RoboKGNet: PDDL File Generation Using Knowledge Graphs to Support Neurosymbolic Sense-Plan-Act Loops

Mark Adamik[1], Paolo Forte[2] and Ilaria Tiddi[1]

[1]Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands
[2]Örebro University, 701 82 Örebro, Sweden

Abstract

We propose a neuro-symbolic sense-perceive-act framework that combines state-of-the-art neural components, namely a VLM-based sensing module and an LLM-based planning module, to generate PDDL domain and problem files. The framework is based on RoboKGNet[1], a knowledge graph that integrates multiple linguistic resources and ontologies to serve as an intermediary background knowledge repository between the sensing and planning components. In the demonstration, users will be able to provide natural language instructions that are then executed by a simulated robot in a virtual environment.

Keywords

Neuro-Symbolic AI, Robotics, AI Planning, Sense-Plan-Act, Generative AI

1. Introduction

Robotic agents operating in real-world environments must act under dynamic, uncertain, and only partially observable conditions. For instance, a household robot preparing breakfast may need to retrieve a mug, but the mug may be inside a closed cupboard, moved by a person during execution, or only partially visible behind other objects. This requires systems that can flexibly interpret sensory data, reason over the current state of the environment, and generate executable actions. Recent generative models, such as Large Language Models (LLMs), Vision-Language Models (VLMs), and Vision-Language Agents (VLAs), have emerged as a promising paradigm to accommodate this flexibility, and have consequently gained increasing prominence within robotics applications [1, 2].

However, several limitations remain. These models are typically large, rendering their direct deployment on robotic platforms generally infeasible and computationally slow [3]. Generative models are also vulnerable to hallucinated outputs and adversarial jailbreak techniques, both of which can compromise overall system safety. This concern is especially acute in cyber-physical systems, where erroneous outputs may directly affect the physical environment and human well-being [4].

To mitigate these limitations, neuro-symbolic approaches have gained increasing attention [5], as they combine the flexibility of neural methods with the transparency, explicit structure, and fast control loops typically associated with symbolic systems. In robotics, such hybrid approaches are especially relevant when perceptual information must be mapped to symbolic representations to support reasoning, planning, and execution.

In this paper, we extend our previous work[6, 7] and introduce a neuro-symbolic Sense-Plan-Act (SPA) framework oriented toward the automated creation of Planning Domain Definition Language (PDDL) [8] domain and problem specifications. We employ two symbolic representation paradigms: (i) PDDL specifications, which are widely adopted within the AI planning community and in robotics, and (ii) knowledge graphs, which semantically enrich data structures in the form of background knowledge and semantic grounding for both virtual and embodied agents.

[1]https://github.com/Dorteel/robokgnet

The 25th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2026) - Satellite Events (Workshops, Tutorials, Demos, Posters).

m.adamik@vu.nl (M. Adamik); paolo.forte@oru.se (P. Forte); i.tiddi@vu.nl (I. Tiddi)
https://dorteel.github.io/ (M. Adamik); https://paoloforte95.github.io/ (P. Forte); https://kmitd.github.io/ilaria/ (I. Tiddi)
0000-0002-7977-3617 (M. Adamik); 0000-0003-4784-5089 (P. Forte); 0000-0001-7116-9338 (I. Tiddi)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-26

CEUR-WS.org/Vol-4340/paper3.pdf

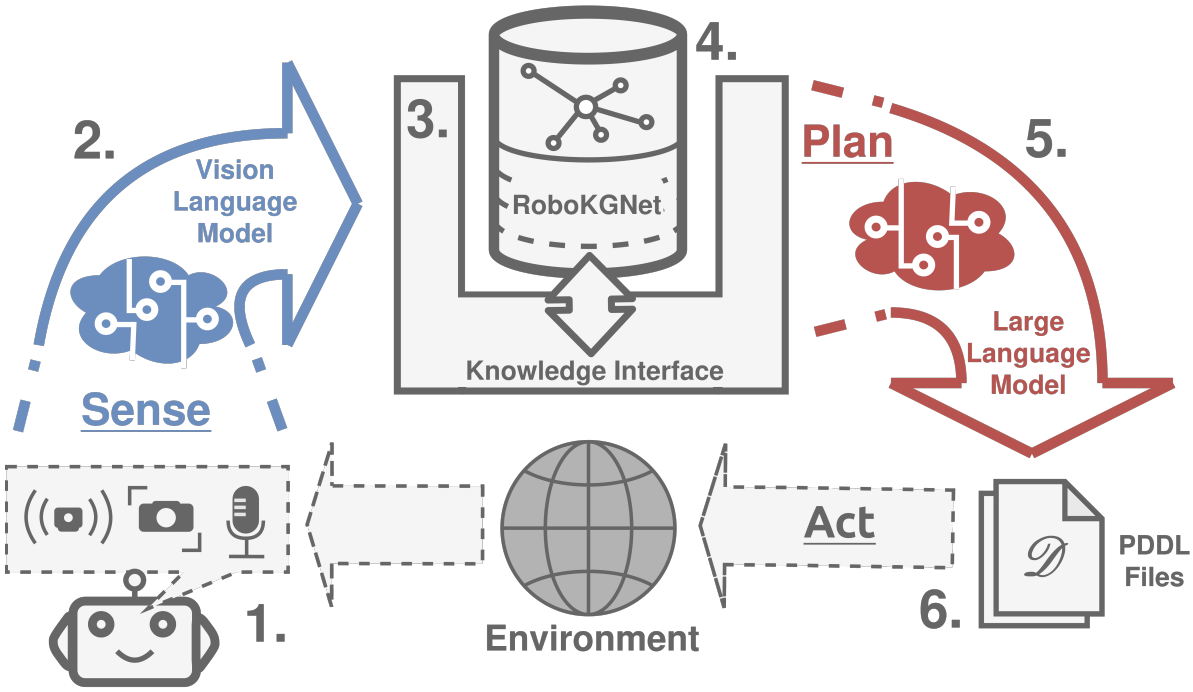


Figure 1: Overview of the proposed modular neurosymbolic SPA framework.

Our system uses a vision–language model as the perception module, which constructs a structured observation graph of the environment from multimodal sensory data. A large language model subsequently generates PDDL specifications based on this representation, which are then used by the robotic system for planning and execution. The two components are interconnected through a commonsense knowledge graph, called RoboKGNet, which serves as an intermediary grounding and validation layer. RoboKGNet facilitates the interaction between perception and planning, while also providing the background knowledge required for generating the corresponding PDDL files.

2. Neurosymbolic SPA Framework

The proposed framework can be modularized into six components, as shown in Figure 1.

The **(1) data acquisiton layer**, which considers the hardware, together with its associated software components such as drivers and controllers, collects sensory data and forwards it to the sensing component. The **(2) sensing component** is implemented using a VLM, which takes a summary of the sensory data as input and produces a symbolic description of the environment in the form of an observation graph [9]. The **(3) knowledge interface** integrates RoboKGNet by establishing correspondences between the observation graph and the underlying RoboKGNet knowledge graph, a process also known as Perceived-Entity Linking [9]. It is also responsible for orchestrating and managing query execution. **(4) RoboKGNet** acts as a repository of background knowledge, integrating multiple resources, as described in detail in the following section. The **(5) planning module** consists of an LLM that takes as input the actions available to the robot, the task-relevant object types detected in the environment, and a predefined set of predicates describing relations between objects, and outputs PDDL domain and problem files that are validated by a symbolic planner. Finally, the **(6) action component** includes the low-level or primitive actions implemented on the robot, which should be aligned with the actions described in the generated PDDL files. This modular architecture is designed to enable developers to adapt the framework and incorporate their own components (e.g., a custom knowledge base) while requiring minimal to no modifications to the remaining modules.

Data Acquisition layer

The data acquisition layer consists of the sensors and associated low-level software components, such as drivers, terminal interfaces, and, in our case, automatic speech recognition algorithms that pre-process the raw sensory input. This layer abstracts over different input modalities and supports both virtual and real-world scenarios. It aggregates the pre-processed data and forwards the natural language instruction, together with the relevant sensory data, to the sensing component. If no natural language instruction is detected, the SPA loop is not initiated.

Sensing Component

The sensing component takes the natural language instruction and the corresponding sensory input frame from the data acquisition layer and inserts them into a structured prompt. The VLM is instructed to generate an observation graph describing the task-relevant objects in the scene, including their labels, relevant properties, and spatial relations. It also identifies a target FrameNet frame and VerbNet class associated with the instruction, which constrains the planning domain to the relevant action context, such as cleaning, cooking, or object arrangement. The resulting observation graph is passed to the knowledge interface for alignment with RoboKGNet.

Knowledge Interface

The knowledge interface identifies the entities in RoboKGNet that correspond to the entities contained in the observation graph provided by the sensing component. This alignment enables more detailed inference over the knowledge graph, for example by retrieving object relations, functional properties, or information about what objects are typically used for. Then, the knowledge interface stores and executes the queries used to create and update plans in the Unified Planning Ontology, as well as to construct the corresponding domain files. This functionality follows the interface between the Unified Planning Ontology and the Unified Planning Library [10], as described in our previous work [7]. Finally, the knowledge interface provides the planning component with the information required to construct the domain and problem files, including relevant object types, predicates, actions, and background knowledge retrieved from RoboKGNet.

RoboKGNet

RoboKGNet comprises an integrated suite of linguistic resources, commonsense knowledge graphs, and ontologies that collectively provide the background knowledge required both for the structured representation and storage of sensory data and for supporting knowledge-intensive planning processes. Specifically, RoboKGNet currently consists of the alignment of the following resources:

- **WordNet** [11] provides the semantic backbone of RoboKGNet by supplying sense-level grounding. Given that WordNet is an extensive lexical database encompassing a broad coverage of English vocabulary, we constrain its use to concepts pertinent to robotic applications. Concretely, we currently include only hyponyms of the `Physical Entity` synset, together with action predicates drawn from VerbNet and location categories obtained from the Places Dataset [12].
- **ConceptNet** [13] is a large-scale, crowd-sourced common-sense knowledge graph. As WordNet provides limited semantic links between the different synsets, ConceptNet is used to retrieve relationships between objects using `LocatedAt` predicate, and the relationships between objects and action verbs using the `UsedFor` predicate.
- **VerbNet** [14] specifies the semantic structure of English verbs by systematically defining their associated thematic roles (e.g., the verb *to cook* presupposes the presence of an agent who performs the action, a patient that constitutes the food being prepared, and potentially one or more instruments employed in the cooking process). In the present work, VerbNet verb classes are used as the basis for specifying the PDDL domains.

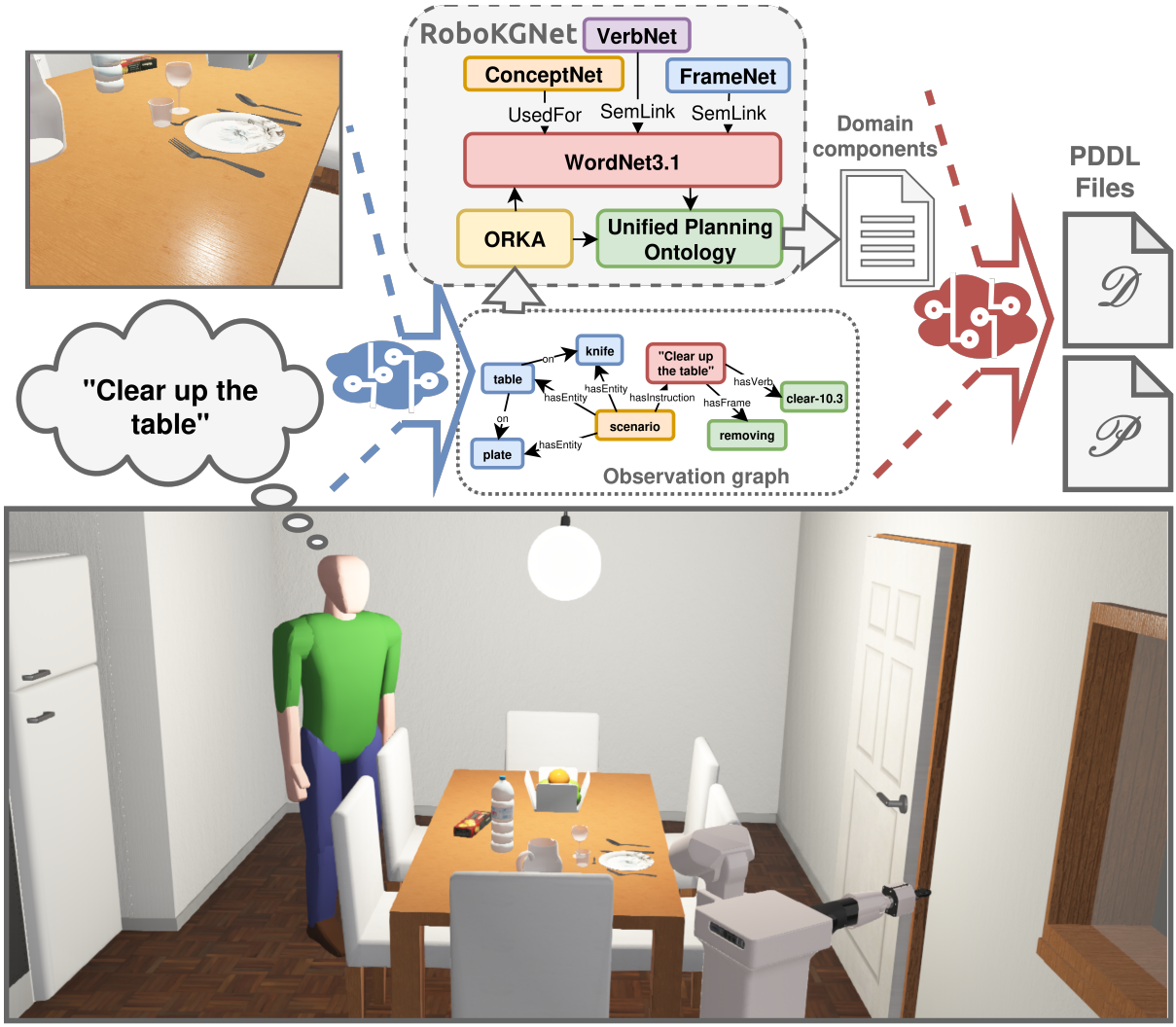


Figure 2: An example from the simulation environment, together with an overview of the components of RoboKGNet.

- Similarly to VerbNet, **FrameNet** [15] provides a semantic representation of situations in terms of frames, with participant roles encoded as frame elements. Within RoboKGNet, FrameNet is employed to supply broader situational context via non-peripheral roles (e.g., specifying the `Place` in which the cooking action occurs). As FrameNet encompasses numerous frames that are not pertinent to robotics applications, we further restrict our scope to the frames provided in the Human-Robot-Interaction Corpus (HURIC) [16]. This refinement also entails the inclusion of all frames linked to the frames contained in the HURIC annotations.
- **The Ontology for Robotic Knowledge Acquisition (ORKA)** [17] serves as a structured repository for perceptual data generated by the sensing component and is employed to align this information with the lexical database WordNet.
- **The Unified Planning Ontology (UPO)** [7] extends the Planning Ontology [18] by providing constructs and mappings that enable interoperability with the Unified Planning Library [10]. Within RoboKGNet, UPO serves as the schema for representing action models, robot configurations, and logical predicates in a manner that is reusable and consistent across heterogeneous application domains, which in turn are also stored together with the problem files.

An overview of these components can be seen in Figure 2.

Planning component

The planning component is implemented using an LLM, whose input prompt contains the available actions, predicates, and relevant object types provided by RoboKGNet through the knowledge interface. This information is extended by the observation graph containing the initial condition. Based on this information, the LLM generates the corresponding PDDL domain and problem files. These files are then validated by a symbolic planner to ensure that they are syntactically correct, semantically consistent, and capable of producing an executable plan. If validation fails, the error can be used to revise the prompt or regenerate the corresponding PDDL files.

Action layer

The action layer contains the controllers and implements low-level actions that correspond to the high-level actions specified in RoboKGNet, the generated PDDL domain file, and the resulting plan. Its main role is to ground the 14 atomic actions currently specified in RoboKGNet, such as `move`, `place`, or `grasp`, into executable robot behaviors. This includes binding symbolic action parameters to perceived entities, invoking the appropriate controllers, and monitoring whether the execution succeeds. If an action cannot be executed, the layer returns a failure signal that can be used to update the planning or perception components.

3. Demonstration

We plan a live demonstration in which participants can issue natural language commands to a simulated robotic agent operating within a virtual kitchen environment. The pipeline is then initiated, executing the following steps:

1. The instruction is integrated, alongside the camera input, into the prompt provided to the VLM that operates as the perception module.
2. The prompt is presented to users, along with the corresponding observation graph generated by the VLM, which encodes the extracted domain associated with the VerbNet verb.
3. The knowledge interface first aligns the observation graph with RoboKGNet and subsequently issues a series of queries that extract the relevant objects, the set of actions available to the robot, and the associated predicates.
4. The outputs of the queries are utilized to formulate the prompt provided to the LLM, which functions as the planning module.
5. The generated PDDL domain and problem files, together with the corresponding sequence of actions comprising the plan, are presented to the user.
6. Finally, once the plan is validated, it is shown to the user, and an attempt is made to execute the sequence of actions specified in the plan.

In the future, we intend to generalize the framework to encompass multiple operational scenarios and incorporate a learning mechanism that adapts to execution failures encountered during plan realization.

Declaration on Generative AI

We acknowledge that generative AI was used to improve writing style and to check grammar.

References

[1] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, L. Fan, VIMA: General Robot Manipulation with Multimodal Prompts, 2022. URL: https://arxiv.org/abs/2210.03094. doi:10.48550/ARXIV.2210.03094, version Number: 2.

[2] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, P. Florence, PaLM-E: An Embodied Multimodal Language Model, 2023. URL: https://arxiv.org/abs/2303.03378. doi:10.48550/ARXIV.2303.03378, version Number: 1.

[3] K. Kawaharazuka, J. Oh, J. Yamada, I. Posner, Y. Zhu, Vision-Language-Action Models for Robotics: A Review Towards Real-World Applications, IEEE Access 13 (2025) 162467–162504. URL: https://ieeexplore.ieee.org/document/11164279/. doi:10.1109/ACCESS.2025.3609980.

[4] T. Zhang, K. Qu, Z. Li, J. Wu, M. Hutter, M. Li, F. Shi, Using large language models for embodied planning introduces systematic safety risks, 2026. URL: https://arxiv.org/abs/2604.18463. doi:10.48550/ARXIV.2604.18463, version Number: 2.

[5] M. Martorana, F. Urgese, M. Adamik, I. Tiddi, Bridging Bots: from Perception to Action via Multimodal-LMs and Knowledge Graphs, 2025. URL: https://arxiv.org/abs/2507.09617. doi:10.48550/ARXIV.2507.09617, version Number: 1.

[6] V. Aregbede, P. Forte, H. Gupta, H. Andreasson, U. Köckemann, A. J. Lilienthal, Here's your PDDL Problem File! On Using VLMs for Generating Symbolic PDDL Problem Files, in: 2025 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Atlanta, GA, USA, 2025, pp. 4455–4461. URL: https://ieeexplore.ieee.org/document/11127832/. doi:10.1109/ICRA55743.2025.11127832.

[7] M. Adamik, P. Forte, PlanOwl: Automated PDDL Files Generation from OWLOntologies and Visual Language Models, Proceedings of the AAAI Symposium Series 7 (2025) 634–643. URL: https://ojs.aaai.org/index.php/AAAI-SS/article/view/36944. doi:10.1609/aaaiss.v7i1.36944.

[8] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, D. Wilkins, PDDL - The Planning Domain Definition Language (1998).

[9] M. Adamik, R. Pernisch, I. Tiddi, S. Schlobach, Advancing Robotic Perception with Perceived-Entity Linking, in: G. Demartini, K. Hose, M. Acosta, M. Palmonari, G. Cheng, H. Skaf-Molli, N. Ferranti, D. Hernández, A. Hogan (Eds.), The Semantic Web – ISWC 2024, volume 15232, Springer Nature Switzerland, Cham, 2025, pp. 192–209. URL: https://link.springer.com/10.1007/978-3-031-77850-6_11. doi:10.1007/978-3-031-77850-6_11, series Title: Lecture Notes in Computer Science.

[10] A. Micheli, A. Bit-Monnot, G. Röger, E. Scala, A. Valentini, L. Framba, A. Rovetta, A. Trapasso, L. Bonassi, A. E. Gerevini, L. Iocchi, F. Ingrand, U. Köckemann, F. Patrizi, A. Saetti, I. Serina, S. Stock, Unified Planning: Modeling, manipulating and solving AI planning problems in Python, SoftwareX 29 (2025) 102012. URL: https://linkinghub.elsevier.com/retrieve/pii/S2352711024003820. doi:10.1016/j.softx.2024.102012.

[11] G. A. Miller, WordNet: a lexical database for English, Communications of the ACM 38 (1995) 39–41. URL: https://dl.acm.org/doi/10.1145/219717.219748. doi:10.1145/219717.219748.

[12] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 Million Image Database for Scene Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (2018) 1452–1464. URL: https://ieeexplore.ieee.org/document/7968387/. doi:10.1109/TPAMI.2017.2723009.

[13] R. Speer, J. Chin, C. Havasi, ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, Proceedings of the AAAI Conference on Artificial Intelligence 31 (2017). URL: https://ojs.aaai.org/index.php/AAAI/article/view/11164. doi:10.1609/aaai.v31i1.11164.

[14] M. Palmer, C. Bonial, J. Hwang, VerbNet, volume 1, Oxford University Press, 2016. URL: https://academic.oup.com/edited-volume/34641/chapter/295177378. doi:10.1093/oxfordhb/9780199842193.013.15.

[15] C. J. Fillmore, C. Baker, A Frames Approach to Semantic Analysis, in: B. Heine, H. Narrog (Eds.), The Oxford Handbook of Linguistic Analysis, 1 ed., Oxford University Press, 2012, pp. 313–340. URL: https://academic.oup.com/edited-volume/38640/chapter/335357314. doi:10.1093/oxfordhb/9780199544004.013.0013.

[16] A. Vanzo, D. Croce, E. Bastianelli, R. Basili, D. Nardi, Grounded language interpretation of robotic commands through structured learning, Artificial Intelligence 278 (2020) 103181. URL: https:

//linkinghub.elsevier.com/retrieve/pii/S0004370218302935. doi:10.1016/j.artint.2019.103181.

[17] M. Adamik, R. Pernisch, I. Tiddi, S. Schlobach, ORKA: An Ontology for Robotic Knowledge Acquisition, in: M. Alam, M. Rospocher, M. Van Erp, L. Hollink, G. A. Gesese (Eds.), Knowledge Engineering and Knowledge Management, volume 15370, Springer Nature Switzerland, Cham, 2025, pp. 309–327. URL: https://link.springer.com/10.1007/978-3-031-77792-9_19. doi:10.1007/978-3-031-77792-9_19, series Title: Lecture Notes in Computer Science.

[18] B. Muppasani, V. Pallagani, B. Srivastava, R. Mutharaju, M. N. Huhns, V. Narayanan, A Planning Ontology to Represent and Exploit Planning Knowledge for Performance Efficiency, 2023. URL: https://arxiv.org/abs/2307.13549. doi:10.48550/ARXIV.2307.13549, version Number: 2.