Research article walkthrough · Embedded cyber-physical control and SoS engineering

Supervisory Small Language Models for Hardware Signal Control and System-of-Systems Optimization

A multi-tier adaptive control, hyper-heuristic, and cross-layer protocol architecture



Hybrid adaptive control theory | Edge TinyML / SSLMs | HIL signal tokenization | Distributed energy and fieldbus

A visual guide to the corrected paper: 6 sections, 6 hypotheses, 22 references

The engineering conflict the paper sets out to resolve

Classical control (PID, MPC)

Strength
Deterministic, microsecond execution on bare-metal processors.

Gap
Cannot re-mathematize its control laws on the fly when global collaborative constraints shift or unmodeled physical wear appears.

The paper's answer

A domain-specific SLM as a supervisory hardware-in-the-loop controller

Its native vocabulary is the device's own signals and communication frames. It synthesizes parameters for classical loops instead of driving pins, inside a multi-agent system of systems.

Standalone cloud LLMs

Strength
Adaptive, context-aware semantic reasoning.

Gap
Millisecond-to-second latency variance, tight edge memory and power, and the risk of hallucinated register instructions.

Model size in scope: quantized SLMs of 500M to 3B parameters, embedded in hardware state machines, MCUs and SoCs as Hybrid Adaptive Controllers.

Lamaakal et al., 2025; Xia et al., 2024; Wagner et al., 2024; Maher, 2025; Wang et al., 2026

Paper at a glance

What the paper contains, section by section

1 Introduction and hypotheses

Why embed SLMs in control hardware, and six falsifiable hypotheses (H1 to H6).

2 Generative control frameworks

CELI, ELF, edge compression mathematics, cascaded loops, HIL benchmarking.

3 Supervisory hybrid adaptive controller

Three-tier edge mesh, L(x) vs G(x) objectives, the deterministic guardrail interlock.

4 HTM, ETL pipeline, metaheuristics

Signals to tokens, hysteresis, training-data taxonomy, the SLM supervising PSO and GA.

5 Solar farm implementation

Three node classes, the node control loop, five micro-scale energy capture paradigms.

6 Cognitive OSI and protocol SLM

7-layer cross-layer optimization, zero-trust detection, Modbus/CANopen/MQTT gateway.

Section 7 lists 22 references (corrected list). Every claim in this deck comes from the paper; later slides cite its sources.

Six falsifiable hypotheses frame the study

H1 · Cognitive-deterministic decoupling

A 500M to 1.5B SLM, walled off from 10 to 100 kHz pins by a C/C++ guardrail, tunes PID, MPPT and current limits every 100 ms to 1 s with zero safety violations and instant fallback.

H2 · Local L(x) vs global G(x)

Local self-preservation lives in the system prompt; global goals arrive as Policy Envelopes. The node arbitrates, and falls back to local survival if comms drop.

H3 · Signal as language

Registers split into Magnitude, Δ/Δt and FFT THD tokens. Dwell and deadband hysteresis stop token thrashing; states compress to 16-bit hex IDs.

H4 · Hyper-heuristic supervision

The SLM bounds the search space, adapts fitness weights and switches solvers over PSO and GA, avoiding local optima and speeding convergence.

H5 · Cognitive OSI co-optimization

All 7 OSI layers plus Modbus-TCP, CANopen and MQTT as one context window via eBPF: reroute ahead of EMI, 80 to 95% compression, spoof detection.

H6 · Micro-scale energy capture

Correlating Layer 1 micro-amps, impedance and switching noise with Layer 7 forecasts recovers energy that classical controllers leave behind.

Sha, 2001; Wang et al., 2026; van Stein & Bäck, 2025; Narimani & Emami, 2026; Soldani et al., 2023; Cooper et al., 2025; Zhang et al., 2021

CELI and ELF: two ways to put control inside the model

CELI · Controller-Embedded Language Model Interactions

Wagner et al., 2024

Conventional

Hard-coded external loop

LLM as passive text generator

CELI

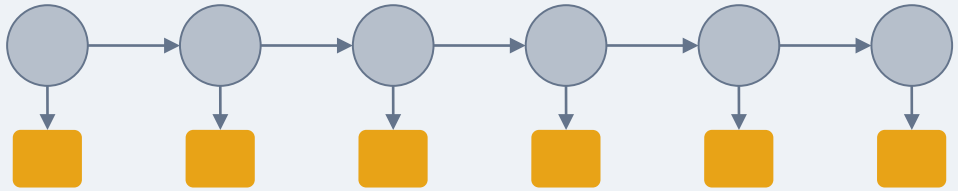
Control loop lives inside the prompt architecture; the model runs multi-stage execution itself

1. **Autonomous state management:** tracks workflow transitions without external scaffolding
2. **Arbitrary function calling:** maps semantic states to tool, register or API actions with validated arguments
3. **State-aware efficiency:** caches immutable action paths; passes only delta state tokens

ELF · Embedded Language Flows

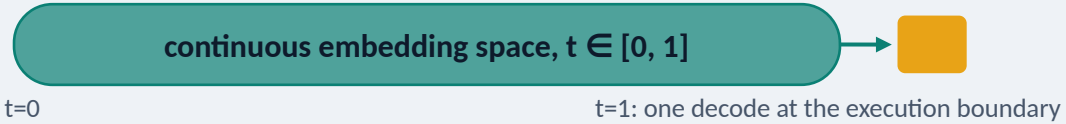
Hu et al., 2026

Autoregressive decoding



Projects to a discrete token at every step: quantization noise and latency at the physical interface

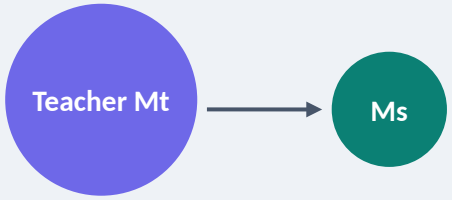
ELF continuous flow



Flow matching replaces token-by-token generation, so live sensor feedback can join the generative trajectory smoothly.

Four pillars for fitting a transformer onto edge silicon

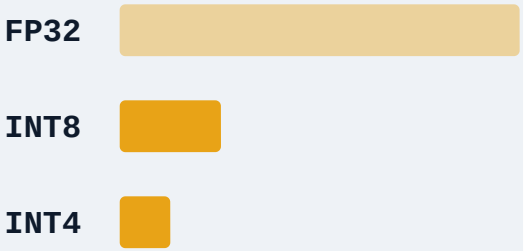
Knowledge distillation



$$L_{total} = (1 - \alpha) \cdot L_{task} + \alpha \cdot T^2 \cdot KL(P_t \parallel P_s)$$

Large teacher's domain reasoning moves into a compact edge student, with softened outputs at temperature T.

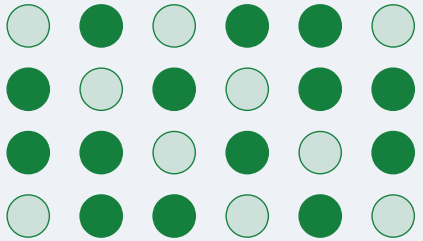
INT4 / INT8 quantization



bit width per weight, to scale

Surveyed figures: over 50% memory reduction and over 2× faster inference with minimal accuracy loss.

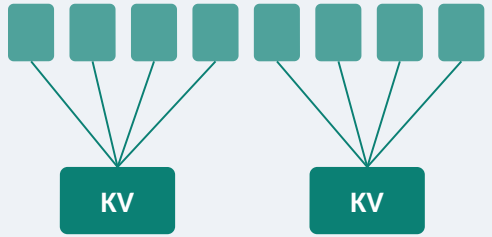
Structured and magnitude pruning



$$w_p = \begin{cases} 0 & \text{if } |w| < \tau \\ w & \text{else} \end{cases}$$

Removes redundant weights, attention heads or channels below a sensitivity threshold τ, giving sparse, low-latency graphs.

Memory-efficient attention



8 query heads share 2 key-value caches

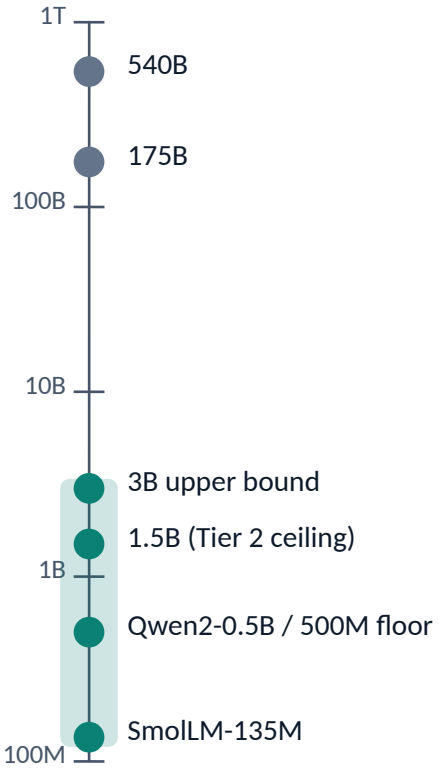
GQA or MLA replaces multi-head attention; with RoPE and RMSNorm it bounds runtime SRAM on edge NPUs.

Lamaakal et al., 2025; Hinton et al., 2015; Scherer et al., 2024; Ainslie et al., 2023; Vaswani et al., 2017

Edge SLMs and cloud models solve different problems

Feature	On-device TinyML / edge SLMs	Cloud-augmented foundation models
Model size	500M to 3B parameters (down to <1B nano-models)	Hundreds of billions (e.g. 175B to 540B)
Hardware targets	ARM Cortex-M, ESP32, STM32MP1, Raspberry Pi 5, local NPUs	Cloud GPU and TPU clusters
Constraints	4 to 8 GB DRAM down to MB-scale SRAM	High compute, distributed VRAM, high power
Optimization	INT4/INT8 quantization, pruning, distillation	None required on the client edge
Core benefit	Zero latency variance, privacy, offline, low power	Broad multi-domain reasoning

Parameter counts named in the paper (log scale)



Lamaakal et al., 2025 (table and nano-model examples as given in the paper)

Small models handle the routine; hard cases cascade upward

Cascaded control loops

On-device nano model (<1B)

Rapid sensory filtering, immediate safety heuristics, routine token tasks

`Qwen2-0.5B` `MobiLlama-0.5B` `SmolLM-135M/360M` `OpenELM-270M`

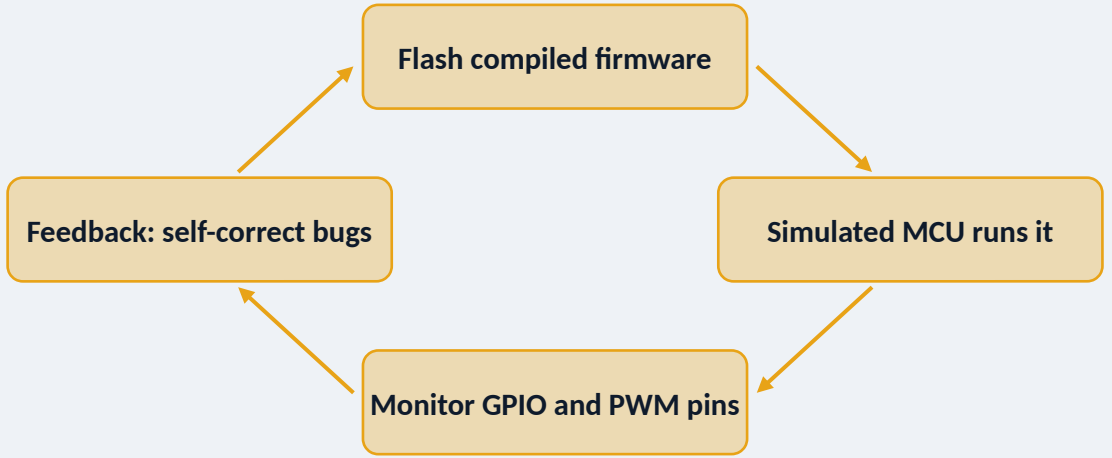
Out-of-distribution exception or system-wide diagnostic

Edge server or cloud-augmented SLM

Resolves high-level planning, then hands results back down

Hardware-in-the-loop development

Closed-loop virtual hardware testing, e.g. Wokwi



EmbedCopilot-Bench: vision-language models judged on hardware-aware development across circuit layouts, schematics and register configurations. Standard code benchmarks miss these electrical and timing variables.

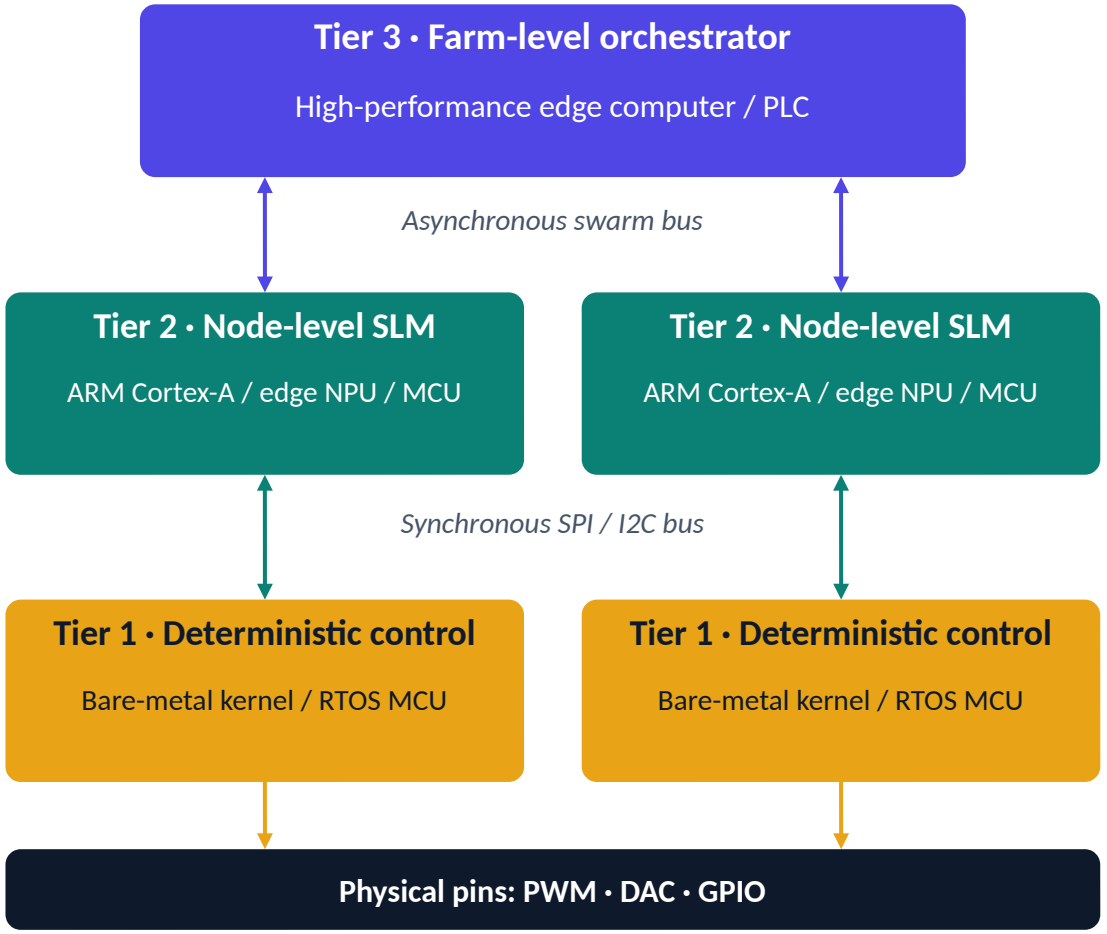
Lamaakal et al., 2025; Yuan et al., 2026

A three-tier hierarchical edge mesh

Global optimization

Context-aware parameters

Microsecond hardware I/O



Tier 3 · multi-second horizon

Industrial PC, PLC or HA edge server. Owns G(x), e.g. farm yield vs battery wear under live grid prices; aggregates Tier 2 state tokens and broadcasts Policy Envelopes.

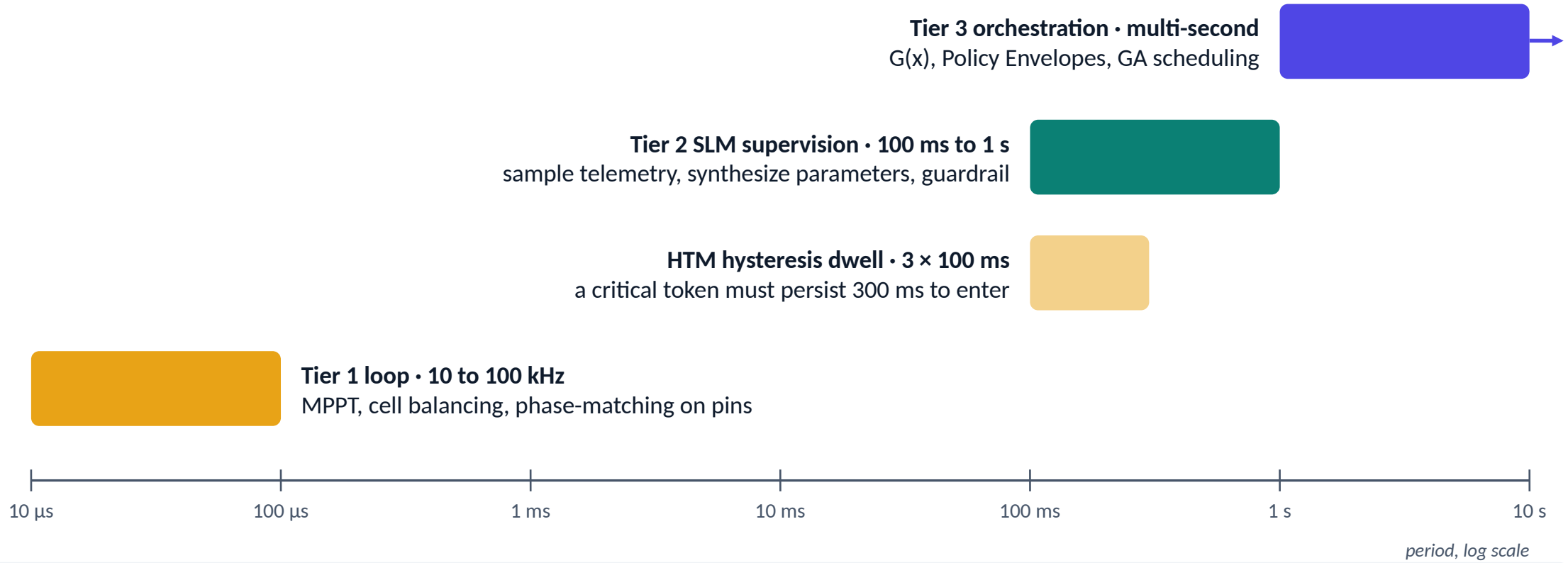
Tier 2 · 100 ms to 1 s

i.MX8M Plus, STM32MP1, RPi CM4 or edge TPU on real-time Linux. Hosts the 500M to 1.5B INT4 SLM, samples Tier 1 telemetry and refines its parameter targets.

Tier 1 · 10 to 100 kHz

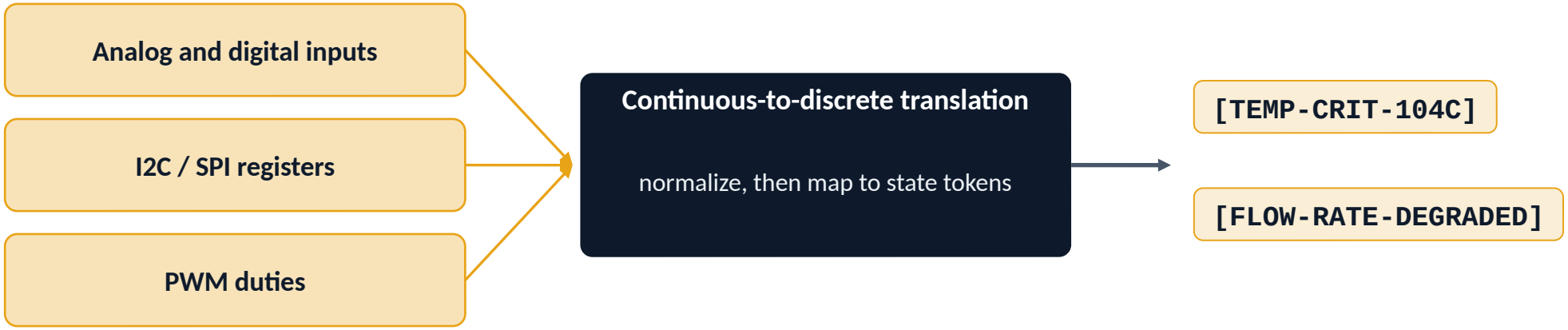
STM32 or TI C2000 DSP, bare metal or RTOS. Runs MPPT, cell balancing and inverter phase-matching. Never runs the SLM; consumes static parameters and writes pins.

Each tier runs on its own clock

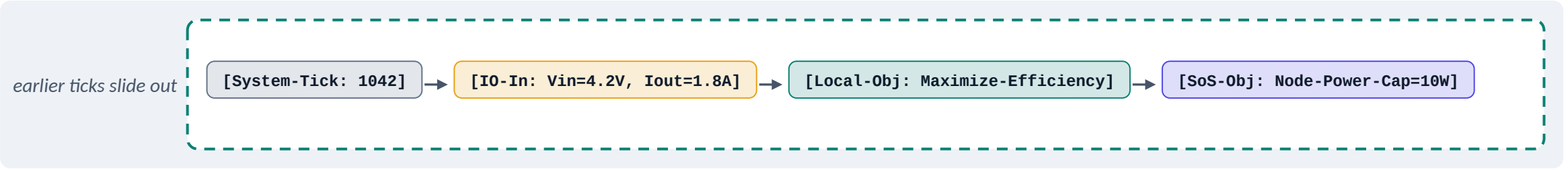


The SLM never sits inside the 10 to 100 kHz loop: three to five orders of magnitude separate Tier 1 from Tier 2.

Hardware I/O becomes a token string the model can read

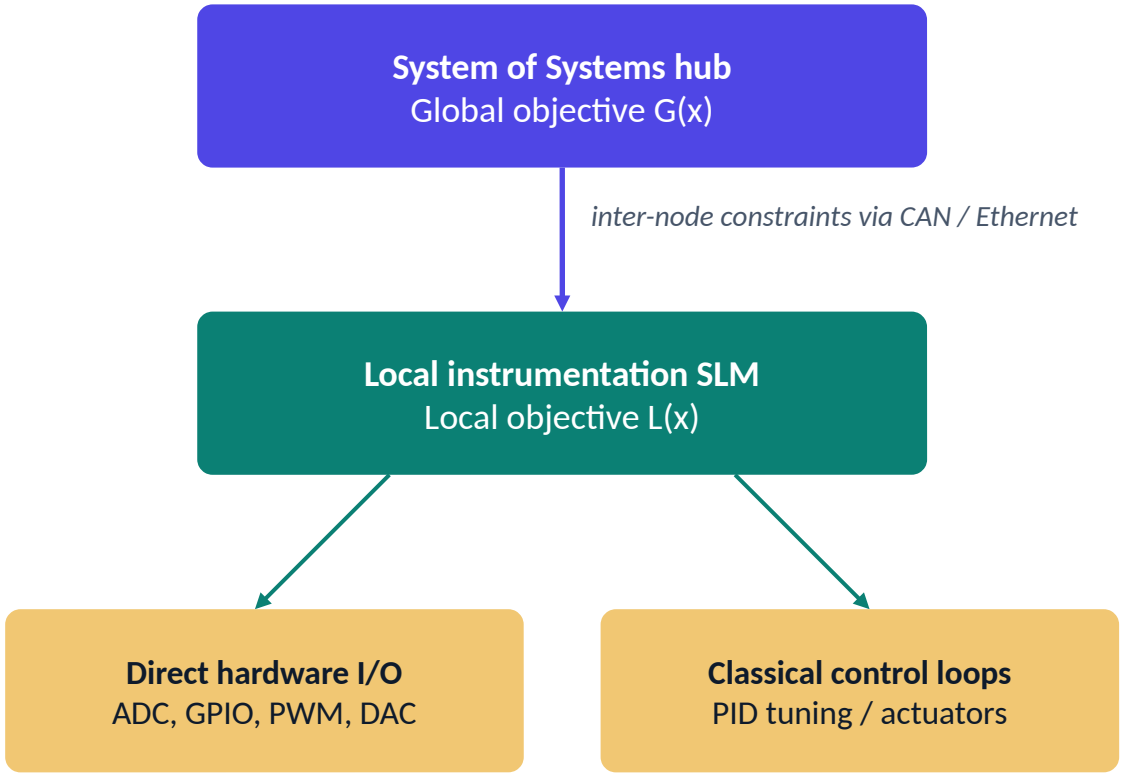


The model treats recent I/O history as a sliding context window



Example tokens and sequence exactly as written in the paper (§3.2, p. 7).

Two objective functions, one hierarchy



L(x) · local objective

Hardcoded as primary attention priorities in the Tier 2 system prompt. Guarantees stability, limits mechanical and thermal wear, meets immediate goals such as pump throughput or avoiding thermal throttling.

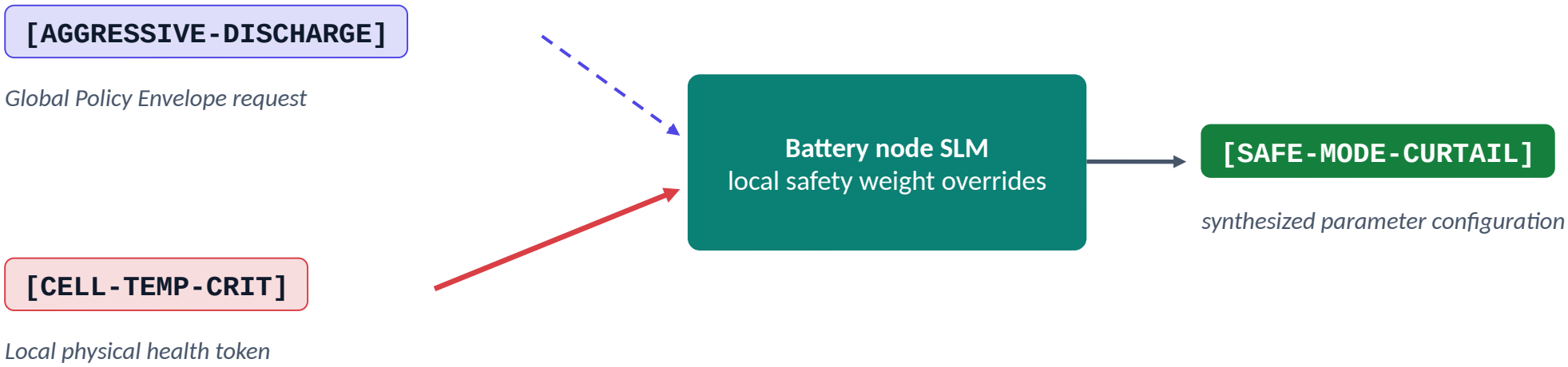
G(x) · global SoS objective

Formulated at Tier 3 from grid pricing, load forecasts and weather. Broadcast as Policy Envelopes over CAN, EtherCAT, Modbus-TCP or ROS2:

```
[MAX-BATTERY-RESERVE-DISCHARGE]
```

```
[LOAD-SHED-PRIORITY-HIGH]
```

When local health and global demand collide, local safety wins



The paper's balancing example · Global SoS objective: maximize total system efficiency and revenue

Node 1 · Battery storage

Local target: minimize degradation

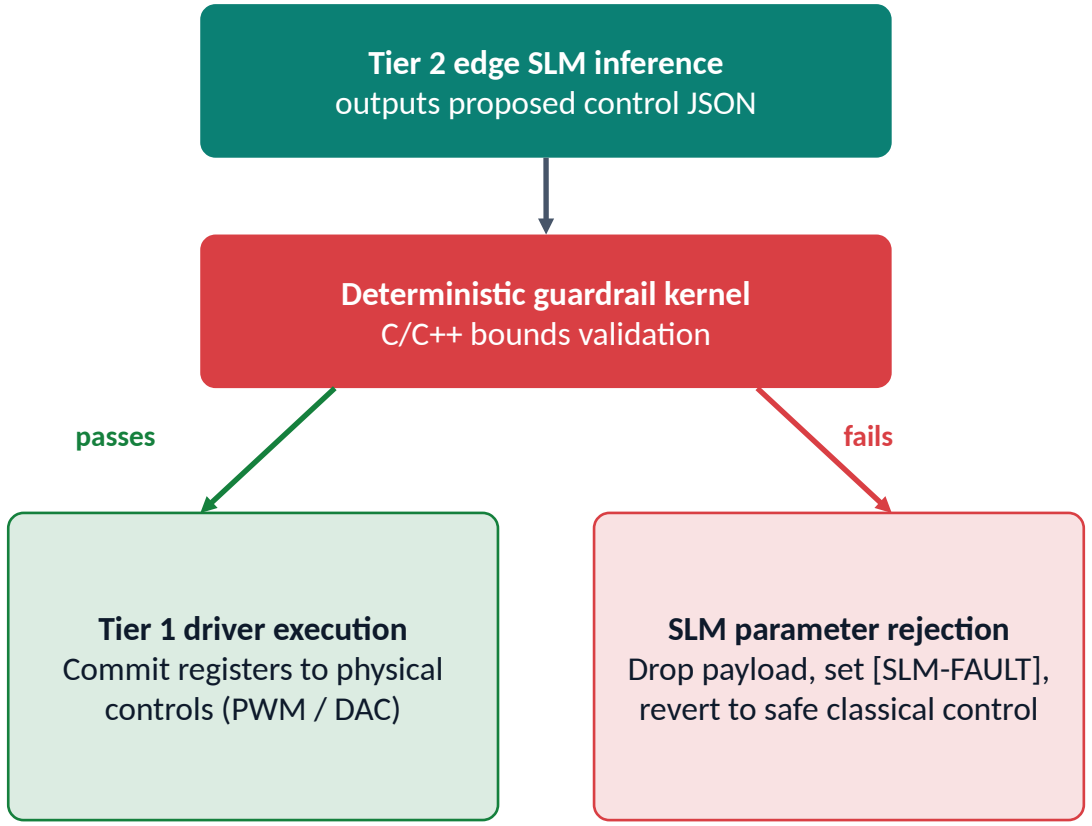
Balanced state: dictates max discharge

Node 2 · Appliance demand

Local target: user comfort

Balanced state: defers HVAC cycle

The interlock: every proposal passes a deterministic kernel



1 **State assessment**

Read the sliding window of tokenized I/O registers plus active Tier 3 SoS constraints.

2 **Parameter synthesis**

Synthesize gains and structure for Tier 1, e.g. Kp, Ki, Kd, an MPPT algorithm, or Lyapunov-based adaptive gains.

3 **Execution guardrails**

A compiled C/C++ verification matrix or bounded SMT verifier checks every value against physical limits.

4 **Execute or fall back**

Valid values go to Tier 1 registers. Out-of-bounds or corrupt output is dropped until the next window.

Example limits from the paper: tracker angle never past the limit switches; current never above wire-gauge ampacity. Sha, 2001; Mohan et al., 2013; Nesti et al., 2026; Maher, 2025; Bosio et al., 2025

Language-model supervision vs traditional MPC / PID

	Language-model-driven supervisory control	Traditional control (MPC / PID)
Complex SoS adaptability	High Translates vague or changing collaborative objectives into working parameters on the fly.	Low Needs heavy re-mathematization if global constraints change drastically.
Handling unmodeled dynamics	High Uses structural semantic reasoning to work around unforeseen sensor faults or wear.	Medium Relies on precise, predetermined models of the physical world.
Execution latency	Slower Varying and slower: milliseconds to seconds at supervisory cadence on an edge NPU.	Fast Deterministic microseconds on the bare-metal loop.

Ratings restate the paper's table (High / Medium / Low and latency) as bars; bar for latency shows speed, so more is faster.

The HTM turns volts into tokens in three steps

1 Telemetry capture
continuous voltages

V	432.1 V
I	12.4 A
Temp	52 C
Wind	45 km/h

2 Semantic map
state slicing

430–440 V →
[VOLT-NOMINAL]

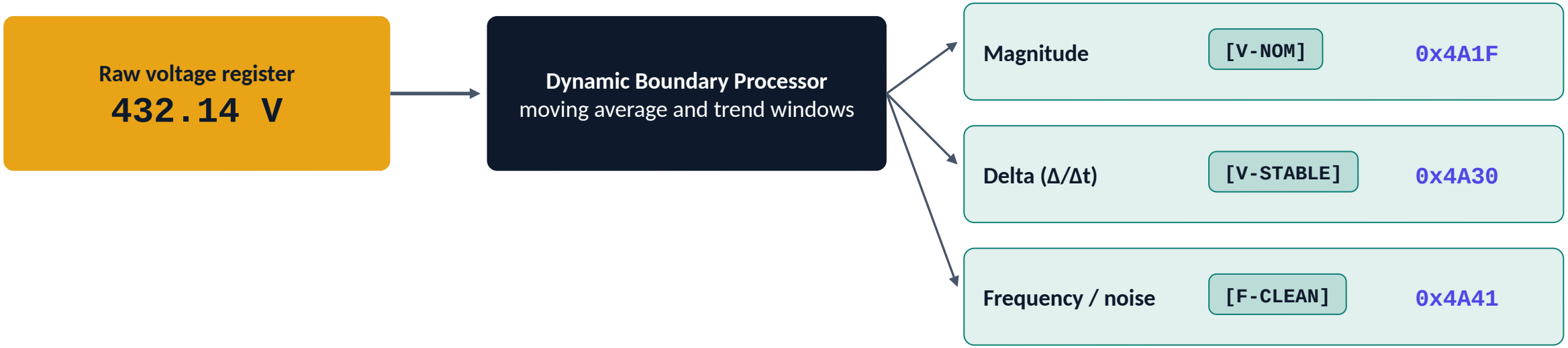
>40 km/h →
[WIND-GUST]

3 Token compilation
SLM ingestion

[INV-01]
[VOLT-NOMINAL]
[WIND-GUST]
[THERM-HIGH]

Where it runs: a deterministic firmware layer inside the Tier 2 processor, acting as an ADC-to-semantic encoder. It reads high-rate signals from an onboard ring buffer and emits structured token arrays.

One register becomes three orthogonal tokens



Why three vectors and hex IDs

Splitting value, trend and spectral quality lets the SLM tell a steady overvoltage from a fast-rising one or a noisy one, which a single magnitude token cannot.

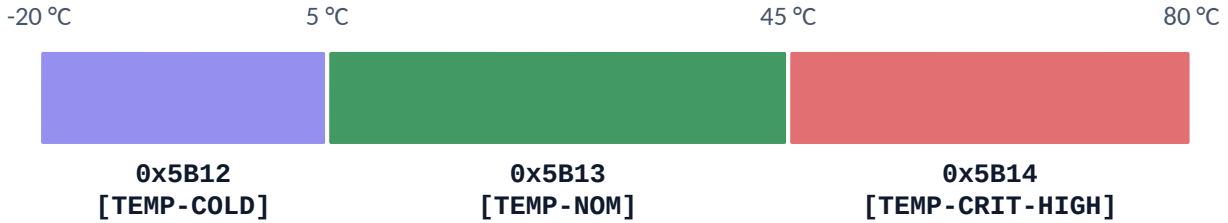
Each discrete token carries a unique 16-bit hexadecimal identifier, so states serialize compactly across fieldbus wires.

The token map: thresholds and hex IDs

DC bus voltage magnitude (0 to 600 V)



Battery cell temperature (−20 to 80 °C)



DC bus ΔV/Δt

0x4A30	[V-STABLE]	within ±0.5 V/s
0x4A31	[V-SPIKE-POS]	> 5.0 V/s
0x4A32	[V-DROP-NEG]	< −5.0 V/s

DC bus FFT ripple (10 kHz)

0x4A41	[F-CLEAN]	THD < 3%
0x4A42	[F-HARMONIC-DIST]	THD ≥ 3%

Battery ΔT/Δt

0x5B25	[THERM-EQUILIBRIUM]	≤ 0.1 °C/min
0x5B26	[THERM-RUNAWAY-RISK]	> 2.0 °C/min

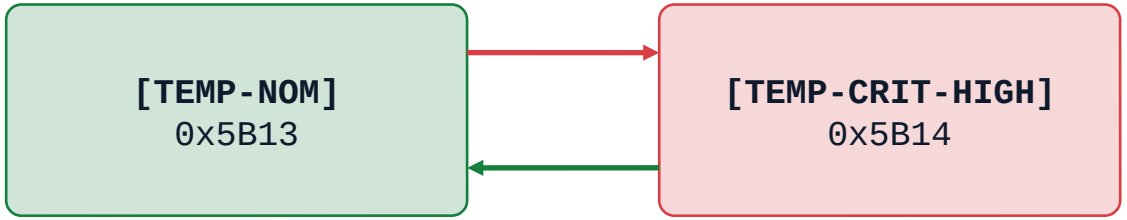
All codes, ranges and thresholds as listed in the paper's HTM table (§4.1, p. 11).

Dwell time and a deadband stop token thrashing

The problem

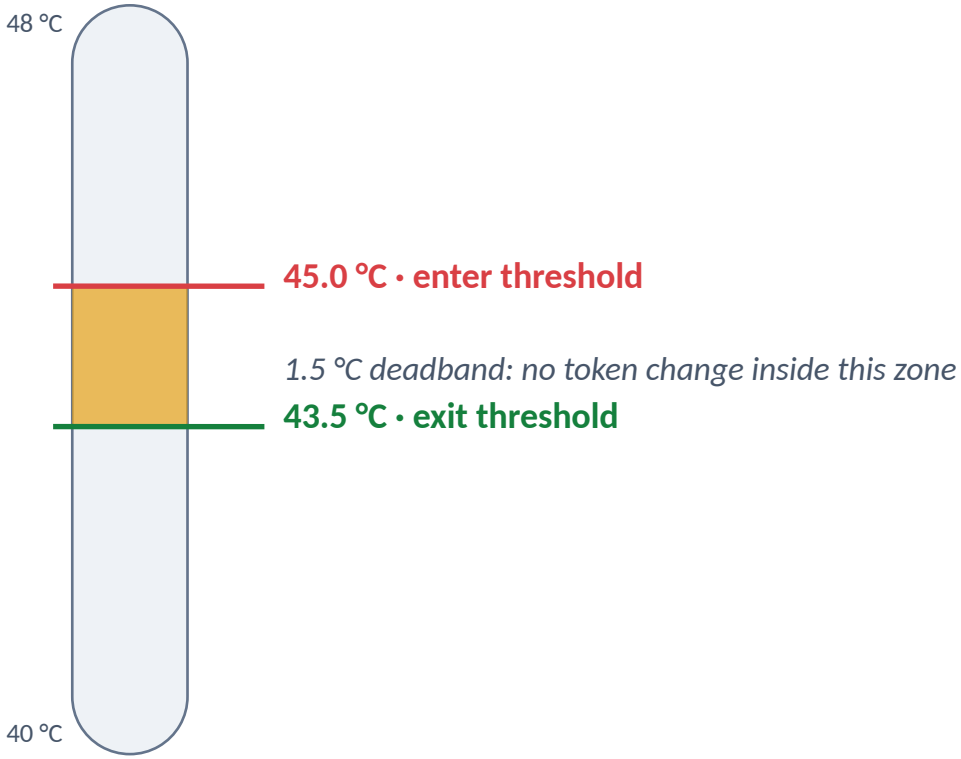
A signal hovering at a boundary, say 429.9 V to 430.1 V, would flood the context window with alternating state tokens.





Enter: cross 45.0 °C and stay there for ≥ 3 consecutive 100 ms samples (300 ms dwell)

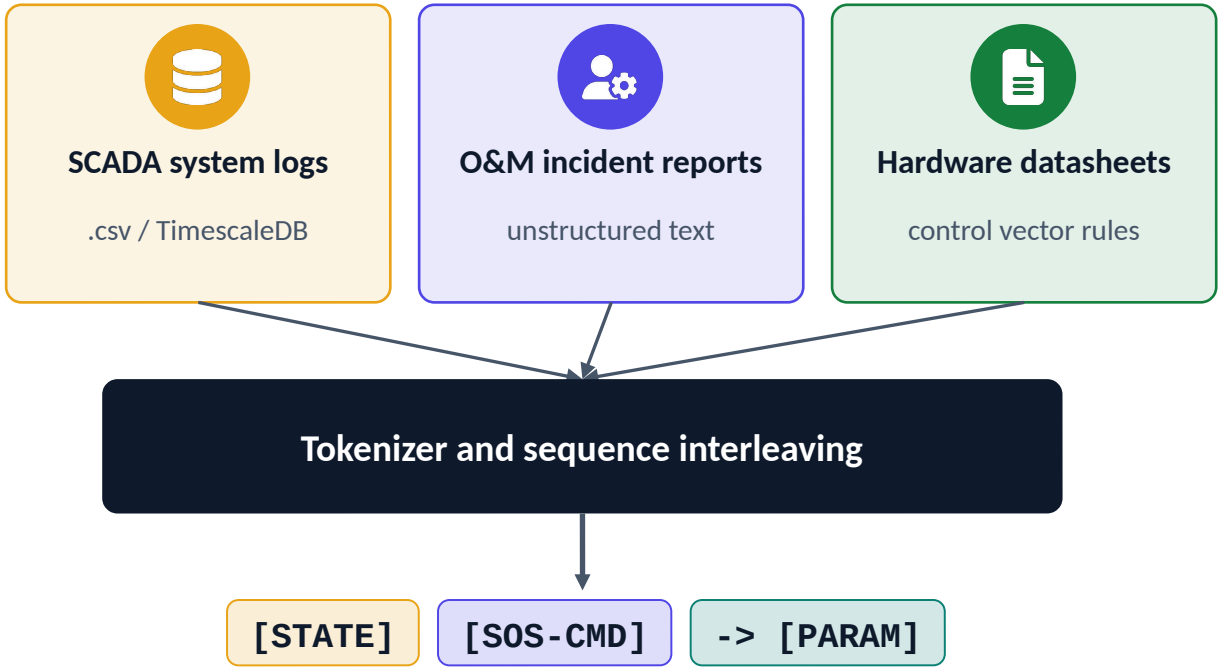
Exit: fall below 43.5 °C, a 1.5 °C deadband



Every compiled payload follows one syntax:

`[SYSTEM-CONTEXT]` `[GLOBAL-OBJECTIVE-TOKEN]`
`[LOCAL-TELEMETRY-TOKENS]` `→ [INFERENCE-TARGET]`

Training data: control-state engineering textures, not web text



token stream that teaches causal links between signal signatures, SoS commands and safe parameters

Pillar 1 · Industrial telemetry interleaving

SCADA archives, historians (TimescaleDB) and inverter logs; the HTM turns floating-point time series into chronological state-token transitions.

Pillar 2 · Hardware domain context

Datasheets, battery thermal manuals, firmware register maps and IEEE 1547 compliance protocols, parsed into structured markdown rule sets.

Pillar 3 · Operator-in-the-loop interventions

Maintenance logs and override tickets, e.g. retarding DC bus engagement 15 minutes when Inverter 4 shows a leakage spike under morning dew.

A four-block reasoning format: justify, then emit registers

[STATE]

`[NODE: INV-04]` `[V-NOM]` `[V-STABLE]` `[TEMP-CRIT-HIGH]` `[THERM-RUNAWAY-RISK]` `[GRID-FREQ: 60.1Hz]`

[SOS-CONTEXT]

`[SOS-PRIORITY: CAPACITY-MAXIMIZATION]` `[MARKET-PRICE: PEAK-SURGE]` `[THERM-CEILING-ALLOWED: TRUE]`

[COGNITIVE-REASONING-CHAIN]

Inverter near its thermal limit, but the SoS window authorizes a higher ceiling. Temperature is rising steeply, so mitigate without a hard shutdown: fans to maximum and gate-drive switching from 16 kHz down to 8 kHz to cut switching losses.

[TARGET-PARAMETERS-SYNTHESIS]

```
{ "fan-duty-pct": 100, "switching-freq-hz": 8000, "mppt-tracking-mode": "CONSERVATIVE" }
```

The worked example

16 → 8 kHz

gate-drive switching frequency

100%

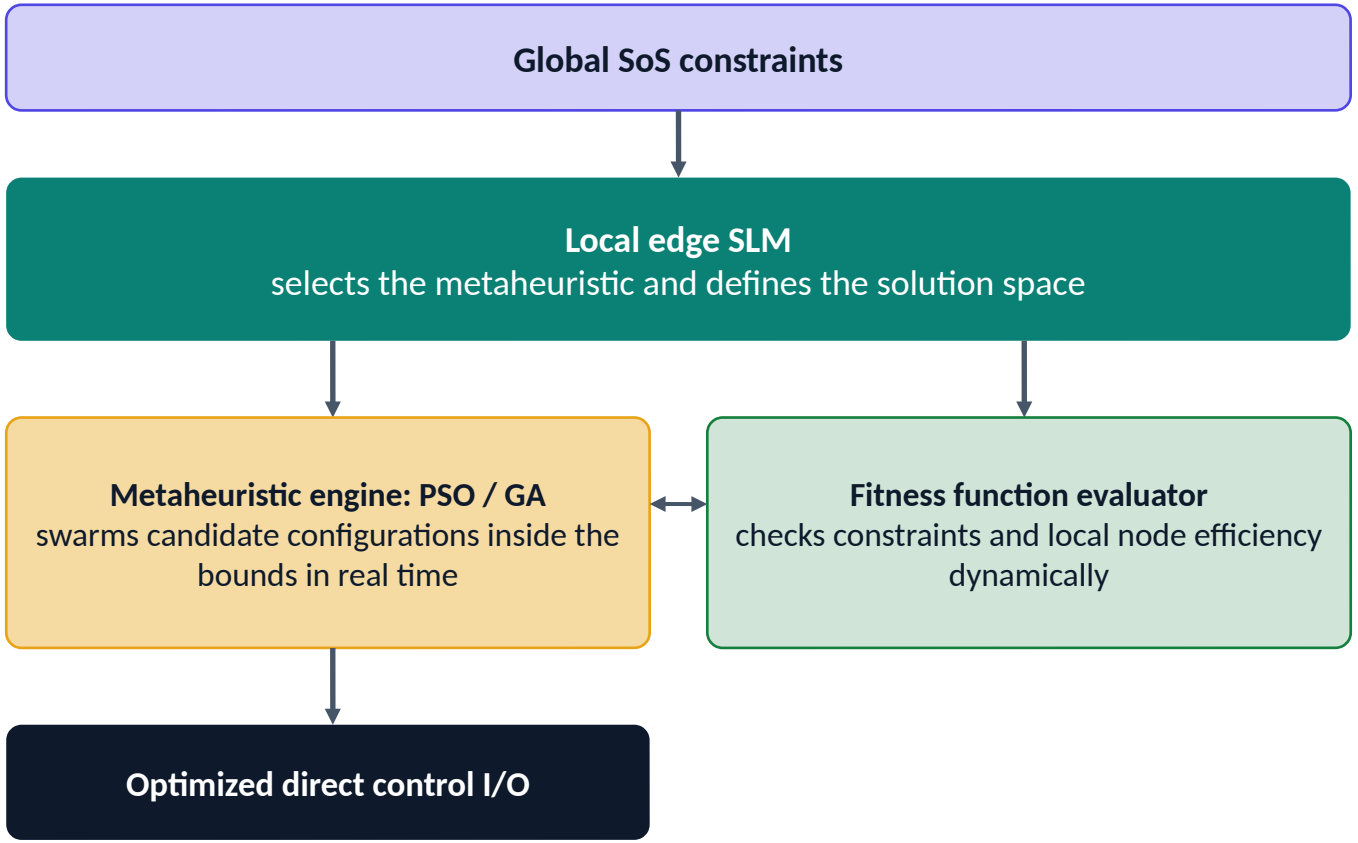
internal fan duty cycle

94%

power injection throughput the example says is maintained

A training-sample illustration from the paper (§4.2, p. 12 to 13), not a measured result.

The SLM supervises the optimizers rather than optimizing itself



PSO at Tier 2 (each node)

Continuous parameter tuning, e.g. finding the multi-peaked global maximum power point on a partially shaded string during cloud transients.

GA at Tier 3 (whole farm)

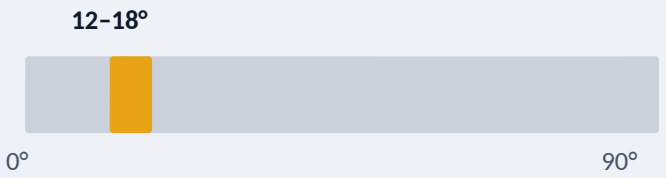
Discrete-event scheduling, e.g. battery discharge sequencing across 50 nodes over a 24-hour horizon.

van Stein & Bäck, 2025; Narimani & Emami, 2026

Three levers the SLM pulls on its solvers

1 Dynamic solution-space bounding

`[CLOUD-SHADING-FRAGMENTED]`



The SLM caps the tracker tilt search to a 12° to 18° band instead of sweeping the full 90° horizon, avoiding local trapping and wasted cycles.

2 Fitness function adaptation

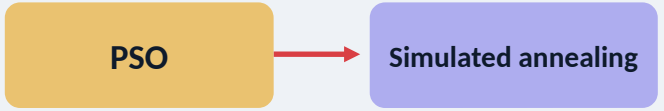
`[MODE: BAT-CONSERVE]`



schematic: grey before, green after

On a battery-conserve SoS token, the SLM rewrites fitness penalty weights so searches heavily penalize battery temperature spikes.

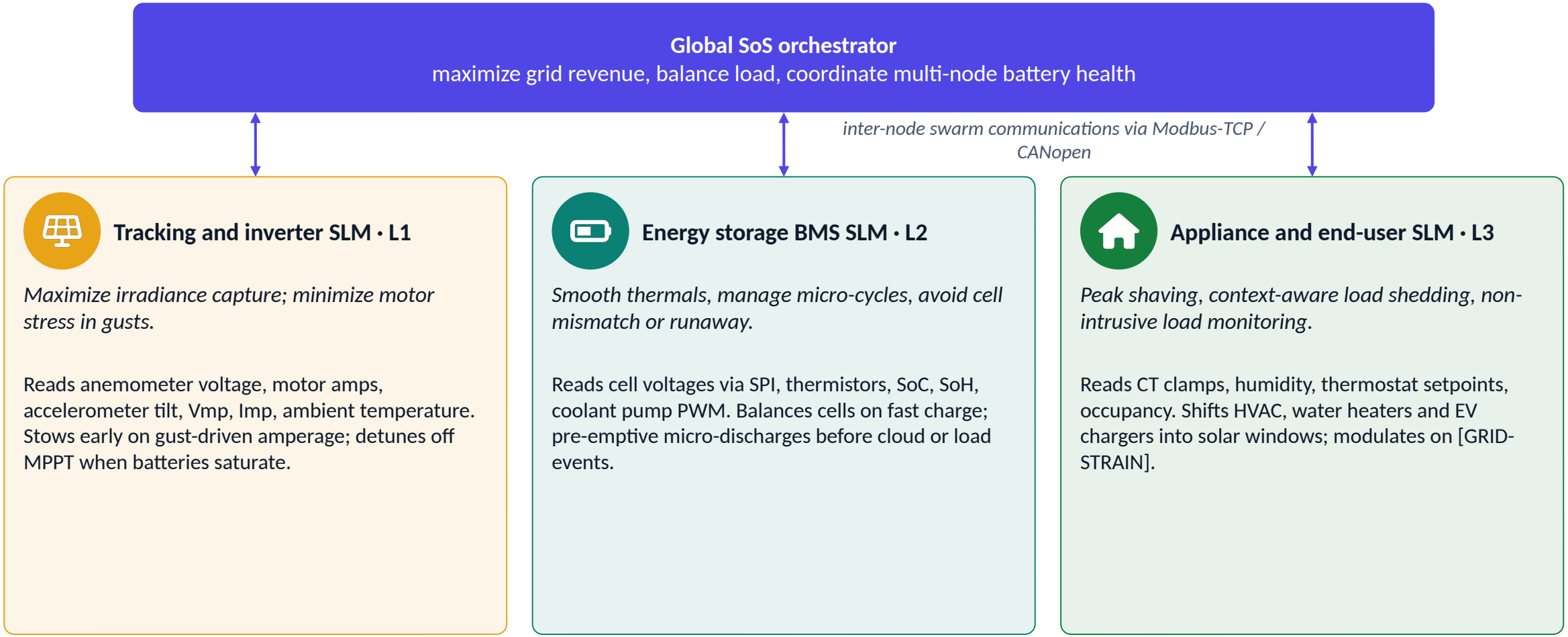
3 Algorithmic switching



no convergence within the timeout token window

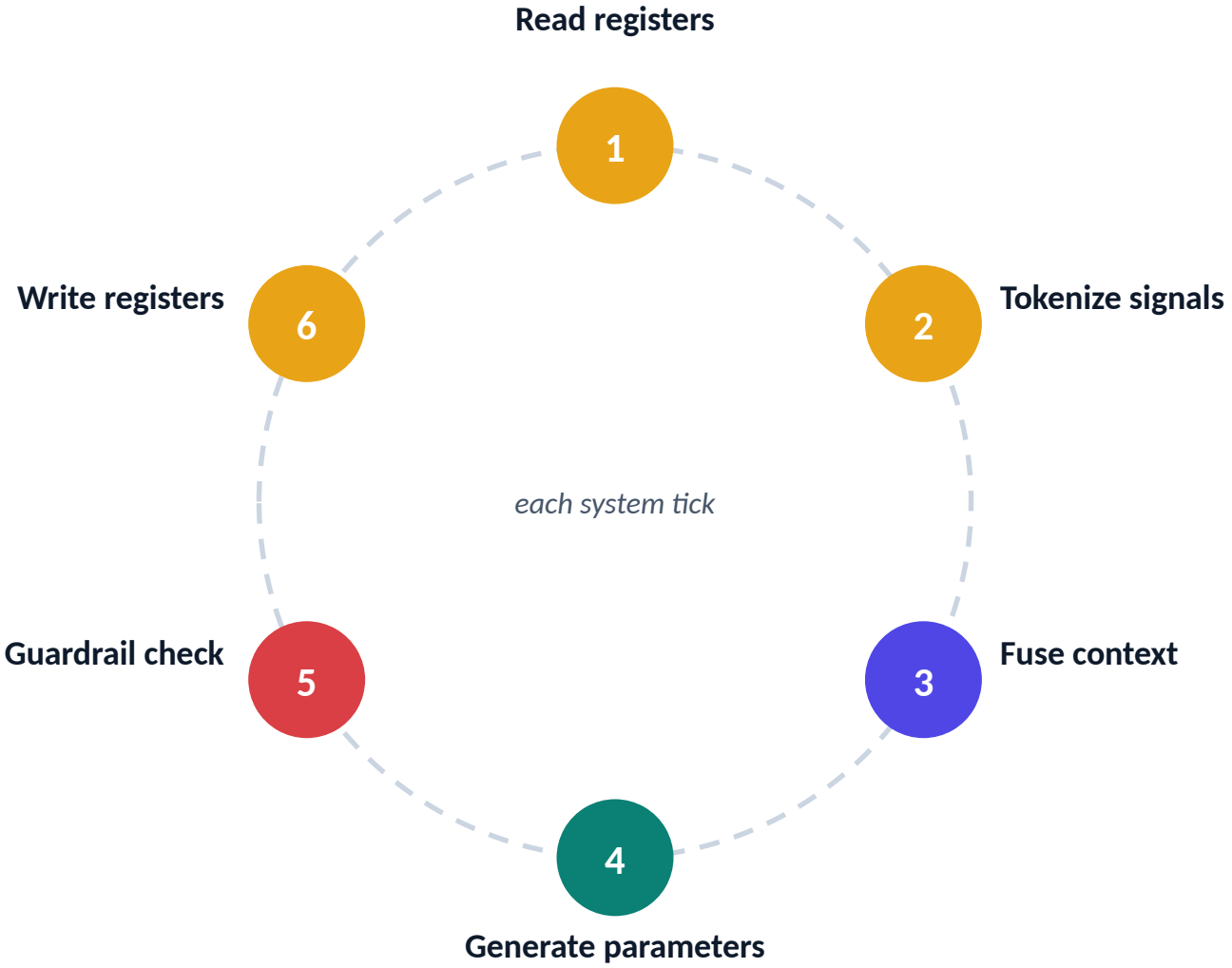
The SLM halts the PSO thread and starts an alternate recovery algorithm, such as a simulated annealing cooling loop, to reset the system smoothly.

A solar farm of cooperating node SLMs



Each node: a quantized 500M to 1.5B SLM on an ARM Cortex-M7, STM32MP1 or local NPU module.

Inside one node: the six-step control loop



1	raw physical telemetry
2	`[NODEID: BAT04] [SOC: 89.2%] [TEMP: 42.1C] [TREND: STEEP-RISE]`
3	`[GLOBALSOS: DEMANDSURGE] [PRICE-SIGNAL: HIGH]`
4	quantized on-device 1B SLM
5	`hard clamp: chargeCurrentLimitA ≤ 120.0 A`
6	DAC, PWM, Modbus registers

Prompt: "Maintain cell safety above all else."

Five places the paper says classical control leaves energy behind



Nano-scale cell degradation

`Layer 1`

Micro-amperage and impedance reveal micro-cracks and soiling; smart junction boxes reconfigure sub-strings to isolate weak cells.



Cognitive albedo and micro-cloud tracking

`Layer 1 + sky camera`

When diffuse light off clouds, panels or ground beats the obscured direct beam, tilt slightly away from the sun.



Predictive thermal-electrical pre-cooling

`Layer 7 + Layer 1`

Before a forecast clear-sky window, cycle MPPT out of phase to cool cells, so peak sun lands on a cooler, more efficient panel.



Swarm wake deflection

`SoS cross-layer`

At low sun angles, front rows flatten their tilt to clear the shadow line for rear rows and raise net farm yield (Cooper et al., 2025).



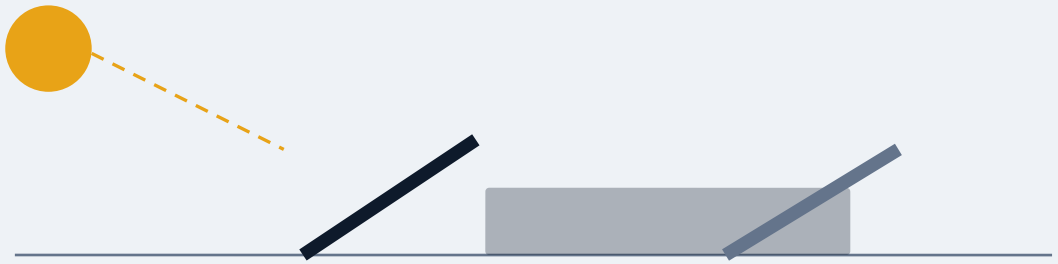
Electromagnetic noise harvesting

`Layers 1 – 2`

Shift inverter switching so IGBT noise cancels or aligns with passive capacitor harvesting loops feeding the DC bus (Zhang et al., 2021).

Two of the paradigms, drawn out

Swarm-level wake deflection

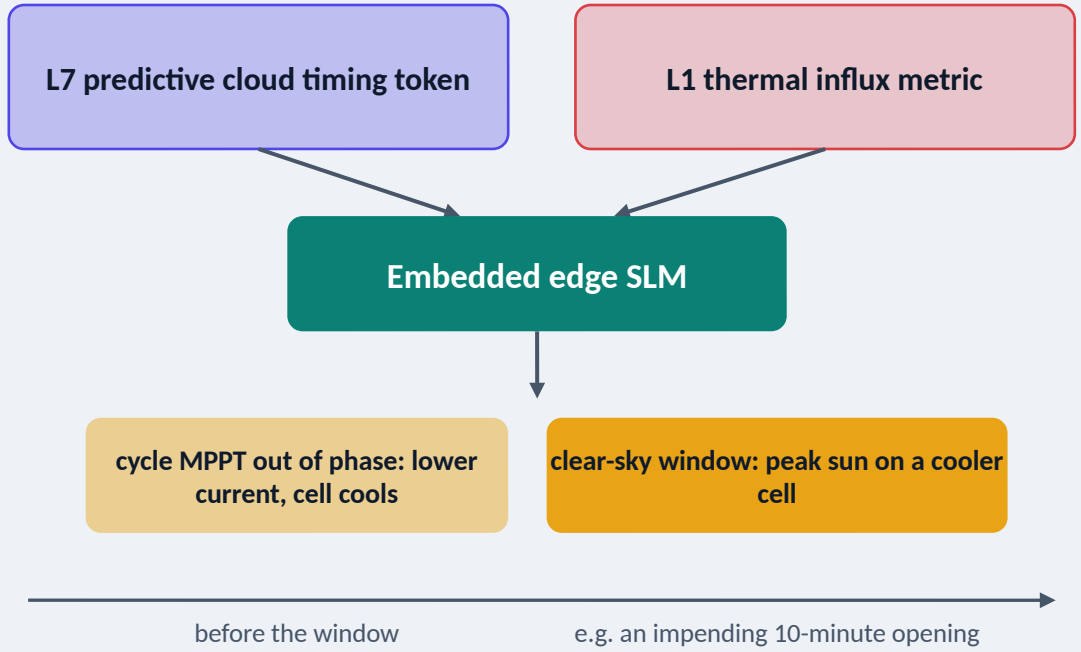


Row 1 aggressive angle: shadow blinds Row 2



Row 1 flattened: light passes, Row 2 captures

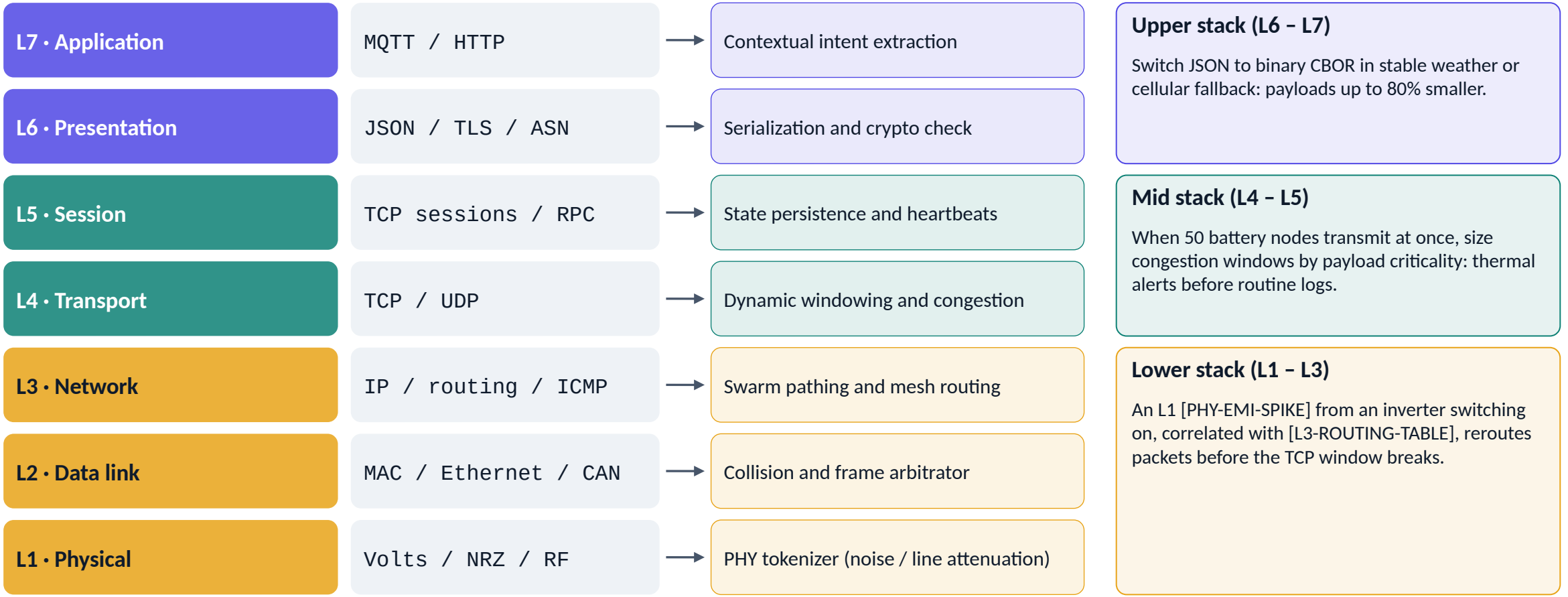
Predictive thermal-electrical co-optimization



Why: cells have a sharp negative temperature coefficient.

Cooper et al., 2025 (backtracking context for wake deflection). Diagrams redrawn from the paper's figures on p. 17 to 18.

The OSI-SLM reads all seven layers as one context window

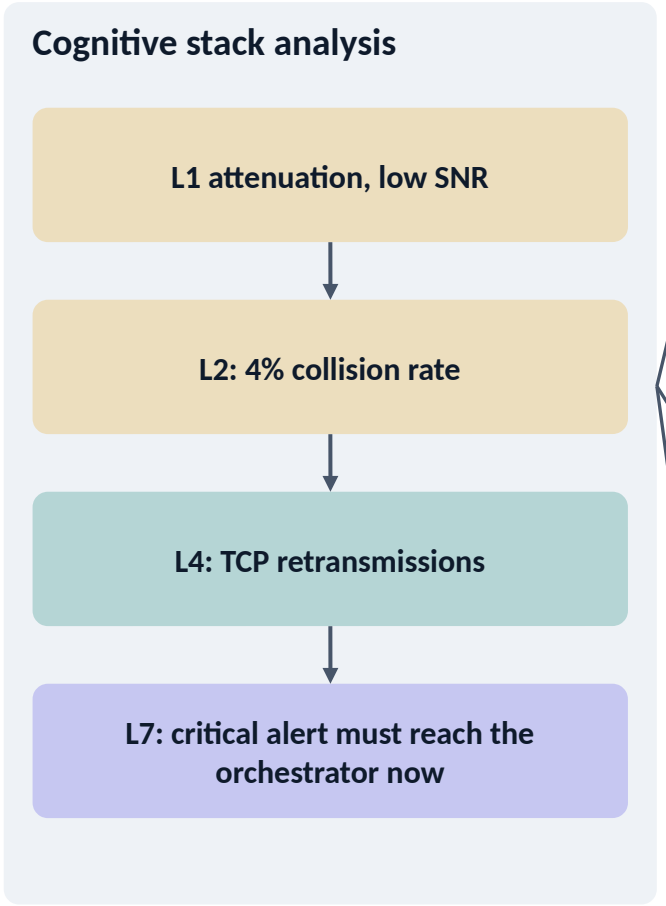


Payloads and metadata are intercepted at each layer boundary through kernel hooks and eBPF (Soldani et al., 2023).

One overheating alert, reasoned across the whole stack

[OSI-STACK-TELEMETRY]

L1	SNR-12dB	ATTN-HIGH
L2	COLLISION-RATE: 4%	PROTOCOL: ETHERNET
L3	ROUTE-COST-MIN	IP-SRC: 10.0.1.45
L4	TCP-WINDOW-SIZE: 1460	RETRANSMIT-DETECTED
L5	SESSION-STATE: INTERMITTENT	
L6	ENCODING: RAW-JSON	
L7	"inverterTemp": "95C", "OVERHEATING"	



[STACK-CONTROL-OUTPUT]

L6 · reformat to CBOR

compact binary instead of raw JSON

L4 · shift to UDP

drop stateful TCP; unacknowledged broadcast

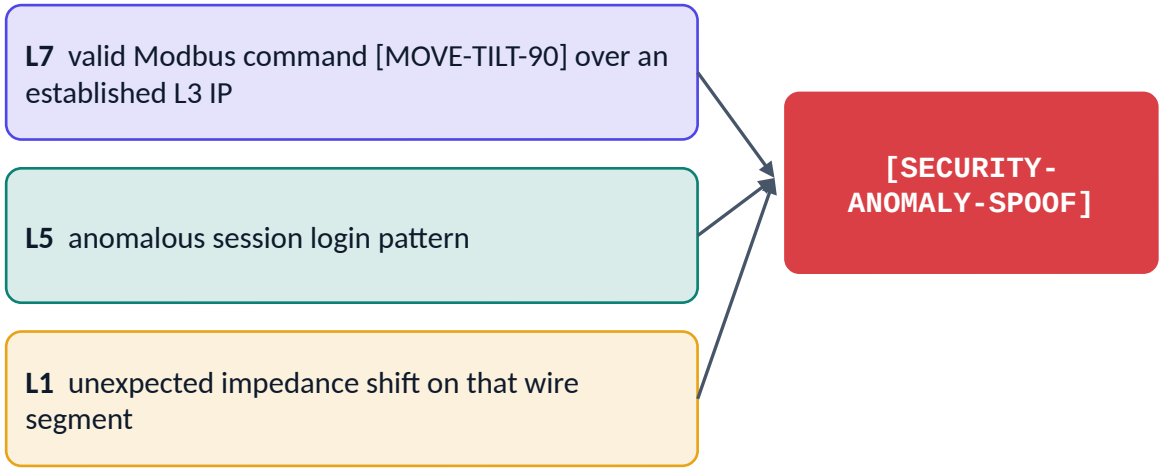
L3 · route via 10.0.2.1

alternate interface for swift delivery

Telemetry, analysis and output from the paper's interleaved example (§6.1, p. 19 to 20).

Seeing every layer catches spoofing and shrinks payloads

Inline zero-trust intrusion detection



Result: parameters rejected at the guardrail and the port isolated, even though the command itself was valid at Layer 7.

up to 80%

smaller payloads switching L6 serialization from JSON to binary CBOR

up to 95%

less data overhead from intent-based data collapsing

Instead of a 2 KB JSON diagnostic every second, send intent tokens; the receiver's SLM rebuilds the context:

`[BAT-04]` `[CELL-STABLE]` `[THERM-CYCLING-NOMINAL]`

Percentages are the figures the paper states (H5); it reports no measurement of them.

Static stack vs unified model, in three scenarios

Scenario	OSI layers	Traditional reaction (static stack)	Unified model co-optimization
High irradiance + high thermal load	L7, L4, L1	Inverter maxes extraction until a hard thermal trip; static TCP window causes bus collisions; fans start late.	Detunes MPPT before the ceiling; packs non-critical frames into multi-second arrays; pre-overclocks cooling early.
Severe weather transients (wind, cloud shadows)	L3, L2	Central cloud loses signal and trackers stall; L2 frames drop with line jitter.	Builds a local peer-to-peer wireless mesh; shortens L2 packet intervals to slip between interference bursts.
Network congestion / swarm inundation	L6, L5	Verbose JSON overflows MCU buffers; heartbeat timeouts trigger re-handshake storms.	Switches JSON to CBOR on the fly; stretches heartbeat intervals by payload criticality.

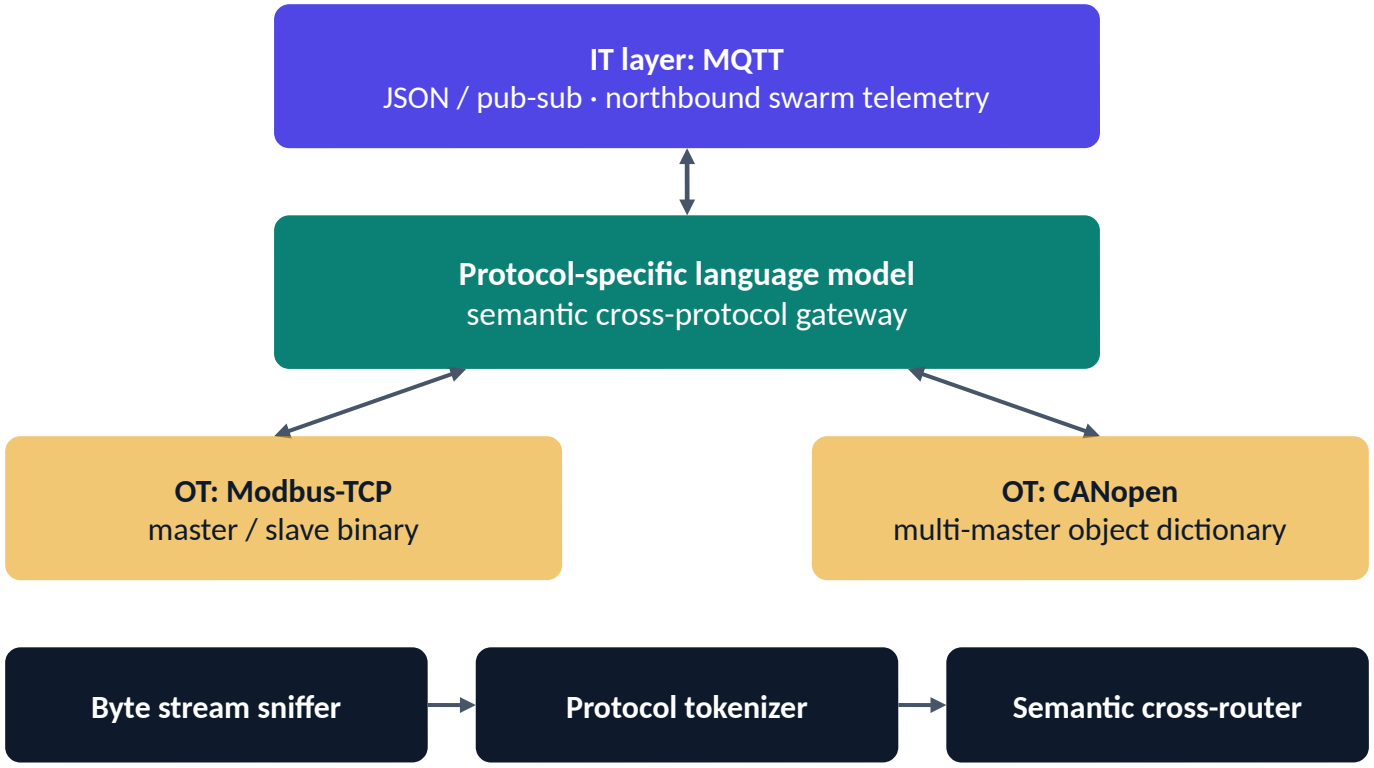
Three design quality principles

Strict state isolation
via the deterministic guardrail interlock

Shared global/local context
scales global priority down to local survival if the orchestrator drops

Graceful degeneracy
under CPU or memory strain, stop SLM inference and revert to PID and static Modbus maps

A protocol SLM bridges OT fieldbuses and IT fabrics



raw hex over RS-485 / Ethernet → semantic blocks → re-encapsulated in the target protocol

Byte-to-token compilation

```
01 03 00 20 00 02 C5 C4
```

`[MODBUS-READ]` `[SLAVE-01]`

`[START-REG-32]` `[LEN-2]` `[CRC-VALID]`

Schemaless Modbus payloads are mapped onto standardized SunSpec Alliance solar data models.

Context-aware error decoding

On `[MODBUS-ERROR-04]` the SLM checks Layer 1 noise to tell line noise from an overloaded inverter MCU, then throttles polling instead of crashing the bus.

One northbound intent becomes two southbound frames

MQTT demand-response (northbound)

```
topic: farm/grid/demand-response

{"strategy": "curtail-battery", "target-reduction-kw": 50}
```

→ **Protocol SLM**

→ **Block A · Modbus-TCP write**
SunSpec register 40021 Active Power Limit, inverter

→ **Block B · CANopen SDO**
lowers discharge current limit across BMS cells

Resiliency via a decoupled Semantic Diary State

Connected
normal MQTT-ACK publishing

→ **Cloud dropout**
the SLM keeps compiling local Modbus and CAN frames into a compressed diary (store-and-forward)

→ **Reconnect**
publish one chronologically continuous state-history token



Instead of flooding the network with millions of raw historical registers, which preserves bandwidth and prevents grid data storms.

Summary

The architecture in one view



The model proposes; deterministic code decides what reaches the hardware.

H1 decoupling | H2 L(x) vs G(x) | H3 signal as language | H4 hyper-heuristics | H5 cognitive OSI | H6 micro-scale yield

Six hypotheses stated for evaluation; the paper presents the architecture and leaves testing to future work.

References (1 to 11 of 22)

1. Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., & Sanghai, S. (2023). GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. arXiv:2305.13245.

2. Anderson, K. (2020). Maximizing Yield with Improved Single-Axis Backtracking on Cross-Axis Slopes: Preprint. NREL/CP-5K00-76023. 47th IEEE PVSC.

3. Anderson, K. S., Jensen, A. R., & Riley, D. M. (2026). A Linear Programming Approach to Backtracking for Single-Axis Trackers on Rolling Terrain. IEEE J. Photovoltaics, 16(2), 291–298. doi:10.1109/JPHOTOV.2025.3645781

4. Bosio, C., Guarrera, M., Sangiovanni-Vincentelli, A., & Mueller, M. W. (2025). Combining Large Language Models and Gradient-Free Optimization for Automatic Control Policy Synthesis. arXiv:2510.00373.

5. Cooper, E., Anderson, K., & Riley, D. (2025). Performance Improvements Through Advanced PV Backtracking on Uneven Terrain. IEEE J. Photovoltaics, 15(4), 593–599. doi:10.1109/JPHOTOV.2025.3558254

6. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. arXiv:1503.02531.

7. Hu, K., Qiu, L., Lu, Y., Zhao, H., Li, T., Kim, Y., Andreas, J., & He, K. (2026). ELF: Embedded Language Flows. arXiv:2605.10938.

8. Lamaakal, I., Maleh, Y., El Makkaoui, K., Ouahbi, I., Pławiak, P., Alfarraj, O., Almousa, M., & Abd El-Latif, A. A. (2025). Tiny Language Models for Automation and Control. Sensors, 25(5), 1318. doi:10.3390/s25051318

9. Maher, G. (2025). LLMPC: Large Language Model Predictive Control. Computers, 14(3), 104. doi:10.3390/computers14030104

10. Mohan, S., Bak, S., Betti, E., Yun, H., Sha, L., & Caccamo, M. (2013). S3A: Secure System Simplex Architecture for Enhanced Security and Robustness of Cyber-Physical Systems. HiCoNS '13, 65–74.

11. Narimani, M., & Emami, S. A. (2026). GA-Agent: Large Language Models as Hyperparameter Optimizers for Evolutionary Controller Synthesis. arXiv:2609.27725.

References (12 to 22 of 22)

12. Nesti, F., Salamini, N., Marinoni, M., Cicero, G., Serra, G., Biondi, A., & Buttazzo, G. (2026). The use of the Simplex architecture to enhance safety in deep-learning-powered autonomous systems. Eng. Appl. of AI, 174, 114583. doi:10.1016/j.engappai.2026.114583

13. Scherer, M., Macan, L., Jung, V. J., Wiese, P., Bompani, L., Burrello, A., Conti, F., & Benini, L. (2024). Deeploy: Enabling Energy-Efficient Deployment of Small Language Models on Heterogeneous Microcontrollers. arXiv:2408.04413.

14. Sha, L. (2001). Using Simplicity to Control Complexity. IEEE Software, 18(4), 20–28. doi:10.1109/MS.2001.936213

15. Soldani, D., Nahi, P., Bour, H., Jafarizadeh, S., Soliman, M. F., Di Giovanna, L., Monaco, F., Ognibene, G., & Risso, F. (2023). eBPF: A New Approach to Cloud-Native Observability, Networking and Security for Current (5G) and Future Mobile Networks (6G and Beyond). IEEE Access, 11, 57174–57202.

16. van Stein, N., & Bäck, T. (2025). LLaMEA: A Large Language Model Evolutionary Algorithm for Automatically Generating Metaheuristics. IEEE Trans. Evol. Comput., 29(2), 331–345. doi:10.1109/TEVC.2024.3497793

17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. NeurIPS 2017, 30.

18. Wagner, J.-S., DeCaprio, D., Muthu Raja, A. C., Holman, J. M., Brady, L. K., et al. (2024). CELI: Controller-Embedded Language Model Interactions. arXiv:2410.14627.

19. Wang, H., Li, W., Zhang, J., & Ren, L. (2026). AI Control Scientist: LLM-driven Agentic System for Automated Control Design. arXiv:2608.26780.

20. Xia, Y., Jazdi, N., Zhang, J., Shah, C., & Weyrich, M. (2024). Control Industrial Automation System with Large Language Model Agents. arXiv:2409.18009.

21. Yuan, D., Deng, Y., & Chen, H. (2026). EmbedCopilot-Bench: Evaluating Vision-Language Models for Hardware-Aware Embedded System Development. ECCV 2026.

22. Zhang, Z., Hu, Y., Chen, X., Jewell, G. W., & Li, H. (2021). A Review on Conductive Common-Mode EMI Suppression Methods in Inverter Fed Motor Drives. IEEE Access, 9, 18345–18360. doi:10.1109/ACCESS.2021.3054514