Dual-Process System: an Architectural Pattern for Assurable Autonomous Robots Inspired by Dual-Process Theory

Krzysztof Czarnecki

Waterloo Intelligent Systems Engineering (WISE) Lab
University of Waterloo, Canada
krzysztof.czarnecki@uwaterloo.ca

Abstract

This paper proposes the 'dual-process system,' an architectural pattern for assurable autonomous robots, inspired by the dual-process theory of the human mind. It seamlessly integrates end-to-end neural architectures with symbolic AI methods, mitigating critical challenges in assurability, such as interpretability, robustness, and the handling of rare inputs, thereby enhancing both performance and safety.

Introduction

The advent of AI has revolutionized the field of autonomous robotics, particularly in tasks like object recognition, where it achieves unprecedented accuracy. This advancement extends to behavior prediction and planning, allowing AI systems, leveraging deep neural networks (DNNs) and trained on extensive datasets, to adapt seamlessly to new environments. Such end-to-end neural architectures allow for holistic system optimization—a solution to the tedious process of updating multiple system components individually. However, these AI systems bring significant assurance challenges: they often lack interpretability and explainability, show unpredictable responses to rare inputs, and suffer from a lack of robustness to shifts in input domains. This uncertainty complicates the application of modular assume-guarantee reasoning and undermines the ability to offer firm assurance guarantees (Salay and Czarnecki 2018).

To address these challenges, this extended abstract introduces a 'dual-process system,' an architectural pattern for autonomous robots, inspired by the dual-process theory of the human mind. It outlines the necessary background, delineates the pattern using the problem, solution, and consequences format (Buschmann et al. 1996), and exemplifies its application. The abstract compares it with existing patterns and explores its advantages, drawbacks, limitations, and open questions, paving the way for future research. By presenting this concept and terminology, this paper aims to spark further discussion and investigation in the field.

Background and Related Work

Dual-process theory and applications. The dual-process theory posits that the human brain operates using two distinct types of processes: Type I, which are fast, non-conscious, and Type II, which are slower, conscious, and capable of deliberate reasoning (Epstein 1994; Kahneman 2011; Evans and Stanovich 2013). Routine tasks, like object recognition in the visual pathway, are managed by default through Type I processes. In contrast, Type II processes, which involve higher-level reasoning in the prefrontal cortex, are activated in situations of high uncertainty, surprise, novelty, or when making high-stakes decisions. This theory has inspired research in robotics (Gurney et al. 2009) and AI (Booch et al. 2021), yet its exploration from an assurance perspective remains limited. Notable exceptions include work by Jha et al. (2020), who advocate for the use of symbolic methods (Type II) in validating perception and planning, with perception itself being handled by subsymbolic neural networks (Type I), and Salay and Czarnecki (2022), who propose a dual-process architecture specifically for perception tasks. This work, however, uniquely develops a dual-process architectural pattern for comprehensive end-to-end neural architectures.

End-to-end neural architectures. These architectures employ DNNs extensively across the sense-perceive-act pipeline in autonomous robots and are designed for end-to-end optimization. They facilitate the discovery of task-specific data representations, allowing for the coordinated improvement and adaptation of perception, prediction, and planning. End-to-end architectures can be categorized into single-task and multi-task models. Single-task architectures utilize a monolithic DNN for the entire pipeline, while multi-task architectures consist of multiple subnetworks, each dedicated to a specific subtask. Notable variants include multi-head and sequential architectures (Hu et al. 2023). For instance, Figure 1 (ignore Type II subsystem for now) illustrates a sequential architecture for autonomous vehicles (AVs): a central backbone extracts features from sensor inputs, and specialized subnetworks handle perception, prediction, and planning tasks. These architectures employ subsymbolic (or latent) intermediate representations, necessitating decoders for interpretable representation extraction and training supervision. A notable feature is the use of skip connections from the backbone to each subnetwork, essential for improved convergence and optimal performance. However, this interconnectivity can compromise modular-

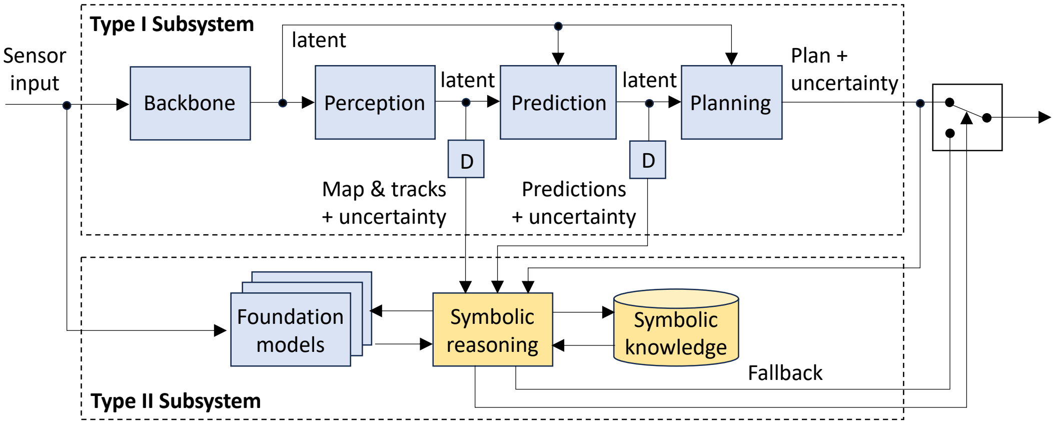

Figure 1: Dual-process system for autonomous driving. Blue boxes represent DNNs and yellow boxes represent symbolic components. ‘D’ denotes ‘decoder.’ Arrows denote data flow. Type I Subsystem architecture is inspired by UniAD (Hu et al. 2023).

ity, as subnetworks might access any information from the backbone features, bypassing sequential processing. This architecture poses several assurance challenges, as discussed by Salay and Czarnecki (2018): the subsymbolic nature of representations and reasoning obscures interpretability and explainability (note that decoding is likely partial); it eludes causal and assume-guarantee reasoning; it is susceptible to spurious features and shortcuts; sensitive to domain shifts; and produces unpredictable outputs for rare inputs.

Architectural Pattern: Dual Process System

Context and problem. End-to-end neural architectures facilitate data-driven representation discovery and adaptation but significantly compromise assurability, due to lack of (i) interpretability and explainability, (ii) robustness to input corruptions and domain shifts, (iii) detection of out-of-scope inputs, (iv) common sense to handle rare inputs, and (v) causal reasoning and guarantees.

Solution. To mitigate these issues, the proposed solution involves a dual-process architecture (Fig. 1). The Type I subsystem, primarily an end-to-end neural architecture, is responsible for executing the overall task with low latency, leveraging the advantages of end-to-end optimization and representation learning. Complementing this, the Type II subsystem, primarily based on symbolic AI methods, is tasked with monitoring assurance targets, such as safety and security, and providing fallback. It uses symbolic reasoning and knowledge, offering interpretability and verifiability, typical examples being rule-based systems and logic reasoners. In the context of AVs, this could include rule books (Censi et al. 2019) and Responsibility-Sensitive Safety (Shalev-Shwartz, Shammah, and Shashua 2018). For tasks that are inherently subsymbolic, such as object recognition in camera images, grounding is provided by decoded representations from the Type I subsystem, enriched with uncertainty estimates, both aleatoric and epistemic (Kendall and Gal 2017). These may be supplemented by additional models, like foundational models, for common sense knowledge (Zhao, Lee, and Hsu 2023). The Type II subsystem assesses the uncertainty, plausibility, and consistency of intermediate representations, ensuring the final plan aligns with assurance targets. Uncertainty can be evaluated using uncertainty wrappers (Kläs and Sembach 2019). If the plan from the Type I subsystem fails to meet targets or lacks confidence, the Type II subsystem intervenes with a fallback plan, such as a minimum risk maneuver. Basic fallbacks, like stopping by the roadside, are immediately available, but when time allows, the Type II subsystem can conduct more detailed analyses of specific detections or situations.

Variations. While the primary flow of information is from Type I to Type II subsystem, the latter can also influence Type I, such as through reinforcement during training or error compensation during runtime. Additionally, the development and refinement of symbolic knowledge can be facilitated by rule learning and distillation, drawing from data and data-driven models (Hu et al. 2016; Bouchard et al. 2022).

Related patterns. There are two related patterns. Simplex is an architecture coupling a primary, complex subsystem with a trusted and simple fallback (Bodson et al. 1994). Shielding is similar, but applying a monitor to a learning agent for reinforcement and protection (Alshiekh et al. 2018). Dual-process system can be seeing as a refinement of simplex for end-to-end neural architectures. In contrast to shielding, which relies on interpretable observations, it recognizes the need for subsymbolic grounding and uncertainty estimation, especially in perception, and integrates with the end-to-end subsystem at multiple stages.

Consequences. The solution mitigates the issues of end-to-end architectures, as already explained. It also offers diversity and increased confidence when Type I and II outputs are consistent. In cases of inconsistency, the Type II subsystem is trusted to provide fallback. The discovered inconsistencies may require re-training in Type I or update in Type II subsystem. Such updates can lead to the discovery of new symbolic knowledge relevant to the tasks. However, a notable drawback of this solution is the increased complexity of the overall system. Future research challenges include achieving sufficient completeness of Type II reasoning in relation to assurance targets and supporting consistency maintenance between the two subsystems.

References

Alshiekh, M.; Bloem, R.; Ehlers, R.; Könighofer, B.; Niekum, S.; and Topcu, U. 2018. Safe Reinforcement Learning via Shielding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Bodson, M.; Lehoczky, J.; Rajkumar, R.; Sha, L.; and Stephan, J. 1994. Analytic Redundancy for Software Fault-Tolerance In Hard Real-Time Systems. In Koob, G. M.; and Lau, C. G., eds., *Foundations of Dependable Computing*, volume 284, 183–212. Boston, MA: Springer US. ISBN 978-0-7923-9485-3. Series Title: The Springer International Series in Engineering and Computer Science.

Booch, G.; Fabiano, F.; Horesh, L.; Kate, K.; Lenchner, J.; Linck, N.; Loreggia, A.; Murgesan, K.; Mattei, N.; Rossi, F.; and Srivastava, B. 2021. Thinking Fast and Slow in AI. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17): 15042–15046.

Bouchard, F.; Sedwards, S.; and Czarnecki, K. 2022. A Rule-Based Behaviour Planner for Autonomous Driving. In Governatori, G.; and Turhan, A., eds., *Rules and Reasoning - 6th International Joint Conference on Rules and Reasoning, RuleML+RR 2022, Berlin, Germany, September 26-28, 2022, Proceedings*, volume 13752 of *Lecture Notes in Computer Science*, 263–279. Springer.

Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; and Stal, M. 1996. *Pattern-oriented software architecture, , Volume 1: A System of Patterns*. Chichester ; New York: Wiley. ISBN 978-0-471-95869-7.

Censi, A.; Slutsky, K.; Wongpiromsarn, T.; Yershov, D.; Pendleton, S.; Fu, J.; and Frazzoli, E. 2019. Liability, Ethics, and Culture-Aware Behavior Specification using Rulebooks. In *2019 International Conference on Robotics and Automation (ICRA)*, 8536–8542. Montreal, QC, Canada: IEEE. ISBN 978-1-5386-6027-0.

Epstein, S. 1994. Integration of the cognitive and the psychodynamic unconscious. *American psychologist*, 49(8): 709.

Evans, J. S. B.; and Stanovich, K. E. 2013. Dual-process theories of higher cognition: Advancing the debate. *Perspectives on psychological science*, 8(3): 223–241.

Gurney, K.; Hussain, A.; Chambers, J.; and Abdullah, R. 2009. Controlled and Automatic Processing in Animals and Machines with Application to Autonomous Vehicle Control. In *ICANN*.

Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; Lu, L.; Jia, X.; Liu, Q.; Dai, J.; Qiao, Y.; and Li, H. 2023. Planning-oriented Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Hu, Z.; Ma, X.; Liu, Z.; Hovy, E.; and Xing, E. 2016. Harnessing Deep Neural Networks with Logic Rules. In Erk, K.; and Smith, N. A., eds., *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2410–2420. Berlin, Germany: Association for Computational Linguistics.

Jha, S.; Rushby, J.; and Shankar, N. 2020. Model-Centered Assurance for Autonomous Systems. In *SAFECOMP*.

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

Kendall, A.; and Gal, Y. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kläs, M.; and Sembach, L. 2019. Uncertainty Wrappers for Data-Driven Models: Increase the Transparency of AI/ML-Based Models Through Enrichment with Dependable Situation-Aware Uncertainty Estimates. In Romanovsky, A.; Troubitsyna, E.; Gashi, I.; Schoitsch, E.; and Bitsch, F., eds., *Computer Safety, Reliability, and Security*, volume 11699, 358–364. Cham: Springer International Publishing. ISBN 978-3-030-26249-5 978-3-030-26250-1. Series Title: Lecture Notes in Computer Science.

Salay, R.; and Czarnecki, K. 2018. Using machine learning safely in automotive software: An assessment and adaption of software process requirements in ISO 26262. *arXiv preprint arXiv:1808.01614*.

Salay, R.; and Czarnecki, K. 2022. A Safety Assurable Human-Inspired Perception Architecture. In *Computer Safety, Reliability, and Security. SAFECOMP 2022 Workshops: DECSoS, DepDevOps, SASSUR, SENSEI, USDAI, and WAISE Munich, Germany, September 6–9, 2022, Proceedings*, 302–315. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-14861-3.

Shalev-Shwartz, S.; Shammah, S.; and Shashua, A. 2018. On a Formal Model of Safe and Scalable Self-driving Cars. *arXiv preprint: 1708.06374*, arXiv:1708.06374.

Zhao, Z.; Lee, W. S.; and Hsu, D. 2023. Large Language Models as Commonsense Knowledge for Large-Scale Task Planning. In *Thirty-seventh Conference on Neural Information Processing Systems*.