



Neurosymbolic Robot Programming

Benjamin Alt¹

Received: 12 March 2026 / Accepted: 6 October 2026
© The Author(s) 2026

Abstract

Programming intelligent robots requires novel approaches that harness the learning abilities of deep neural architectures while allowing humans to specify, monitor and understand robot behavior. This dissertation conceptualizes robot programs as predictive models of robot tasks and proposes a neurosymbolic program representation that allows robots to learn and optimize their own behavior, while remaining parameterizable and interpretable by human users. It offers a blueprint for AI-enabled robot programming that bridges the apparent divides between symbolic and subsymbolic representations, designed and learned behavior, and capabilities and safety. The framework was evaluated on a range of robots and manipulation tasks from wire harness installation to kitchen clean-up and shopping assistance in retail stores.

Keywords Robot programming · Neurosymbolic AI · Program optimization · Program synthesis

1 Introduction

For robots to be useful assistants, they must be intelligent agents that can learn models of the world and optimize their own behavior. At the same time, human users need to understand and control their robotic assistants to ensure safe execution. *Programs* enable humans to communicate their intent to machines, and can be read and understood by users beyond the original programmer. Programming constructs that introduce symbolic abstractions, such as classes and functions, allow programmers to express *what* the machine should do rather than *how* it should do it. Deep learning over subsymbolic representations, on the other hand, allows robots to learn from unstructured data, yet makes it difficult for human programmers to understand and modify the learned behavior. This dissertation bridges the gap between subsymbolic and symbolic representations by casting robot programs as *predictive models* of robot tasks, and introduces a neurosymbolic program representation that affords learning, optimization, and human control (Sect. 2). Together with algorithms for program optimization (Sect. 3) and interactive program synthesis (Sect. 4), it realizes

AI-enabled robot programming that allows the use of subsymbolic learning even in safety-critical domains (Fig. 1).

2 Neurosymbolic Robot Programs

The central representational contribution are Neurosymbolic Robot Programs (NRPs). An NRP pairs a symbolic *source program* that specifies the robot's behavior in human-readable form, with a *shadow program*, a differentiable surrogate with partially learned components that mirrors the source program's structure and predicts its execution outcomes. Source programs are hierarchical compositions of parameterized *skills*, and the framework supports any underlying skill representation. For any given source program, its shadow program is automatically constructed. Shadow programs are differentiable computation graphs whose nodes (*shadow skills*) are either neural networks trained on real-world or simulated execution data, or differentiable path planners. Each shadow skill predicts the expected robot trajectory, including end-effector wrenches and gripper states, given skill parameters as input. Because shadow programs are differentiable, gradients of a task-level objective can be propagated through the entire program via backpropagation, making NRP parameters amenable to first-order optimization. The fundamental novelty of NRPs is that they can make *existing* robot programs optimizable.

✉ Benjamin Alt
benjamin.alt@uni-bremen.de

¹ AICOR Institute for Artificial Intelligence, University of Bremen, Bremen, Germany

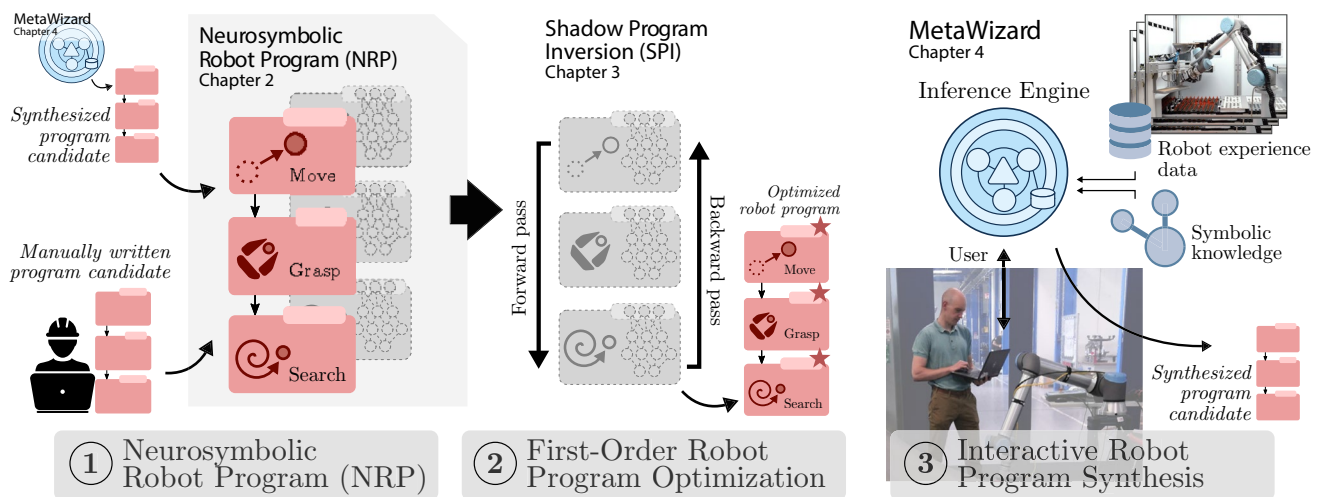


Fig. 1 NRPs associate differentiable surrogates (gray) to robot source programs (red), permitting learning and optimization across the full robot program lifecycle from programming to operation. Figure adapted from [1].

3 First-Order Robot Program Optimization

The dissertation introduces Shadow Program Inversion (SPI), a first-order optimizer for parameterized robot programs. SPI performs gradient-based optimization over the shadow program to find skill input parameters that minimize a task objective such as cycle time, task success, or distance to a reference trajectory, such as a human demonstration. Optimized parameters are written back to the source program for execution. Because SPI operates on the learned shadow program, it is applicable to tasks where analytical forward models are unavailable, such as contact-rich manipulation under process noise, as well as safety-critical tasks for which the certifiability of the source program matters.

SPI is extended in two directions. A *lifelong learning* scheme based on sequential transfer learning continuously updates shadow programs to reflect changing dynamics, enabling data-efficient adaptation. *SPI-DP* integrates a differentiable motion planner into the shadow program framework to perform double-loop gradient descent: an outer loop optimizes task-level parameters, while an inner loop optimizes motion trajectories with respect to collision avoidance, smoothness, and proximity to a human demonstration. SPI is the first gradient-based approach to optimize the program parameters of arbitrary, originally non-differentiable robot programs.

SPI has been shown to optimize force-controlled motion parameters to improve task success and reduce cycle time in several industrial assembly scenarios. SPI-DP plans collision-free pick-and-place trajectories one-shot from single human demonstrations and jointly optimizes trajectory and program parameters for a poka-yoke quality assurance task, improving task success by up to 186% and reducing cycle times by up to 75%.

4 Interactive Robot Program Synthesis

While SPI optimizes existing robot programs, the initial creation of robot source programs remains a challenge. This dissertation introduces the *MetaWizard* family of knowledge-driven program synthesis systems.

MetaWizard synthesizes NRP source programs by reasoning over a semisymbolic knowledge base combining foundational and application ontologies with a digital twin of the environment. Given a Virtual Reality (VR) demonstration of the task, it abstracts the intended actions by parsing the demonstrated force-dynamic events and grounds them in the digital twin to produce an executable source program. MetaWizard was evaluated on retail assistance tasks.

MetaWizard2 extends this toward interactive programming via natural-language dialogue. It models tasks as hierarchical *metatasks* and incrementally grounds them by proactively asking the user for information about the task. MetaWizard2 was validated on industrial surface treatment tasks. MetaWizardLLM, a third variant, leverages pretrained LLMs for task reasoning and was validated on industrial gear assembly. The core contribution of MetaWizard is the use of reasoning over hybrid knowledge bases that combine both symbolic and subsymbolic information. Like SPI, MetaWizard produces robot programs that afford deterministic behavior guarantees, making it suitable for safety-sensitive applications.

5 Conclusion and Further Work

The dissertation proposes a neurosymbolic framework for robot programming that bridges the divides between symbolic and subsymbolic representations, designed and learned

behavior, and capabilities and safety. SPI on NRPs enables gradient-based optimization while preserving symbolic structure for human control. MetaWizard demonstrates that structured knowledge representations support interactive program synthesis from intuitive user inputs.

NRP!s are particularly suited for industrial applications, as source programs remain auditable and engineers can continue using established programming toolchains. Conditioning shadow programs on environment images via multimodal transformer architectures [3] enables optimization with respect to current percepts without finetuning. Combining LLM reasoning with structured knowledge is a promising evolution of program synthesis [2]. More broadly, the interplay between optimization and synthesis, whereby synthesized programs are optimized and results inform future synthesis, points toward systems that continuously improve through experience.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability Much of the work of this dissertation is available open-source. Corresponding links are provided in [1].

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format,

as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alt B (2025) Neurosymbolic Robot Programming: A Framework for AI-Enabled Programming of Robot Manipulation Tasks. Ph.D. thesis, Bremen, Germany
2. Kienle C, Alt B, Arenz O, Peters J (2025) LODGE: joint hierarchical task planning and learning of domain models with grounded execution. <https://doi.org/10.48550/arXiv.2505.13497>
3. Kienle C, Alt B, Celik O, Becker P, Katic D, Jäkel R, Neumann G (2024) MuTT: A Multimodal Trajectory Transformer for Robot Skills. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 9644–9651. IEEE, Abu Dhabi, United Arab Emirates. <https://doi.org/10.1109/IROS58592.2024.10802198>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Benjamin Alt received his Ph.D. in Computer Science from the University of Bremen in 2025, supervised by Prof. Michael Beetz. He studied Computer Science and Political Science at the Karlsruhe Institute of Technology and the Institut d'Études Politiques de Paris. His research interests include robot programming, planning, and neurosymbolic AI.