Scalable Design-for-Calibration of Programmable Silicon Photonics

Lawrence Schlitt, Priyank Kalla, and Steve Blair
Department of Electrical & Computer Engineering, University of Utah, Salt Lake City, UT, USA

***Abstract*—Process variation, thermal cross-talk, and environmental drift make large programmable silicon photonic integrated circuits (PICs) difficult to program and maintain. We present a design-for-calibration (DfC) framework that co-designs partitioning, modeling, and control for scalable calibration. DfC combines hypergraph-based partitioning to expose weakly interacting calibration blocks, sparse per-block surrogate models to accelerate local optimization, and hierarchical feedback including dither-based ring locking for microring banks. Simulation studies on a 256-tuner mesh-inspired benchmark and an 8-ring filter bank show reduced wall-clock-equivalent probing versus an unpartitioned global surrogate, reduced committed actuator writes versus sequential tune-and-check, and robust operation under thermal disturbance.**

***Index Terms*—silicon photonics, photonic integrated circuits, design-for-calibration, design-for-test, hypergraph partitioning, surrogate modeling**

I. Introduction

Programmable silicon photonic integrated circuits (PICs) built from tunable Mach–Zehnder interferometers (MZIs) and microring resonators (MRRs) are emerging as a general-purpose platform for optical signal processing, communications, and photonic computing. Interferometer meshes can implement arbitrary linear transformations for universal multiport processors [1], [2], while microring banks provide compact wavelength-selective primitives central to filtering and wavelength-division multiplexing (WDM)-style programmability.

As these architectures scale, the dominant challenge shifts from functional synthesis to *calibration and lifetime maintenance*. Fabrication-induced phase, coupling, and loss variations displace the as-fabricated operating point from design intent, while ambient drift and heater-induced thermal cross-talk perturb many tunable elements during operation [3], [4]. For large-scale PICs, sequential "tune-and-check" procedures become slow and can fail when tuning actions interact through coupling, creating a system-level calibration bottleneck [5], [6].

A second barrier is *observability*. Closed-loop calibration requires internal feedback, yet dense test access points (TAPs) can impose insertion loss and routing overhead. Non-invasive monitors such as Contactless Integrated Photonic Probes (CLIPPs) provide distributed internal power sensing with minimal optical penalty [7], and dither-based ring locking stabilizes individual MRRs against drift [8]. Existing techniques, however, are usually tailored to specific subsystems or tuning sequences rather than a unified, topology-agnostic calibration methodology for large heterogeneous PICs.

In this work, we propose a *design-for-calibration (DfC)* paradigm that treats calibratability as a co-design objective spanning architecture, instrumentation, modeling, and control. The key idea is to represent multi-way optical, thermal, and electrical coupling with a hypergraph abstraction [9], and partition [10] the circuit into weakly interacting calibration blocks. Within each block, data-efficient surrogate models learned[1] from sparse probing enable fast local optimization and expose Jacobian information for calibratability assessment. Setpoints are then maintained by hierarchical feedback, including practical dither-based locking for microring banks, while a slower supervisory layer coordinates residual inter-block interactions. The main contributions are:

- **Coupling-aware partitioning for parallel bring-up:** a weighted hypergraph representation of multi-way coupling and a balanced k-way partitioning that concentrates global coordination on a small set of inter-block interfaces.
- **Surrogate-assisted calibration with diagnostics:** per-block surrogates trained from sparse, information-rich probing to accelerate local optimization and provide Jacobian-based indicators of calibratability (sensitivity and conditioning).
- **Hierarchical in-field maintenance:** a two-timescale control structure that combines fast local feedback (including ring-bank locks) with slower supervisory setpoint refresh to reject drift while containing cross-talk.

The remainder of the paper is organized as follows. Section II summarizes relevant background and previous work. Section III presents the proposed DfC framework, including partitioning, surrogate learning, and hierarchical maintenance. Section IV covers the implementation of the methodology into pseudocode. Section V evaluates the approach using representative simulation case studies, and Section VI concludes with limitations and future directions.

II. Background & Previous Work

Programmable photonic architectures. Universal programmable photonic processors based on interferometer meshes are well established, with canonical decompositions enabling arbitrary linear transformations (including unitaries) [1], [2]. In practice, these processors are realized using large arrays of tunable MZIs and phase shifters. Their utility

[1]In this work, "trained" or "learned" refers to system identification and curve fitting from measured calibration data, i.e. non-AI/ML models.

as a general-purpose platform is ultimately limited not by whether a target function *can* be synthesized, but by whether it can be *reliably programmed and maintained* in the presence of fabrication variation, drift, and coupling among tuning elements. A generic view of a PIC is depicted in Fig. 1, where laser power, electrical and thermal control inputs may also act as *tuning/calibration knobs*.

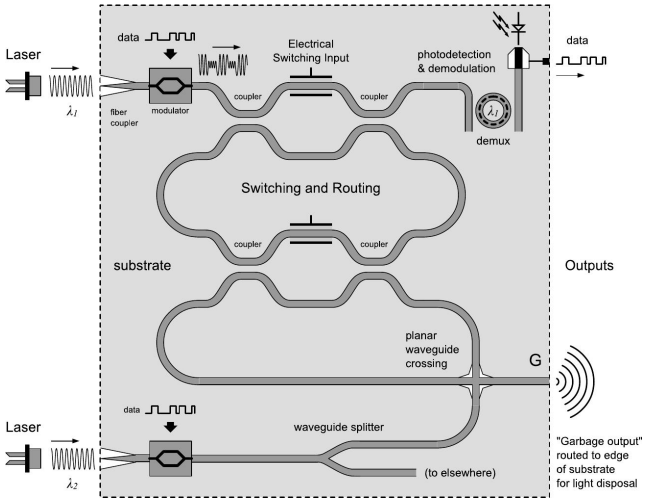

Fig. 1. Generic heterogeneous PIC illustration showing representative actuation and sensing paths.

Figure 1 is a generic system-level illustration of available calibration knobs and monitors, not the exact benchmark topology used in Section V-A.

Calibration and automation of large meshes. A substantial amount of literature addresses automated bring-up and programming of interferometric meshes. Self-configuration, progressive routing, and local tuning rules have demonstrated robust configuration for small- to medium-scale meshes [11], and newer methods improve observability and robustness to imperfect components [12]. At larger scales, however, thermal cross-talk and shared optical paths induce multi-parameter interactions that slow convergence, cause oscillation between partial solutions, and require careful sequencing and revisiting of previously tuned elements.

Observability and monitoring. Closed-loop calibration depends critically on internal sensing. Dense TAP monitors and photodetectors (PDs) can provide high-SNR signals but impose insertion loss, routing overhead, and layout constraints that become prohibitive as circuits scale. CLIPP sensors provide distributed, non-invasive power monitoring with minimal optical penalty and have enabled scalable control demonstrations in silicon photonics [7]. Monitor placement, readout strategy, and the resulting identifiability/conditioning materially impact calibration speed and reliability in large PICs [5], [6].

Resonant subsystems and thermal cross-talk. MRRs provide compact, wavelength-selective primitives but are highly sensitive to thermal drift, making feedback-based stabilization essential in practical systems. Dither-based locking with lock-in demodulation (paired with integrated sensing) remains a dominant approach for stabilizing MRR-based filter banks [8], [13]. As integration density increases, thermal cross-talk among nearby heaters and resonators becomes a primary limitation. Recent work shows that learned thermal models can predict and compensate cross-talk with high accuracy in programmable PICs [14].

Gap and motivation. Taken together, prior work establishes effective *components*—mesh programming algorithms, internal monitoring technologies, ring-locking loops, and cross-talk compensation methods—but these elements are often developed in isolation or assume specific topologies and tuning sequences that become difficult to scale. What remains lacking is a *unified, system-level calibration methodology* that (i) explicitly leverages coupling structure to decompose the tuning problem, (ii) co-designs calibration with sparse-yet-sufficient observability, and (iii) supports both fast bring-up and continuous in-field maintenance for large, heterogeneous programmable PICs. This gap motivates the DfC framework developed in the remainder of this paper.

III. Design-for-Calibration Framework

A. Overview: bring-up vs. in-field maintenance

Our DfC flow treats calibration as a scalable pipeline with two operating modes: *bring-up* (factory/wafer/package calibration) and *in-field maintenance*. Bring-up computes a baseline set of actuator setpoints by decomposing a globally coupled tuning problem into weakly interacting domains and then accelerating per-domain tuning using compact *surrogate* models fit from sparse calibration probes (i.e., lightweight local input–output regression/system-identification models used to support optimization). In-field maintenance keeps the circuit at its operating point under drift and thermal cross-talk using lightweight always-on local feedback and occasional supervisory setpoint refresh.

Concretely, bring-up proceeds in five stages: (i) construct a weighted coupling hypergraph from optical/thermal/electrical interactions, (ii) compute a balanced k-way partition to expose weakly interacting calibration blocks, (iii) collect sparse, structured probing data and fit a local surrogate per block, (iv) solve per-block constrained local optimization problems (in parallel), and (v) perform a small number of *boundary-only* polishing updates over the inter-block interface. Fig. 2 summarizes the workflow.

B. Model, objectives, and calibratability

We model a programmable PIC as a nonlinear, partially observed system with uncertain (and drifting) parameters. Let $u \in \mathbb{R}^m$ be actuator inputs (e.g., thermo-optic heaters or carrier-based phase shifters) and $y \in \mathbb{R}^p$ be on-chip monitor readings. Let $\theta \in \mathbb{R}^q$ collect lumped, slowly varying uncertainties such as fabrication-induced phase/coupling/loss offsets and environmental state (e.g., local temperature) that can drift over time. The measurement model is

$$y \;=\; h(u,\theta) + \nu \;\triangleq\; \mathcal{M}(F(u,\theta)) + \nu, \tag{1}$$

where $F(u,\theta)$ denotes the underlying circuit response (e.g., transfer characteristics), $\mathcal{M}(\cdot)$ maps that response to the available monitor channels (TAP PDs, CLIPPs, etc.), θ captures fabrication offsets and drift, and ν is measurement noise.

Given target readings $y^\star$ (derived from the desired circuit function and monitor map), bring-up calibration seeks a feasible actuation $u^\star \in \mathcal{U}$ that minimizes the measured residual:

$$u^\star \in \arg\min_{u\in\mathcal{U}} \; \|h(u,\theta) - y^\star\|_2^2 , \tag{2}$$

where $\mathcal{U}$ captures practical constraints such as actuator bounds and slew-rate limits.

We use the squared ℓ_2 residual in (2) because (i) under an approximately Gaussian monitor-noise model it corresponds to a maximum-likelihood objective; (ii) it yields a smooth, twice-differentiable cost with well-behaved gradients/curvature, enabling efficient Gauss–Newton (GN) or Levenberg–Marquardt (LM)-style numerical solving/updates in closed loop.

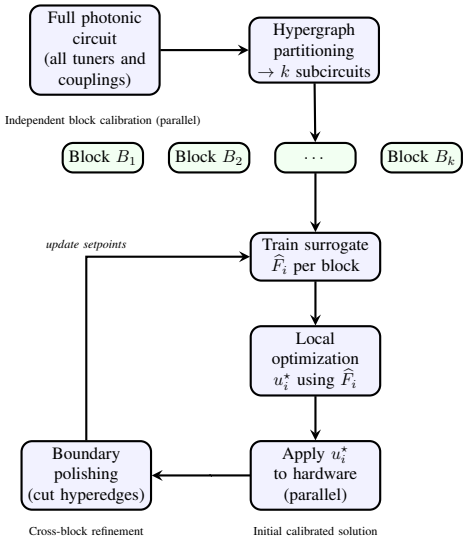

Fig. 2. Block-wise optimization of a photonic circuit using surrogate models.

In practice, (2) is solved iteratively in closed loop using measurements, e.g., via a small number of damped GN/LM steps based on estimated (measured or surrogate) Jacobians, with projection/clipping to enforce $u \in \mathcal{U}$ (bounds and slew limits). Because θ is unknown and time-varying, DfC never relies on a single global linear model: it fits each block locally around the current operating point using small Hadamard-coded perturbations, applies bounded LM-style updates, and then re-probes and re-linearizes as conditions shift.

Linearizing Eq. (1) about an operating point $(\bar{u},\bar{\theta})$ yields

$$\Delta y \approx J_u\,\Delta u + J_\theta\,\Delta\theta, \qquad J_u = \tfrac{\partial h}{\partial u},\; J_\theta = \tfrac{\partial h}{\partial \theta}, \tag{3}$$

which links monitor placement, actuation authority, and drift sensitivity to practical calibratability while handling nonlinear effects piecewise rather than assuming them away.

C. Hypergraph partitioning for parallel tuning

Calibration-aware hypergraph construction. To expose weak coupling structure relevant to calibration, we represent a programmable PIC as a *weighted hypergraph* $H = (V, E)$. Each vertex $v \in V$ corresponds to a tunable primitive with one effective control input. Hyperedges encode *multi-way* interactions that couple several tuners through optical paths, thermal diffusion, or shared electrical resources.

A practical way to construct E is to start from an estimated local sensitivity (coupling) map obtained from layout/PDK models and/or a short identification sweep (e.g., small finite-difference heater steps or coded perturbations/dithers around the current bias point). For a monitor channel y_ℓ (TAP PD, CLIPP, etc.), define a pin set

$$\mathrm{pins}(e_\ell) \triangleq \left\{ v_j \in V : \left|\frac{\partial y_\ell}{\partial u_j}\right| \geq \tau_\ell \right\}, \tag{4}$$

and include a hyperedge e_ℓ connecting all tuners that materially influence that measurement. Here τ_ℓ is a *sensitivity threshold* used to sparsify the hypergraph. Thermal interactions can be incorporated similarly by forming hyperedges over tuners within a heater footprint or neighborhood whose thermal coupling exceeds a threshold.

Edge and vertex weights. Hyperedge weights $w_e(e)$ quantify the expected *strength* (and therefore calibration impact) of the interaction represented by e.

$$w_e(e_\ell) \triangleq \sum_{v_j \in \mathrm{pins}(e_\ell)} \left|\frac{\partial y_\ell}{\partial u_j}\right|. \tag{5}$$

Vertex weights $w_v(v)$ capture calibration "mass" (e.g., expected probing effort for that tuner group, or a proxy for actuation cost). In our simulations we use uniform w_v and synthetic coupling weights; the same construction supports layout/PDK-derived coupling maps or identification-based estimates.

Balanced k-way partitioning objective. Given $H = (V, E)$, we seek a balanced k-way partition $\{B_i\}_{i=1}^k$ that keeps strongly coupled tuners within the same block while minimizing cross-block interactions:

$$\min_{\{B_i\}} \sum_{e\in E} w_e(e)\big(\lambda(e) - 1\big) \tag{6}$$
$$\text{s.t.} \sum_{v\in B_i} w_v(v) \leq (1+\epsilon)\frac{1}{k}\sum_{v\in V} w_v(v),$$

where $\lambda(e) \triangleq |\{\, i : e \cap B_i \neq \emptyset \,\}|$ is the number of blocks spanned by hyperedge e, and ϵ the balancing factor for partitioning.

Interface definition and polishing set. The partition induces an explicit *interface* where global coordination is concentrated:

$$V_{\mathrm{int}} \triangleq \{\, v \in V : \exists e \in E \text{ incident on } v, \lambda(e) > 1 \,\}. \tag{7}$$

DfC uses this set to restrict global polishing updates to interface tuners only, rather than re-solving a monolithic full-chip problem. Concretely, *boundary polishing* refers to one or two refinement iterations (e.g., damped Gauss–Newton/least-squares updates on the residuals associated with cut hyperedges) in which only $u_{V_{\mathrm{int}}}$ is adjusted while the intra-block (non-interface) tuners are held fixed.

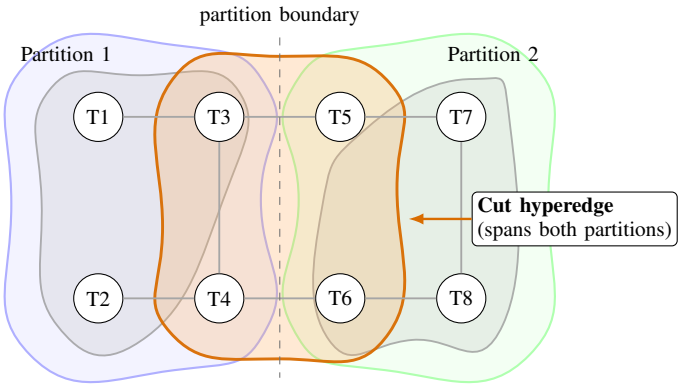

Fig. 3. A cut hyperedge represents a multi-way optical/thermal interaction spanning multiple calibration blocks. Nodes are tuners; shaded regions denote hyperedges; the highlighted region crosses the partition boundary.

Parallel wall-clock proxy. To summarize parallelism without committing to a specific runtime scheduler, we report an idealized full-parallel probe-round metric in which blocks within the same outer round are treated as if they could be probed concurrently. This is an optimistic wall-clock proxy rather than a graph-colored execution schedule.

Implementation note. Solving (6)–(7) optimally is NP-hard, so we use standard multilevel heuristic partitioners developed for large netlists/hypergraphs [10]. Reported runs use a deterministic multilevel heuristic partitioner. The contribution here is the calibration-aware hypergraph construction and weighting, and the explicit use of the induced interface for boundary-only polishing and the parallelism proxy within the DfC pipeline.

D. Surrogate-assisted bring-up

For each block B_i, we *fit* a lightweight local surrogate around the current operating point. Let $u_i \in \mathbb{R}^{m_i}$ be the block's tuners and $y_i \in \mathbb{R}^{p_i}$ be the corresponding local monitor outputs.

We fit

$$y_i \approx \widehat{F}_i(u_i, \xi_i), \tag{8}$$

where ξ_i optionally includes slow context features (e.g., estimated local temperature or a compact summary of neighboring block heater powers) to capture boundary cross-talk and reduce bias in local gradients. Training uses a small number of information-rich probes (typically scaling with m_i): small perturbations about the current bias point, orthogonal patterns such as Hadamard-coded $\pm$ steps, and/or scheduled dithers, followed by regularized regression/learning [15]. In the reported experiments, $\widehat{F}_i$ is an affine-plus-sparse-quadratic ridge surrogate centered at the current operating point and fit from Hadamard-coded $\pm$ probe data. In the representative runs reported here, ξ_i is omitted; it is retained only to indicate a natural extension for future context-aware surrogates.

Bring-up tuning then solves a small local problem on the surrogate:

$$u_i^\star \in \arg\min_{u_i \in \mathcal{U}_i} \left\| \widehat{F}_i(u_i, \xi_i) - y_i^\star \right\|_2^2, \tag{9}$$

with actuator bounds and slew-rate guardrails. Because $\widehat{F}_i$ is differentiable, we compute $J_i = \partial \widehat{F}_i / \partial u_i$ to accelerate local updates (e.g., a few damped LM-steps with projection onto $\mathcal{U}_i$) and to detect ill-conditioned calibration (e.g., insufficient sensing or overly coupled tuners). Block solutions are applied in parallel. Finally, a few *polishing* rounds (boundary-only corrections over interface tuners) adjust only boundary/interface tuners to reduce residual mismatch along cut hyperedges without re-solving the full-chip problem.

E. In-field maintenance: feedback, observability, cross-talk

Bring-up yields an initial programmed state, but practical systems require always-on feedback to reject drift and parameter migration. We use a two-timescale hierarchy:

- **Fast local loops (inner):** per-element or per-cell regulation using nearby monitors (TAP PD, CLIPP, or embedded sensing), implemented with incremental digital proportional-integral (PI)-family feedback, similar in spirit to [16].
- **Ring-bank locks:** per-ring dither + lock-in demodulation to generate a signed detuning error (with sign indicating red/blue detuning relative to the lock point), followed by a local PI-family update to null it [8].
- **Slow supervisory updates (outer):** block-level setpoint refresh using surrogate Jacobians (e.g., damped least-squares / coordinate corrections) that *update* inner-loop targets rather than closing one monolithic controller around the full chip [17].

After each supervisory setpoint refresh, affected blocks are re-probed around the current operating point, re-linearized through an updated local surrogate/Jacobian, and locally re-optimized before the next outer update. Thus local optimization is used both during bring-up and as the block-level correction mechanism inside the outer feedback loop.

IV. Implementation

Our evaluation is implemented in Python as outlined in Algo. 1. The bring-up pipeline follows Sec. III: (i) build a weighted coupling hypergraph, (ii) compute a balanced k-way partition with a deterministic multilevel heuristic partitioner [10], (iii) fit per-block surrogates from sparse structured probes, (iv) compute per-block constrained updates using a damped LM-step with bound projection and slew limits, and (v) perform 1–2 boundary-only polishing rounds over cut-edges. We report accuracy and effort metrics.

A. Metrics and effort accounting

Residual and calibration error. Given target monitor readings $y^\star$ and measured outputs $y = h(u, \theta) + \nu$, define

$$r(u) \triangleq h(u, \theta) - y^\star, \tag{10}$$

$$\mathrm{MSE}(u) \triangleq \frac{1}{p} \|r(u)\|_2^2, \tag{11}$$

where p is the number of monitored constraints used in the objective, and MSE denotes the mean squared error.

Probe counts (measurement effort). A *probe* is one monitor acquisition under a chosen perturbation/stimulus. Let $P_{i,t}$

Algorithm 1 DfC bring-up and in-field maintenance (implementation view)

Require: Coupling/sensitivity data, monitor map $\mathcal{M}$, target readings $y^\star$, constraints $\mathcal{U}$

Ensure: Calibrated setpoints $u^\star$ and always-on maintenance loops

Construct weighted hypergraph $H = (V, E)$ from optical/thermal/electrical coupling
Partition H into blocks $\{B_i\}_{i=1}^{k}$ (balance factor ϵ); form interface set V_{int}
for $t = 1, \ldots, T$ **do** ▷ bring-up outer rounds
  for all blocks $\{B_i\}_{i=1}^{k}$ **in parallel do**
    Probe (Hadamard $\pm$ steps), fit surrogate $\widehat{F}_i$, compute J_i
    Compute constrained LM update for u_i; apply bounds/slew; write block setpoints
  end for
end for
Boundary polishing: update only V_{int} for S rounds to reduce cut-hyperedge residuals
In-field maintenance: fast local PI-family feedback / ring-bank dither locks; periodic supervisory refresh triggers re-probe, re-linearization, and local block re-optimization

be probes collected for block B_i during outer round t and let P_{pol} be probes used during polishing:

$$P_{\text{tot}} \triangleq \sum_{t=1}^{T} \sum_{i=1}^{k} P_{i,t} + P_{\text{pol}}, \tag{12}$$

$$P_{\text{eff}} \triangleq \sum_{t=1}^{T} \max_{i \in \{1,\ldots,k\}} P_{i,t} + P_{\text{pol}}, \tag{13}$$

where P_{eff} is an idealized full-parallel wall-clock proxy rather than a colored scheduling bound.

Hardware writes (actuation effort). A *hardware write* is one committed setpoint update applied to an actuator group:

$$W \triangleq \sum_{t=1}^{T} |\mathcal{S}_t| + W_{\text{pol}}, \tag{14}$$

where $\mathcal{S}_t$ is the set of actuator groups updated in round t.

V. Results

This section evaluates DfC on a mesh bring-up benchmark and a microring filter-bank maintenance benchmark. For bring-up we report final residual MSE, total probes P_{tot}, effective probe rounds P_{eff}, and hardware writes W; for ring maintenance we report settle time, peak detuning error, and steady-state root-mean-square (RMS) detuning error.

A. Coupled-tuner mesh bring-up benchmark ($N = 16$)

Setup and objective. To isolate calibration scaling effects without committing to a specific foundry layout, we use a mesh-inspired weighted-hypergraph benchmark with an $N \times N$ lattice of MZI tuners and local multi-way coupling constraints. Here $N = 16$, $m = N^2 = 256$, and $p = 450$ monitored coupling relations. The benchmark is a stand-in rather than a foundry-exact layout: hyperedges represent local 2×2 patches and L-shaped triplets to isolate locality, coupling, and partitioning effects. In this benchmark, $\mathcal{M}$ maps the synthetic plant response to the monitored hyperedge-residual channels used in the calibration objective. The bring-up goal is to drive measured residual signatures to their targets, consistent with Eq. (2).

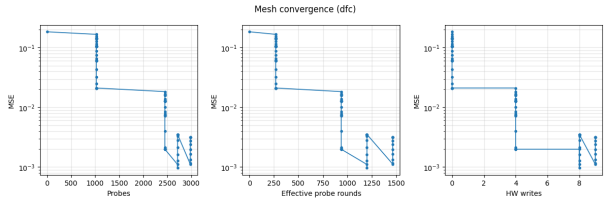

Fig. 4. DfC mesh convergence: residual MSE vs. total probes, effective probe rounds, and hardware writes.

Metric: We report the normalized global residual mean-squared error

$$\text{MSE}(u) = \frac{1}{p} \|y(u) - y^\star\|_2^2, \qquad y^\star = \mathbf{0}, \tag{15}$$

and use $\text{MSE} \leq 1 \times 10^{-4}$ as a stretch target rather than a threshold achieved in the representative run.

Methods compared.

- **DfC (this work):** $k = 4$ balanced partitions, Hadamard-coded probing, per-block surrogate fitting and parallel bounded LM updates, followed by two boundary-only polishing rounds on interface tuners incident to cut hyperedges; we report $k = 4$ as a representative operating point balancing parallelism against cut-interface size.
- **Global surrogate ($k = 1$ ablation):** a single surrogate trained and optimized without block decomposition or polishing to isolate the benefit of partitioning and parallelism.
- **Tune-and-check baseline:** sequential coordinate descent over a random 30% tuner subset per round for 3 rounds, using central finite-difference probing at $u_v \pm 0.05$ and update gain 0.5; for $N = 16$, this is 77 visited tuners per round and 462 total probes.

Results. Table I summarizes a representative run (seed 7). Tune-and-check uses fewer raw probes (462 vs. 2864 for DfC) but leaves much larger residual error. DfC attains the lowest residual error (2.06×10^{-4}), cuts committed hardware-write effort versus tune-and-check (10 vs. 231 writes, $\approx 23\times$ lower), and, relative to the global surrogate ablation, reduces both total probes (2864 vs. 6144) and idealized full-parallel probe rounds (1352 vs. 6144, $\approx 4.5\times$ lower). These results support the central DfC claim: most calibration work stays within weakly interacting blocks, while global coordination is limited to a small interface set during polishing. Reported times are simulator-side compute+calibration runtimes; the code records probe, fit, solve, and polishing components separately.

Fig. 4 shows convergence of DfC MSE as a function of total probes, effective probe rounds, and hardware writes. The sharp reductions in MSE align with the end of each outer round and the subsequent boundary-only polishing step.

TABLE I
REPRESENTATIVE MESH BRING-UP SUMMARY.

Calibration strategy	k	Total probes	Effective probe rounds	HW writes	Final MSE	Time
Tune-and-check (coordinate descent)	1	462	462	231	1.76×10^{-1}	0.62 s
Global surrogate (no partition)	1	6144	6144	2	2.30×10^{-2}	12.50 s
DfC: partition + surrogates + polish	4	2864	1352	10	2.06×10^{-4}	6.17 s

Note: "Total probes" counts all measurements across all blocks. "Effective probe rounds" is the idealized full-parallel wall-clock proxy. "Time" is simulator-side compute+calibration runtime.

B. Microring filter-bank drift rejection ($N_r = 8$)

Setup and metrics. We simulate an $N_r = 8$ microring bank subject to an ambient thermal step disturbance of $\Delta T = 0.5\,\mathrm{K}$ at $t = 0$, with thermal cross-talk level 0.06 to represent dense integration where neighbor heaters perturb nearby resonators. We report **settle time** (re-lock time until detuning stays within threshold for a dwell interval), **peak detuning error** during the transient, and **steady-state RMS detuning error** after re-lock. Section V-B evaluates the fast per-ring locking layer; the slower supervisory refresh is part of the proposed architecture but is not isolated quantitatively here.

Controllers compared:

- **Parallel dither-locking (DfC inner loop):** per-ring heater dither with lock-in demodulation and an incremental PI-family update, with staggered tones (5 Hz base, 0.7 Hz spacing), 50 Hz control updates, and a 0.15 pm lock threshold.
- **Sequential scan-and-retune (baseline):** rings are visited one at a time; the baseline measures all rings at 1 kHz but updates only one active ring, with gain 0.02, update clip ± 0.01, and 150 ms switch dwell.

Results. Table II shows that parallel locking re-acquires the operating point within $\approx 2.88\,\mathrm{s}$ with very low steady-state error, while sequential scan-and-retune does not re-lock within the $5\,\mathrm{s}$ simulation horizon and exhibits large RMS detuning under cross-talk. This illustrates the scaling limit of sequential maintenance: correction latency grows with ring count, whereas parallel local loops keep errors bounded.

TABLE II
RING-BANK DISTURBANCE REJECTION SUMMARY.

Method	Settle time (ms)	Peak error (pm)	RMS error (pm)
Parallel dither-locking	2881	6.96	0.0146
Sequential scan-and-retune	> 5000	16.0	8.05

VI. CONCLUSION

We introduced a *design-for-calibration (DfC)* framework for scalable bring-up and maintenance of programmable silicon photonic integrated circuits. DfC combines coupling-aware partitioning, data-efficient local surrogates, and hierarchical feedback to localize most tuning work while confining global coordination to limited interfaces. The present results are algorithmic validation on synthetic, topology-agnostic benchmarks; foundry-/PDK-derived or measured device models are the immediate next step.

REFERENCES

[1] W. Bogaerts, D. Pérez, J. Capmany, D. A. B. Miller, J. Poon, D. Englund, F. Morichetti, and A. Melloni, "Programmable photonic circuits," *Nature*, vol. 586, no. 7828, pp. 207–216, Oct. 2020.

[2] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walsmley, "Optimal design for universal multiport interferometers," *Optica*, vol. 3, no. 12, p. 1460, Dec. 2016.

[3] W. Bogaerts, Y. Xing, and U. Khan, "Layout-aware variability analysis, yield prediction, and optimization in photonic integrated circuits," *IEEE J. Sel. Topics Quantum Electron.*, vol. 25, no. 5, pp. 1–13, Sep. 2019.

[4] D. S. Boning, S. I. El-Henawy, and Z. Zhang, "Variation-aware methods and models for silicon photonic design-for-manufacturability," *J. Lightwave Technol.*, vol. 40, no. 6, pp. 1776–1783, Mar. 2022.

[5] M. Rossetto, "Programmable integrated photonic circuits: Automatic testing and calibration techniques," Ph.D. dissertation, Politecnico di Milano - School of Industrial and Information Engineering, 2022.

[6] J.-H. Zheng, Q.-Q. Wang, L.-T. Feng, Y.-Y. Ding, X.-Y. Xu, X.-F. Ren, C.-F. Li, and G.-C. Guo, "Global calibration of large-scale photonic integrated circuits," *Phys. Rev. Applied*, vol. 22, no. 5, p. 054011, Nov. 2024.

[7] S. Grillanda, M. Carminati, F. Morichetti, P. Ciccarella, A. Annoni, G. Ferrari, M. Strain, M. Sorel, M. Sampietro, and A. Melloni, "Non-invasive monitoring and control in silicon photonics using cmos integrated electronics," *Optica*, vol. 1, no. 3, p. 129, Aug. 2014.

[8] K. Padmaraju, D. F. Logan, T. Shiraishi, J. J. Ackert, A. P. Knights, and K. Bergman, "Wavelength locking and thermally stabilizing microring resonators using dithering signals," *Journal of Lightwave Technology*, vol. 32, no. 3, pp. 505–512, Feb. 2014.

[9] L. Schlitt, P. Agnihotri, P. Kalla, and S. Blair, "Silicon photonic test-point selection by integrating design parameters with hypergraph partitioning," in *IEEE International Test Conference, ITC 2025, San Diego, CA, USA, September 20-26, 2025*. IEEE, Sep. 2025, pp. 262–271.

[10] L. Gottesbüren, T. Heuer, N. Maas, P. Sanders, and S. Schlag, "Scalable high-quality hypergraph partitioning," *ACM Trans. Algorithms*, vol. 20, no. 1, pp. 1–54, Jan. 2024.

[11] A. Annoni, E. Guglielmi, M. Carminati, S. Grillanda, P. Ciccarella, G. Ferrari, M. Sorel, M. J. Strain, M. Sampietro, A. Melloni, and F. Morichetti, "Automated routing and control of silicon photonic switch fabrics," in *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 22, no. 6. Institute of Electrical and Electronics Engineers (IEEE), Nov. 2016, pp. 169–176.

[12] X. Xu, G. Ren, T. Feleppa, X. Liu, A. Boes, A. Mitchell, and A. J. Lowery, "Self-calibrating programmable photonic integrated circuits," *Nat. Photonics*, vol. 16, no. 8, pp. 595–602, Jul. 2022.

[13] J. C. C. Mak, W. D. Sacher, T. Xue, J. C. Mikkelsen, Z. Yong, and J. K. S. Poon, "Automatic resonance alignment of high-order microring filters," *IEEE J. Quantum Electron.*, vol. 51, no. 11, pp. 1–11, 2015.

[14] I. Teofilovic, A. Cem, D. Sanchez-Jacome, D. Pérez-López, and F. D. Ros, "Thermal crosstalk modelling and compensation methods for programmable photonic integrated circuits," *J. Lightwave Technol.*, vol. 42, no. 22, pp. 7816–7824, Nov. 2024.

[15] Z. Li, Z. Zhou, C. Qiu, Y. Chen, B. Liang, Y. Wang, L. Liang, Y. Lei, Y. Song, P. Jia, Y. Zeng, L. Qin, Y. Ning, and L. Wang, "The intelligent design of silicon photonic devices," *Adv. Opt. Mater.*, vol. 12, no. 7, p. 2301337, Feb. 2024.

[16] K. Padmaraju, J. Chan, L. Chen, M. Lipson, and K. Bergman, "Thermal stabilization of a microring modulator using feedback control," *Opt. Express*, vol. 20, no. 27, pp. 27 999–28 008, Dec. 2012.

[17] M. Milanizadeh, S. Ahmadi, M. Petrini, D. Aguiar, R. Mazzanti, F. Zanetto, E. Guglielmi, M. Sampietro, F. Morichetti, and A. Melloni, "Control and calibration recipes for photonic integrated circuits," *IEEE J. Sel. Topics Quantum Electron.*, vol. 26, no. 5, pp. 1–10, Sep. 2020.