Hierarchical Dual-System Reinforcement Learning for Long-Horizon Autonomous Planning with Large Language Models

Nathan R. Lawrence
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA.
lawrence1990@oregonstate.edu

Kaihui Shao
Department of Computer Science, University of North Texas, Denton, TX, USA.
kaihuishao76@unt.edu

Abstract

This paper introduces a hierarchical dual-system reinforcement learning framework designed to address the challenges of long-horizon autonomous planning in environments where large language models serve as both reasoning components and planning priors. The proposed architecture draws upon the cognitive distinction between fast, intuitive reasoning and slow, deliberative reasoning, adapting it to a two-tier reinforcement learning hierarchy. At the lower level, a high-frequency control system learns primitive actions and local policies through trial-and-error interaction, while the upper level employs a deliberative system that leverages pretrained large language models to generate abstract subgoals, evaluate long-term consequences, and restructure task representations. The integration of large language models into this hierarchy introduces both opportunities and structural tensions, including issues of computational cost, semantic grounding, real-time adaptability, and ethical governance. This paper examines the system-level trade-offs inherent in such an architecture, focusing on deployment robustness, fairness in planning outcomes, sustainability of large-scale inference, and the policy implications of embedding generative models within autonomous planning pipelines. Through case illustrations in domains such as robotic navigation, logistics scheduling, and automated scientific experimentation, we analyze how the dual-system hierarchy can mitigate the brittleness of purely language-driven planning while retaining the flexibility of neural reasoning. The paper concludes by outlining a research agenda for improving the transparency, reliability, and scalability of hierarchical dual-system RL systems in real-world infrastructures.

Keywords

hierarchical reinforcement learning, dual-system theory, large language models, long-horizon planning, autonomous systems, socio-technical infrastructure, governance.

1. Introduction

The emergence of large language models has fundamentally altered the landscape of autonomous planning, enabling systems that can interpret natural language instructions, reason about abstract goals, and generate coherent sequences of actions over extended time horizons. However, the deployment of such models in real-world, long-horizon planning tasks reveals critical limitations in reliability, computational efficiency, and adaptability to dynamic

environments. Reinforcement learning, which enables agents to learn optimal behaviors through interaction, offers a complementary paradigm but struggles with the combinatorial complexity of long-horizon tasks and the sparse reward signals that often characterize them. Hierarchical reinforcement learning has been proposed as a remedy, structuring decision-making into temporally abstract subproblems. Yet existing hierarchies typically lack the semantic reasoning capabilities that large language models provide. This paper proposes a novel synthesis: a hierarchical dual-system reinforcement learning architecture in which a fast, learned control system operates at a low level, while a slow, language-model-driven system oversees strategic decomposition and re-planning. By drawing on the dual-system framework popularized in cognitive science [1], we argue that such a separation can address both the sample inefficiency of monolithic RL and the hallucination and instability of pure language-based planning. We structure this paper around system-level considerations: the architectural trade-offs between speed and deliberation, the integration of language models as planning oracles versus critics, the infrastructure demands of maintaining both systems in real time, and the broader governance and fairness implications of autonomous systems that mix learned experience with generative reasoning.

2. Background and Related Work

Long-horizon planning remains a central challenge in artificial intelligence. Traditional reinforcement learning methods, from tabular approaches to deep Q-networks and policy gradient algorithms [2][3], have achieved remarkable results in simulated environments but often require millions of interactions to converge on effective policies. Hierarchical reinforcement learning attempts to mitigate this by introducing a hierarchy of policies that operate at different temporal scales, with higher-level policies selecting subgoals for lower-level controllers [4]. This abstraction reduces the effective horizon and enables credit assignment across longer sequences. Meanwhile, the arrival of large language models such as GPT-3 and their successors has demonstrated the capacity for few-shot and zero-shot reasoning about plans, particularly when prompted with chain-of-thought sequences [5][6]. However, these models are prone to generating plausible but infeasible plans and lack closed-loop interaction with the environment. Dual-system theory, originating in cognitive psychology, posits two modes of thought: System 1, which is fast, automatic, and intuitive, and System 2, which is slow, deliberative, and analytical [1]. In artificial intelligence, this dichotomy has inspired hybrid architectures that combine learned intuitions with explicit reasoning [7]. Recent work has begun to explore the integration of language models into reinforcement learning pipelines, using them as reward models, plan generators, or action samplers [8][9]. Yet most of these approaches do not embed the language model within a hierarchical dual-system framework that explicitly separates fast and slow reasoning loops. Our work builds on these foundations but emphasizes the structural and infrastructural challenges of such an integration, moving beyond algorithmic novelty to consider deployment, governance, and sustainability.

3. Conceptual Architecture of Hierarchical Dual-System RL

The proposed architecture consists of two interacting reinforcement learning subsystems organized in a hierarchy. The lower-level system, analogous to System 1, is a learned controller trained with standard deep reinforcement learning algorithms, such as proximal policy optimization or soft actor-critic [3][10]. This system operates at a high frequency, mapping environmental observations to low-level actions within a few milliseconds. Its policy is optimized for speed and local optimality, focusing on immediate rewards and short-

horizon subgoals. The upper-level system, analogous to System 2, is a deliberative planner that operates on a slower timescale. It receives higher-level observations and the agent's current progress, consults a large language model to generate candidate subgoals or task decompositions, and selects among them using a learned meta-controller that evaluates expected long-term returns. This meta-controller itself may be trained through model-free or model-based reinforcement learning, but its action space consists of abstract subgoal specifications rather than primitive motor commands. The language model serves as a generative prior, providing semantically meaningful subgoals that are grounded in the agent's symbolic knowledge base. However, the language model's outputs are not executed directly; they are passed to the lower-level system as commands, which then learns to achieve them through trial and error. This separation has several structural advantages. First, it prevents the language model from directly driving low-level actions, thus reducing the risk of catastrophic failures due to model hallucinations. Second, it allows the lower-level system to adapt to local dynamics that the language model cannot anticipate, promoting robustness. Third, the hierarchical structure enables the upper-level system to replan only when necessary, conserving the computational cost of invoking a large language model. The trade-off lies in the choice of abstraction granularity: if subgoals are too coarse, the lower-level system may struggle to achieve them; if too fine, the computational overhead of the upper-level system reduces the benefit of hierarchy. Balancing this granularity requires careful meta-tuning and may vary across tasks.

4. Integration with Large Language Models

Integrating large language models into the deliberative planning layer introduces several design decisions regarding when and how to query the model, how to interpret its outputs, and how to handle uncertainty. In our framework, the language model is not used for online fine-tuning; instead, it remains a static or periodically updated module that provides planning suggestions based on pretrained knowledge. This approach reduces the need for environment-specific reinforcement learning on the language model itself, which is computationally prohibitive for large models. Instead, the upper-level meta-controller learns to select among candidate subgoals generated by the language model, effectively treating the model as a stochastic generator of plausible plans. This is reminiscent of methods that use language models as world models or as sources of prior knowledge in reinforcement learning [11]. However, a key difference is the dual-system separation: the language model operates only during deliberative phases, such as when the agent encounters a novel situation or detects a significant discrepancy between expected and observed outcomes. During routine execution, the fast lower-level system handles the majority of interactions without invoking the language model. This reduces latency and energy consumption, which are critical for real-time deployment in robotics or autonomous driving. The language model can also be used to generate intrinsic reward signals by evaluating the semantic similarity between achieved states and intended subgoals, providing a dense reward proxy in sparse environments. Yet reliance on language-model-generated rewards introduces risks of misalignment: the model may reward superficial linguistic similarity rather than functional completion. To mitigate this, we propose a two-stage verification process in which the lower-level system's experience is used to calibrate the language model's outputs, akin to a critic model that learns to predict whether a subgoal has been achieved [12]. This calibration loop is essential for maintaining grounding and preventing drift.

5. System-Level Considerations: Governance, Robustness, and Fairness

The deployment of hierarchical dual-system RL in autonomous planning systems raises important governance questions. Because the upper-level system relies on a large language model trained on vast internet corpora, its suggestions may embed biases, stereotypes, or unsafe behaviors that are not immediately apparent. For instance, in a logistics planning scenario, the language model might generate routes that systematically avoid certain neighborhoods due to biased training data, leading to inequitable service distribution. The lower-level reinforcement learning system, being purely data-driven from the environment, may amplify these biases if the history of interactions is skewed. Governance mechanisms must therefore include fairness audits that assess the distribution of outcomes across demographic groups, geographic regions, or temporal periods. One approach is to incorporate a fairness constraint into the meta-controller's objective function, penalizing subgoals that lead to disparate impacts. Another is to use the language model's own capability for reflective reasoning to flag potential biases before selection [13]. Robustness is another critical dimension. The dual-system architecture is designed to be resilient to failures of either component. If the language model generates an infeasible subgoal, the lower-level system will either fail to achieve it, signaling a need for replanning, or find an alternative path that still satisfies the higher reward. Similarly, if the lower-level system encounters distributional shift that degrades its performance, the upper-level system can detect the deviation and adjust the subgoal hierarchy. However, this resilience is contingent on the proper functioning of the detection mechanisms, which themselves require uncertainty quantification. Without reliable confidence estimates, the system may either replan too frequently, wasting computational resources, or too infrequently, leading to persistent failures. Ensuring robustness also demands that the language model's outputs are not treated as authoritative; the reinforcement learning framework must be able to override or ignore language model suggestions when they lead to poor outcomes. This requires that the meta-controller be trained with diverse counterfactual experiences, including scenarios where the language model provides misleading advice.

6. Deployment and Infrastructure Implications

Deploying a hierarchical dual-system architecture in production environments entails significant infrastructure challenges. The lower-level system typically runs on edge devices or local processors to meet latency requirements, while the upper-level system, especially the large language model, may require cloud-based GPU clusters or specialized hardware accelerators. This creates a network dependency that can become a single point of failure. Redundancy strategies, such as caching frequent subgoal queries or maintaining a smaller distilled model for common cases, are necessary to maintain uptime. The energy footprint of repeatedly invoking a large language model for planning tasks is substantial. Estimates from large-scale deployments indicate that a single query to a 175-billion-parameter model can consume several kilowatt-hours of electricity, which is unsustainable for continuous operation in autonomous systems [14]. Our architecture mitigates this by limiting the language model to deliberative phases, which may occur only a few times per hour or per day depending on the task. However, even infrequent queries can accumulate significant costs over long-duration deployments, such as autonomous farming or space exploration. One potential solution is to use a cascade of models: a smaller, faster language model for routine high-level reasoning and a larger, more capable model for complex or anomalous situations. This hierarchical model selection parallels the dual-system hierarchy and can be tuned to balance cost and performance. Additionally, the infrastructure must support continuous monitoring and logging of both systems' decisions for auditability. In safety-critical applications such as autonomous

driving or medical planning, regulatory requirements may mandate that every high-level decision be recorded and explainable. The dual-system architecture facilitates this because the upper-level system's reasoning steps can be traced through its subgoal selections and the language model's generated plans, while the lower-level system's actions are recorded as raw interaction data. However, the interpretability of language model outputs remains an active research area; ensuring that generated plans can be reliably linked to logical reasoning chains is essential for accountability.

7. Case Illustrations and Cross-Domain Comparisons

To illustrate the practical implications of the hierarchical dual-system RL architecture, we consider three domains with different planning challenges. In robotic navigation, a mobile robot must traverse unknown terrain to reach a distant goal. The lower-level system learns local control policies for collision avoidance and short-distance movement, while the upper-level system uses a language model to decompose the path into semantic waypoints such as "go to the red building" or "cross the bridge" [15]. The language model leverages map descriptions and prior knowledge to suggest a sequence of waypoints. The lower-level system then learns to navigate between waypoints, adapting to unforeseen obstacles. This approach has been shown to outperform both pure RL and pure language-based planning in terms of success rate and adaptability, though at higher computational cost during the deliberative planning phase. In logistics scheduling, a warehouse management system must coordinate multiple robots to fulfill orders over an eight-hour shift. The lower-level system handles individual robot movements and collision avoidance, while the upper-level system generates daily schedules by querying a language model trained on logistics optimization literature. The language model proposes schedules, and the meta-controller selects the one that minimizes idle time and energy use based on historical data. A key trade-off here is that the language model may propose schedules that are theoretically optimal but practically infeasible due to unmodeled constraints, such as battery charging cycles. The lower-level system's failure to follow the schedule triggers replanning, which gradually improves the meta-controller's ability to filter unrealistic proposals [12]. In automated scientific experimentation, an AI system must design and execute a sequence of laboratory procedures to discover a new chemical compound. The lower-level system controls the robotic arm and microfluidic devices, while the upper-level system generates experimental hypotheses and selects the next experiment. The language model, fine-tuned on scientific literature, suggests plausible hypotheses. The reinforcement learning meta-controller values not only the outcome of the experiment but also the information gain, using a Bayesian approach to explore the hypothesis space. This domain highlights the need for fairness in resource allocation across different research groups or experiments, as the language model may be biased toward well-studied chemical families. Governance structures must ensure that the system does not systematically ignore underrepresented research directions.

8. Forward-Looking Perspectives

The future development of hierarchical dual-system RL for long-horizon planning will likely focus on closing the loop between the two systems more tightly. One promising direction is to allow the lower-level system to provide feedback to the upper-level system in the form of learned state abstractions that can be used to refine the language model's subgoal generation. For example, if the lower-level system repeatedly fails to achieve a particular type of subgoal, the upper-level system can learn to generate alternative formulations that are more grounded in the agent's actual capabilities. This co-adaptation resembles the process of mutual

accommodation in human cognition, where intuitive and deliberative systems learn from each other over time [1]. Another direction is the development of specialized hardware that can efficiently run both fast RL policies and large language models on the same device, reducing latency and improving energy efficiency. Emerging neuromorphic chips and in-memory computing architectures may enable this integration. From a policy perspective, the deployment of such systems in public infrastructure will require standards for transparency, accountability, and fairness. Regulatory bodies may need to mandate that autonomous planning systems maintain a "dual-system black box" that logs both the fast and slow reasoning steps in an auditable format. The research community also has a responsibility to develop benchmarks and evaluation suites that test not only task completion accuracy but also the robustness, fairness, and sustainability of the dual-system approach. Finally, the interplay between language models and reinforcement learning in the dual-system framework raises deep questions about the nature of intelligence and planning. While not the focus of this paper, it is worth noting that the cognitive science underpinnings of the dual-system theory are still evolving, and the extent to which our artificial systems mirror human cognition is an open empirical question.

9. Conclusion

This paper has presented a hierarchical dual-system reinforcement learning architecture for long-horizon autonomous planning that integrates large language models as deliberative reasoning components while maintaining a fast, learned control system for local execution. We have examined the structural trade-offs inherent in this integration, including the choice of abstraction granularity, the computational cost of language model inference, and the mechanisms for ensuring robustness and fairness. Through case illustrations in robotics, logistics, and scientific experimentation, we have shown how the dual-system approach can mitigate the fragility of pure language-based planning while leveraging the semantic flexibility that language models provide. The infrastructure and governance implications of deploying such systems at scale are substantial, requiring careful design of hardware, software, and regulatory frameworks. We anticipate that future research will refine the co-adaptation between the two systems, develop energy-efficient inference methods, and establish standards for equitable and accountable autonomous planning. The hierarchical dual-system framework offers a promising pathway toward autonomous systems that combine the speed of learned intuition with the depth of deliberative reasoning, provided that the associated socio-technical challenges are addressed in parallel.

References

1. Kahneman, D. (2011). Thinking, fast and slow. Farrar, Straus and Giroux.
2. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.
3. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
4. Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. Discrete Event Dynamic Systems, 13(4), 341-379.
5. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877-1901.

6. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... & Chi, E. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35, 24824-24837.

7. Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019). Reinforcement learning, fast and slow. Trends in Cognitive Sciences, 23(5), 408-422.

8. Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., ... & Zhang, A. (2022). Do as I can, not as I say: Grounding language in robotic affordances. arXiv preprint arXiv:2204.01691.

9. Huang, W., Abbeel, P., Pathak, D., & Mordatch, I. (2022). Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. arXiv preprint arXiv:2201.07207.

10. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv preprint arXiv:1801.01290.

11. Lu, Y., Zhong, Y., & Sievert, S. (2023). Language models as world models for reinforcement learning. In International Conference on Machine Learning (pp. 23156-23182). PMLR.

12. Dou, Z., Cui, D., Yan, J., Wang, W., Chen, B., Wang, H., ... & Zhang, S. (2025). Dsadf: Thinking fast and slow for decision making. arXiv preprint arXiv:2505.08189.

13. Schramowski, P., Turhan, C., Jentzsch, S., Rothkopf, C., & Kersting, K. (2022). The moral debate: Language models as moral judges. In Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society (pp. 620-630).

14. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3645-3650).

15. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sunderhauf, N., ... & van den Hengel, A. (2018). Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3674-3683).

16. Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., & Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36, 53728-53741.

17. Finn, C., Levine, S., & Abbeel, P. (2016). Guided cost learning: Deep inverse optimal control via policy optimization. In International Conference on Machine Learning (pp. 49-58).

18. Duan, Y., Chen, X., Houthooft, R., Schulman, J., & Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In International Conference on Machine Learning (pp. 1329-1338).

19. Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., ... & Michalewski, H. (2019). Model-based reinforcement learning for atari. arXiv preprint arXiv:1903.00374.

20. Hanna, J. P., & Stone, P. (2017). Grounded action transformation for robot learning from demonstration. In Proceedings of the 2017 International Conference on Autonomous Agents and Multiagent Systems (pp. 876-884).

21. Shu, T., Bhandwaldar, A., Gan, C., Smith, K., Liu, S., Gutfreund, D., ... & Ullman, T. (2020). AGENT: A benchmark for core psychological reasoning. In International Conference on Machine Learning (pp. 8830-8841).