The Dual Role of Abstracting over the Irrelevant in Symbolic Explanations: Cognitive Effort vs. Understanding

Zeynep G. Saribatur (zeynep.saribatur@tuwien.ac.at)[1], **Johannes Langer**[2] **& Ute Schmid**[2]
[1]Institute of Logic and Computation, TU Wien
[2]Cognitive Systems, University of Bamberg

Abstract

Explanations are central to human cognition, yet AI systems often produce outputs that are difficult to understand. While symbolic AI offers a transparent foundation for interpretability, raw logical traces often impose a high extraneous cognitive load. We investigate how formal abstractions, specifically removal and clustering, impact human reasoning performance and cognitive effort. Utilizing Answer Set Programming (ASP) as a formal framework, we define a notion of irrelevant details to be abstracted over to obtain simplified explanations. Our cognitive experiments, in which participants classified stimuli across domains with explanations derived from an answer set program, show that clustering details significantly improve participants' understanding, while removal of details significantly reduce cognitive effort, supporting the hypothesis that abstraction enhances human-centered symbolic explanations.

Keywords: symbolic AI ; explanations; abstraction ; understanding ; cognitive effort

Introduction

Symbolic AI refers to the methods in AI that are based on explicitly describing the knowledge of the world and the problem through logical or related formal languages, and finding possible solutions through search and reasoning. Unlike subsymbolic systems (largely neural networks), symbolic and rule-based representations are inherently more transparent, offering a strong foundation for explainable AI (XAI). Despite the dominance of deep learning, the need for symbolic methods has never been more critical. Purely statistical models, such as Large Language Models (LLMs), frequently suffer from black-box opacity, making their internal decision-making processes impossible to audit. In high-stakes environments, such as healthcare, autonomous aviation, and law, this lack of transparency is a prohibitive barrier to trust. Furthermore, statistical systems often fail at multi-step logical reasoning and are prone to hallucinations, where they generate plausible-sounding but factually incorrect information. Symbolic AI provides the necessary guidance by encoding explicit rules, ensuring that system behavior remains within defined ethical, safety, and logical boundaries.

This synergy is central to the emerging direction of Neuro-symbolic AI which aims to bridge Kahneman's System 1 (fast, intuitive thinking) and System 2 (slow, logical deliberation) (Kahneman, 2011). Neuro-symbolic systems use neural layers for perception—such as interpreting visual data or natural language—and symbolic layers for high-level reasoning and decision-making (Mao et al., 2019), which is shown to have very successful applications while also obtaining explanations (Hitzler & Sarker, 2021).

One challenge in adopting a symbolic view on understandability and explainability is that providing humans with rules and axioms may not be enough to achieve a clear understanding of how a system reached a decision. Initial steps towards providing explanations have been by showing the trace of rules that were applied in computing a decision (Clancey, 1983). However, if the decision-making rules are too large and contain many distracting details, then it is challenging for humans to understand the decision process. In the cognitive science of explanation, it is well-established that humans prefer simple, contrastive explanations over exhaustive traces (Lombrozo, 2007; Miller, 2019). Just as image classification explanations use saliency maps to highlight relevant pixels while treating the rest as irrelevant (Ribeiro et al., 2016), symbolic representations must distinguish between essential logical pivots and distracting details adhering with Grice's Maxim of Quantity (Grice, 1975).

The role of abstraction is thus central to making complex systems interpretable, with established theories spanning both symbolic and subsymbolic domains. In symbolic reasoning, abstraction often involves forgetting or projecting away non-essential details to distill the essence of an explanation (Chakraborti et al., 2020; Siebers & Schmid, 2019), as well as simplifying solution spaces to foster better human-AI alignment (Ho et al., 2022; Poesia Reis e Silva & Goodman, 2022). These structural refinements are frequently achieved through predicate invention to streamline rule-based representations (Fürnkranz et al., 2020; Muggleton et al., 2018; Schmid et al., 2016) or domain clustering to group related logical entities into higher-level concepts (Eiter et al., 2019). By reducing the granularity of the data, these methods aim to get a symbolic output that is not just technically sound, but also cognitively manageable for a human observer. Beyond purely symbolic systems, abstraction serves as a bridge for interpreting neural architectures, utilizing causal abstractions (Geiger et al., 2021) or distilling opaque weights into readable decision trees (Confalonieri et al., 2021).

In this work, we focus on Answer Set Programming (ASP) (Brewka et al., 2011), a prominent logic-based formalism in symbolic AI known for its declarative expressivity and efficient solvers. ASP is a highly popular language widely used not only in AI but in Computer Science to solve a variety of

In G. Boleda, I. Dautriche, M. Maldonado, C. Saldana and T. Torrent (Eds.), *Proceedings of the 48th Annual Conference of the Cognitive Science Society*

problems such as combinatorial optimisation, logical reasoning, planning, bioinformatics, and data integration to name a few (Schaub & Woltran, 2018), and has recently shown great potential in neuro-symbolic reasoning (Barbara et al., 2023; Eiter et al., 2022) and as a reasoning layer for LLMs (Kareem et al., 2024; Rajasekharan et al., 2023). While obtaining explanations for solutions (i.e., answer sets) of an answer set program is a well-studied topic, achieving concise explanations that aid human understanding remains a challenge (Fandinno & Schulz, 2019). Recent theoretical work in ASP has explored *removal* (Saribatur & Woltran, 2023) and *clustering* (Saribatur et al., 2024) of irrelevant details, via preserving the correspondence of answer sets for any potential extensions of the program, but these formalisms are too restricted to bring to applications and, more importantly, lack empirical validation regarding their impact on human cognition.

Our contributions are as follows: We propose a notion of abstraction in ASP based on irrelevancy within a problem space, by relaxing the previous restricted notions. We then use these abstractions to obtain explanations on the decision-making. We empirically evaluate how removal and clustering of irrelevant details help in human understanding and cognitive effort. Our results demonstrate a double benefit: clustering details significantly improves performance (accuracy), while removal of irrelevant details significantly reduces cognitive effort (answer time).

The rest of the paper is organized as follows. We begin with a high-level background on ASP and explanations in ASP. We then present the notion of abstracting over details that are irrelevant w.r.t. a set of potential problem instances and illustrate its use in obtaining simplified explanations. Then we describe our empirical study, and present the results. We then conclude with a discussion on future work.

Background

Answer Set Programming (ASP) ASP is a popular declarative modeling and problem solving framework in artificial intelligence, and generally in computer science, with roots in non-monotonic logic (Brewka et al., 2011).

The problem specifications are described by a set of rules using propositional atoms, meaning that an atom a can either be true or false, and the solutions to the problem instances get represented by the models (i.e., answer sets) of the program. The roots of ASP come from formalisms that aimed at representing and reasoning about commonsense knowledge and beliefs of agents, by relying on the ability to represent non-monotonic reasoning, which is the ability to withdraw previous conclusions about the world when receiving new information. Readers interested in the use of the ASP paradigm in modeling human reasoning principles can see (Dietz et al., 2022). Here, we focus on the use of ASP for logical reasoning in AI with real-world applications (Erdem et al., 2016).

Let us present a toy example that shows the expressiveness of ASP in modeling problems, through the "guess and check" approach. The idea is to use choice rules to guess for solutions

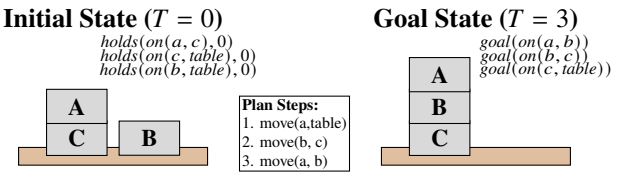

Figure 1: Blocksworld example

to the problem, which are then pruned by the constraints of the problem that a correct solution should not satisfy.

Example 1. Consider the blocksworld planning problem, where from a given initial state, the aim is to find a sequence of actions, i.e., plan, that arranges the blocks so that the desired goal state is reached.

We begin by guessing potential actions for each time step. The choice rule

$$1\{occurs(A,T) : action(A)\}1 \leftarrow step(T)$$

guesses for each step T the occurrence of exactly one action A. The effect of taking an action is described with the rule

$$holds(on(B,L),T) \leftarrow occurs(move(B,L),T)$$

which states that if moving block B to location L occurs at time T, it holds that block B is on location L.

In order to reason over a sequence of time steps, we specify what properties carry over time.[1] The following rule states that if F holds at time step T and in the next time step $T{+}1$ it is not the case that F was made false, i.e., we do not have evidence of a change, then we can infer that F holds at $T{+}1$.

$$holds(F,T{+}1) \leftarrow holds(F,T), not\ \neg holds(F,T{+}1), step(T).$$

The executability conditions for actions are defined using constraints, which are rules of the form

$$\bot \leftarrow occurs(move(B,C),T), unclear(B,T)$$

that forbids the movement of block B at time step T if it is not clear.[2] Here $unclear(B,T)$ is an auxiliary predicate defined with the rule $unclear(C,T) \leftarrow holds(on(B),C,T)$ stating that a block is unclear if it has another block on top. In order to enforce that the goal condition is reached in the computed plan, constraints are used, e.g.,

$$\bot \leftarrow goal(on(B,L)), not\ holds(on(B,L),T)), maxTime(T).$$

Once we have an ASP description of the problem, we can use an ASP system (e.g., Clingo (Gebser et al., 2019)) to solve a given problem instance described through a set of facts. In case of a planning problem, these facts describe the initial state and the desired goal state. The answer sets of the program then contain the solutions to the problem.

[1] An action usually changes a few things but leaves most of the world state untouched. Representing this efficiently without explicitly listing everything that does not change is called the Frame Problem (McCarthy & Hayes, 1969).

[2] In ASP, constraints, which are rules with $\bot$ (falsum) in the head denoting a contradiction, are important as they eliminate non-valid solutions. If the guessing rule generates a plan where block a is moved while block b is on top of it, this constraint removes that entire sequence from the list of possible solutions.

Example 2 (Ex. 1 cont.)**.** For initial and goal states shown in Figure 1, the answer set contains $occurs(move(a, table), 1)$, $occurs(move(b, c), 2)$, $occurs(move(a, b), 3)$ which describes the solution to the planning problem.

Explanations in ASP The main focus on explanations in ASP is explaining the reasoning behind an obtained answer set. In particular, given an answer set I of a program P, a user might ask questions about the presence or absence of certain atoms in I. Interested readers are refered to Fandinno and Schulz (2019) for an overview of the explanation approaches. The main idea is to provide a justification of the reached solution via tracing the relevant rules/atoms and showing them through a graph or a tree structure. There are several tools that can be used for this purpose, including a recent one `xclingo` (Cabalar & Muñiz, 2024). For example, for Example 1 depending on how the problem is encoded (Cabalar & Muñiz, 2023), the explanation for the atom $h(on(a), b, 3)$ to hold true in the answer set is a trace with the actions $o(move(a, b), 3)$ $o(move(a, table), 1)$ and the changes of locations of block a.

However, as the complexity of the domain rises, these explanations suffer from information density. Consider a scenario where only colored blocks can be moved. If all blocks in every possible initial state are colored, the *colored* attribute would be part of the logical trace while providing no discriminative detail. To a human observer, these redundant details are distracting and increases cognitive load. This motivates the need for formal notions of irrelevancy to be removed or clustered over, which we define in the next section.

Abstracting the Irrelevant in Problems

To systematically reduce the complexity of explanations, we must first define which parts of a logical program are "safe" to simplify. We focus on abstracting over details that are irrelevant for decision-making within specific problem contexts. For this, we relax previous notions of abstractions (Saribatur & Woltran, 2023; Saribatur et al., 2024), by focusing on the consistency of answer sets under a set of potential scenarios.

Definition 1 (χ-irrelevance)**.** *Given a program P over $\mathcal{U}$ and a set $\chi = \{I_1, \ldots, I_n\}$ with $I_i \subseteq \mathcal{U}$, the sets $A_1, \ldots, A_n \subseteq \mathcal{U}$ of atoms are* χ-irrelevant *if for a (surjective) mapping $m : \mathcal{U} \to \mathcal{U}' \cup \top$ with $|m(A_i)| = 1, 1 \leq i \leq n$, for any $F \in \chi$*

$$m(AS(P \cup F)) = AS(m(P) \cup m(F)). \tag{1}$$

Here, $m(P)$ is the (χ-)abstracted program.

The aim is to define atoms that are irrelevant in P w.r.t. the provided problem instances, via the existence of an (abstracted) program that can be obtained with applying a mapping to P. Here, the mapping m performs two primary cognitive operations: (i) Removal, which is mapping atoms to $\top$ (logical truth), represents *forgetting* a detail that lacks discriminative power, and (ii) Clustering: Mapping multiple distinct atoms to a single representative atom, represents *chunking* and reducing the granularity of the domain.

```
|__scent
| |__spiky            |__scent
| |__headLargerLeaf   | |__headLargerLeaf
| |__needsWater       | |__needsWater
| | |__habitatWater   | | |__habitatWaterOrMud
```

(a) Default (b) Abstracted (cluster&removal)

Figure 2: Explanations obtained by `xclingo`

To illustrate how these notions relate to obtaining explanations, consider a toy classification task.

Example 3. Let us consider the program P

$$needsWater \leftarrow habitatWater$$
$$needsWater \leftarrow habitatMud$$
$$scent \leftarrow spiky, needsWater, headLargerLeaf$$

which states if the flower has a larger head than its leaves, if if lives in a habitat which shows it needs water, and it if has spiky leaves then we can classify it as having a scent.

The set $\chi = \{\{spiky, habitatWater, headLargerLeaf\}, \{spiky, habitatMud, headSmallerLeaf\}, \{spiky, habitatSand, headLargerLeaf\}\}$ presents three flowers. All these flowers have spiky leaves, but they vary in their habitat, and head size vs. leaf size.

Observation on *removal*: Since *spiky* are true for all the flowers, it is not decisive in classification, and can be removed ($m(spiky) = \top$). Observation on *clustering*: Since both, *habitatWater* and *habitatMud* (but not *habitatSand*), reach to the same outcome *needsWater*, they can be clustered into a single abstract concept $m(habitatWater) = m(habitatMud) = habitatWaterOrMud$ Applying the mapping m yields the program $m(P)$ below.

$$needsWater \leftarrow habitatWaterOrMud$$
$$scent \leftarrow needsWater, headLargerLeaf$$

It can be observed that $m(P)$ satisfies (1) for any $F \in \chi$.

We are interested in making use of this notion to generate more concise justifications. The idea is to obtain justifications from the abstracted program, i.e., $m(P)$. Using the `xclingo` system (Cabalar & Muñiz, 2024), we can visualize the difference between a default justification and an abstracted one.

Example 4 (Ex. 3 ctd)**.** Let us look at the scenario $\chi_1 = \{spiky, habitatWater, headLargerLeaf\} \in \chi$. As seen in Figures 2a and 2b, the default explanation for the atom *scent* to occur in the answer set of $P \cup \chi_1$ includes the *spiky* leaf and the specific habitat. The abstracted explanation for *scent* appearing in the answer set of $m(P) \cup m(\chi_1)$, however, prunes the redundant detail and generalizes the habitat.

Now, with our formal notions at hand, we can ask the central question of our empirical study: Do these theoretically simpler traces actually result in measurable improvements in human understanding and reductions in cognitive effort?

Empirical Study

We conduct an empirical study to observe whether abstract explanations help in human understandability, cognitive effort and confidence. The idea is to present the participants with explanations of a reasoning task, and then evaluate their performance for determining the outcome when encountered with new instances. For measuring understanding and effort, we look at accuracy and answer times. Subjective confidence is assessed via participant self-reports collected after each classification task.

We formulate our research questions as follows:

Q1. Do explanations abstracted over irrelevant details improve participant accuracy when performing transfer tasks on new instances?

As with abstraction by removal and by clustering, we obtain simplified explanations with only the relevant details that reaches the outcome, we hypothesize that these explanations allow participants to better understand the underlying logic of the reasoning trace, leading to fewer errors when applying that logic to novel scenarios.

Q2. Do explanations with irrelevant details removed reduce the answer time when deciding on the new instances?

The aim of the removal abstraction is to keep the decisive details. We expect that participants will spend less time parsing redundant information, leading to more efficient decision-making and faster answer times.

Q3. Do explanations abstracted over irrelevant details increase human confidence in their decision making?

If an explanation is concise and highlights only the relevant path to an outcome, we expect the participants feel more confident in their understanding, correlating with their objective improvements in performance (Q1 and Q2).

Method

Task The experiment utilized concept-learning tasks in which participants performed a binary classification task after being presented with explanations (e.g., Figure 3). Explanations in the context of symbolic AI approaches rely on human-understandable concepts and relations between them (Poeta et al., 2023). As such, they are closely related to concept learning (Lake et al., 2015). In the context of rule-based classification, concept-based explanations support humans in understanding the relevant aspects of instances to make them belong to a specific class or category. This has been shown, for instance, in the context of inductive logic programming (Muggleton et al., 2018; Rabold et al., 2022).

Domains. For the classification task, we designed three different domains, each of which being of different biological specimen: flowers, mushrooms, and cacti. Each domain has six domain-specific decision attributes and one binary target label. We used domain-specific neutral target terms, such as flower having a *scent* in contrast to it being *poisonous*, with no implied consequence to the decision. We decided on having tasks from similar but different domains in order to prevent proactive interference (Underwood, 1957), where learning one rule hinders the ability to learn the next. We designed 8 instances for each domain: 2 positive instances to be used for the learning phase, and 3 positive/3 negative instances for the test phase.

Cactus 1

Cactus Attributes:

Spines	Shape	Arms	Stem	Flower	Height	Slow Growth
few	flattened-padded	many	thick	yes	short	yes

Explanation:

This cactus has slow growth because all of the following attributes apply:

- adaptive because
  - **many** arms
  - **short** height
- no leaf-like shape, since **flattened-padded** shape
- **few** spines
- **has** flowers
- **thick** stem

Figure 3: Example of a learning phase instance as presented in the study. The table lists all attributes and values. The provided explanation is the default without any abstraction.

Classification rules. Each classification task is based on an answer set program representing the conditions needed to reach the target class. As a representative we show below the program for the cactus domain which determines when a cactus has *slow-growth*.[3]

$$adaptive(X) \leftarrow arms(X, C), height(X, short).$$
$$slowGrowth(X) \leftarrow adaptive(X), spines(X, C), flower(X, yes),$$
$$- shape(X, leaflike), stem(X, thick).$$

Figure 3 shows an example explanation for a cactus to be classified as having *slow-growth*. The program for the flower domain extends that from Example 3, and the mushroom domain is represented similarly to define the conditions for a mushroom to be *tolerant*. We calculated for each answer set program of the three domains the χ-irrelevant atoms where χ is the set of all positive/negative instances of the respective domain, for removal and for clustering. We use the abstracted answer set programs to obtain the abstracted explanations.

Study Design and Procedure The empirical online study is based on a complete between-subject design, where participants were randomly assigned to one of the four groups: `default`, `cluster`, `removal`, and `cluster and removal`, and the data between these groups were compared to calculate the effect on our dependent variables.[4] The experiment consists of a learning and a test phase (see Figure 4).

Learning Phase. At the beginning of each domain, participants receive 2 positive examples together with an explanation,

[3] Here $-shape(X, leaflike)$ is true whenever $shape(X, cylindrical)$ or $shape(X, padded)$ is true. We use the notation for *strong negation* ($-$), since `xclingo` does not provide explanations of default negation (*not*), due to its underlying theory.

[4] Informed consent about data protection, anonymity and the right to leave the study at any time was given. At the end of the study, participants had the opportunity to leave further comments and notes.

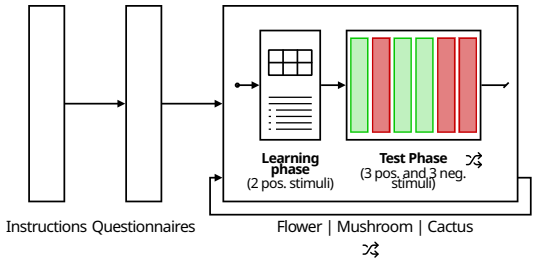

Figure 4: Study Procedure Overview. The order of the three domains and the test phase, i.e. the order of the positive (illustrative in green) and negative (resp. in red) stimuli were random to balance out possible position effects.

a textual form of the output from `xclingo`, on why the instance was classified as such (see Figure 3). These explanations differ depending on the assigned group of the experiment. The `default` group receives a full justification trace including all technical attributes leading to the classification (see Figure 3). The `cluster` and `removal` groups receive abstracted explanations based on clustering (e.g. 'water or mud') or removal of χ-irrelevant atoms, respectively. The `cluster_removal` group sees abstracted explanations resulting from the application of both strategies (e.g., Figure 2b).

Test Phase. After the learning phase, unseen instances (3 positive, 3 negative) are presented without explanations. Participants need to decide *stimulus valence* that is whether the instance is classified as positive or not and give a rating about their confidence with a slider bar (range 0-100), or choose the option "I don't know". The attribute values were evenly distributed across all instances to avoid accumulations of a single attribute value.

Subjective Assessments. After the last domain, participants gave ratings on the perceived usefulness of the explanations (5 items, based on a subset of Bahel et al., 2024), their memorization efforts (1 item) and their familiarity with the domain terminologies (1 item). All these items were rated on a Likert-type scale ranging from 1 ('strongly disagree') to 7 ('strongly agree'). Prior knowledge in computer science, logic, mathematics and programming as well as the three domains (7 items) is rated on a scale from 0 ('none') to 4 ('expert').

Participants We recruited participants via the online platform *Prolific*. In order to ensure that performance metrics reflected genuine reasoning rather than fatigue or random guessing, we implemented comprehension checks and attention checks. We informed the participants that the survey can only be taken once, in order to avoid familiarity with the previously learned rules. The survey screened-out those that restarted the survey after failing a check. Among the 157 participants that have started to survey, 103 have completed the survey and the questionnaire. Furthermore, we had to exclude from the analysis, 3 participants due to restarting the survey, which missed the online checks during the survey, and 2 participants, due to low answering times (more than three standard deviations from the mean of the assigned group). The final sample size comprised of 99 participants (age mean = 35.9, sd = 14.0; 64 female, 32 male, 1 other). Participants were randomly assigned to one of four conditions: Default (n = 19), Cluster (n = 19), Removal (n = 15), or Cluster-Removal (n = 44). While final group sizes were unequal due to unbalanced group sizes resulting from the real-time random assignment process, Levene's test confirmed that the assumption of homogeneity of variance was maintained ($p > .05$) for two of the dependent variables (Accuracy, Confidence). For Answer Time, where the assumption was violated, we utilized the robust Welch ANOVA and Games-Howell post-hoc tests to ensure statistical validity. There is no significant difference in the distribution of gender, age, or self-reported knowledge in computer science among the abstraction groups. Participants were paid for their participation.

Results

Analysis We report our statistical tests regarding our previously formulated research questions. We tested effects for accuracy, answer time and confidence. No effects of the domain could be observed on our dependent variables. Therefore we report results aggregated over all domains (see Figure 5).

Q1 Accuracy: A one-factor ANOVA considering the different types of abstraction shows a significant effect of abstraction on mean accuracy ($F = 2.72, p = 0.049$). Posthoc pairwise comparison tests show significant effect for `cluster` vs. `default` ($T = 2.75, p = 0.009$) and for `cluster_removal` vs. `default` ($T = 2.52, p = 0.016$).

Q2 Answer Time: A one-factor Welch ANOVA considering the different types of abstraction shows a significant effect of abstraction on mean answer time ($F = 6.41, p = 0.001$). Posthoc tests show significant effect for `cluster` vs. `cluster_removal` ($T = 3.25, p = 0.017$) and for `cluster_removal` vs. `default` ($T = -3.23, p = 0.018$).

Q3 Confidence: A one-factor ANOVA shows no significant effect of abstraction over mean reported confidence ($F = 1.58, p = 0.19$). In the postdoc pairwise comparison tests we see marginal improvement for `default` vs. `removal` ($T = -1.71, p = 0.096$) and `cluster` vs. `default` ($T = 1.84, p = 0.072$). Confidence was reported higher by the participants in their correct answers (mean 80 ± 1.56) than in their incorrect answers (mean 69 ± 2.31).

Exploratory Findings The exploratory ANOVAs showed a main effect of stimulus valence ($F = 9.89, p = 0.002$), where accuracy for positive stimuli ($M = 0.82 \pm 0.02$) was significantly higher than for negative stimuli ($M = 0.70 \pm 0.02$). This is inline with the findings that negative instances are more difficult when the concept is conjunctive (Bourne Jr & Guy, 1968). No significant interaction with abstraction was observed, which suggests that these symbolic abstractions provide a consistent cognitive benefit regardless of the stimulus valence. Regarding subjective assessments on the usefulness of the explanations, we observe no difference between the conditions.

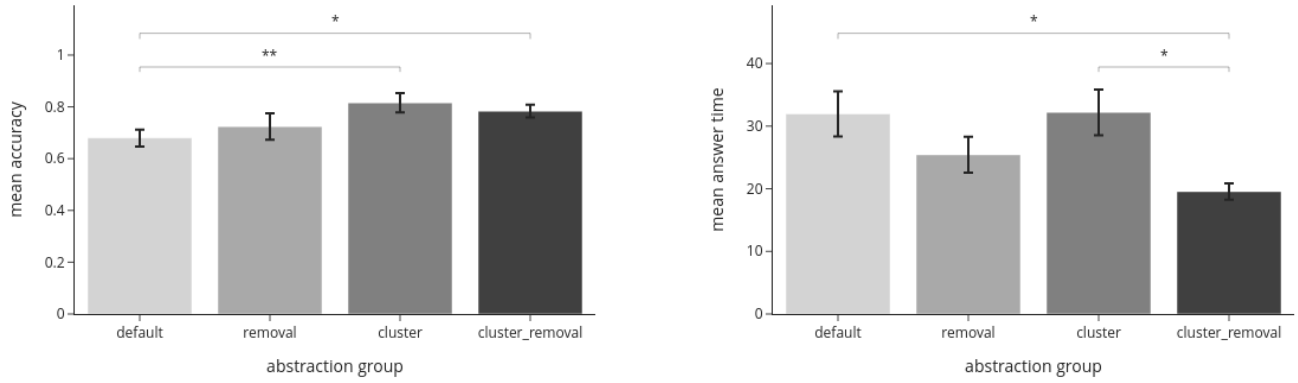

Figure 5: Mean accuracy (left) and mean answering times (right) across the abstraction levels. Error bars represent ±1 SEM. Significant effect observed for pairwise comparison between the groups are marked with ⋆ ($p < 0.05$) and ⋆⋆ ($p < 0.01$)

Discussion

The findings confirm that abstractions help with understanding of the explanations and reduces the cognitive effort, though the type of abstraction influences the nature of the improvement.

The results for Accuracy (Q1) show a significant effect of clustering. When explanations were abstracted into clusters, participants made significantly fewer errors. Interestingly, the way we presented the clusters, by grouping the features without providing a high-level semantic label, closely resembles the lists described by Rissman and Lupyan, 2024. They showed that providing exemplar lists does not necessarily activate the concept (e.g., from *water,mud* to *wet*), though for our setting, these clusters likely provided structural organization that allowed participants to form more clear models of the classification rules, leading to fewer errors.

The lack of a significant effect of removal on accuracy suggests two distinct possibilities: (i) The original explanations may have already been below a complexity threshold for our participants, so that further pruning offered no benefit; (ii) Our formal notion of χ-irrelevance for removal might be capturing *specificity* (Bolognesi et al., 2020), as it removes details that are common across instances, and focuses on the distinctive details. This resembles deciding on whether a cat is Siamese by looking at its eye color, rather than existence of whiskers or paws as they are common to all cats. For the participants, this type of removal might not have been clear in the presented explanations, preventing them to generalize to new cases.

The results for Answer Time (Q2), on the other hand, show significance for removal of the irrelevant details in explanations for participants to reach decisions faster. The finding that the `cluster_removal` group was significantly faster than both `default` and `cluster` groups suggests a synergetic effect. While clustering alone helps in accuracy, it is the removal of irrelevant details that acts as the primary engine for speed. Yet, clustering neither hurts response time, nor does removal of irrelevant details hurt accuracy. From a cognitive perspective, since the formal abstraction already removed the irrelevant details, they were not displayed in the explanations. This then helped in the participants to ignore the undecisive attributes when making the classification decisions.

A notable finding is the lack of a significant effect of abstraction on Confidence (Q3). Although participants in abstracted groups performed better and faster, this did not reflect in their perceived confidence. They had a reliable sense of their own reasoning accuracy, regardless of the explanation type.

Conclusion

In this work, we presented an experimental evaluation on how the formal notions of removal and clustering of irrelevant details in explanations help in human understanding and cognitive effort. For this, we utilized ASP as a formal foundation to establish the notion of irrelevancy within a set of problem instances, by relaxing the notions of dependency-preserving abstractions. Our study provides empirical evidence for the double benefit of symbolic abstraction in explanations: structural organization via clustering facilitates understanding, while the removal of irrelevant details reduces cognitive effort. We observe that there is more than just obtaining *simpler* explanations, as different abstraction operations serve distinct cognitive effects.

Since tasks require to be of a certain intermediate complexity for explanations to be helpful (Ai et al., 2021), we plan to explore how complexity of the default explanation plays a role, for removal abstraction to become significant in accuracy, and for abstraction to significantly improve confidence. Another direction will be to investigate whether χ-irrelevancy indeed captures specificity rather than abstraction.

Our findings highlight the importance of tailoring symbolic explanations to cognitive processes, bridging formal logic-based AI with theories of explanation in cognitive science. Our work positions ASP as a system for studying how symbolic abstraction can foster cognitively aligned AI.

Acknowledgments

This research was funded in whole or in part by the Austrian Science Fund (FWF) grant 10.55776/T1315.

References

Ai, L., Muggleton, S. H., Hocquette, C., Gromowski, M., & Schmid, U. (2021). Beneficial and harmful explanatory machine learning. *Machine Learning*, *110*, 695–721.

Bahel, V., Sriram, H., & Conati, C. (2024). Personalizing explanations of ai-driven hints to users' cognitive abilities: An empirical evaluation. *CoRR*, *abs/2403.04035*. https://doi.org/10.48550/ARXIV.2403.04035

Barbara, V., Guarascio, M., Leone, N., Manco, G., Quarta, A., Ricca, F., & Ritacco, E. (2023). Neuro-symbolic AI for compliance checking of electrical control panels. *TPLP*, *23*(4), 748–764. https://doi.org/10.1017/S1471068423000170

Bolognesi, M., Burgers, C., & Caselli, T. (2020). On abstraction: Decoupling conceptual concreteness and categorical specificity. *Cogn. Process.*, *21*(3), 365–381. https://doi.org/10.1007/S10339-020-00965-9

Bourne Jr, L. E., & Guy, D. E. (1968). Learning conceptual rules: Ii. the role of positive and negative instances. *Journal of Experimental Psychology*, *77*(3p1), 488.

Brewka, G., Eiter, T., & Truszczyński, M. (2011). Answer set programming at a glance. *Commun. ACM*, *54*(12), 92–103.

Cabalar, P., & Muñiz, B. (2023). Commonsense explanations for the blocks world. *Proc. Workshop on Challenges and Adequacy Conditions for Logics in the New Age of Artificial Intelligence*. https://lianda23.uma.es/ACLAI23/resources/ACLAI23_Cabalar_muniz.pdf

Cabalar, P., & Muñiz, B. (2024). Model explanation via support graphs. *Theory and Practice of Logic Programming*, *24*(6), 1109–1122. https://doi.org/10.1017/S1471068424000048

Chakraborti, T., Sreedharan, S., & Kambhampati, S. (2020). The emerging landscape of explainable automated planning & decision making. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 4803–4811. https://doi.org/10.24963/ijcai.2020/669

Clancey, W. J. (1983). The epistemology of a rule-based expert system–a framework for explanation. *Artificial Intelligence*, *20*(3), 215–251.

Confalonieri, R., Weyde, T., Besold, T. R., & del Prado Martín, F. M. (2021). Using ontologies to enhance human understandability of global post-hoc explanations of black-box models. *Artificial Intelligence*, *296*, 103471.

Dietz, E., Fichte, J. K., & Hamiti, F. (2022). A quantitative symbolic approach to individual human reasoning. In J. Culbertson, H. Rabagliati, V. C. Ramenzoni, & A. Perfors (Eds.), *Proceedings of the 44th annual meeting of the cognitive science society, cogsci 2022, toronto, on, canada, july 27-30, 2022*. cognitivesciencesociety.org. https://escholarship.org/uc/item/5rv6124b

Eiter, T., Higuera, N., Oetsch, J., & Pritz, M. (2022). A neuro-symbolic ASP pipeline for visual question answering. *Theory and Practice of Logic Programming*, *22*(5), 739–754.

Eiter, T., Saribatur, Z. G., & Schüller, P. (2019). Abstraction for zooming-in to unsolvability reasons of grid-cell problems. *Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence (XAI@IJCAI)*.

Erdem, E., Gelfond, M., & Leone, N. (2016). Applications of answer set programming. *AI Magazine*, *37*(3), 53–68. http://www.aaai.org/ojs/index.php/aimagazine/article/view/2678

Fandinno, J., & Schulz, C. (2019). Answering the "why" in answer set programming - A survey of explanation approaches. *Theory and Practice of Logic Programming*, *19*(2), 114–203. https://doi.org/10.1017/S1471068418000534

Fürnkranz, J., Kliegr, T., & Paulheim, H. (2020). On cognitive preferences and the plausibility of rule-based models. *ML*, *109*(4), 853–898.

Gebser, M., Kaminski, R., Kaufmann, B., & Schaub, T. (2019). Multi-shot ASP solving with clingo. *Theory Pract. Log. Program.*, *19*(1), 27–82. https://doi.org/10.1017/S1471068418000054

Geiger, A., Lu, H., Icard, T., & Potts, C. (2021). Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, *34*, 9574–9586.

Grice, H. P. (1975). Logic and conversation. *Syntax and semantics*, *3*, 43–58.

Hitzler, P., & Sarker, M. K. (Eds.). (2021). *Neuro-symbolic artificial intelligence: The state of the art* (Vol. 342). IOS Press. https://doi.org/10.3233/FAIA342

Ho, M. K., Abel, D., Correa, C. G., Littman, M. L., Cohen, J. D., & Griffiths, T. L. (2022). People construct simplified mental representations to plan. *Nature*, *606*(7912), 129–136.

Kahneman, D. (2011). *Thinking, fast and slow*. Macmillan.

Kareem, I., Gallagher, K., Borroto, M., Ricca, F., & Russo, A. (2024). Using learning from answer sets for robust question answering with LLM. *Logic Programming and Nonmonotonic Reasoning - 17th International Conference, LPNMR 2024, Dallas, TX, USA, October 11-14, 2024, Proceedings*, *15245*, 112–125. https://doi.org/10.1007/978-3-031-74209-5_9

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, *350*(6266), 1332–1338.

Lombrozo, T. (2007). Simplicity and probability in causal explanation. *Cognitive psychology*, *55*(3), 232–257.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., & Wu, J. (2019). The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

McCarthy, J., & Hayes, P. (1969). *Some philosophical problems from the standpoint of artificial intelligence*. Stanford University USA.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, *267*, 1–38.

Muggleton, S. H., Schmid, U., Zeller, C., Tamaddoni-Nezhad, A., & Besold, T. (2018). Ultra-strong machine learning: Comprehensibility of programs learned with ILP. *Machine Learning*, *107*(7), 1119–1140.

Poesia Reis e Silva, G., & Goodman, N. (2022). Left to the reader: Abstracting solutions in mathematical reasoning. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *44*(44).

Poeta, E., Ciravegna, G., Pastor, E., Cerquitelli, T., & Baralis, E. (2023). Concept-based explainable artificial intelligence: A survey. *ACM Computing Surveys*.

Rabold, J., Siebers, M., & Schmid, U. (2022). Generating contrastive explanations for inductive logic programming based on a near miss approach. *Machine Learning*, *111*(5), 1799–1820.

Rajasekharan, A., Zeng, Y., Padalkar, P., & Gupta, G. (2023). Reliable natural language understanding with large language models and answer set programming. *Proceedings 39th International Conference on Logic Programming, ICLP 2023, Imperial College London, UK, 9th July 2023 - 15th July 2023*, *385*, 274–287. https://doi.org/10.4204/EPTCS.385.27

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you?: Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Rissman, L., & Lupyan, G. (2024). Words do not just label concepts: Activating superordinate categories through labels, lists, and definitions. *Language, cognition and neuroscience*, *39*(5), 657–676.

Saribatur, Z. G., Knorr, M., Gonçalves, R., & Leite, J. (2024). On abstracting over the irrelevant in answer set programming. In P. Marquis, M. Ortiz, & M. Pagnucco (Eds.), *Proceedings of the 21st international conference on principles of knowledge representation and reasoning, KR 2024, hanoi, vietnam. november 2-8, 2024*. https://doi.org/10.24963/KR.2024/61

Saribatur, Z. G., & Woltran, S. (2023). Foundations for Projecting Away the Irrelevant in ASP Programs. *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning*, 614–624. https://doi.org/10.24963/kr.2023/60

Schaub, T., & Woltran, S. (2018). Special issue on answer set programming. *Künstliche Intelligenz*, *32*, 101–103.

Schmid, U., Zeller, C., Besold, T., Tamaddoni-Nezhad, A., & Muggleton, S. (2016). How Does Predicate Invention Affect Human Comprehensibility? *Inductive Logic Programming. ILP 2016*, 52–67. https://doi.org/10.1007/978-3-319-63342-8_5

Siebers, M., & Schmid, U. (2019). Please delete that! why should I? *Künstliche Intelligenz*, *33*(1), 35–44.

Underwood, B. J. (1957). Interference and forgetting. *Psychological Review*, *64*(1), 49.