semantics of the domain are misaligned and misleading, as these action specifications better align with a `put-down` action. External knowledge, which is found based on the semantics of the representation, could help detect this misalignment and attempt to align the representation with the dynamics of the represented environment. This can be achieved by modifying the task description or, during execution, by leveraging the perceived state of the environment.

**Typing.** Untyped domains may benefit from incorporation of semantic knowledge. Implicit types may exist in the domain, and those could be inferred from the existing semantics and added to the domain description, or used to eliminate nonsensical actions during planning, post planning or execution. Additional constraints based on the semantics of the types can also be added to the domain description. Nevertheless, typed domains may benefit from a better integration of knowledge due to a richer semantic representation, which could provide a better context for finding relevant knowledge and constraints based on causal or temporal information.

**Relations between actions.** We define similar actions $a$ and $a'$ as actions which are not identical, but share a significant amount of their preconditions and effects. We express that $a$ is similar to $a'$ as $a \sim a'$, under a predefined threshold $t$ $(0 < t < 1)$, as:

$$\begin{gathered} a \sim a' \\ \Updownarrow \\ a \neq a' \wedge \left(\frac{|\mathrm{pre}(a) \cap \mathrm{pre}(a')|}{|\mathrm{pre}(a) \cup \mathrm{pre}(a')|} > t\right) \wedge \left(\frac{|\mathrm{eff}(a) \cap \mathrm{eff}(a')|}{|\mathrm{eff}(a) \cup \mathrm{eff}(a')|} > t\right) \end{gathered}$$

If a domain contains at least two similar actions, i.e., $\exists a, a' \in A : a \sim a'$, the most appropriate choice between using $a$ or $a'$ could be improved either by incorporating semantics and additional facts about each action to the description, by augmenting the current state with information during search, or by choosing from possible plans after planning, attempting to amplify the distinction between the actions and making them less similar.

Following the definition for independent actions by Ghallab, Nau, and Traverso (2004), two actions $a$ and $a'$ are independent of each other iff:

$$\begin{gathered} (\mathrm{del}(a) \cap [\mathrm{pre}(a') \cup \mathrm{add}(a')] = \emptyset) \wedge \\ (\mathrm{del}(a') \cap [\mathrm{pre}(a) \cup \mathrm{add}(a)] = \emptyset) \end{gathered}$$

That is, the propositions removed by one action do not affect the applicability of the other, and the effects added by one action are not removed by the other. If two actions in a domain are independent of each other, they can often occur in various orders or, in temporal planning, concurrently. Further knowledge about the implications of actions may improve the choice of action order and make it more appropriate for real-world execution by making the actions dependent, forcing a certain order in the plans. This could be done by updating the description and replanning, or through post-planning analysis.

**Action characteristics.** Actions with more than $n$ effects, $(|\mathrm{eff}(a)| > n)$, or, in temporal planning, taking more than $m$ time steps $(\mathrm{dur}(a) > m)$, may be broken down into multiple smaller actions, which could potentially improve efficiency and thus the optimality of a plan, by providing additional ordering options. For instance, making food and washing dishes are two actions that humans often execute concurrently. An action representing washing dishes could be broken into washing $k$ dishes, where the duration of the action is determined by $k$ and can fit in the food-making timeline. However, adding multiple actions to the task description could significantly increase the search space; it may therefore be more beneficial to split the actions during search or when analysing the plans given by the planner.

External knowledge can be useful when one or more actions in the domain have physical features as preconditions, which are also explicitly defined in the initial state. For example, the Elevator domain has levels which are required to be explicitly stated, and the initial state must include predicates to represent the relations between those levels, even when those can be implicitly understood by humans. Another example is the Blocksworld variant in which blocks have different sizes (Winograd 1972). In this domain, the size of the blocks is explicitly represented, and their implications are considered. However, many objects in real-world domains, such as household items, have different shapes and sizes, and that is usually not explicitly represented in the PDDL domain. External knowledge could, given object names, automatically and dynamically provide the implied constraints without explicitly representing those for each object. In addition, if an object is composed of other smaller objects and can be taken apart, disassembling it may have implications on the agent's manipulation abilities with each component. For example, a jar may be deemed a fragile object, but if its lid has been removed, the lid itself is not fragile, which could impact the agent's interaction with it.

Another realisation of this notion utilises spatial knowledge; if the domain includes predicates that represent spatial relations between locations, the names of the locations, when those have a distinct semantic meaning, can be used to augment the planning process with spatial knowledge to detect nonsensical paths that are not feasible.

In domains where actions could lead to a harmful scenario and are thus safety-critical, incorporating CSK can help strike a balance between optimality and risk. For example, in the Climber domain as presented by Little and Thiebaux (2007), the person has a 40% chance of falling in the optimal plan. External knowledge can help identify when it is appropriate to take a risk and when a higher cost should be tolerated to prevent a dangerous action. This can be achieved through contingent planning with conditional cases, during search with contextual knowledge based on the current state, by selecting between potential plans, or monitoring execution and determining when a plan should be revised.

**General domain characteristics.** When domains are incomplete or contain incorrect preconditions or effects, external knowledge can be leveraged to detect flaws in the description. For instance, when a domain models containers and represents their capacities using a numeric function, if the symbolic capacity values do not match the real physical capacities of the containers, the planner may generate infeasible plans (e.g., placing an object into a container that is too small). Incorporating external spatial knowledge, such as the actual dimensions or volume of the containers, can correct these discrepancies or flag the domain description as inconsistent with the real world. A function could also be wrongly used in an action's precondition, for example, if an action for passing an object through a door includes a precondition `(>= (width ?obj) (width ?door)`, the representation would be incorrect and infeasible, as the object must be less wide than the door to pass, rather than the opposite. This could be detected using CSK and fixed to a precondition `(<= (width ?obj) (width ?door))`. A wrong domain representation may go undetected at the representation stage, for example, when the task lacks sufficient semantic information to infer its intended meaning. However, it may still be possible to detect causal and temporal inconsistencies during search or execution by exploiting the problem representation and sensor data.

When fluents represent resources, CSK can provide additional information on how to use them efficiently. For instance, in transportation domains (Helmert 2001), the agent's speed may impact consumption. The vehicle's speed also depends on the type of vehicle in use, and could be predicted more accurately when context is considered. A better understanding of resource consumption could help to prevent the search heuristic from under- or overestimating the plan's true cost, if the resource is part of the cost function.

Whilst enriching the domain description with additional knowledge could contribute to finding a better plan, the domain must not be too large. Zhao, Lee, and Hsu (2023) show that using a common-sense world model in large-scale tasks is not always effective due to the size of the search space. Adding actions, predicates, or objects can make the search space too large, even for simple problems, if the domain is already quite complex, with a large number of actions and predicates. CSK could still be beneficial in large-scale domains, but its integration may be more appropriate and effective in later stages of the planning process.

## Problems

The semantics, structure, and complexity of a problem, can all impact the incorporation of knowledge into the task. We first detail the features concerning how a problem is described, followed by problem characteristics that involve the search process, and finally, the features of problem solutions.

**Object representation.** Ideally, the names of instantiated objects convey their meanings, but that is not always the case in planning problem descriptions. When objects are numbered, modelling them with meaningful names instead, during task description or plan analysis, may inform on preferences in action order. For example, in Barman (López, Celorrio, and Olaya 2015), ingredients may have a preferred order of addition to improve the cocktail's quality. Object names can also be used to infer their physical features, which can indicate whether certain actions are feasible and how they should be executed. For instance, (Conti, Varde, and Wang 2020) take into account that during vehicle assembly, parts are made of varying materials, and the type of material (e.g., glass) should affect the decision-making process when allocating tasks to a robot and a human. When objects are given names, these considerations could be implemented to improve the execution and success of the task.

**Finding an entity.** A problem's goal state can include a proposition related to an entity with an unspecified location, i.e., the goal requires knowing the location of an agent or an object, and the location is unknown at the initial state. The process can be improved by prioritising more probable locations for the required entity, based on task semantics and CSK. This may be during the search for a plan or during execution, leveraging the perceived environment. For example, Kaiser et al. (2014) using spatial relations from texts to encode the most probable locations of kitchen items.

**Multi-agent problems.** When a problem has multiple agents, CSK could support the decision for which actions can be parallelised. Specifically, when actions can only be parallelised under specific circumstances and in a certain context, it can be unsafe or too complex to model the constraints in the task description. CSK can be used to provide a richer expression of the current state for more appropriate decision-making when finding a plan. Moreover, external knowledge can be used to determine which actions are more appropriate for one agent to execute than another, leading to improved optimality and collaboration. If a semantic distinction is created between the agents, e.g., by naming agents based on their real-world entities, the semantics may imply different capabilities between the agents. Yu et al. (2025) utilise previous experience and object attributes to determine which actions can be automated and executed by a robot, to improve human-robot collaboration for objects disassembly.

**Invariants and mutual exclusion.** When no invariants are found for a given problem, or when there is a gap between the problem's complexity and the number of invariants, the semantics of the task can help discover implicit invariants that are not found by the planner using causal, spatial and temporal knowledge. Finding additional invariants can also be used to prove task unsolvability where no solution has been found, but unsolvability has not be proven with the existing invariants. Another case is where a plan has been found but is not actually executable in the real world. This can potentially be addressed by finding an invariant during search based on external knowledge, bridging the gap between the task description and the real-world environment.

In a problem with multiple mutex groups or groups that have many variables, external causal knowledge can be used to eliminate potential states during search. Fišer and Komenda (2018) show that inferring mutex groups is at least NP-hard, and can help in detecting dead-ends. Establishing mutex groups from CSK or previously encountered knowledge can help in finding additional mutex groups or potenitally making the search process faster in some cases.

**Landmarks.** When some landmarks have yet to be discovered or cannot be identified using certain algorithms, CSK can be leveraged to attempt to locate them. Additional knowledge may reveal these landmarks by introducing explicit constraints to the task description or during the search process. A problem may have resource limitations which are not modelled in the domain, or are represented too simplistically in the problem. Representing these constraints more accurately could introduce new landmarks which would restrict the search. For instance, by incorporating temporal constraints based on additional temporal knowledge. Furthermore, causal information may help uncover implicit dependencies between goal states. This will result in one goal state that must be achieved before the other, thus finding a new ordering landmark. This implies that if plans were found where the ordering does not abide by the new ordering constraints, they should not be able to execute successfully in a real scenario, and should thus be discarded.

**Dead ends.** CSK can be used to identify avoidable dead ends and indicate a path that is more likely to lead to those dead ends; based on the context of the current environment and the next actions in the trajectory, the path can be deemed less safe or risky. When the problem is probabilistic, the retrieved knowledge can be used to tweak the probabilities in transitions. If the problem is deterministic, the knowledge can be used as a heuristic to attempt to avoid risky paths.

**Problem solutions.** Trivially, a solvable problem may benefit from knowledge integration, i.e., $|\Pi| > 0$, where $\Pi$ is the set of plans that solve the problem. When a problem is unsolvable, it may be possible to use external knowledge to identify flaws in the task representation or the cause for unsolvability in the search space. For instance, if there is an unavoidable dead-end caused by an effect that misrepresents the real world.

We consider a set of plans $\Pi_{\text{sub}} = \langle \pi_1, \dots, \pi_n \rangle$ $(n > 1)$ and an optimal plan $\pi_{\text{opt}}$ with cost $C(\pi_{opt}) = c$. $\Pi_{\text{sub}}$ contains acceptable plans when $\forall i \in \{1, \dots, n\}, c \leq C(\pi_i) \leq c + t$, where $t$ is a predefined threshold for cost tolerance. In problems where $n > 2$, that is, there are several plans with a cost within tolerance $t$, incorporating external knowledge could aid in finding the plan that aligns best with real-world expectations and execution.

A problem may also have multiple solutions, where independent actions $a_1$ and $a_2$ appear with the same parameters in two or more solutions, in a different order. This structure may imply that reordering these actions could be possible. External knowledge may influence the preference for ordering actions, which can be considered during search or when analysing the found plans.

## Plans

Structural features of plans may indicate CSK use-cases for a better-aligned execution in a real-world environment. In this section, we discuss the notions of state overlap in plan solutions and the use of CSK when plan execution fails.

**State overlap.** Let $S_\pi = \langle s_0, s_1, \dots, s_m \rangle$ be the states reached during the execution of a plan (i.e., $s_m \in G$). We define state overlap between two states as the intersection of their true propositions,

$$\text{overlap}(s_i, s_j) := \big|\mathcal{P}(s_i) \cap \mathcal{P}(s_j)\big|$$

the distance between two states as the number of actions which are executed to transition from one to another,

$$\text{dist}(s_i, s_j) := |i - j|$$

and a threshold function for overlap significance as

$$\text{thresh}(s_i, s_j) := \max\big(|\mathcal{P}(s_i)|,\ |\mathcal{P}(s_j)|\big)\ -\ \text{dist}(s_i, s_j)$$

Two non-consecutive states $s_i$ and $s_j$ are two states in a plan, where $|i - j| \geq 2$. If a plan $\pi$ contains at least two non-consecutive states which share a significant portion of the propositions, i.e., $\text{overlap}(s_i, s_j) \geq \text{thresh}(s_i, s_j)$, there is a higher chance of redundancy and reversible behaviour, with actions further in the plan undoing the effects of previous actions. This could imply that the plan could be restructured. Causal knowledge can be used to identify and provide further context for these cases at the plan analysis stage, to modify the given plan, or replan with additional context. Another case would be when some actions have an insignificant effect on the state of the environment, or when their effects are not adequately represented in the PDDL description. Causal knowledge could be used to infer the effects of these actions and add those the domain description or to augment states during the search process.

**Plan execution failure.** Bezrucav et al. (2022) explain how, when plan execution fails during action execution, improper states can occur, in which not all action effects have taken place or have been applied. The states may thus not conform with all invariants, which can prevent successful replanning. They discuss making execution more robust by ensuring proper states are maintained during failure. The finite state machines used to model action execution can be automatically completed using CSK by inferring the state after failure, whilst taking context into account; CSK can aid in determining which effects of the actions have been fulfilled and which have not, making the state more consistent with the real-world environment. For instance, if a robot navigating up an incline gets stuck because the incline has become too steep, spatial and causal knowledge could help identify the fault and adapt the state accordingly. CSK can also be used to adapt the task description to appropriately replan from an improper state, using causal information. For instance, if a robot navigates between two points and cannot reach its destination because a door which was open is now closed, it might be more sensible to ask a person to reopen the door than to take an alternative path. The current domain description might have human-robot interaction capabilities modelled for the agent, but it does not specifically include an action to ask a person to open a door. CSK could help in recognising this implicit capability and modelling it as an action the agent could take.

Lastly, when the task's goal changes during execution, CSK can be leveraged to consider the context of the current state, the previously executed actions, and the new goal, and provide useful information for replanning or repairing the plan, to effectively achieve the new goal.

| **Scope** | **Characteristic** | **Description** | **TD** | **S** | **A** | **E** |
|---|---|---|---|---|---|---|
| Paradigm | Temporal planning | Planning with temporal constraints and durations | X | X | X | |
| | Planning w/ preferences | Planning with soft constraints and goals | X | X | | |
| | Probabilistic planning | Planning under uncertainty and probabilities | X | X | | X |
| | Contingent planning | Planning under uncertainty with sensing | X | X | | X |
| | Conformant planning | Planning under uncertainty without sensing | X | X | | X |
| | Partial-observability | Planning with partial-observability and stochastic transitions | X | X | | X |
| Domain | ADL | Domain is defined using ADL | X | | | |
| | STRIPS | Domain is defined using STRIPS | X | X | | |
| | Meaningful representation | Domain semantics are meaningful and reflect real-world use | X | X | X | X |
| | Misaligned representation | Domain representation is meaningful but inconsistent | X | | | X |
| | Untyped domains | Domain does not specify object types | X | X | X | X |
| | Typed domains | Domain does specifies object types | X | X | X | X |
| | Similar actions | Domain contains at least two similar actions | X | X | X | |
| | Independent actions | Domain contains independent actions | X | | X | |
| | Actions with more than $n$ effects | Domain has actions with many effects or long duration | | X | X | |
| | Physical features in preconditions | Domain has actions with physical features as preconditions | X | X | X | X |
| | Spatial relations between locations | Domain has predicates for spatial relations between locations | X | X | X | X |
| | Safety-critical tasks | Domain has actions that may lead to harmful scenarios | X | X | X | X |
| | Flawed domains | Domain is incomplete or incorrect | X | X | | X |
| | Resources as fluents | Domain has fluents that represent resources | X | X | X | X |
| | Large-scale domains | Domain has many actions and/or predicates | | X | X | X |
| Problem | Meaningful object names | Problem's object names convey their real meaning | X | | X | |
| | Entity with unknown location | Problem includes finding an entity for its goal state | | X | | X |
| | Multi-agent problem | Problem has multiple agents to execute actions | X | X | X | X |
| | Unfound invariants | Problem has invariants that were not found by the planner | | X | | |
| | Multiple mutex-groups | Problem has many mutex groups or many variables in groups | | X | | |
| | Undiscovered landmarks | Problem has landmarks that were not found by the planner | X | X | X | |
| | Avoidable dead ends | Problem search space has dead ends but also viable paths | X | X | | |
| | Solvable problems | Problem has at least one solution | X | X | X | X |
| | Unsolvable problems | Problem has no solution | X | X | | |
| | $n$ acceptable (sub-)optimal plans | Problem has $n$ solutions within a cost threshold | | | X | |
| | Independent actions reordered | Problem has solutions with the same actions reordered | | X | X | |
| Plan | Overlap in non-consecutive states | Plan execution leads to similar states multiple times | X | | X | |
| | Plan failure in improper state | Plan fails before all effects of the action are achieved | | | | X |
| | Goal changes during execution | Plan is no longer valid as the goal has changed | | | | X |

Table 1: Features and the stage of modification they apply to. The stages are: TD (task description), S (search), A (post-planning analysis), and E (execution). An `X` depicts the cases addressed in the paper.

## Conclusion

We systematically record planning characteristics that affect the benefits of integrating external knowledge into the planning process. Our work sheds light on the impact of representation and structure of planning tasks on the value of external knowledge, and on the importance of identifying these features for meaningful evaluation of knowledge integration methods. Table 1 consolidates the characteristics presented in the paper, in order, and specifies the mentioned stages in which they could be applied. The table shows that different characteristics can benefit from different integration approaches, and that the most effective integration stage for tasks depends on all its features, at all levels of abstraction. We thus propose this table as a guide for choosing the most appropriate integration stage for a certain task, based on the combination of its features. Our analysis was done systematically, based on existing approaches and investigation of planning domains and problems. While we aim at a thorough and complete analysis of features that make planning benefit from CSK, it is possible these could be extended in the future. Therefore, we hope that the taxonomy could be used and expanded with further use by the community.

Our findings demonstrate that further work can be done on integrating CSK and semantics in different stages of the planning process, and we hope to facilitate a more mindful choice of tasks for assessing the quality of these works. Future work could focus on improving the evaluation tasks themselves, either through refining existing PDDL domains to better align with the intended semantics of the task, or by curating a suite of environments with diverse properties tailored for the stage of modification for which they are built. Additional research is also needed on knowledge integration methods, including assessing how well current methods generalise across tasks, and developing new techniques that leverage characteristics not yet addressed by existing methods at various stages of modification.

## Acknowledgments

This work has been supported by UK Research and Innovation (EP/S023356/1), in the UKRI Centre for Doctoral Training in Safe and Trusted Artificial Intelligence (www.safeandtrustedai.org).

## References

Al-Moadhen, A.; Qiu, R.; Packianather, M.; Ji, Z.; and Setchi, R. 2013. Integrating Robot Task Planner with Common-sense Knowledge Base to Improve the Efficiency of Planning. *Procedia Computer Science*, 22: 211–220.

Beetz, M.; Beßler, D.; Haidu, A.; Pomarlan, M.; Bozcuoğlu, A. K.; and Bartels, G. 2018. Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents. In *2018 IEEE international conference on robotics and automation (ICRA)*, 512–519. IEEE.

Bezrucav, S.-O.; Canal, G.; Coles, A.; Cashmore, M.; and Corves, B. 2022. Towards automatic state recovery for replanning. In *ICAPS 2022 Workshop on Integrated Planning, Acting, and Execution (IntEx)*.

Bian, N.; Han, X.; Sun, L.; Lin, H.; Lu, Y.; He, B.; Jiang, S.; and Dong, B. 2024. Chatgpt is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 3098–3110.

Bouguerra, A.; Karlsson, L.; and Saffiotti, A. 2008. Monitoring the execution of robot plans using semantic knowledge. *Robotics and autonomous systems*, 56(11): 942–954.

Conti, C. J.; Varde, A. S.; and Wang, W. 2020. Robot action planning by commonsense knowledge in human-robot collaborative tasks. In *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 1–7. IEEE.

Fišer, D.; and Komenda, A. 2018. Fact-alternating mutex groups for classical planning. *Journal of Artificial Intelligence Research*, 61: 475–521.

Fox, M.; and Long, D. 2003. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research*, 20: 61–124.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning: theory and practice*. Elsevier.

Ghallab, M.; Nau, D.; and Traverso, P. 2016. *Automated planning and acting*. Cambridge University Press.

Helmert, M. 2001. On the complexity of planning in transportation domains. *Cesta, & Borrajo (Cesta & Borrajo, 2001)*, 349–360.

Hoffmann, J.; Edelkamp, S.; Thiébaux, S.; Englert, R.; Liporace, F.; and Trüg, S. 2006. Engineering benchmarks for planning: the domains used in the deterministic part of IPC-4. *Journal of Artificial Intelligence Research*, 26: 453–541.

Howey, R.; Long, D.; and Fox, M. 2004. VAL: Automatic plan validation, continuous effects and mixed initiative planning using PDDL. In *16th IEEE International Conference on Tools with Artificial Intelligence*, 294–301. IEEE.

Ilievski, F.; Oltramari, A.; Ma, K.; Zhang, B.; McGuinness, D. L.; and Szekely, P. 2021. Dimensions of commonsense knowledge. *Knowledge-Based Systems*, 229: 107347.

Ilievski, F.; Szekely, P.; and Zhang, B. 2021. CSKG: The CommonSense Knowledge Graph. In Verborgh, R.; Hose, K.; Paulheim, H.; Champin, P.-A.; Maleshkova, M.; Corcho, O.; Ristoski, P.; and Alam, M., eds., *The Semantic Web*, Lecture Notes in Computer Science, 680–696. Springer International Publishing. ISBN 978-3-030-77385-4.

Kaiser, P.; Lewis, M.; Petrick, R. P.; Asfour, T.; and Steedman, M. 2014. Extracting common sense knowledge from text for robot planning. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 3749–3756. IEEE.

Koehler, J.; and Schuster, K. 2000. Elevator Control as a Planning Problem. In *AIPS*, 331–338.

Lenat, D. B. 1995. CYC: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11): 33–38.

Li, X. L.; Kuncoro, A.; Hoffmann, J.; de Masson d'Autume, C.; Blunsom, P.; and Nematzadeh, A. 2022. A systematic investigation of commonsense knowledge in large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11838–11855.

Little, I.; and Thiebaux, S. 2007. Probabilistic planning vs replanning. In *International Conference on Automated Planning and Scheduling (ICAPS 2007)*, 1–10. AAAI Press.

Liu, P.; Orru, Y.; Vakil, J.; Paxton, C.; Shafiullah, N. M. M.; and Pinto, L. 2024. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*.

López, C. L.; Celorrio, S. J.; and Olaya, Á. G. 2015. The deterministic part of the seventh international planning competition. *Artificial Intelligence*, 223: 82–119.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL - The Planning Domain Definition Language.

Meli, D.; Nakawala, H.; and Fiorini, P. 2023. Logic programming for deliberative robotic task planning. *Artificial Intelligence Review*, 56(9): 9011–9049.

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; and Miller, A. 2019. Language models as knowledge bases? In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2463–2473.

Puig, X.; Ra, K.; Boben, M.; Li, J.; Wang, T.; Fidler, S.; and Torralba, A. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8494–8502.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Tenorth, M.; and Beetz, M. 2009. KnowRob—knowledge processing for autonomous personal robots. In *2009 IEEE/RSJ international conference on intelligent robots and systems*, 4261–4266. IEEE.

Winograd, T. 1972. Understanding natural language. *Cognitive psychology*, 3(1): 1–191.

Yu, W.; Lv, J.; Zhuang, W.; Pan, X.; Wen, S.; Bao, J.; and Li, X. 2025. Rescheduling human-robot collaboration tasks under dynamic disassembly scenarios: An MLLM-KG collaboratively enabled approach. *Journal of Manufacturing Systems*, 80: 20–37.

Zhao, Z.; Lee, W. S.; and Hsu, D. 2023. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems*, 36: 31967–31987.