

PRIOR ART REPORT

Application Title: Computer-Implemented Systems and Methods for Closed-Loop Optimization of Hybrid Compute Architectures Comprising Under-Specified Elements Using Language Model-Driven Search Space Shaping and Hardware-in-the-Loop Evaluation

Applicant: Kernel Keys LLC

Inventor(s): David H. Silver

Priority Application: 63/927,859 (November 30, 2025)

Report Date: December 26, 2025

EXECUTIVE SUMMARY

This report identifies and analyzes prior art relevant to the claimed invention, which covers:

1. Hardware-in-the-loop optimization of under-specified compute architectures
2. Language model-driven code and configuration generation
3. Dynamic search-space shaping based on physical hardware telemetry
4. Distributed artificial neuron nodes for robotic sensing and control

Conclusion: The identified prior art establishes context in (a) language model optimization for text/speech, (b) dynamic grammar models for constrained devices, (c) automatic code generation from models, and (d) hardware-in-the-loop simulation. However, **no prior art teaches or suggests** the combination of:

- Using language models to generate artifacts for *under-specified* hardware (analog, neuromorphic, approximate)
 - Modifying the optimization *search space itself* (IR operators, grammars, constraints) based on *physical* hardware telemetry
 - Deploying LM-optimized artifacts to distributed neuron-like nodes in robotic systems with per-node calibration and safety envelopes
-

CATEGORIZED PRIOR ART REFERENCES

Category 1: Language Model Optimization (Text/Speech Domain)

These references teach adaptation of language models or grammar models for improved text/speech recognition, but operate purely on symbolic/textual data—not hardware telemetry or under-specified compute.

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
1	US 20150325235 A1	Language Model Optimization for In-Domain Application	2015-11-12	Microsoft Technology Licensing LLC	Google Patents	Medium — Adapts LMs based on usage statistics for speech recognition. Does NOT address hardware telemetry or under-specified compute.
2	US 9,972,311 B2	Language Model Optimization for In-Domain Application	2018-05-15 (grant)	Microsoft Technology Licensing LLC	Google Patents	Medium — Granted version of above. Same distinctions apply.
3	WO 2015/171875 A1	Language Model Optimization for In-	2015-11-12	Microsoft Technology Licensing LLC	Google Patents	Low — PCT family of above.

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
		Domain Application				
4	EP 2026327 A1	Language Model Learning System, Language Model Learning Method, and Language Model Learning Program	2009-02-18	Emori, Tadashi; Onishi, Yoshifumi	Google Patents	Low — Language model learning for text processing. No hardware feedback.
5	EP 2996045 A1	Language Model with Structured Penalty	2016-03-16	Nelakanti et al.	Google Patents	Low — Structured regularization for LMs. Pure software optimization. Family includes US 20160070697 A1, US 9,684,650 B2.

Key Distinction: All references optimize LMs for textual/linguistic data. None addresses hardware-in-the-loop feedback from under-specified compute elements.

Category 2: Dynamic Grammar Models for Resource-Constrained Devices

These references teach updating grammar/language models on resource-constrained devices, but for speech/voice recognition—not hardware code generation or telemetry-driven optimization.

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
6	WO 2016/191313 A1	Dynamically Updatable Offline Grammar Model for Resource-Constrained Offline Device	2016-12-01	Google Inc.	Google Patents	High — Updates grammars on constrained devices based on usage. Does NOT modify grammar based on hardware telemetry or generate executable artifacts for under-specified hardware. Inventors: Sung, Gao, Murugesan.
7	US 20160350320 A1	Dynamically Updatable Offline Grammar Model for Resource-	2016-12-01	Google LLC	Google Patents	High — US publication of above.

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
		Constrained Offline Device				
8	US 9,922,138 B2	Dynamically Updatable Offline Grammar Model for Resource-Constrained Offline Device	2018-03-20 (grant)	Google LLC	Google Patents	High — Granted version. Teaches grammar updates on devices, but for voice recognition, not hardware optimization.
9	EP 3385946 A1	Dynamically Updatable Offline Grammar Model for Resource-Constrained Offline Device	2018-10-10	Google LLC	Google Patents	Medium — EP family. Same distinctions.
10	WO 2017/146803 A1	Facilitation of Offline Semantic Processing in a Resource-	2017-08-31	Google	Google Patents	Medium — Offline semantic processing on constrained devices. Inventors:

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
		Constrained Device				Gao, Sung, Moreno Mengibar. No hardware-in-the-loop optimization.

Key Distinction: These references update grammars for speech/voice recognition based on linguistic usage patterns. Our invention updates *intermediate representation operators, constraints, and generation priors* based on *physical hardware telemetry* from under-specified compute elements—a fundamentally different domain and mechanism.

Category 3: Automatic Code Generation and HDL Synthesis

These references teach automatic code/HDL generation from high-level models, but assume deterministic, fully-specified target architectures.

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
11	WO 2006/034352 A3	Automatic Generation of Code for Component Interfaces in Models	2006	The MathWorks, Inc.	Google Patents	High — Model-based code generation (Simulink/Stateflow style). Assumes deterministic target semantics.

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
						Does NOT address under-specified hardware or hardware-in-the-loop calibration.
12	WO 2008/033344 A3	Hardware Definition Language Generation for Frame-Based Processing	2008	The MathWorks, Inc.	Google Patents	High — HDL generation from models. Assumes fully-specified hardware behavior. No per-device calibration or telemetry feedback.
13	EP 0709773 A1	System and Method for Generating Target Language Code Utilizing an Object Oriented Code Generator	1996	(check EPO)	Google Patents	Medium — Early OO code generation patent. Deterministic compilation model.

Key Distinction: Model-based code generation and HDL synthesis assume the target hardware has fully deterministic, simulable semantics. Our invention explicitly addresses *under-specified* architectures where the functional mapping varies per device and must be characterized through hardware-in-the-loop evaluation—rendering traditional compilation approaches inapplicable.

Category 4: AutoML and Neural Architecture Search

These references teach automated optimization of neural architectures, but assume deterministic, simulable hardware execution.

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
14	WO 2024/156237 A1	Neural Architecture Search Optimization	2024	(check WIPO)	Google Patents	Medium — AutoML/NAS for neural networks. Assumes deterministic training and inference. Does NOT address per-device calibration of under-specified hardware.
15	EP 4339843 A1	Automated Machine Learning Optimization	2023	(check EPO)	Google Patents	Medium — AutoML optimization. No hardware-in-the- loop feedback from analog/

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
						neuromorphic elements.

Key Distinction: AutoML and NAS optimize models *before* deployment based on simulated or deterministic training runs. Our invention performs *continuous* optimization *on physical hardware* whose behavior cannot be accurately simulated, with per-device calibration and telemetry-driven search-space adaptation.

Category 5: Hardware-in-the-Loop Systems (Control/Power)

These references teach hardware-in-the-loop testing for control systems and power electronics, but do not use language models or dynamically reshape search spaces.

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
16	WO 2020/102450 A1	Decentralized Hardware-in-the-Loop Scheme / HIL Simulation Systems	2020	(check WIPO)	Google Patents	High — HIL simulation for control systems. Uses fixed parameterization, not LM-driven generation. Does NOT modify search space based on telemetry.
17	US 10,971,931 B2	Hardware-in-the-Loop Testing System	2021-04-06 (grant)	(check USPTO)	Google Patents	High — HIL for power systems testing. Fixed test configurations,

Ref #	Publication	Title	Pub Date	Patentee	Link	Relevance
						not adaptive optimization with LMs.

Key Distinction: Traditional HIL systems evaluate fixed configurations or controller parameterizations. They do not employ language models to generate code/configuration artifacts, and they do not dynamically reshape the optimization search space (operators, grammars, constraints) based on telemetry. Our invention uses the LM as a *meta-optimizer* that modifies the search space itself.

Category 6: Robotic Sensing and Control (Non-Patent Literature)

These references teach robotic sensing systems but lack LM-driven optimization and per-node calibration.

Ref #	Publication	Title	Year	Relevance
18	Dahiya et al.	"Tactile sensing—from humans to humanoids," IEEE Trans. Robotics, 26(1)	2010	Survey of tactile sensing. No LM-driven optimization or distributed node calibration.
19	Various	Distributed robotic skin sensing literature	Various	General distributed sensing approaches. Centralized processing or hand-coded distributed logic—not adaptive LM-optimized per-node artifacts.

Key Distinction: Existing robotic sensing uses centralized processing (high latency) or hand-coded distributed logic (inflexible). Our invention provides LM-driven generation and

adaptation of per-node artifacts with hardware-in-the-loop calibration compensating for manufacturing variability, event-based messaging, bounded-latency reflexes, and safety envelopes with rollback.

NON-PATENT LITERATURE (NPL) REFERENCES

Ref #	Author(s)	Title	Publication	Year	Relevance
NPL-1	Mead, C.	Analog VLSI and Neural Systems	Addison-Wesley	1989	Foundational analog computing. Context for under-specified compute.
NPL-2	Shen et al.	Deep learning with coherent nanophotonic circuits	Nature Photonics, 11(7), pp. 441-446	2017	Photonic computing with device variability. Context for under-specified compute.
NPL-3	Aho, Lam, Sethi, Ullman	Compilers: Principles, Techniques, and Tools (2nd Ed.)	Pearson	2006	Traditional compilation. Assumes deterministic ISA—our invention addresses systems where this

Ref #	Author(s)	Title	Publication	Year	Relevance
					assumption fails.
NPL-4	Lattner & Adve	LLVM: A Compilation Framework for Lifelong Program Analysis and Transformation	Proceedings of CGO	2004	LLVM framework. Assumes semantic preservation—undefined for under-specified hardware.
NPL-5	Chen et al.	Evaluating Large Language Models Trained on Code	arXiv:2107.03374	2021	Codex LLM code generation. Does not address hardware telemetry feedback.
NPL-6	Li et al.	Competition-level code generation with AlphaCode	Science 378(6624)	2022	Advanced LLM code generation. No hardware-in-the-loop adaptation.
NPL-7	Scholak et al.	PICARD: Parsing Incrementally for Constrained Auto-	EMNLP	2021	Grammar-constrained decoding. Pure software domain.

Ref #	Author(s)	Title	Publication	Year	Relevance
		Regressive Decoding			
NPL-8	Poesia et al.	Synchromesh: Reliable code generation from pre-trained language models	ICLR	2022	Constrained LLM decoding. No hardware feedback.
NPL-9	Hutter et al.	Automated Machine Learning: Methods, Systems, Challenges	Springer	2019	AutoML survey. Assumes deterministic execution.
NPL-10	Dahiya et al.	Tactile sensing —from humans to humanoids	IEEE Trans. Robotics, 26(1)	2010	Robotic tactile sensing survey. No LM-driven optimization.
NPL-11	Davies et al.	Loihi: A neuromorphic manycore processor with on-chip learning	IEEE Micro, 38(1)	2018	Intel Loihi neuromorphic chip. Context for under-specified compute.
NPL-12	Merolla et al.	A million spiking-neuron integrated circuit with a scalable	Science 345(6197)	2014	IBM TrueNorth. Context for neuromorphic

Ref #	Author(s)	Title	Publication	Year	Relevance
		communication network and interface			compute variability.
NPL-13	Yang et al.	Memristive devices for computing	Nature Nanotechnology, 8(1)	2013	Memristor computing with device variation. Context for under-specified compute.

SUMMARY OF DISTINCTIONS FROM PRIOR ART

Distinction 1: Target Domain (Under-Specified Compute)

Prior art in LM optimization, grammar adaptation, and code generation assumes fully deterministic, simulable target architectures. Our invention explicitly targets **under-specified compute elements** (analog, neuromorphic, approximate, stochastic) whose behavior varies per device and cannot be accurately modeled in software.

Distinction 2: Hardware-in-the-Loop with LM Generation

Traditional HIL systems (WO 2020/102450, US 10,971,931) evaluate fixed configurations. Our invention combines:

- **LM-driven artifact generation** (code + configuration)
- **Physical hardware execution** (not simulation)
- **Telemetry-driven adaptation** of the generation strategy

Distinction 3: Search-Space Shaping (Key Innovation)

No prior art teaches the LM as a **meta-optimizer** that modifies:

- Intermediate representation operators
- Grammar production rules

- Constraints and safety bounds
- Generation priors

...based on patterns discovered in **physical hardware telemetry**. This differs fundamentally from prior dynamically updatable grammar models (US 9,922,138, EP 3385946, WO 2016/191313), which adapt grammars for speech or text recognition rather than for generating executable artifacts for under-specified hardware. This enables discovery of hardware-specific optimizations not expressible in the initial formulation.

Distinction 4: Distributed Robotic Neuron Nodes

Prior robotic sensing uses centralized processing or fixed distributed logic. Existing HIL and code-generation techniques (WO 2020/102450, US 10,971,931, WO 2006/034352) do not describe LM-optimized, telemetry-driven calibration of distributed neuron-like nodes with bounded-latency reflex behavior and per-node safety constraints. Our invention provides:

- LM-optimized per-node artifacts
 - Per-node calibration for manufacturing variability
 - Event-based messaging with rate limiting
 - Bounded-latency local reflexes
 - Safety envelopes with rollback
-

CONCLUSION

The prior art establishes that language model optimization, dynamic grammar adaptation, automatic code generation, AutoML, and hardware-in-the-loop testing are known techniques in their respective domains. However, **no prior art teaches or suggests** the claimed combination of:

1. Using a language model to generate candidate artifacts (code + configuration) for **under-specified compute elements** whose behavior must be characterized on physical hardware
2. Dynamically **reshaping the optimization search space** (IR operators, grammar rules, constraints, priors) based on **physical hardware telemetry**—enabling discovery of optimizations not expressible in the original formulation
3. Deploying LM-optimized artifacts to **distributed artificial neuron nodes** in robotic systems with per-node calibration, event-based messaging, bounded-latency reflexes, and safety envelopes

The claimed invention represents a non-obvious combination of techniques from disparate fields (language models, hardware-in-the-loop testing, grammar-guided generation, robotic sensing) applied to a novel problem domain (programming under-specified compute) with novel mechanisms (search-space shaping from hardware telemetry).

REFERENCES LIST (For IDS Form PTO/SB/08)

U.S. Patent Documents

Cite #	Document Number	Publication Date	First Named Inventor / Patentee	Title
US1	US 20150325235 A1	2015-11-12	Levit et al. / Microsoft	Language Model Optimization for In-Domain Application
US2	US 9,972,311 B2	2018-05-15	Levit et al. / Microsoft	Language Model Optimization for In-Domain Application
US3	US 20160350320 A1	2016-12-01	Sung et al. / Google	Dynamically Updatable Offline Grammar Model for Resource-Constrained Offline Device
US4	US 9,922,138 B2	2018-03-20	Sung et al. / Google	Dynamically Updatable Offline Grammar Model for Resource-Constrained Offline Device
US5	US 10,971,931 B2	2021-04-06	(verify USPTO)	Hardware-in-the-Loop Testing System

Foreign Patent Documents

Cite #	Document Number	Country	Publication Date	Patentee/Applicant	Title
FP1	WO 2015/171875 A1	WO	2015-11-12	Microsoft	Language Model Optimization for In-Domain Application
FP2	WO 2016/191313 A1	WO	2016-12-01	Google Inc.	Dynamically Updatable Offline Grammar Model for Resource-Constrained Offline Device
FP3	WO 2017/146803 A1	WO	2017-08-31	Google	Facilitation of Offline Semantic Processing in a Resource-Constrained Device
FP4	WO 2006/034352 A3	WO	2006	The MathWorks, Inc.	Automatic Generation of Code for Component Interfaces in Models
FP5	WO 2008/033344 A3	WO	2008	The MathWorks, Inc.	Hardware Definition Language Generation for Frame-Based Processing
FP6		WO	2020		

Cite #	Document Number	Country	Publication Date	Patentee/Applicant	Title
	WO 2020/102450 A1			(verify WIPO)	Decentralized Hardware-in-the-Loop Scheme
FP7	WO 2024/156237 A1	WO	2024	(verify WIPO)	Neural Architecture Search Optimization
FP8	EP 2026327 A1	EP	2009-02-18	Emori, Onishi	Language Model Learning System
FP9	EP 2996045 A1	EP	2016-03-16	Nelakanti et al.	Language Model with Structured Penalty
FP10	EP 3385946 A1	EP	2018-10-10	Google LLC	Dynamically Updatable Offline Grammar Model
FP11	EP 4339843 A1	EP	2023	(verify EPO)	Automated Machine Learning Optimization
FP12	EP 0709773 A1	EP	1996	(verify EPO)	Object-Oriented Code Generator

Non-Patent Literature

Cite #	Author(s)	Title	Publication	Date
NPL1	Mead, C.	Analog VLSI and Neural Systems	Addison-Wesley	1989
NPL2	Shen et al.	Deep learning with coherent nanophotonic circuits	Nature Photonics 11(7):441-446	2017

Cite #	Author(s)	Title	Publication	Date
NPL3	Aho, Lam, Sethi, Ullman	Compilers: Principles, Techniques, and Tools (2nd Ed.)	Pearson	2006
NPL4	Lattner & Adve	LLVM: A Compilation Framework for Lifelong Program Analysis and Transformation	CGO Proceedings	2004
NPL5	Chen et al.	Evaluating Large Language Models Trained on Code	arXiv:2107.03374	2021
NPL6	Li et al.	Competition-level code generation with AlphaCode	Science 378(6624)	2022
NPL7	Scholak et al.	PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding	EMNLP	2021
NPL8	Poesia et al.	Synchromesh: Reliable code generation from pre-trained language models	ICLR	2022
NPL9	Hutter et al.	Automated Machine Learning: Methods, Systems, Challenges	Springer	2019
NPL10	Dahiya et al.	Tactile sensing—from humans to humanoids	IEEE Trans. Robotics 26(1)	2010
NPL11	Davies et al.	Loihi: A neuromorphic manycore processor with on-chip learning	IEEE Micro 38(1)	2018
NPL12	Merolla et al.	A million spiking-neuron integrated circuit...	Science 345(6197)	2014

Cite #	Author(s)	Title	Publication	Date
NPL13	Yang et al.	Memristive devices for computing	Nature Nanotech. 8(1)	2013

QUICK REFERENCE: PATENT LINKS

For examiner convenience (Google Patents direct links):

Document	Link
US 20150325235 A1	https://patents.google.com/patent/US20150325235A1
US 9,972,311 B2	https://patents.google.com/patent/US9972311B2
WO 2015/171875 A1	https://patents.google.com/patent/WO2015171875A1
WO 2016/191313 A1	https://patents.google.com/patent/WO2016191313A1
US 20160350320 A1	https://patents.google.com/patent/US20160350320A1
US 9,922,138 B2	https://patents.google.com/patent/US9922138B2
EP 3385946 A1	https://patents.google.com/patent/EP3385946A1
WO 2017/146803 A1	https://patents.google.com/patent/WO2017146803A1
WO 2006/034352 A3	https://patents.google.com/patent/WO2006034352A3
WO 2008/033344 A3	https://patents.google.com/patent/WO2008033344A3
EP 0709773 A1	https://patents.google.com/patent/EP0709773A1
WO 2020/102450 A1	https://patents.google.com/patent/WO2020102450A1
US 10,971,931 B2	https://patents.google.com/patent/US10971931B2
EP 2026327 A1	https://patents.google.com/patent/EP2026327A1

Document	Link
EP 2996045 A1	https://patents.google.com/patent/EP2996045A1
WO 2024/156237 A1	https://patents.google.com/patent/WO2024156237A1
EP 4339843 A1	https://patents.google.com/patent/EP4339843A1

Report prepared by: Kernel Keys LLC

Date: December 26, 2025