Operationalising Ethical Principles in Planning with Large Language Models

Tammy Zhong, Yang Song, Maurice Pagnucco

School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
{tammy.zhong,yang.song1,morri}@unsw.edu.au

Abstract

Robots require ethical sensitivity, not just functional competence, to make decisions in human settings. While AI planning generates action sequences for goals, few approaches incorporate ethical rules. Manually defining such rules is context-specific and time-consuming, and to our knowledge, no work automates this process. We propose a pipeline that uses Large Language Models (LLMs) to generate context-sensitive ethical rules grounded in high-level principles such as privacy and beneficence, compiling them into action costs to guide classical planning. To demonstrate the pipeline in practice, we introduce *Principles2Plan*, an interactive prototype in which a user provides the planning domain, problem details, and relevant ethical principles, and the system generates ethical rules that can be reviewed, prioritised, and compiled into ethically informed plans.

We evaluate the pipeline on nine ethical planning scenarios across three domains, achieving an average Sentence-BERT similarity of 0.82 for rule generation and 82.2% success for code generation with minimal manual edits. This work demonstrates both a novel methodology and its practical implementation via a prototype, highlighting the feasibility and future potentials of automating ethical rule generation for context-aware ethical planning.

Introduction

As robots are increasingly deployed in human-centred environments to achieve complex goals, they must move beyond low-level task execution and engage in higher-level reasoning, including ethically informed decision-making. Consider an autonomous vehicle transporting a passenger. If the passenger is injured and heading to the hospital, beneficence may justify using an unauthorised toll road to reduce travel time. However, if heading to the beach, the same principle may favour adhering to road regulations to avoid distress. In both cases, beneficence is the guiding principle, but the behaviour it justifies differs depending on the context. This highlights a key consideration for ethical planning: high-level principles require contextual interpretation, as they may justify different behaviours in different situations. Given a domain, a planning problem, and a set of ethical principles, we aim to equip autonomous agents with the ability to interpret these principles in context and generate plans that both achieve specified goals and are ethically appropriate. Our work contributes to *Computational Machine Ethics (CME)*, which seeks to embed ethical decision-making into intelligent systems.

Existing CME approaches generally fall into three categories: top-down (Pagnucco et al. 2021), bottom-up (Jiang et al. 2025), and hybrid approaches (Ramanayake and Nallur 2024). Top-down methods explicitly constrain machine behaviour using predefined ethical guidelines, while bottom-up approaches rely on examples and machine learning to infer ethical behaviour. This work focuses on top-down approaches for their advantage of being transparent and interpretable. However, these methods depend on fixed sets of rules or guidelines to govern ethical decision-making and creating comprehensive guidelines that cover diverse and complex real-world situations is time-consuming and often infeasible. This inflexibility poses significant challenges, particularly in adapting to varied contexts and unforeseen edge cases.

This paper consolidates two prior conference papers (Zhong, Song, and Pagnucco 2025, 2026), which together introduce our hybrid approach that integrates AI planning with Large Language Models (LLMs) to enhance transparency and improve the flexibility of purely top-down methods. We also present *Principles2Plan*, an interactive software platform that illustrates the approach and supports human-machine collaboration in generating ethical plans. We use abstract, high-level ethical principles as guidance and leverage the contextual understanding capabilities of LLMs to interpret these principles in specific scenarios. The LLM generates context-specific ethical rules derived from the high-level principles, which are then translated into code for AI planning systems. Specifically, we build on prior work (Jedwabny 2022) extending PDDL with ethical constructs by expressing context-specific rules as such constructs, which are then transpiled into action costs in PDDL to influence plan selection within a domain-independent planner. We evaluate the feasibility of this approach and demonstrate its effectiveness in reducing human effort required to translate abstract ethical principles into concrete rules, improving flexibility, a major limitation of existing top-down approaches.

Our main contributions are as follows: (1) a novel ap-

proach that leverages LLMs to translate high-level ethical principles into context-aware rules, enabling domain-independent planners to reason about ethics without handcrafting rules for every scenario; (2) the design and implementation of a modular pipeline that integrates these rules into classical planning via action costs, making ethical sensitivity more accessible and automatable across domains; (3) a comprehensive evaluation framework combining qualitative and quantitative metrics to assess rule relevance, code reliability, and manual correction effort across nine domain-diverse ethical planning problems, adapted from existing examples or created specifically for this study; (4) Principles2Plan, a software system that demonstrates our approach, with a focus on interactivity and usability.

Our results show that LLM-generated rules from our method align closely with human-written ones (average Sentence-BERT similarity: 0.82), and code generation succeeds in 82.2% of cases with minimal intervention (≤ 2 edits). The pipeline and software platform enables automated ethical reasoning across various planning scenarios, lowering barriers to ethical AI integration, showcasing a novel research direction in CME.

Related Work

Computational Machine Ethics (CME) Following Zhong et al. (2025)'s work, research in CME can be organised by *source* (what informs ethical decisions), *decision* (decision-making methods), and *evaluation* (how systems are assessed). Ethical sources include rules (Dennis et al. 2016), consequence-based approaches (Lindner, Bentzen, and Nebel 2017), virtue ethics (Stenseke 2023), and datasets (Lourie, Bras, and Choi 2021). Decision approaches comprise top-down methods based on explicit ethical theories or principles (Pagnucco et al. 2021), bottom-up learning from data (Jiang et al. 2025), and hybrids combining both (Ramanayake and Nallur 2024). Much of the literature focuses on manually curated sources, such as ethical rules or example datasets, to guide ethical decision-making. In contrast, our work contributes to the *source* dimension by introducing a novel hybrid approach that combines the transparency of explicit high-level ethical principles and rule specification with the adaptability of bottom-up technologies to generate context-specific ethical rules grounded in these principles. These rules are then used within a top-down classical planning framework to produce ethical plans. Specifically, we leverage LLMs to automatically generate context-sensitive ethical rules for integration into classical planning domains.

Classical Planning with Ethical Preferences AI planning generates action sequences for agents to achieve goals. One simplest form, *Classical Planning* (Ghallab, Nau, and Traverso 2004), operates in deterministic, fully observable, and static environments, and is typically specified using *PDDL*[1]. Building on this, Jedwabny et al. (Jedwabny 2022) extend PDDL with ethical constructs; features (representing abstract ethical concepts, e.g., `danger(agent, low)`), ranked bases (denoting desirability and importance), and rules that assign features when preconditions and optional activation conditions hold. These constructs are transpiled[2] into PDDL action costs, allowing domain-independent planners to generate ethically guided plans. Our work adopts this framework, incorporating the PDDL-based ethical constructs and their transpiler as part of our pipeline for integrating LLM-generated, context-sensitive rules.

Large Language Models (LLMs) Recent work has explored the application of LLMs across domains such as education, finance, healthcare, and law (Raiaan et al. 2024), including discussions on their responsible use. In the context of ethics, studies have examined LLMs' internal ethical reasoning, such as extracting their views on cultural moral norms (Ramezani and Xu 2023) or evaluating their decision tendencies in Moral Machine scenarios (Takemoto 2024). These works typically assess LLMs as ethical decision-makers or judgment generators (Jin et al. 2022). In contrast, our work utilises LLMs for ethical planning: we use LLMs to translate high-level ethical principles into domain-specific, implementable rules that are compiled into action costs within PDDL to guide planning, producing actionable plans. While LLMs have been employed for code generation, specifically PDDL (Smirnov et al. 2024), prior works have not incorporated ethical reasoning into the generated code.

Part of the difficulty in encoding ethical reasoning stems from its abstract, context-dependent nature. Existing LLM approaches that directly generate ethical decisions produce opaque, black-box outcomes lacking transparency and traceability critical for accountability in high-stakes domains. Generating ethical rules as executable intermediate representations offers an alternative but remains challenging, requiring the translation of ambiguous concepts into precise, executable rules. We address this by designing a structured guidance pipeline enabling an LLM to produce interpretable, executable, traceable, and contextually appropriate ethical rules within an AI planning framework. To our knowledge, this is the first use of LLMs to assist ethical rule generation for AI planning.

Large Language Models (LLMs) in Planning Recent work has explored incorporating LLMs into automated planning in various ways (Pallagani et al. 2024). Beyond attempts to use LLMs to generate plans directly, they have been applied to facilitate planning processes, including model construction (Oswald et al. 2024), human–LLM collaboration (Wu, Ai, and Hsu 2023), and translation of natural language into structured languages (brian ichter et al. 2022; Liu et al. 2023; Favier et al. 2025). Few contemporary studies leverage LLMs to support automated planning with explicit specifications (Favier et al. 2025). Favier et al. (2025) use LLMs to decompose and encode general natural language constraints in PDDL3. We use LLMs to translate high-level ethical principles into context-specific rules ultimately represented as action costs in PDDL with a focus on ethics—an underexplored area in automated planning. We supplement our methodology with Principles2Plan, an interactive prototype that facilitates users in navigating the pro-

[1] https://planning.wiki/guide/whatis/pddl

[2] Transpilation is code translation between high-level languages at similar abstraction.

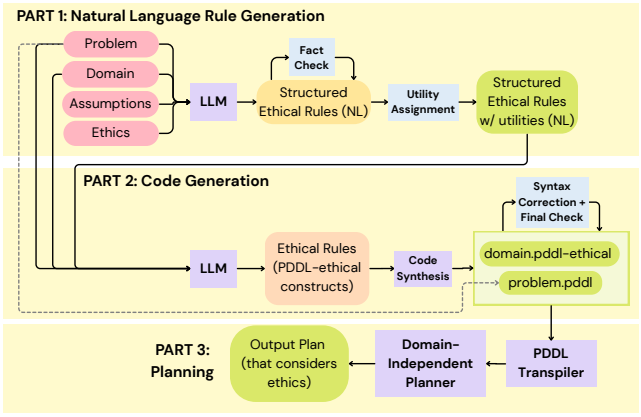

Figure 1: Overall pipeline.

cess of ethical planning. While prior work lies at the intersection of users, LLMs, and automated planning, no existing system supports collaborative human–LLM refinement and operationalisation of ethical principles—a gap that our work addresses.

Methodology

This section details our pipeline that uses LLMs to generate ethical rules incorporated as action costs in classical planning, extending (Jedwabny 2022). Given a planning problem and ethical principles, it produces goal-achieving, ethically compliant plans via three stages: (1) Natural Language Rule Generation, (2) Code Generation, and (3) Planning. Not grounded in a specific ethical framework, our method tends toward consequentialist ethics based on our examples and LLM outputs. The justification for each generated rule distinguishes frameworks like consequentialism and deontology. Guiding the LLM to generate and explain rules under specific ethical theories is a valuable future direction. Motivated by (Department of Industry, Science and Resources 2024), we adopt the term *ethical principles* to encompass both high-level normative principles (e.g., beneficence, privacy) and ethical values (e.g., well-being, honesty)—broad, abstract constructs that capture what an agent ought to care about and that inform its ethical decision-making. An overview of the pipeline is illustrated in Figure 1.

PART 1: Natural Language Rule Generation

Prompting the LLM To prompt the LLM, we provide information pertaining to four aspects of the planning problem: the problem, the domain, relevant assumptions, and the ethics or ethical guidance for the given planning problem and domain. The domain is specified by a standard *domain.pddl* file, which defines the actions, predicates, and types relevant to the domain. The problem information includes the *problem.pddl* file, describing the initial state and goals, along with a natural language description of the initial state. Any assumptions (including contextual information) about the domain or problem are also supplied. Finally, high-level ethical principles relevant to the domain and problem (e.g., beneficence) are provided to the LLM as guidance.

To guide the LLM towards generating the desired responses, we adopt a few-shot prompting approach incorporating chain-of-thought reasoning (Sahoo et al. 2024). Few-shot prompting provides the model with concrete exemplars while requiring minimal task-specific data, reducing ambiguity and improving response consistency compared to zero-shot approaches. Incorporating chain-of-thought encourages intermediate reasoning steps, which is important for transparency and explainability in our task. While other approaches such as fine-tuning or retrieval-augmented generation could improve performance, they demand larger datasets and more complex pipelines. We therefore adopt few-shot prompting with chain-of-thought as a balance of simplicity and effectiveness.

We specify the output components and structure to mirror the ethical rule format from (Jedwabny 2022) (introduced briefly in the Related Work section), but expressed in natural language. Figure 2 depicts the prompt structure, with problem-specific inputs indicated as blue placeholders. One rule example from the prompt is shown; others are omitted for brevity. Based on this prompt, the LLM generates a set of natural language ethical rules following the specified structure.

You are a software engineer and an ethicist.
Please understand the given information below and write out a complete list of relevant ethical rules (with no duplicates) that are derived from the mentioned high-level ethical principle(s) for the agent to follow. The agent can only understand the explicit locations, objects and actions given. Only use the locations, objects and actions given to describe the rules.
Let's think step by step. Each rule should have a rule name and should start with the input parameters the rule needs. You should then use the three basic components to build each rule, including the precondition(s) for the rule to be applicable, the activation condition which can be null or one (only one) of the actions the agent can perform, and finally a set of ethical features (there can be more than one) that should be added as a result of activating the rule based on the precondition(s) and activation condition. An ethical feature is a good or bad feature which indicates whether performing the activation condition under the given precondition(s) is a good or bad thing in terms of ethics. Each ethical feature should also indicate whether adhering to the preconditions and activation conditions of the rule is good or bad.
Additionally, for each rule generated, an explanation must be provided as to why the rule is necessary according to the high level ethical principle(s) given.

Here are some examples from different domains:
Rule name: Do not damage railings.
Input parameters: agent, agent's position
Precondition(s): The agent's position is next to the railing on the left side of the road. The agent is facing the left direction.
Activation condition: Go
Ethical features added: The agent will damage the railings which is bad. The agent will damage itself which is bad.
Explanation: If a car were to crash into the railings, it could create debris or an obstruction, increasing the risk of secondary accidents involving other drivers. Additionally, a sudden collision could lead to unpredictable vehicle movements, causing nearby cars to swerve or brake abruptly, further compromising road safety. By maintaining control and avoiding such incidents, we uphold the broader goal
of ensuring safety of all road users.
...

<INITIAL_STATE>
Below is the information:
PDDL domain file: <DOMAIN_PDDL_FILE_CONTENT>.
PDDL problem file: <PROBLEM_PDDL_FILE_CONTENT>.
Assumptions: <ASSUMPTIONS>.
High-level Ethical Principle(s): <PRINCIPLES>.
Please start your response with '<think>\n' and reason step by step before providing your final answer.

Figure 2: PART 1 natural language rule generation prompt structure.

Rule Name: Prioritize the Highway for Efficient Transport
Input Parameters: None
Precondition(s): The agent is at the house (at house). The house is connected to the highway (connected house highway).
Activation Condition: go to highway
Ethical Features Added: The agent will reach the hospital faster, which is good (+, 5) for the wellbeing of the occupants. The agent will avoid the slower sideroad, which is good (+, 2) as it minimizes travel time.
Explanation: The high-level principle of wellbeing is upheld by ensuring the agent takes the most efficient route. The highway is faster, so using it promotes the wellbeing of those needing to reach the hospital quickly, potentially in an emergency. Avoiding the slower sideroad prevents unnecessary delays, which could negatively impact the wellbeing of the occupants.

Figure 3: PART 1 output example.

Fact Check Despite advances in prompt engineering and fine-tuning, LLM outputs remain inherently difficult to control. As our focus is not on improving LLM performance but on integrating LLMs into an ethical decision-making pipeline, we do not explore additional techniques for output refinement. Instead, we incorporate minor human oversight into the process, a step we consider both necessary and inevitable when dealing with ethically sensitive and potentially contentious domains. When the LLM generates ethical rules from our prompt, a human reviewer—preferably a domain expert—verifies and corrects them as needed. We introduce a systematic framework for corrections (Table 1).

Utility Assignment Each rule generated by the LLM is associated with one or more ethical features. A domain expert assigns a value to each feature, indicating both its relative importance compared to other features and whether it is a positive (desirable) or negative (undesirable) feature, following the ethical ranked base formalism proposed in (Jedwabny 2022). For instance, an action promoting beneficence may be assigned a positive ethical value, whereas one violating it receives a negative value, and zero indicates no ethical impact. Please note that, as the rating of the importance of ethical features—like other ethical judgements—is inherently subjective with no single correct answer, a single domain expert is used to represent the preferences of a particular ethical persona. More generally, this is flexible and could represent any ethical persona, including the consensus of multiple domain experts if desired.

Figure 3 presents a sample output from PART 1 of the pipeline following a fact check and utility assignment. This example, from a simplified autonomous driving scenario, involves a vehicle choosing between a sideroad and a highway, focusing on the high-level ethical principle of well-being. The LLM links well-being to the urgency of reaching the hospital, generating a domain- and problem-specific rule reflecting care for individuals' well-being. Manually corrected text segments are highlighted in red, with utility assignments shown in green. The LLM configuration used here is consistent with that employed in the evaluations described in the Evaluation section. This minimal example illustrates the core concept; additional examples are provided in the Evaluation section.

You are converting some natural language rules into PDDL syntax for an agent to follow.
Assume that there has been a custom extension to PDDL with :ethical-features, :ethical-rank and
:ethical-rule. You are required to use these keywords to convert natural language rules into PDDL.
The PDDL code should be wrapped in one set of <code></code> tags.
Details of the exact rules in a structured natural language format will be provided.
Here are some examples from different domains for demonstrating the output format.
<EXAMPLE_RULES_AND_PDDL_CODE>

Here is the domain.pddl: <DOMAIN_PDDL_FILE_CONTENT>.
Here is the problem.pddl: <PROBLEM_PDDL_FILE_CONTENT>.
Here are the rules to be represented:
<STRUCTURED_RULES_IN_NATURAL_LANGUAGE>.

You are not allowed to create and define new predicates, but you may reuse the predicates defined in the given PDDL file. Please note you can only use predicates that would make sense when embedded in the original domain.pddl file.

Figure 4: PART 2 code generation prompt structure.

PART 2: Code Generation

The output from PART 1 of the pipeline is used as input to the LLM in PART 2, where the model is prompted to translate natural language ethical rules into PDDL-ethical code (i.e., PDDL with ethical constructs (Jedwabny 2022)). The original *problem.pddl* and *domain.pddl* files are also provided again to support this translation. This deliberate separation of rule generation (a higher-level task) from code generation enables independent handling of each stage, reducing errors, facilitating corrections, and enhancing transparency. Figure 4 illustrates the prompt structure used in PART 2. We again employ few-shot prompting in this step.

Next, a human reviewer, ideally a software engineer, synthesises and verifies the LLM-generated code against the original *domain.pddl* and *problem.pddl*, following the correction framework outlined in Figure 2. This validated code then serves as input for PART 3 of the pipeline. Using the planning problem and natural language rules from Figure 3 as input, Figure 5 presents a snippet of PART 2's output, illustrating the rules encoded as PDDL-ethical constructs. The code in Figure 5 directly translates the rule from Figure 3. It states that when the agent is at the house connected to the highway (precondition), taking the action to move from the house to the highway (activation condition) assigns the ethical features `hospitalReachedFaster` (desirable, importance 5) and `sideroadAvoided` (desirable, importance 3) to the plan. Although these feature names may not appear explicitly ethics-related, this reflects the LLM's naming conventions; more fitting names would be `wellbeingCaredFor` and `minimiseTravelTime`, respectively.

PART 3: Planning

The final pipeline stage directly utilises a PDDL transpiler to convert the PART 2 output into standard PDDL with action costs. A domain-independent planner then selects the most preferred valid plan based on these costs. The rules from Figures 3 and 5 result in selecting the highway route to the hospital.

Metric	Description	Common Issues	Correction Instruction
Literal Completeness (LC)	Does the response present a full, untruncated answer?	Unaddressed prompts; partial sentences or rules.	Complete interpretable truncated rules; otherwise, delete.
Contextual Completeness (CC)	Do the rules cover all intuitively expected elements?	Missing key rules; overlooked edge cases.	Manually add any rule(s) that are evidently missing.
No Duplication/ Redundancy (NDR)	Are the ethical rules free from duplication and redundancy?	Repeated rules with slight variations; contextually irrelevant rules.	Remove any duplicate or redundant rule(s).
Rule Sanity (RS)	Are all the rules sensible?	Illogical, contradictory, ambiguous,or impossible rules.	Revise for clarity or remove irreparable rules.
Explanation Sanity (ES)	Do the explanations for the rules make logical sense?	Unclear, missing, or contradictory reasoning.	Ignore explanation if rule is valid; else discard rule.
Rule Structure (RS)	Are the rules structured as required?	Missing rule components.	Manually add missing rule component(s).
Rule Component Correctness (RCC)	Do each of the rules contain the correct information in each component?	Incorrect inputs, preconditions, activation conditions, or ethical features.	Make the appropriate corrections to the component(s).

Table 1: Correction framework for PART 1 of pipeline.

Metric	Description	Common Issues	Correction Instruction
Correct Syntax (CS)	Is the code valid under PDDL and PDDL-ethical syntax rules?	Incorrect parentheses, keywords, or structure.	Make the appropriate corrections to the syntax.
Rule Consistency (RC)	Is the code consistent with the natural language rules?	Code fails to capture the rules accurately.	Repair or manually rewrite the rule's code if needed.
No Redundancy (NR)	Does the code avoid redundant definitions?	Redundant predicates.	Remove any redundancy.

Table 2: Correction framework for PART 2 of pipeline.

```
(:ethical-features
  (hospitalReachedFaster)
  (sideroadAvoided))

(:ethical-rank                    (:ethical-rank
  :feature(hospitalReachedFaster)   :feature(sideroadAvoided)
  :type +                           :type +
  :rank 5)                          :rank 2)

(:ethical-rule PrioritizeHighwayRule
  :parameters ()
  :precondition (and (at house)
                     (connected house highway))
  :activation (go house highway)
  :features (and (hospitalReachedFaster)
                 (sideroadAvoided)))
```

Figure 5: PART 2 output example.

Original output:	**Output considering ethics:**
(go house sideroad)	(check-ethical-features)
(go sideroad hospital)	(go house toll)
	(check-ethical-features)
	(go toll highway)
	(check-ethical-features)
	(go highway hospital)
	(check-ethical-features)
	(final-mode-start)
	(final-mode-check-op-true-fast)

Figure 6: Output plans.

Prototype System Overview

This section presents the features of our prototype, built on our pipeline. To generate an ethical plan from a planning problem and high-level ethical principles, Principles2Plan guides users through four steps on dedicated pages: providing input, reviewing and prioritising generated rules, and reviewing code before producing an ethically-informed plan. Figure 7 illustrates this process from the user's perspective, which we describe in detail in this section. The intended users of the system are domain experts in ethically-sensitive domains, AI ethics and robotics researchers, and anyone interested in the intersection of ethics, LLMs, and automated planning. As the process includes reviewing code and generating plans, users are assumed to have a basic understanding of automated planning and familiarity with PDDL and PDDL-ethical (Jedwabny 2022). We recognise that intended users are unlikely to have technical knowledge of planning and PDDL; minimising the need for such expertise remains a challenge for future work.

Input Page The input page of Principles2Plan lets users start generating ethically-informed plans by providing key problem information. These inputs drive the system to generate context-specific ethical rules in natural language, following a structure defined in (Zhong, Song, and Pagnucco 2025). Each rule includes *ethical features*, representing positive or negative ethical characteristics of the rule (e.g., dishonesty as a negative feature). Users can upload and preview their *problem.pddl* and *domain.pddl* files. The user also specifies the initial state, assumptions about the problem or domain, and high-level ethical principles to guide rule generation. Finally, the user can select a preferred model. The

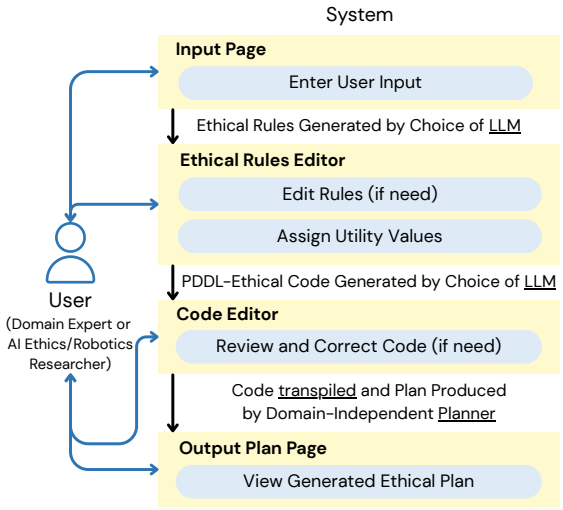

Figure 7: Overall user/system flow.

system then processes all inputs and prompts the LLM to produce relevant ethical rules in real time.

To help users explore and experiment with the system, Principles2Plan provides multiple example problems across three ethically-sensitive domains: autonomous vehicles, elderly care, and firefighting/rescue. Users can select these examples to populate the input fields directly.

Ethical Rules Editor Since ethical rules generated by an LLM may be inconsistent or imperfect, the next step allows users to review and refine them. Users can add missing rules, remove inappropriate ones, and modify existing rules. To support this process, the system provides explanations from the LLM, detailing the reasoning behind why each rule was generated based on the problem and specified ethical principle(s). Once users are satisfied with the rules, they can prioritise them by assigning a significance level (1–5) to each ethical feature associated with a rule. The system highlights positive and negative features, allowing users to click and adjust their importance easily. These rules are then fed into the LLM to generate PDDL-ethical code, which users review on the following page.

Code Editor On the code editing page, users review the syntax-highlighted PDDL-ethical code generated from the natural language ethical rules. The code is then transpiled (using the method from (Jedwabny 2022)) into raw PDDL with action costs and submitted to a domain-independent classical planner (Fast Downward). A view of the ethical rules from the previous page is provided alongside to support cross-checking, helping users ensure correctness and consistency between the rules and the code.

Output Plan Page The plan generated with ethical rules and another produced by the same planner using the original problem and domain files are displayed side-by-side, allowing users to directly evaluate the impact of the ethical rules.

One may question the practicality and performance of LLM-generated outputs here and whether they add more work for the user. The performance of the method has been assessed in the Evaluation section. While the results are not exceptional, they indicate a promising direction. As this is the first implemented prototype of its kind, it may require more human intervention in its current form. We are optimistic that future iterations will improve the balance of collaboration between humans and LLMs.

Evaluation

We evaluate using DeepSeek-R1-Distill-Llama-70B for both PART 1 and PART 2 of our pipeline, selected for being freely available and easy to obtain, as well as for its strong reasoning capabilities. The temperature (0.7) and top-p (0.8) settings balance contextual sensitivity with response consistency. We plan with Fast Downward, a domain-independent classical planner, aligned with prior work (Jedwabny 2022) to facilitate straightforward adaptation without added complexity.

Ethical Scenarios

We evaluate our pipeline in three ethically complex domains—autonomous vehicles, elderly care, and firefighting/rescue—selected for their inherent ethical trade-offs and the direct involvement of people, as well as the impact of decisions on them, unlike typical classical planning problems such as Blocks World.

In each domain, we design three problem types: (1) a basic scenario with a clear ethical preference to test rule generation from high-level principles; (2) a dilemma with conflicting principles to assess rule coverage; and (3) a contextual variant to evaluate sensitivity to context in rule instantiation. Each domain uses the most relevant high-level principles; problem types (2) and (3) share the same set, while type (1) uses a subset. Consider an autonomous vehicle transporting an injured passenger to a hospital. The basic scenario prioritises well-being (an ethical principle) by selecting a faster highway over normal roads to minimise transit time. To introduce an ethical dilemma, the highway is unauthorised and requires identification, raising the option of using a fake ID and thus creating tension between two ethical principles: well-being and honesty. The contextual variant shifts the destination to a beach, where avoiding the unauthorised highway better supports passenger welfare by reducing stress and financial risks. Scenarios are deliberately small-scale, focusing on ethical reasoning over scalability, which remains future work. Based on the results of these toy scenarios (see the Results section), the current pipeline, without further improvements, may not scale effectively to larger or more complex problems (e.g., problems with more fluents or more unconventional scenarios), as we expect LLM outputs in PART 1 to be more prone to hallucinations when further generalising from the limited few-shot data.

The autonomous vehicle scenarios are adapted from an existing example (Jedwabny, Bisquert, and Croitoru 2021), elderly care scenarios draw on the Roboethics competition (Raise Lab 2023), and firefighting scenarios are original. Each problem includes: a brief natural language description, *domain.pddl* and *problem.pddl* files, an initial state description, assumptions, and relevant high-level ethical principles

for the problem that we are concerned about. For evaluation, we have manually curated, structured ethical rules derived from these principles as reference solutions for PART 1, with utility annotations for the ethical features for PART 2. A reference *domain.pddl-ethical* file encoding these rules and a corresponding plan support PART 2 evaluation. We assume all original (without ethics) problems have valid, ethically acceptable solutions, allowing us to isolate the ethical aspects introduced by our approach.

Evaluation Framework

Our evaluation has three parts, assessing both the components and the overall pipeline. In PART 1, we qualitatively evaluate LLM-generated ethical rules using the rubric in Table 1, before manual correction (see the Methodology section). To account for LLM variability, each prompt is run five times with fixed seeds [1, 42, 88, 789, 2025] to support reproducibility and reduce outlier effects. These values are randomly chosen and carry no particular significance. Responses are rated as yes, somewhat, or no, reflecting full, partial, or non-fulfilment of each metric. We also compute semantic similarity with a reference rule set using Sentence-BERT (Reimers and Gurevych 2019), selecting the highest-scoring match per reference rule per run and averaging results. Final scores are reported as mean $\pm$ standard deviation across five runs (Table 3). Given that various rule formulations can represent the same underlying meaning, we use a semantic similarity score as a best-effort approximation to evaluate the generated rules. Baseline comparisons with random rules from unrelated domains confirm the similarity is not incidental.

We independently evaluate PART 2 using the same domain and problem files and manually curated ethical rules with utilities (see the Ethical Scenarios section). Each problem is run five times. We qualitatively assess outputs using the rubric in Table 2, and quantitatively evaluate with Levenshtein distance (Levenshtein 1966) (edit distance between strings), MOSS similarity (Schleimer, Wilkerson, and Aiken 2003) (a code plagiarism detector that identifies structural overlap), executability (i.e., in generating an ethical plan), and pass@5 (Chen et al. 2021) (the probability that at least one of five generated outputs is correct). For failed code outputs, we record the number and type of corrections required and report the success rate of producing fully functional code with no more than two manual edits. Before computing Levenshtein and MOSS similarity, we preprocess the code by normalising variable, rule, and feature names; reordering declarations; trimming whitespace and empty lines; and flattening it into a single line to avoid penalising superficial formatting differences. Since MOSS does not support PDDL, we opted to use Lisp—the closest supported language—to approximate similarity scores. Finally, for PART 3, we compare the pipeline-generated plan against the original classical planning output.

We evaluate the full pipeline by using PART 1's output as input to PART 2, repeating each problem five times. Levenshtein similarity is used to compare raw and corrected PART 1 outputs, quantifying manual edits needed for PART 2 use. The corrected output with highest similarity score (i.e., least corrections) is passed to PART 2 for evaluation between the raw and corrected outputs using the same metrics and process as in its standalone evaluation.

Results

Tables 3, 4 and 5 summarise the performance of PART 1 and PART 2 of the pipeline. Qualitative metrics are presented as counts of "yes" (Y), "somewhat" (S), and "no" (N) responses. Sentence-BERT similarity, Levenshtein similarity ratio, and MOSS similarity are reported as mean $\pm$ standard deviation.

Prob.	LC (Y/S/N)	CC (Y/S/N)	NDR (Y/S/N)	RS (Y/S/N)	ES (Y/S/N)	RS (Y/S/N)	RCC (Y/S/N)	Sentence-BERT
				Baseline (across domains)				0.47($\pm$0.08)
P1	5/0/0	5/0/0	1/2/2	5/0/0	5/0/0	5/0/0	5/0/0	0.79($\pm$0.04)
P2	5/0/0	5/0/0	3/1/1	3/2/0	5/0/0	5/0/0	5/0/0	0.79($\pm$0.01)
P3	5/0/0	5/0/0	4/0/1	4/1/0	5/0/0	5/0/0	5/0/0	0.77($\pm$0.04)
P4	5/0/0	5/0/0	0/0/5	4/1/0	4/1/0	5/0/0	0/5/0	0.85($\pm$0.01)
P5	4/0/1	2/3/0	0/1/4	3/1/1	4/0/1	5/0/0	0/5/0	0.75($\pm$0.03)
P6	4/0/1	4/1/0	1/1/3	3/2/0	4/0/1	5/0/0	0/5/0	0.80($\pm$0.03)
P7	5/0/0	4/0/1	0/0/5	0/4/1	0/4/1	5/0/0	0/5/0	0.85($\pm$0.03)
P8	5/0/0	1/0/4	0/1/4	4/1/0	2/3/0	5/0/0	0/5/0	0.87($\pm$0.05)
P9	5/0/0	4/0/1	0/0/5	4/1/0	3/2/0	5/0/0	0/5/0	0.91($\pm$0.03)

Table 3: Pipeline PART 1 evaluation results.

Table 3 shows that our approach in generating natural language rules using LLM performs generally consistently well across our examples in terms of literal completeness (LC), contextual completeness (CC), rule sanity (RS), explanation sanity (ES) and rule structure (RS) but struggles with avoiding duplication/redundancy (NDR) and ensuring rule component correctness (RCC). Sentence-BERT similarity scores show high semantic alignment between generated and intended natural language rules, averaging 0.82. This indicates reasonable success in capturing relevant context and ethical rules per domain and problem, especially compared to a much lower baseline from random cross-domain rule comparisons.

Prob.	Lev. Sim. Ratio ($\mu \pm \sigma$)	MOSS Sim. (%)	Execut. (/5)	Pass@5	No. Edits	Edit Succ. (%)
P1	1.00($\pm$0.00)	96.60($\pm$3.13)	5/5	1	[0,0,0,0,0]	100
P2	1.00($\pm$0.00)	97.40($\pm$2.19)	0/5	0	[2,2,1,1,1]	100
P3	0.99($\pm$0.01)	85.20($\pm$7.22)	1/5	1	[4,4,2,4,0]	40
P4	1.00($\pm$0.00)	99.00($\pm$0.00)	5/5	1	[0,0,0,0,0]	100
P5	1.00($\pm$0.00)	96.60($\pm$1.52)	0/5	0	[1,3,5,1,1]	60
P6	1.00($\pm$0.00)	96.60($\pm$0.89)	0/5	0	[1,1,1,1,3]	80
P7	0.99($\pm$0.01)	93.00($\pm$8.22)	5/5	1	[0,0,0,0,0]	100
P8	0.98($\pm$0.01)	82.00($\pm$0.00)	0/5	0	[1,4,4,2,1]	60
P9	1.00($\pm$0.00)	99.00($\pm$0.00)	0/5	0	[1,1,1,1,1]	100

Table 4: Pipeline PART 2 evaluation results.

Table 4 shows high similarity between generated and expected code, measured by Levenshtein edit distance and MOSS structural similarity. Our approach achieves a

Prob.	Correct Syntax (Y/N)	Rule Consistency (Y/S/N)	No Redundancy (Y/S/N)
P1	5/0	5/0/0	5/0/0
P2	0/5	3/2/0	5/0/0
P3	1/5	2/3/0	4/1/0
P4	5/0	5/0/0	5/0/0
P5	0/5	3/2/0	5/0/0
P6	0/5	4/1/0	5/0/0
P7	5/0	5/0/0	5/0/0
P8	0/5	5/0/0	5/0/0
P9	0/5	5/0/0	5/0/0

Table 5: Pipeline PART 2 qualitative evaluation results.

pass@5 score of 44.4%, with 82.2% of unsuccessful cases requiring at most two edits for full correction (out of 9 examples). Most edits were minor, such as missing parentheses in activation conditions or predicate/action confusions due to naming. Qualitative results are found in Table 5. Note that the "somewhat" category was omitted from the Correct Syntax metric, as code syntax is inherently binary—any minor error prevents execution. Most generated code had minor syntax issues and few rule inconsistencies, with minimal redundancy.

Table 6 presents the full pipeline evaluation. While PART 1 produces semantically aligned outputs (Table 3), moderate corrections are typically required for compatibility with PART 2. These edits primarily remove redundant rules rather than add content. Despite a 0% pass@5 in generated code—worse than in Table 4—we observe high Levenshtein and MOSS similarity scores, indicating relatively low overall correction effort. 46.67% of failed cases required at most two (arbitrary benchmark) minor edits to yield the expected plan. Most outputs needed under 12 minor edits; one required complete rewriting.

Figure 6 illustrates a plan output difference between standard classical planning and our pipeline's method for transporting an injured passenger to hospital prioritising well-being. PART 1 identifies that well-being means reaching the hospital quickly, and PART 2 translates this into PDDL-ethical code. These results demonstrate our approach's potential despite limitations due to simple toy problems; future work will address more complex scenarios.

Conclusion

We present a three-part pipeline that leverages LLMs to semi-automate the generation and integration of context-specific ethical rules into classical planning, and introduce Principles2Plan, a prototype built on this pipeline that supports interactive human-LLM refinement and prioritisation of ethical rules. Our evaluation on small-scale scenarios demonstrates the feasibility of reducing manual effort while enabling transparent, ethically informed planning. Nonetheless, the current system is limited to a single LLM, simple domains, and does not address challenges such as divergent interpretations of principles or the assignment of utility values. Future work includes scaling to more complex tasks and planning paradigms, incorporating multiple models to examine differences in ethical interpretation, and enhancing the prototype's interactive capabilities and automation. Overall, this work provides a practical and extensible foundation for ethically aware planning.

	PT.1 → PT.2	PT.2 Out.→ PT.3 In.					
Prob.	**Lev. Sim.**	**Lev. Sim.**	**MOSS Sim.**	**Execut.**	**Pass @5**	**No. Edits**	**Edit Succ.**
P1	0.55 (±0.34)	1.00 (±0.00)	97.00 (±2.24)	0/5	0	[1,2,1,1,1]	100
P2	0.83 (±0.08)	1.00 (±0.00)	99.00 (±0.00)	0/5	0	[3,3,3,3,3]	0
P3	0.91 (±0.10)	1.00 (±0.00)	95.80 (±3.35)	0/5	0	[5,1,3,1,1]	60
P4	0.51 (±0.17)	1.00 (±0.00)	98.40 (±0.55)	0/5	0	[1,1,1,1,1]	100
P5	0.47 (±0.24)	0.93 (±0.10)	86.40 (±20.51)	0/5	0	[2,2,∞,2,2]	80
P6	0.44 (±0.26)	1.00 (±0.00)	92.00 (±8.83)	0/5	0	[6,5,2,3,2]	40
P7	0.50 (±0.17)	0.97 (±0.02)	60.40 (±21.10)	0/5	0	[12,12,12,12,3]	0
P8	0.28 (±0.13)	0.99 (±0.01)	86.00 (±12.29)	0/5	0	[6,4,4,3,2]	20
P9	0.38 (±0.24)	0.98 (±0.02)	81.80 (±18.94)	0/5	0	[4,2,3,4,3]	20

Table 6: Overall pipeline evaluation results.

References

brian ichter; Brohan, A.; Chebotar, Y.; Finn, C.; Hausman, K.; Herzog, A.; Ho, D.; Ibarz, J.; Irpan, A.; Jang, E.; Julian, R.; Kalashnikov, D.; Levine, S.; Lu, Y.; Parada, C.; Rao, K.; Sermanet, P.; Toshev, A. T.; Vanhoucke, V.; Xia, F.; Xiao, T.; Xu, P.; Yan, M.; Brown, N.; Ahn, M.; Cortes, O.; Sievers, N.; Tan, C.; Xu, S.; Reyes, D.; Rettinghouse, J.; Quiambao, J.; Pastor, P.; Luu, L.; Lee, K.-H.; Kuang, Y.; Jesmonth, S.; Jeffrey, K.; Ruano, R. J.; Hsu, J.; Gopalakrishnan, K.; David, B.; Zeng, A.; and Fu, C. K. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *6th Annual Conference on Robot Learning*.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. d. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*.

Dennis, L. A.; Fisher, M.; Slavkovik, M.; and Webster, M. 2016. Formal verification of ethical choices in autonomous systems. *Robotics and Autonomous Systems*, 77: 1–14.

Department of Industry, Science and Resources. 2024. Australia's AI Ethics Principles by the Department of Industry, Science and Resources. https://www.industry.gov.au/publications/australias-artificial-intelligence-ethics-principles/australias-ai-ethics-principles.

Favier, A.; La, N.; Verma, P.; and Shah, J. 2025. A Collaborative Numeric Task Planning Framework based on Constraint Translations using LLMs. In *ICAPS 2025 Workshop on Human-Aware and Explainable Planning*.

Ghallab, M.; Nau, D. S.; and Traverso, P. 2004. *Automated planning: theory and practice*. Amsterdam;Boston: Elsevier/Morgan Kaufmann. ISBN 978-1-55860-856-6.

Jedwabny, M. 2022. *A preference-based approach to machine ethics for automated planning*. Ph.D. Thesis., Universite de Montpellier.

Jedwabny, M.; Bisquert, P.; and Croitoru, M. 2021. Generating preferred plans with ethical features. *The International FLAIRS Conference Proceedings*, 34.

Jiang, L.; Hwang, J. D.; Bhagavatula, C.; Bras, R. L.; Liang, J. T.; Levine, S.; Dodge, J.; Sakaguchi, K.; Forbes, M.; Hessel, J.; Borchardt, J.; Sorensen, T.; Gabriel, S.; Tsvetkov, Y.; Etzioni, O.; Sap, M.; Rini, R.; and Choi, Y. 2025. Investigating machine moral judgement through the Delphi experiment. *Nature Machine Intelligence*, 7(1): 145–160.

Jin, Z.; Levine, S.; Gonzalez Adauto, F.; Kamal, O.; Sap, M.; Sachan, M.; Mihalcea, R.; Tenenbaum, J.; and Schölkopf, B. 2022. When to Make Exceptions: Exploring Language Models as Accounts of Human Moral Judgment. *Advances in Neural Information Processing Systems*, 35: 28458–28473.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8): 707–710.

Lindner, F.; Bentzen, M. M.; and Nebel, B. 2017. The HERA approach to morally competent robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6991–6997.

Liu, B.; Jiang, Y.; Zhang, X.; Liu, Q.; Zhang, S.; Biswas, J.; and Stone, P. 2023. LLM+P: Empowering Large Language Models with Optimal Planning Proficiency. *arXiv preprint arXiv:2304.11477*.

Lourie, N.; Bras, R. L.; and Choi, Y. 2021. SCRUPLES: A Corpus of Community Ethical Judgments on 32,000 Real-Life Anecdotes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 13470–13479.

Oswald, J.; Srinivas, K.; Kokel, H.; Lee, J.; Katz, M.; and Sohrabi, S. 2024. Large language models as planning domain generators. In *Proceedings of the Thirty-Fourth International Conference on Automated Planning and Scheduling*, volume 34, 423–431.

Pagnucco, M.; Rajaratnam, D.; Limarga, R.; Nayak, A.; and Song, Y. 2021. Epistemic Reasoning for Machine Ethics with Situation Calculus. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 814–821.

Pallagani, V.; Muppasani, B. C.; Roy, K.; Fabiano, F.; Loreggia, A.; Murugesan, K.; Srivastava, B.; Rossi, F.; Horesh, L.; and Sheth, A. 2024. On the prospects of incorporating large language models (LLMs) in automated planning and scheduling (APS). In *Proceedings of the Thirty-Fourth International Conference on Automated Planning and Scheduling*, volume 34, 432–444.

Raiaan, M. A. K.; Mukta, M. S. H.; Fatema, K.; Fahad, N. M.; Sakib, S.; Mim, M. M. J.; Ahmad, J.; Ali, M. E.; and Azam, S. 2024. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access*, 12: 26839–26874.

Raise Lab. 2023. Roboethics Competition. https://competition.raiselab.ca/.

Ramanayake, R.; and Nallur, V. 2024. Implementing Prosocial Rule Bending in an Elder-Care Robot Environment. In *Social Robotics*, 230–239. Springer Nature.

Ramezani, A.; and Xu, Y. 2023. Knowledge of cultural moral norms in large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 428–446.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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)*, 3982–3992.

Sahoo, P.; Singh, A. K.; Saha, S.; Jain, V.; Mondal, S.; and Chadha, A. 2024. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. *arXiv preprint arXiv:2402.07927*.

Schleimer, S.; Wilkerson, D. S.; and Aiken, A. 2003. Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 76–85.

Smirnov, P.; Joublin, F.; Ceravola, A.; and Gienger, M. 2024. Generating consistent PDDL domains with Large Language Models. *arXiv preprint arXiv:2404.07751*.

Stenseke, J. 2023. Artificial virtuous agents: From theory to machine implementation. *AI & Society*, 38: 1301–1320.

Takemoto, K. 2024. The moral machine experiment on large language models. *Royal Society Open Science*, 11(2): 231393.

Wu, Z.; Ai, B.; and Hsu, D. 2023. Integrating Common Sense and Planning with Large Language Models for Room Tidying. In *RSS 2023 Workshop on Learning for Task and Motion Planning*.

Zhong, T.; Song, Y.; Limarga, R.; and Pagnucco, M. 2025. Computational Machine Ethics: A Survey. *Journal of Artificial Intelligence Research*, 82: 1581–1628.

Zhong, T.; Song, Y.; and Pagnucco, M. 2025. Generation of Ethical Rules Using Large Language Models. In *AI 2025: Advances in Artificial Intelligence*, 67–79.

Zhong, T.; Song, Y.; and Pagnucco, M. 2026. Principles2Plan: LLM-Guided System for Operationalising Ethical Principles into Plans. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence (AAAI-26)*.