

CLAIMS

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

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

What is claimed is:

Independent Claims (4 Total)

Claim 1 — Hardware-in-the-Loop Co-Optimization Method

1. A computer-implemented method for optimizing a hybrid compute system comprising at least one deterministic digital processing core and at least one under-specified compute element, wherein the under-specified compute element exhibits device-dependent, time-varying functional behavior that is not accurately captured by a static simulation model, the method comprising:
 - (a) receiving, by one or more processors, a specification of the under-specified compute element, the specification comprising a set of configurable parameters and an indication that the functional mapping varies across devices due to at least one of: manufacturing variability, environmental drift, component tolerances, or intrinsic stochasticity;
 - (b) generating, by a language model executing on the one or more processors, a candidate artifact using grammar-constrained decoding, wherein the candidate artifact specifies at least one of executable code for the deterministic digital processing core and configuration data including device-specific calibration parameters for the under-specified compute element;
 - (c) deploying the candidate artifact to the physical hybrid compute system and executing at least a portion of a workload on the physical hardware rather than on a simulation;

- (d) collecting telemetry data from the physical hybrid compute system during execution, the telemetry data comprising at least two of: power consumption, execution latency, output accuracy, error rate, temperature, temporal drift metrics, or cross-execution variance metrics;
 - (e) computing a score for the candidate artifact using a learned evaluator model executing on or near the hybrid compute system;
 - (f) updating a generation strategy of the language model based on the computed score and the collected telemetry, wherein updating comprises modifying at least one of: a grammar mask specifying allowed production rules, operator usage priors, hardware-conditioned context tokens encoding quantized telemetry features, or sampling parameters; and
 - (g) upon satisfaction of a termination criterion, outputting a deployable artifact that satisfies predefined constraints and is specific to the physical device on which telemetry was collected.
-

Claim 2 — Search-Space Shaping Method (Key Differentiator)

1. A computer-implemented method for dynamically modifying an optimization search space during hardware-in-the-loop optimization of an under-specified compute system, wherein the search space modifications are driven by a language model analyzing physical hardware telemetry, the method comprising:
 - (a) defining, by one or more processors, an initial search space comprising an intermediate representation (IR) with a finite set of operators, a context-free grammar specifying valid artifact structures, and a set of constraints bounding resource usage and safety properties;
 - (b) generating, by a language model, a plurality of candidate artifacts that are valid instances of the grammar;
 - (c) deploying the plurality of candidate artifacts to a physical target compute substrate comprising at least one under-specified element and executing workloads thereon;
 - (d) collecting performance telemetry from the physical target compute substrate during execution;
 - (e) analyzing, by the language model, patterns in the collected telemetry across the plurality of candidate artifacts;
 - (f) generating, by the language model based on the analysis, a proposed modification to the search space, the proposed modification comprising at least one of:

- (i) adding a new composite operator to the IR encapsulating a pattern discovered in high-performing candidates,
 - (ii) removing an operator associated with low-performing or unstable candidates,
 - (iii) adding a new grammar production rule enabling previously inexpressible structures,
 - (iv) tightening constraints in regions of feature space with unstable telemetry, or
 - (v) shifting prior probability distributions toward high-performing patterns;
 - (g) validating the proposed modification by generating test candidates, verifying safety constraints, and evaluating on physical hardware;
 - (h) upon successful validation, adopting the modification to form an updated search space; and
 - (i) generating subsequent candidates within the updated search space, whereby the process discovers hardware-specific patterns not expressible in the initial search space and not discoverable by simulation.
-

Claim 3 — System Claim

1. A system for optimizing hybrid compute architectures comprising under-specified elements, the system comprising:
 - (a) a hybrid compute substrate comprising a deterministic digital core and at least one under-specified compute element exhibiting device-dependent behavior not fully characterized by a deterministic model;
 - (b) instrumentation configured to collect telemetry from the hybrid compute substrate during physical execution;
 - (c) a candidate generator comprising a language model configured to generate candidate artifacts using grammar-guided decoding;
 - (d) a search controller configured to guide optimization using at least one of reinforcement learning, Bayesian optimization, or evolutionary search, adapting based on physical hardware telemetry;
 - (e) an on-device evaluator model trained online using discrepancies between predicted and measured metrics;
 - (f) a constraint enforcer configured to verify resource, timing, and safety constraints; and

(g) a deployment component configured to output deterministic runtime artifacts with rollback capability.

Claim 4 — Robotic System with LM-Optimized Distributed Neuron Nodes

1. A robotic system comprising:

- (a) a plurality of artificial neuron nodes (ANN-nodes) physically distributed over a robot body, each ANN-node comprising at least one sensor, a local compute element, and a communication interface;
 - (b) a communication bus coupling the ANN-nodes to at least one hub controller, supporting event-based messaging and rate limiting;
 - (c) an optimization system comprising a language model configured to generate per-node artifacts specifying code, configuration, calibration parameters, or message-passing rules, and to adapt artifacts using search-space shaping based on telemetry from physical robot operation;
 - (d) instrumentation configured to collect telemetry from the ANN-nodes during physical operation; and
 - (e) a safety enforcement subsystem configured to verify safety envelopes before deployment, bound actuator commands and message rates, and trigger rollback upon constraint violation.
-

Dependent Claims (26 Total)

Claims Dependent on Claim 1 (HIL Method)

- 1. The method of claim 1, wherein the under-specified compute element comprises at least one of: an analog circuit with component tolerances exceeding $\pm 5\%$, a memristor-based crossbar array, a neuromorphic circuit, an approximate arithmetic unit, a stochastic logic element, a near-threshold circuit, or a photonic processor.
- 2. The method of claim 1, wherein the language model is a transformer-based model with fewer than 50 million parameters.

3. The method of claim 1, wherein updating the generation strategy comprises updating a grammar mask that disables production rules associated with candidates exhibiting high telemetry variance or constraint violations.
 4. The method of claim 1, wherein the learned evaluator model is trained online during the optimization process using discrepancies between predicted scores and measured performance.
 5. The method of claim 1, wherein the hardware-conditioned context tokens encode at least one of: quantized temperature, supply voltage, detected drift magnitude, or device age estimate.
 6. The method of claim 1, wherein the deployable artifact comprises bytecode for the deterministic core, configuration bits for the under-specified element, per-device calibration parameters, and a safety envelope specification.
 7. The method of claim 1, further comprising verifying safety constraints by invoking a formal verification engine selected from model checking or SMT solving before deployment.
 8. The method of claim 1, wherein the language model is a quantized model stored and executed on the same microcontroller as the deterministic core, enabling fully on-device adaptation.
 9. The method of claim 1, further comprising initializing priors for a new device using telemetry and artifacts from a previously calibrated device of the same manufacturing batch, enabling transfer learning.
 10. The method of claim 1, wherein the hybrid compute system is part of a robotic system and the telemetry includes embodied task performance metrics.
-

Claims Dependent on Claim 2 (Search-Space Shaping)

1. The method of claim 2, wherein the new composite operator encapsulates a subgraph comprising at least three primitive operators that co-occur in high-performing candidates.
2. The method of claim 2, wherein tightening constraints specifically targets regions of feature space where telemetry indicates thermal instability or output variance exceeding a threshold.

3. The method of claim 2, wherein the grammar modification enables expression of hardware-specific idioms discovered through telemetry analysis that have no analog in standard programming patterns.
 4. The method of claim 2, wherein the intermediate representation comprises operators for: sensor reading, actuator control, temporal integration, event detection with adaptive thresholds, and sparse message passing.
 5. The method of claim 2, wherein the validation rejects modifications that cause any test candidate to violate safety constraints, even if mean performance improves.
 6. The method of claim 2, wherein the search-space shaping enables discovery of optimizations that exploit device-specific quirks not documented in device specifications.
-

Claims Dependent on Claim 3 (System)

1. The system of claim 3, wherein the under-specified compute element comprises an analog neural network accelerator with per-device synaptic weight variation exceeding 10%.
 2. The system of claim 3, wherein the constraint enforcer employs SMT solving to prove absence of unbounded loops, stack overflow, and forbidden state reachability.
 3. The system of claim 3, wherein the deployment component produces versioned artifacts with automatic rollback upon detecting performance degradation or safety violation.
-

Claims Dependent on Claim 4 (Robotic System)

1. The robotic system of claim 4, wherein the ANN-nodes are implemented on a flexible substrate and distributed conformally over a robotic gripper, limb, or skin.
2. The robotic system of claim 4, wherein each ANN-node transmits event messages using spike encoding representing threshold crossings with compressed sensory representations.
3. The robotic system of claim 4, wherein at least one node artifact defines a bounded-latency reflex action executed locally within 10 milliseconds without requiring round-trip communication to the hub controller.

4. The robotic system of claim 4, wherein the optimization system learns per-node calibration parameters compensating for manufacturing variability in sensor sensitivity, mounting-dependent signal coupling, or aging-induced drift.
 5. The robotic system of claim 4, wherein the safety enforcement subsystem bounds actuator commands to $F_{\max} < 2N$ per node and caps message frequency to prevent bus congestion.
 6. The robotic system of claim 4, wherein the node artifacts implement leaky integrate-and-fire neuron dynamics with learned time constants.
 7. The robotic system of claim 4, wherein the language model-driven search-space shaping discovers message-passing patterns and local reflex rules not specified in the initial artifact grammar.
-

Abstract of the Disclosure

A computer-implemented system and method for optimizing hybrid compute architectures comprising deterministic digital cores and under-specified compute elements whose functional behavior is device-dependent and not accurately captured by simulation. The system employs a language model to generate candidate artifacts, deploys candidates to physical hardware, collects telemetry during execution, and iteratively refines generation based on hardware-in-the-loop feedback. A key innovation is search-space shaping, wherein the language model dynamically modifies intermediate representation operators, grammar rules, and constraints based on patterns discovered in physical hardware telemetry—enabling discovery of hardware-specific optimizations not expressible in the initial formulation. The system produces device-specific deployable artifacts meeting resource, timing, and safety constraints. In robotic embodiments, optimized artifacts are deployed to distributed artificial neuron nodes with per-node calibration, event-based messaging, bounded-latency reflexes, and safety envelopes with rollback capability.