The Journal of Open Source Software

Benchmarking Hierarchical Reasoning with HierarchyCraft

Mathis Fédérico [1,2] **and Matthew E. Taylor** [1,3]

1 Department of Computing Science, University of Alberta, Canada **2** CentraleSupelec, University of Paris-Saclay, France **3** Alberta Machine Intelligence Institute (Amii), Canada

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Tristan Miller

Reviewers:

- @lwu9
- @Christopher-Henry-UM

Submitted: 14 January 2024
Published: unpublished

Summary

Hierarchical reasoning poses a fundamental challenge in the field of artificial intelligence. Existing methods may struggle when confronted with hierarchical tasks, yet there is a scarcity of suitable environments or benchmarks designed to comprehend how the structure of the underlying hierarchy influence a task difficulty. Our software represents a crucial initial step in the development of tools aimed at addressing research questions related to hierarchical reasoning.

We introduce **HierarchyCraft**, a lightweight environment builder designed for creating hierarchical reasoning tasks with arbitrary width and depth and that do not necessitate feature extraction. This includes tasks containing pixel images, text, sound, or any data requiring deep-learning based feature extraction. HierarchyCraft serves a dual purpose by offering a set of pre-defined hierarchical environments and simplifying the process of creating customized hierarchical environments.

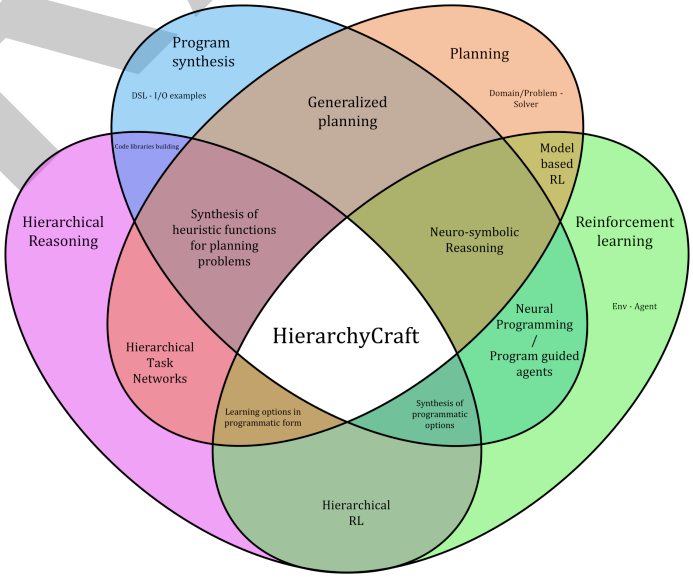

Figure 1: HierarchyCraft is at the intersection of Reinforcement learning, Planning, Hierarchical reasoning and Program synthesis.

The Journal of Open Source Software

Statement of need

HierarchyCraft is designed as a user-friendly Python library for constructing environments tailored to the study of hierarchical reasoning in the contexts of reinforcement learning, classical planning, and program synthesis as displayed in Figure 1.

Analysis and quantification of the impacts of diverse hierarchical structures on learning agents is essential for advancing hierarchical reasoning. However, current hierarchical benchmarks often limit themselves to a single hierarchical structure per benchmark, and present challenges not only due to this inherent hierarchical structure but also because of the necessary representation learning to interpret the inputs.

We argue that arbitrary hierarchical complexity can emerge from simple rules without the need for learning a representation. To the best of our knowledge, no general frameworks currently exist for constructing environments dedicated to studying the hierarchical structure itself, underscoring the necessity for the development of tools like HierarchyCraft. We compare five particularly related benchmarks to HierarchyCraft.

GridWorld

GridWorld, a general class of 2D grid-based environments, is frequently utilized in hierarchical reinforcement learning research, notably within the options framework (Sutton et al., 1999).

Minigrid (Chevalier-Boisvert et al., 2018) is a user-friendly Python library that not only implements a GridWorld engine but also expands its capabilities. This allows researchers to create more intricate scenarios by introducing additional rooms, objectives, or obstacles, as illustrated in Figure 2. Unfortunately, GridWorld environments typically exhibit a **limited hierarchical structure** and primarily focus on navigation tasks.

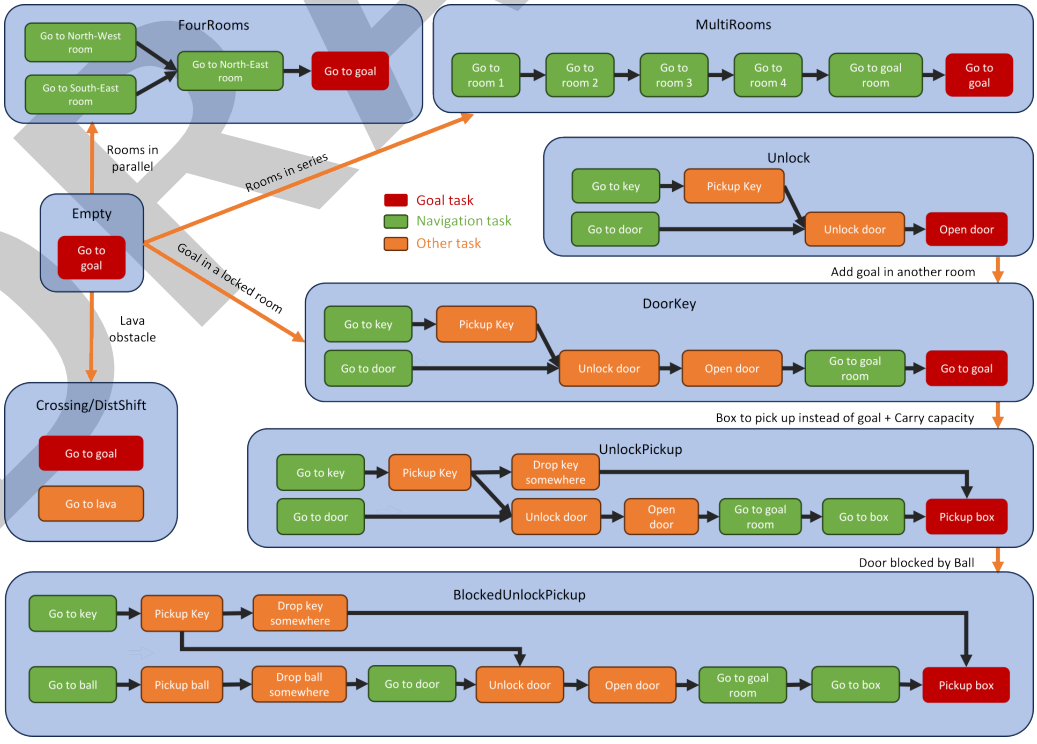

Figure 2: Example of Minigrid environments hierarchical structures and their relationships. There is only a few possible sub-tasks and most of them are navigation tasks (in green).

Minecraft

An exemplary instance of a hierarchical task is the collection of diamonds in the popular video game Minecraft, as showcased in the MineRL competition (Guss et al., 2021), where hierarchical reinforcement learning agents have dominated the leaderboard(Milani et al., 2020).

Due to the sparse rewards, exploration difficulty, and long time horizons in this procedurally generated sandbox environment, DreamerV3 (Hafner et al., 2023) recently became the first algorithm to successfully collect diamonds in Minecraft without prior training or knowledge. Unfortunately, DreamerV3 required training on an Nvidia V100 GPU for 17 days, gathering around 100 million environmental steps. Such **substantial computational resources are inaccessible to most researchers**, impeding the overall progress of research on hierarchical reasoning.

Moreover, although Minecraft has a undeniably complex hierarchical structure, **this underlying hierarchical structures is fixed** and cannot be modified without modding the game, a complex task for researchers.

Crafter

Crafter (Hafner, 2022) presents a lightweight grid-based 2D environment, with game mechanics akin to Minecraft and poses similar challenges including exploration, representation learning, rewards sparsity and long-term reasoning. Although Crafter offers 22 different tasks displayed in Figure 3, the **fixed underlying hierarchical structure** restricts how researchers can investigate the impacts of changes in this structure.

Furthermore, the tasks considered by the authors do not include navigation subtasks (e.g., Find water, look for a cow, wait for a plant to grow, go back to a table...) or certain optional but useful subtasks (e.g., Swords and the skill of dodging arrows contribute to making the task of killing skeletons easier.), leading to abrupt drops in success rates in the hierarchy instead of a more gradual increase in difficulty.

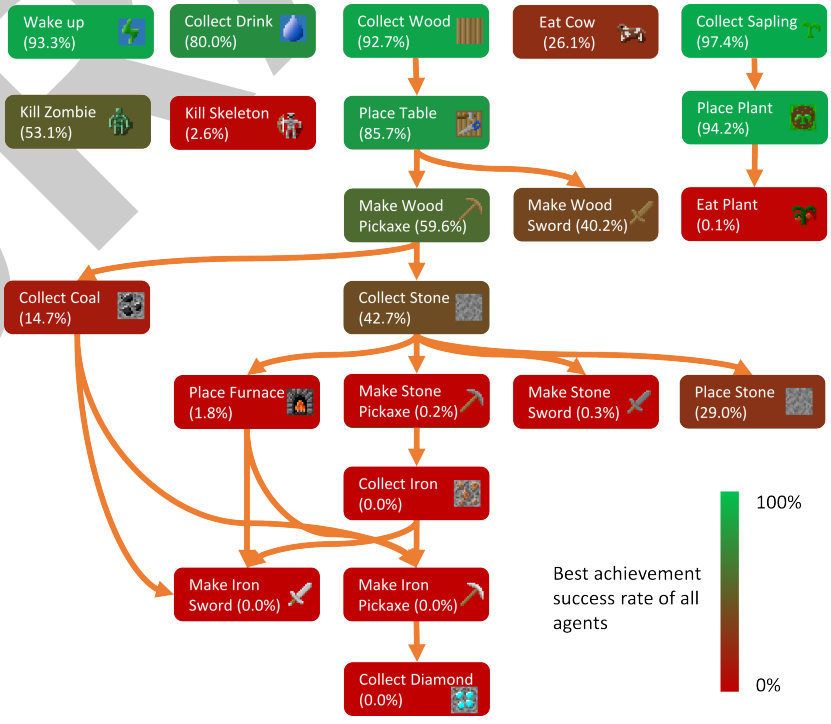

Figure 3: Partial Hierarchical structure of the Crafter environment. Inspired from Figure 4 of (Hafner, 2022)

PDDLGym

PDDLGym (Silver & Chitnis, 2020) is a Python library that automatically constructs Gym environments from Planning Domain Definition Language (PDDL) domains and problems. PDDL (Drew McDermott, 1998) functions as a problem specification language, facilitating the comparison of different symbolic planners. However, constructing PDDL domains and problems with a hierarchical structure is challenging and time-consuming, especially for researchers unfamiliar with PDDL-like languages. Additionally, PDDLGym is **compatible only with PDDL1** and does not support numeric-fluents introduced in PDDL 2.1 that are required to represent quantities in the inventories of HierarchyCraft environments.

Arcade Learning Environment (Atari)

The arcade learning environment (Bellemare et al., 2013) stands as a standard benchmark in reinforcement learning, encompassing over 55 Atari games. However, **only a few of these games, such as Montezuma's Revenge and Pitfall, necessitate hierarchical reasoning**. Each Atari games has a **fixed hierarchy that cannot be modified** and agents **demand substantial computational resources** to extract relevant features from pixels or memory, significantly slowing down experiments.

Design goals

HierarchyCraft aims to be a fruitful tool for investigating hierarchical reasoning, focusing on achieving the following four design goals.

1. Hierarchical by design

The action space of HierarchyCraft environments consists of sub-tasks, referred to as *Transformations*, as opposed to detailed movements and controls. But each *Transformations* has specific requirements to be valid (eg. have enought of an item, be in the right place), and these requirements may necessitate the execution of other *Transformations* first, inherently creating a hierarchical structure in HierarchyCraft environments.

This concept is visually represented by the *Requirements graph* depicting the hierarchical relationships within each HierarchyCraft environment. The *Requirements graph* is directly constructed from the list of *Transformations* composing the environement, as illustrated in Figure 4.

Requirements graphs should be viewed as a generalization of previously observed graphical representations from related works, including Figure 3 and Figure 2.

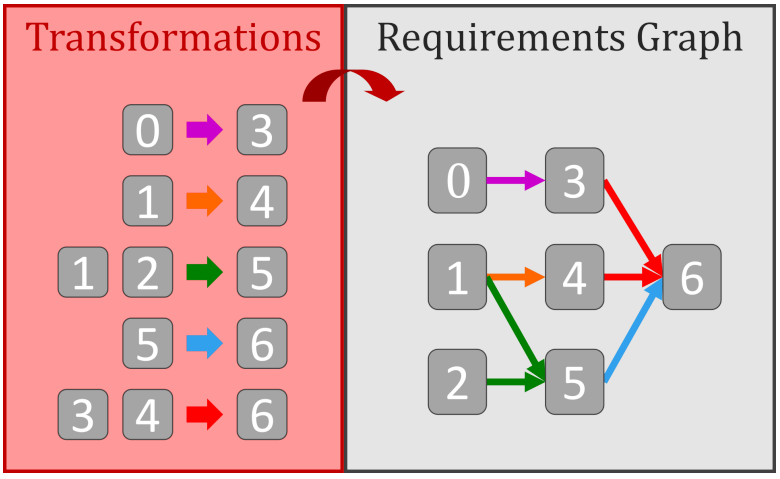

Figure 4: How sub-tasks build a hierarchical structure.

2. No feature extraction needed

In contrast to benchmarks that yield grids, pixel arrays, text, or sound, HierarchyCraft directly provides a low-dimensional latent representation that does not require learning, as depicted in Figure 5. This not only saves computational time but also enables researchers to concentrate on hierarchical reasoning while additionally allowing for the utilization of classical planning frameworks such as PDDL (Drew McDermott, 1998) or ANML (Smith et al., 2007).

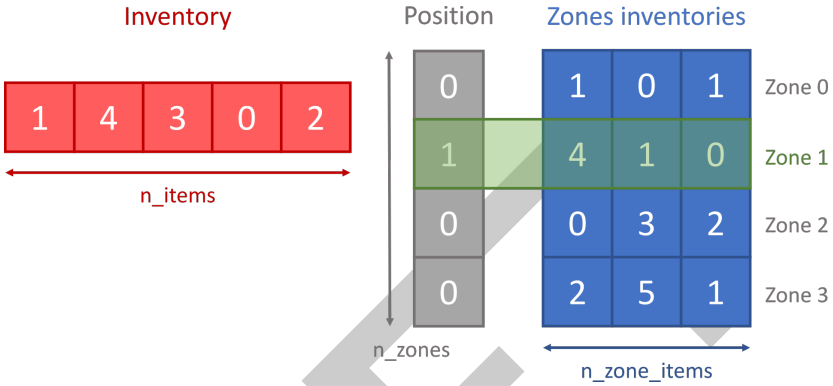

Figure 5: HierarchyCraft state is already a compact representation.

3. Easy to use and customize

HierarchyCraft is a versatile framework enabling the creation of diverse hierarchical environments. The library is designed to be simple and flexible, allowing researchers to define their own hierarchical environments with detailed guidance provided in the documentation. To showcase the range of environments possible within HierarchyCraft, multiple examples are provided.

4. Compatible with domains frameworks

HierarchyCraft environments are directly compatible with both reinforcement learning through OpenAI Gym (Brockman et al., 2016) and planning through the Unified Planning Framework ("Unified Planning Framework," 2023) (see Figure 6). This compatibility facilitates usage by both the reinforcement learning and planning communities.

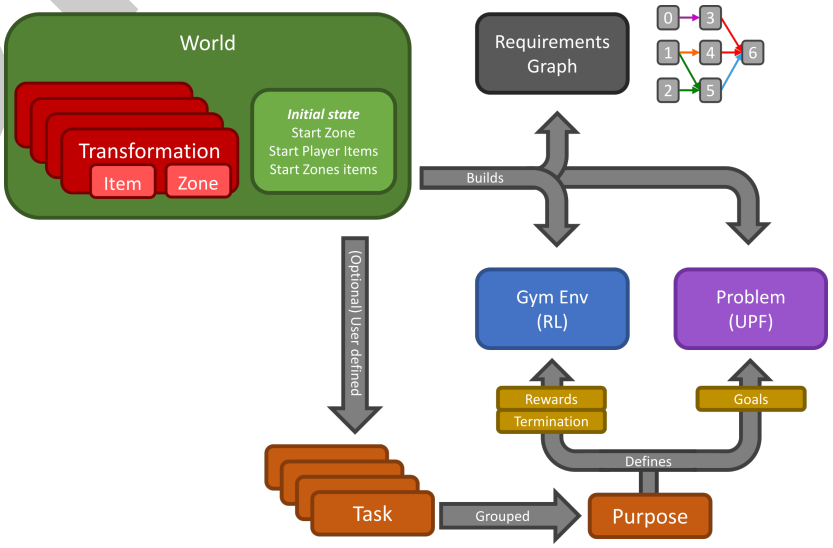

Figure 6: HierarchyCraft pipeline into different representations.

Acknowledgements

This work was made possible by the research program of the engineering cursus at CentraleSupelec, University of Paris-Saclay, France.

The research was conducted at the Intelligent Robot Learning (IRL) Lab, University of Alberta, which is supported in part by research grants from the Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; Compute Canada; Huawei; Mitacs; and NSERC.

Special thanks to Shang Wang, Yuxuan Li, and Laura Petrich from the IRL Lab for their assistance in finding and describing related works, their critical thinking on the project, and their contributions to the revisions of the documentation and report.

References

Bellemare, M. G., Naddaf, Y., Veness, J., & Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, *47*, 253–279.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI gym. *CoRR*, *abs/1606.01540*. http://arxiv.org/abs/1606.01540

Chevalier-Boisvert, M., Willems, L., & Pal, S. (2018). *Minimalistic gridworld environment for gymnasium*. https://github.com/Farama-Foundation/Minigrid

Drew McDermott. (1998). PDDL - the planning domain definition language. *Artificial Intelligence Planning Systems, TR98003/DCS TR1165*.

Guss, W. H., Castro, M. Y., Devlin, S., Houghton, B., Kuno, N. S., Loomis, C., Milani, S., Mohanty, S. P., Nakata, K., Salakhutdinov, R., Schulman, J., Shiroshita, S., Topin, N., Ummadisingu, A., & Vinyals, O. (2021). The MineRL 2020 competition on sample efficient reinforcement learning using human priors. *CoRR*, *abs/2101.11071*. https://arxiv.org/abs/2101.11071

Hafner, D. (2022). Benchmarking the spectrum of agent capabilities. *International Conference on Learning Representations*. https://openreview.net/forum?id=1W0z96MFEoH

Hafner, D., Pasukonis, J., Ba, J., & Lillicrap, T. (2023). Mastering diverse domains through world models. *arXiv Preprint arXiv:2301.04104*.

Milani, S., Topin, N., Houghton, B., Guss, W. H., Mohanty, S. P., Nakata, K., Vinyals, O., & Kuno, N. S. (2020). Retrospective analysis of the 2019 MineRL competition on sample efficient reinforcement learning. In H. J. Escalante & R. Hadsell (Eds.), *Proceedings of the NeurIPS 2019 competition and demonstration track* (Vol. 123, pp. 203–214). PMLR. https://proceedings.mlr.press/v123/milani20a.html

Silver, T., & Chitnis, R. (2020). PDDLGym: Gym environments from PDDL problems. *CoRR*, *abs/2002.06432*. https://arxiv.org/abs/2002.06432

Smith, D. E., Frank, J., & Cushing, W. (2007). *The ANML language*. https://api.semanticscholar.org/CorpusID:14116191

Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, *112*(1-2), 181–211.

Unified planning framework. (2023). In *GitHub repository*. GitHub. https://github.com/aiplan4eu/unified-planning