

Article

Permutation-Equivariant Graph Reinforcement Learning for Thermal-Aware Microservice Scheduling in Co-Packaged Optics Data Centers

Zhaoqi Qiu ¹ , Linya Peng ², Fuming Fan ^{3,4}, Haoran Zuo ³, Wenjie Qiu ³, Bo Xu ^{3,5} and Tianping Deng ^{3,*} 

¹ School of Electronic Information, Wuhan University, Wuhan 430072, China; qiu Zhaoqi@whu.edu.cn

² Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China; penglinya@mails.ccn.edu.cn

³ Hubei Key Laboratory of Internet of Intelligence, School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China; d202487126@hust.edu.cn (F.F.); zhr2002@hust.edu.cn (H.Z.); qwj1037229249@hust.edu.cn (W.Q.); xubosite@hust.edu.cn (B.X.)

⁴ Fiberhome Telecommunication Technologies Co., Ltd., Wuhan 430074, China

⁵ Hubei Chutianyun Co., Ltd., Wuhan 430074, China

* Correspondence: dengtp@hust.edu.cn

Abstract

As data-center interconnects move to co-packaged optics (CPO), high-power application-specific integrated circuits (ASICs) and heat-sensitive optical engines share a single interposer, and the resulting intra-module thermal coupling overwhelms conventional schedulers. Thermal-aware microservice directed acyclic graph (DAG) scheduling on CPO modules is recast here as a question of symmetry. Whereas a homogeneous-graph policy assumes the full node-permutation symmetry, that symmetry is broken twice: by the distinct task and processor node types, and by the asymmetric ASIC–engine coupling. We therefore propose a heterogeneous-graph Proximal Policy Optimization (PPO) scheduler in which the placement head is permutation-equivariant, the delay head is permutation-invariant, and the parameterization stays invariant to the processor count N . Because these symmetries hold by construction, the policy transfers zero-shot across module sizes. Heterogeneous edge typing and the resistor–capacitor (RC) coupling edge attribute are isolated by a six-test ablation chain. Evaluated on the Alibaba 2021 microservice trace across all module sizes and ambient regimes under the standard auto-cool budget, the proposed scheduler cuts the thermal-violation rate from roughly 98% under the Heterogeneous Earliest-Finish-Time (HEFT) heuristic to about 0.3%; at the hot operating point it lowers peak temperature by about 25% and raises DAG completion from about 26% to 100%, with the rare residual violations most frequent in the extreme-ambient band. With the env auto-cool budget disabled, a controlled single-axis comparison shows that removing the RC-coupling edge attribute raises the violation rate by over an order of magnitude, isolating its contribution. A single parameter set serves every N without retraining.

Keywords: permutation equivariance; symmetry-aware learning; size invariance; co-packaged optics; thermal-aware scheduling; deep reinforcement learning; heterogeneous graph neural network; microservice DAG scheduling



Academic Editor: Jian-Qiang Wang

Received: 28 May 2026

Revised: 25 June 2026

Accepted: 5 July 2026

Published: 13 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The exponential growth of large language model training, high-performance computing, and generative AI workloads has pushed modern hyperscale data centers to

fundamental physical limits in their interconnect subsystems [1]. Traditional pluggable optics architectures, which connect optical modules to switch application-specific integrated circuits (ASICs) through long printed-circuit board (PCB) traces, are approaching their signal integrity ceiling at per-lane speeds of 112 Gbps and beyond. Maintaining reliable transmission requires power-hungry digital signal processors (DSPs) and retimers, driving both the I/O power wall and end-to-end communication latency to unsustainable levels [2].

Co-packaged optics (CPO) has emerged as the consensus next-generation interconnect technology [3–5], integrating optical engines (OEs) and switch ASICs on a shared silicon interposer to improve interconnect energy efficiency from ~ 20 pJ/bit to below 5 pJ/bit [1]. Production deployments include Broadcom’s Tomahawk 5-Bailey [6] (1 ASIC + 8 OEs, 51.2 Tb/s) and Tomahawk 6-Davisson [7] (1 ASIC + 16 OEs, 102.4 Tb/s), the latter motivating the reference $N = 17$ configuration evaluated in this paper, with Section 5.4 reporting zero-shot generalization to $N \in \{9, 13, 17, 24, 33\}$. CPO is one facet of a broader trajectory in which computation and communication increasingly converge at the optical layer—from photonic accelerators now reaching 160 TOPS/W [8] to integrated optical-computing-communication networks proposed as a next-generation data-center fabric [9]—a paradigm in which on-die thermal coupling becomes a shared, first-order design concern.

This tight integration creates a severe thermal challenge: silicon photonic devices drift at ~ 0.1 nm/°C in resonance wavelength [10], while the co-packaged ASIC exceeds 1.5 W/mm² in power density [1] and induces intense thermal crosstalk into adjacent optical engines [2]. Thermal management in CPO systems becomes a first-order constraint on software-layer scheduling logic, not a cooling-engineering afterthought.

Thermal-aware task scheduling on a CPO module is a *sequential decision problem under a thermal-safety constraint* that jointly couples directed acyclic graph (DAG) precedence, heterogeneous ASIC/OE placement, and nonlinear leakage-temperature feedback on millisecond timescales. We cast it as a finite-horizon Markov Decision Process and shape the reward with a soft penalty on the peak-temperature constraint, so that the policy must respect the constraint over the full episode horizon and not merely at the moment of each dispatch. Thermal inertia, with resistor–capacitor (RC) time constants of ~ 244 ms (ASIC) and ~ 110 ms (OE), makes the binding constraint surface with a delay relative to the dispatch that causes it. Whether explicit anticipation adds value over placement-only scheduling, given the module’s reactive auto-cool budget, is an empirical question examined directly in Section 5.6.

Modern microservice workloads compound this challenge: cloud applications are decomposed into DAGs of fine-grained tasks whose dispatch pattern concentrates compute load onto specific processors over short windows [11,12]. Classical heuristic DAG schedulers such as Heterogeneous Earliest-Finish-Time (HEFT) [13] optimize solely for makespan and greedily route compute-intensive tasks to the highest-capacity node (the ASIC), oblivious to the resulting thermal hotspots; on CPO hardware they trigger hardware throttling or thermal emergency shutdowns, defeating the very latency advantage CPO was designed to deliver. In our evaluation regime, 9821 of 10,000 HEFT-scheduled episodes cross the $T_{\text{pen}} = 80$ °C safety threshold, motivating the thermally-aware scheduling formulation of this paper.

Deep reinforcement learning (DRL) for DAG scheduling, with graph neural network (GNN) encoders as the dominant policy backbone [14], has matured for cluster- and VM-scale workloads but does not directly address three properties of the CPO regime: (i) the processor side is *small but heterogeneous* (one high-power ASIC plus $N - 1$ OEs with distinct thermal capacitances); (ii) lateral thermal coupling between processors is a physically explicit RC matrix that can be supplied as edge attributes to a graph encoder, rather than being learned implicitly; and (iii) the CPO product family spans $N \in [9, 33]$ in

current designs, motivating an encoder whose forward pass is parametric in N for zero-shot size transfer.

These three properties are unified by a single principle: an effective policy must respect the *symmetry structure* of the CPO scheduling problem. The optical engines are exchangeable up to their physical state and coupling (relabeling them should permute the scheduler's per-engine placement decisions identically and leave its engine-agnostic delay decisions unchanged), while the module's defining asymmetry (one high-power ASIC against $N - 1$ engines on a star-plus-ring thermal graph) breaks the full node-permutation symmetry that a homogeneous-graph policy implicitly assumes. We design a policy that is permutation-equivariant in placement, permutation-invariant in delay, and invariant to the processor count N , and we argue (Section 4.4) that matching the policy's symmetries to the problem's makes a single policy well-defined and label-consistent across module sizes, a necessary precondition for the zero-shot transfer we then report empirically. Our contribution is not yet another equivariant architecture, as permutation-equivariant graph networks are by now standard, but rather the identification and exploitation of the specific symmetry that CPO scheduling *breaks*: typed task/processor heterogeneity over an asymmetric ASIC-plus-ring coupling graph that a single equivariant policy can then respect across module sizes.

In this paper, we propose a thermal-aware DAG scheduling framework for CPO data centers based on Proximal Policy Optimization (PPO) with a heterogeneous-graph encoder and a cross-attention factored actor. The architecture is *size-parametric*: a single trained policy operates on CPO modules of varying processor count N without retraining. Our contributions are as follows:

- **A physically grounded scheduling MDP coupled with an RC thermal network.** The full per-processor thermal field is exposed in the observation, and temperatures evolve under a discretized RC matrix with leakage-power feedback so the policy can internalise thermal inertia from reward signals rather than hand-tuned heuristics.
- **A heterogeneous-graph encoder with cross-attention factored actor.** The state is a heterogeneous graph with task and processor nodes and three edge relations (DAG dependencies, RC coupling, task-to-processor candidates carrying estimated thermal-rise as an edge feature). A two-layer GATv2 encoder produces fixed-dimension embeddings independent of N , enabling zero-shot generalization across processor counts. The actor decouples processor placement (cross-attention scoring) from an optional anticipatory pre-execution delay action (pooled-context MLP).
- **A controlled six-test ablation chain.** We decompose the safety gap along a six-test chain: HEFT $\rightarrow$ Decima-vanilla (homogeneous GCN + REINFORCE) $\rightarrow$ Decima-thermal (same encoder under our thermal reward); Decima-thermal $\rightarrow$ HGATE-PPO [15] (heterogeneous edge typing under the published recipe of Wu et al. [15]—reported as a recipe-level outcome, not an architectural verdict, because the reproduction exhibits early entropy collapse, see Section 5.3); Decima-thermal $\rightarrow$ D2 (cross-attention actor on a homogeneous trunk); Decima-thermal $\rightarrow$ Ours-NoThermal (the full proposed architecture); and Ours-NoThermal $\rightarrow$ Ours-auto_only (adding per-processor thermal observations on top of the thermal-aware reward). The chain isolates each design factor in turn; because the multi-axis links also vary the optimizer regime, the CPO-specific RC-coupling edge attribute is attributed by an optimizer-matched single-axis comparison (Ours-NoRCEdge, Section 5.6). Paired McNemar exact tests on the binary safety indicator, with Holm–Bonferroni family-wise control across the six tests, quantify the contribution of every component.
- **Empirical validation with explicit bounded claims.** Across paired evaluation on the Alibaba 2021 microservice trace, the proposed Ours-auto_only scheduler recorded

30 unsafe episodes per 10,000 paired runs across the full 5×4 size-by-ambient envelope ($N \in \{9, 13, 17, 24, 33\}$ processors, T_0 drawn from cold [25, 40], warm [40, 65], hot [60, 75], and extreme [70, 80] °C bands), at 100% DAG completion throughout. The 30 residual failures spanned all four ambient bands and were most frequent under the extreme ambient (5 cold, 5 warm, 5 hot, 15 extreme). Every classical heuristic baseline retained a violation rate above 0.94 across the same envelope, and our reproduction of HGATE-PPO under the published Wu et al. 2025 recipe and our paired-seed budget retained 0.97 (we attribute the high rate to recipe-level training instability and report it as a recipe-level outcome, not an architectural verdict; see Section 5.3). A single-axis isolation experiment (Section 5.6) further showed that the deployed Ours-auto_only policy was *policy-intrinsically safe* at hot $N = 17$ (formal definition in Section 3.4): removing the env’s lookahead auto-cool budget left its violation rate unchanged ($0.002 \rightarrow 0.002$; within-policy exact McNemar across cool = 100 vs cool = 0:1 discordant pair, $p = 1.000$). Under the same training recipe, removing the RC-coupling edge attribute (Ours-NoRCEdge) degraded to 0.056 (paired McNemar $p < 10^{-6}$). The envelope-wide rates reported in this paper are under the standard env auto-cool budget; whether the policy-intrinsic property extends across the full 5×4 envelope is left to future work.

2. Related Work

2.1. Classical Heuristic and DVFS-Based Schedulers

List-based heuristics, exemplified by HEFT [13] and its multi-objective extension MOHEFT (Multi-Objective HEFT) [16], rank DAG tasks by upward priority and place each on the processor that minimizes its earliest finish time, with MOHEFT producing makespan–energy Pareto sets on static cost tables. Anticipatory refinements such as PEFT [17] and the lookahead variant of Bittencourt et al. [18] score one step ahead to reduce makespan further. None of these methods reason about transient thermal state. A complementary line of work intervenes *after* a hot trajectory has been initiated, through DVFS throttling and workload migration [19,20]; such mechanisms are reactive and impose millisecond-to-second latency penalties that negate the low-latency advantage of CPO. Meta-heuristic methods (genetic, particle-swarm, simulated annealing) offer global makespan–thermal optimization but require seconds to minutes of iterative search per DAG, making them unsuitable for online scheduling.

2.2. Deep Reinforcement Learning for DAG Scheduling

DRL with graph-structured policies is now the dominant alternative to list-based heuristics. The canonical baseline is Decima [14], which couples a homogeneous graph convolutional network with REINFORCE for Spark-style DAG scheduling, achieving up to 21% job-completion-time reduction over hand-tuned policies. We used Decima as a reference architecture under two instantiations in our experiments: Decima-vanilla (original makespan-only reward, isolating the contribution of the RL backbone alone) and Decima-thermal (the same encoder under our thermal-aware reward, isolating the contribution of the reward signal on top of the homogeneous-GCN architecture). Subsequent work has elaborated the Decima paradigm along two axes. The first is more expressive graph encoders, including the deep-learning workflow scheduler of Chandrasiri and Mee-deniya [21], the multi-relation HGNN of Wan et al. [22] and HGNNR-FJSP [23] for flexible job-shop scheduling, the graph-encoder DRL scheduler of Sun et al. [24], the interface-matched dependency-graph GNN of Zhao et al. [25] for microservice orchestration, and the heterogeneous-GAT plus PPO architecture HGATE-PPO of Wu et al. [15] for vehicular cloud offloading; we reproduce HGATE-PPO as the closest architectural prior to ours

(heterogeneous edge typing with a different placement scorer and no RC-edge attribute). Pointer-style attention over a variable set originates in the pointer-network literature [26]; we use it through a cross-attention factored variant that places one task per decision step. More broadly, the permutation-equivariant graph encoders these methods rely on are instances of the geometric-deep-learning principle that a network should be equivariant to the symmetries of its input [27,28]; our aim was to identify and exploit the specific broken symmetry of the CPO scheduling problem rather than to propose a new equivariant layer. The second axis is multi-agent or hierarchical formulations [29,30], the latter exemplified by the hierarchical-DRL CPU scheduler of Xing et al. [31] for mixed database workloads; we do not pursue multi-agent control. Related microservice and mobile-edge scheduling formulations also target latency- and energy-constrained task placement, including the flexible offloading scheme of Sun et al. [32] for dynamic multi-access edge environments and the service-mesh orchestration line of Peng et al. [33–35], which includes a symmetry-based formulation of edge-cloud orchestration; none of these model the intra-module thermal coupling that is central here. A complementary strand explicitly couples DRL to thermal control: Yi et al. [19] target compute-intensive job allocation in data centers with DRL and a learned thermal model, while industrial systems TAPAS [36], D3T [37], and TAWS [38] demonstrate rack- and cluster-level thermal-aware scheduling for LLM inference, all targeting independent workloads rather than coupled microservice DAGs. Reactive throttling and migration represent a third strand, exemplified by Throttled-HEFT and similar policies that respond to temperature excursions *after* the placement is committed; we include Throttled-HEFT (hybrid mode) as a classical reactive-control baseline.

A separate body of work treats constraint satisfaction directly, via the constrained policy optimization of Achiam et al. [39], Reward-Constrained Policy Optimization (RCPO) [40], and the PID Lagrangian formulation of Stooke et al. [41]. These methods learn a feasible policy under a hard or expected-value constraint and apply most cleanly when the cost signal can be evaluated without simulator-side recovery. In our regime the thermal safety condition is enforced inside the env through a reward-shaping path (a soft penalty above T_{pen} and a hard penalty plus episode truncation above T_{crit} , Section 4.1.3) and through a placement-time lookahead auto-cool mechanism that we disclose and isolate explicitly in Section 5.6. A formal Lagrangian treatment of the safety constraint, with the auto-cool budget folded in as an explicit constraint multiplier, is left to future work.

2.3. Thermal Management of Co-Packaged Optics

CPO thermal management has historically been studied as a packaging-cooling problem. Wu et al. [1] characterized a 51.2 Tb/s microchannel-cooled CPO module via coupled CFD simulation and experimental validation, establishing the canonical limits taken as hard constraints in our work (ASIC junction below $\sim 100^\circ\text{C}$, inter-OE gradient below 4°C). Related photonic and packaging studies confirm millisecond-scale ASIC-to-OE crosstalk consistent with the hundred-millisecond time constants of our RC network ($\tau_{\text{ASIC}} \approx 244\text{ ms}$, $\tau_{\text{OE}} \approx 110\text{ ms}$, Section 3.2) [2,10]. Industry has more recently begun to characterize CPO design as a six-way coupled multi-physics problem spanning EMIR, timing, thermal, signal/power integrity, optical, stress, and aging analyzes [42], with concrete design-time tooling emerging on the thermal axis: Alam et al. [43] simulated the steady-state thermal field of a fixed 3DIC + CPO package and imported the resulting temperature map into FDTD-level photonic device models to predict grating-coupler and ring-modulator spectrum drift. These works held the workload fixed and designed the package; we hold the package fixed and design the operational dispatch policy that drives the thermal trace. The two perspectives are complementary rather than competing: none of the design-time tool-

ing exposes a knob at the operational scheduling layer where transient task-level decisions drive the heat trajectory.

2.4. Positioning and Gap Statement

The systems reviewed above each address a strict subset of the problem regime: graph-structured DAG scheduling without thermal state (Decima and its descendants), heterogeneous edge typing without CPO physics (HGATE-PPO), thermal-aware DRL at rack scale on independent jobs (Yi, TAPAS, D3T, TAWS), reactive throttling after the placement has been committed (DVFS-based and Throttled-HEFT variants), and design-time CPO thermal physics on static workloads (Wu, Alam). To the best of our knowledge, no prior work combines microservice-DAG scheduling, PPO with a heterogeneous-graph encoder and cross-attention factored actor, an RC-coupling edge relation supplied as an explicit edge attribute, and the CPO intra-module thermal regime in a single framework that generalizes zero-shot across module sizes $N \in \{9, 13, 17, 24, 33\}$.

3. System Model

We model a CPO module of N heterogeneous processors as a coupled power-thermal-workload system. In a co-packaged-optics module the optical engines are electro-optic transceiver dies rather than general-purpose compute cores; throughout this paper we use *processor* in the scheduling sense of an assignable, heat-generating active die—the switch ASIC or an OE transceiver—onto which a unit of microservice data-movement workload (switching and aggregation on the ASIC, per-flow electro-optic conversion on an OE) is dispatched for a bounded interval. The scheduler thus allocates the placement and timing of this workload across the dies under their mutual thermal coupling rather than floating-point computation. Figure 1 sketches the reference $N = 17$ topology used at training time (one 51.2 Tb/s switch ASIC n_0 plus sixteen 6.4 Tb/s COUPE-class OEs $n_1, \dots, n_{16}$ [6,44], co-packaged on a shared silicon interposer). All per-processor parameters introduced below are defined as functions of N , which is what permits zero-shot transfer of the trained policy across the $N \in \{9, 13, 17, 24, 33\}$ size sweep evaluated in Section 5.4.

3.1. Power Consumption Model

The instantaneous total power of node n_i is the sum of a temperature-dependent leakage term and a workload-driven dynamic term:

$$P_i^{\text{total}}(t) = P_i^{\text{leak}}(T_i(t)) + P_i^{\text{dyn}}(t), \quad P_i^{\text{leak}}(T_i) = P_i^{\text{leak},0} \exp(\beta (T_i - T_{\text{ref}})), \quad (1)$$

with $\beta = 0.015 \text{ K}^{-1}$ for 5 nm-class CMOS, baseline leakage $P_0^{\text{leak},0} = 30 \text{ W}$ (ASIC) and $P_{i>0}^{\text{leak},0} = 3 \text{ W}$ (OE), and with reference $T_{\text{ref}} = 25^\circ\text{C}$ [19]. Dynamic power is non-zero only on the node currently executing a task, $P^{\text{ASIC,dyn}} = 150 \text{ W}$ and $P^{\text{OE,dyn}} = 40 \text{ W}$, a $3.75\times$ asymmetry in active dynamic power between the ASIC and any single OE transceiver. The exponential leakage–temperature coupling closes a positive feedback loop: higher load raises T_i , which raises P_i^{leak} , which accelerates temperature rise. An effective scheduler must account for this nonlinearity rather than ignore it.

3.2. RC Thermal Network

We use a lumped-parameter RC equivalent network [1,2]: each node has thermal capacitance C_i and conductances G_i^{env} (to the cold plate at $T_{\text{amb}} = 25^\circ\text{C}$) and G_{ij}^{cross} (lateral coupling to neighbors):

$$C_i \frac{dT_i}{dt} = P_i^{\text{total}} - G_i^{\text{env}}(T_i - T_{\text{amb}}) - \sum_{j \neq i} G_{ij}^{\text{cross}}(T_i - T_j). \quad (2)$$

We discretize (2) by forward Euler with $\Delta t = 1$ ms, yielding the state-space form $\mathbf{T}(t + \Delta t) = \mathbf{A}\mathbf{T}(t) + \mathbf{B}\mathbf{P}(t) + \mathbf{D}T_{\text{amb}}$ used inside the scheduling loop. With the parameters of Table 1, the ASIC thermal time constant is $\tau_0 = C_{\text{ASIC}}/G_{\text{ASIC}}^{\text{eff}}$ with $G_{\text{ASIC}}^{\text{eff}} = G_{\text{ASIC}}^{\text{env}} + (N - 1)G_{\text{ASIC-OE}}^{\text{cross}}$ giving $\tau_0 \approx 244$ ms at the training $N = 17$ (ranging from ~ 303 ms at $N = 9$ to ~ 175 ms at $N = 33$ as the fan-out grows); the OE local time constant is $\tau_{\text{OE}} = C_{\text{OE}}/G_{\text{OE}}^{\text{eff}} \approx 110$ ms with $G_{\text{OE}}^{\text{eff}} = G_{\text{OE}}^{\text{env}} + G_{\text{ASIC-OE}}^{\text{cross}} + 2G_{\text{OE-OE}}^{\text{cross}}$. Both time constants are at least two orders of magnitude above Δt , so the forward-Euler scheme is stable across the eigenvalue spectrum.

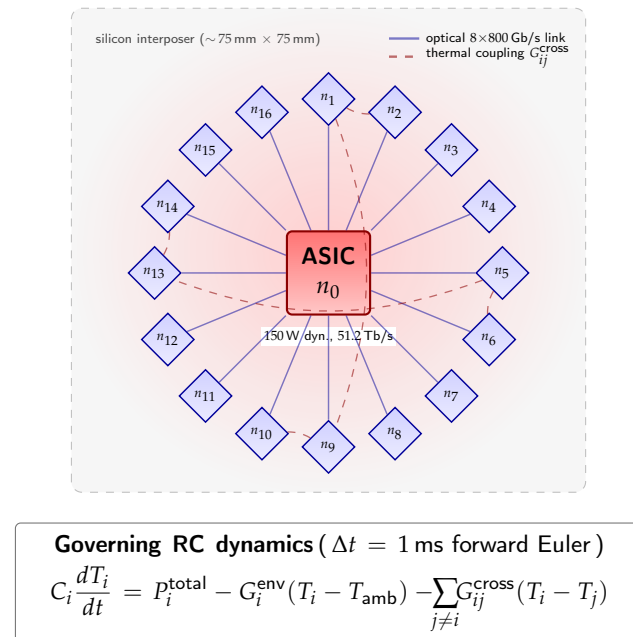


Figure 1. Reference $N = 17$ CPO module: one switch ASIC (n_0 , central red) plus sixteen optical engines (n_1 – n_{16} , blue diamonds) on a shared silicon interposer. Solid blue lines: optical links; dashed red curves: lateral thermal couplings G_{ij}^{cross} (the full coupling matrix is dense, only a few representatives drawn).

Table 1 lists the calibrated per-node parameters that fully determine the discretized system across all evaluated sizes N . The processor graph topology is a star plus a ring: the ASIC (n_0) couples to every OE ($G_{0j}^{\text{cross}} = G_{j0}^{\text{cross}} = G_{\text{ASIC-OE}}^{\text{cross}}$ for $j \geq 1$), and each OE couples to its two ring neighbors ($G_{ij}^{\text{cross}} = G_{ji}^{\text{cross}} = G_{\text{OE-OE}}^{\text{cross}}$ for $|i - j| \equiv 1 \pmod{N - 1}$, $i, j \geq 1$). All other G_{ij}^{cross} entries are zero. The same per-node constants are re-used at every $N \in \{9, 13, 17, 24, 33\}$; only the matrix dimension and the ASIC fan-out term $(N - 1)G_{\text{ASIC-OE}}^{\text{cross}}$ in $G_{0,0}^{\text{env}}$ scale with N .

Table 1. Calibrated RC-network parameters used at every N .

Symbol	Value	Description
C_{ASIC}	2.0 J/K	ASIC heat capacitance
C_{OE}	0.2 J/K	OE heat capacitance
$G_{\text{ASIC}}^{\text{env}}$	5.0 W/K	ASIC $\leftrightarrow$ cold plate
$G_{\text{OE}}^{\text{env}}$	1.5 W/K	OE $\leftrightarrow$ cold plate
$G_{\text{ASIC-OE}}^{\text{cross}}$	0.2 W/K	ASIC $\leftrightarrow$ each OE
$G_{\text{OE-OE}}^{\text{cross}}$	0.05 W/K	adjacent OE $\leftrightarrow$ OE
Δt	1 ms	forward-Euler step
T_{amb}	25 °C	cold-plate reference

Physical Justification and Calibration

The single-node-per-die lumped abstraction does not discard the package: each C_i and G_i^{env} is an *aggregate* that folds the off-die thermal mass and conduction path of node

i —substrate, lid, copper spreader, heat pipe, and, for the ASIC, the active heatsink—into one effective heat capacitance and one effective cold-plate conductance. This is the standard reduced-order model for system-level thermal *control*, where the controlled quantity is the per-die junction temperature on the $10\text{--}10^2$ ms scheduling timescale, not the intra-die spatial field that detailed package-geometry or finite-element models resolve. The constants of Table 1 are calibrated rather than free-fitted: C_i and G_i^{env} are set so that each die’s transient junction rise under its rated active power (Equation (1)) is $\approx 0.08^\circ\text{C/ms}$ for the ASIC and $\approx 0.18^\circ\text{C/ms}$ for an OE—rates consistent with published thermal characterizations of cold-plate-cooled integrated-photonics modules and within the $<100^\circ\text{C}$ junction budget of commercial CPO parts [1,2]—while the lateral G_{ij}^{cross} are set to a magnitude that yields a steady-state ASIC–OE gradient of the same order as the Tx–Rx thermal differentials reported for cold-plate-cooled silicon-photonics interposers under sustained transceiver load [45]. We further verified that the scheduling conclusions do not hinge on the precise calibrated values: a decoupled calibration-error sweep (Section 5.7), in which the thermal plant was perturbed by up to $\pm 30\%$ while the policy kept observing the nominal calibration, left the safety outcome essentially unchanged for perturbations within $\pm 10\%$ of the heat-balance parameters and fully robust against errors in the coupling strength.

3.3. Microservice DAG Task Model

A microservice workflow is a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each task v_i carries workload w_i (ms, normalized to ASIC speed) and slack $s_i = \text{LST}(v_i) - \text{EST}(v_i)$ from a standard forward/backward topological pass; tasks with $s_i = 0$ lie on the critical path. Each edge e_{ij} carries a data volume d_{ij} and contributes a communication delay $c_{ij} = d_{ij}/B + L^{\text{O/E}}$ when its endpoints are assigned to different processors, with CPO link bandwidth $B = 800\text{ Gb/s}$ and a fixed $L^{\text{O/E}} = 5\text{ }\mu\text{s}$ optical–electrical conversion delay. The DAGs used in this work are drawn from the Alibaba 2021 microservice trace [11], filtered to 3000 acyclic graphs with task counts in $[3, 20]$ and workloads in $[0.5, 50]$ ms; the makespan of a schedule π is the completion time of its last exit task.

3.4. Thermal Safety Constraints and Operational Definitions

CPO hardware specifications [1] impose a hard limit $T_i(t) \leq T_{\text{crit}} = 85^\circ\text{C}$ (violation is a thermal crash) and a soft advisory threshold $T_{\text{pen}} = 80^\circ\text{C}$ above which a continuous reward penalty engages (Section 4.1.3). Because the paper refers to “thermal safety” and “violation” in multiple operational senses, we fix terminology here for use throughout the rest of the manuscript (Table 2).

The underlying scheduling objective is the constrained makespan problem

$$\min_{\pi} \mathbb{E}[\text{Makespan}(\pi)] \quad \text{s.t.} \quad T_i(t) \leq T_{\text{crit}} \quad \forall i, t. \quad (3)$$

We do not solve (3) as a formal constrained MDP; the MDP formulation developed in Section 4 approximates it through a shaped reward that combines a soft penalty above T_{pen} , a hard penalty plus episode truncation above T_{crit} , and a completion bonus on makespan. Figure 2 illustrates the problem on a representative DAG and contrasts a HEFT-driven thermal trajectory against the proposed scheduler in one hot $N = 17$ episode.

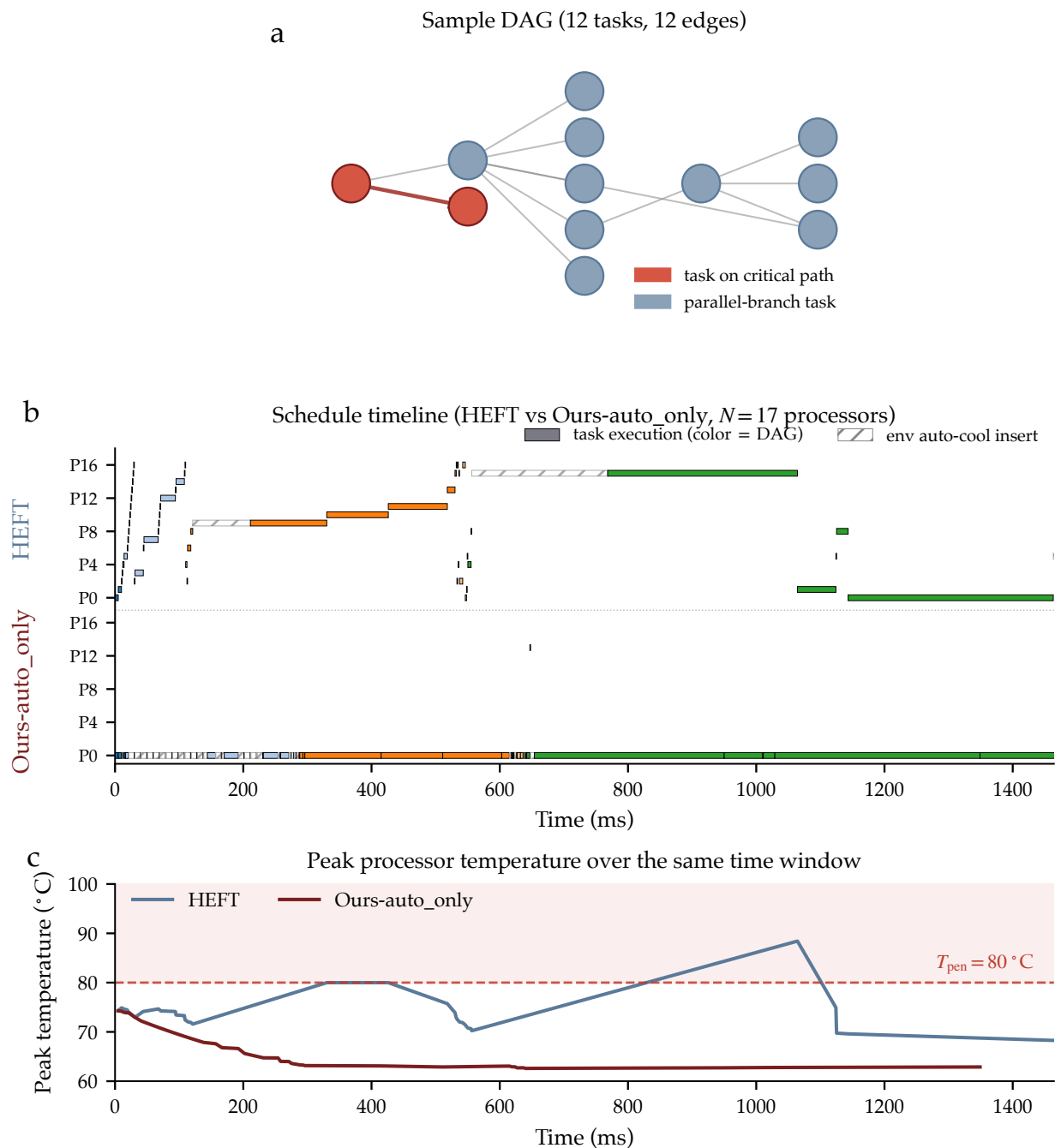


Figure 2. Problem at a glance: **(a)** A representative Alibaba 2021 microservice DAG drawn from the evaluation trace (12 tasks, 12 inter-task dependency edges). Node size encodes the per-task workload; the critical-path tasks are highlighted in red, the parallel-branch tasks in slate-blue. **(b)** Schedule timeline for the first ~ 1.5 s of one hot-ambient $N = 17$ episode (env seed shared between the two schedulers): HEFT [13] in the upper band distributes the workload uniformly across the processor ring and triggers two large env auto-cool inserts (hashed); Ours-auto_only in the lower band concentrates dispatches on a small set of processors and interleaves many short auto-cool inserts. Block colors in **(b)** encode the parent DAG of each task execution, matching the in-panel legend. **(c)** Peak processor temperature across the same time window. HEFT crosses the $T_{\text{pen}} = 80^\circ\text{C}$ threshold near $t = 1065$ ms and peaks at $\sim 88^\circ\text{C}$ inside the unsafe band (shaded); Ours-auto_only stays at $62\text{--}65^\circ\text{C}$ throughout. The scheduling decisions of Section 4 therefore control whether the same workload drives the chip into a violation regime or holds it at a comfortable thermal margin.

Table 2. Operational definitions used consistently throughout this paper. Subsequent sections reference these entries when ambiguity is possible.

Term	Operational Definition
<i>Unsafe episode</i>	An episode is <i>unsafe</i> if and only if at least one committed placement would, absent any cooling intervention (neither the agent’s anticipatory delay nor the env’s auto-cool), drive the peak processor temperature above $T_{\text{pen}} = 80^\circ\text{C}$ —the per-step <i>would-violate</i> condition, aggregated over the episode. We adopt this counterfactual placement-quality criterion rather than the realised post-cooling peak so that the metric attributes safety to the <i>placement policy</i> and not to the reactive cooling that backstops it; it is the more conservative of the two, additionally flagging near-boundary placements that cooling subsequently held just below T_{pen} , so the reported unsafe-episode rates upper-bound the realised-threshold-crossing rates (for Ours-auto_only, 30 versus 20 per 10,000 over the full envelope, the gap confined to the extreme band). This is the binary indicator used in every paired-McNemar test and every headline “violation rate” reported in Sections 5.2–5.6.
<i>Thermal crash</i>	The strictly stronger event $\max_i \max_t T_i(t) > T_{\text{crit}} = 85^\circ\text{C}$, which triggers episode truncation and a hard reward penalty (Section 4.1.3). Crash rates are uniformly lower than unsafe rates.
<i>Underlying constraint</i>	The hard constraint $T_i(t) \leq T_{\text{crit}}$ in Equation (3); we never solve it as a formal CMDP. Section 4 approximates it through reward shaping.
<i>Env auto-cool floor</i>	A simulator-side lookahead mechanism (default <code>max_cooling_steps</code> = 100 ms per dispatch) that inserts idle cooling before a placement when the predicted execution peak would cross T_{pen} . Independent of any scheduler.
<i>Policy-intrinsic safety</i>	The property, established for a specific checkpoint via Section 5.6, that the unsafe-episode rate is unchanged or improved when the env auto-cool floor is removed (<code>max_cooling_steps</code> = 0). <i>Ours-auto_only</i> carries this label at hot $N = 17$ (as does its identical hybrid extension), the cell where the auto-cool ablation was run (Section 5.6). The envelope-wide rates of Sections 5.2–5.4 are reported under the standard auto-cool budget; whether the policy-intrinsic property extends across the full 5×4 envelope is left to future work.
<i>Size-parametric</i>	The encoder forward pass is parametric in N : parameter tensor shapes depend on d and the head count only, not on N , so a single trained policy produces a valid forward pass at every N without retraining. Zero-shot transfer across N is the <i>empirical</i> consequence reported in Section 5.4.

4. Proposed Method

We cast thermal-aware DAG scheduling as a finite-horizon Markov Decision Process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with $\gamma = 0.99$, in which the thermal-safety requirement is enforced through a shaped reward (a soft penalty above T_{pen} and a hard penalty plus episode truncation above T_{crit} , Section 4.1.3), not as an explicit constraint, and we learn a stationary policy π_θ via PPO [46] maximizing $J(\theta) = \mathbb{E}_{\pi_\theta} [\sum_{k=0}^{H-1} \gamma^k r_{t+k}]$, where H is the episode length, bounded above by `dags_per_episode` and below by the thermal-truncation condition. Section 4.1 defines the state, action, and reward; Sections 4.2 and 4.3 describe the heterogeneous-graph encoder and the cross-attention factored actor; Section 4.5 summarizes the PPO training loop. A central architectural commitment is *size-parametric generalization*: all module dimensions are independent of the processor count N , enabling the zero-shot size-transfer result of Section 5.4. Figure 3 gives a panel-level overview of the architecture, the training curriculum, and the deployment modes.

4.1. State, Action, and Reward

4.1.1. State as a Heterogeneous Graph

The state at step t is a graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ comprising two node types and three edge relations:

- **Task nodes** ($\mathcal{V}_\tau$): one per currently-pending DAG task. Each has a 4-dimensional feature vector $\mathbf{x}_i^\tau \in \mathbb{R}^4$ encoding normalized workload, slack, ready-flag, and current-task indicator.
- **Processor nodes** ($\mathcal{V}_p$): one per active die (the switch ASIC plus the $N-1$ OE transceivers). Each has a 7-dimensional feature vector $\mathbf{x}_j^p \in \mathbb{R}^7$ comprising busy flag, remaining-time, type indicator (ASIC vs OE), and four *thermal observables*: T_j , $\dot{T}_j$, leakage current, and headroom $T_{\text{crit}} - T_j$.
- **Task–task edges** ($\mathcal{E}_{\tau\tau}$): one per DAG dependency, with edge attribute $\mathbf{e}_{ij}^{\tau\tau} \in \mathbb{R}^1$ encoding normalized inter-task traffic.
- **Processor–processor edges** ($\mathcal{E}_{pp}$): fully connected, with edge attribute $\mathbf{e}_{ij}^{pp} \in \mathbb{R}^1$ derived from the RC-coupling matrix A_{ij} of Section 3.2. This relation lets the encoder

propagate thermal coupling structure to placement decisions. Such a property is absent from any homogeneous-graph encoding.

- **Task–processor edges** ($\mathcal{E}_{\tau p}$): bipartite cross-relation from each task to each candidate processor, with edge attribute $\mathbf{e}_{ij}^{\tau p} \in \mathbb{R}^2$ encoding the estimated execution time of task i on processor j and the estimated thermal rise that placement would cause. The estimated ΔT is the second feature dimension that Section 5.1's controlled ablation Ours-NoThermal removes.

A homogeneous graph would collapse these two relations into one edge type with a unified attribute, losing the geometry-vs-combinatorics distinction; the heterogeneous formulation lets the encoder learn distinct propagation dynamics for the two phenomena.

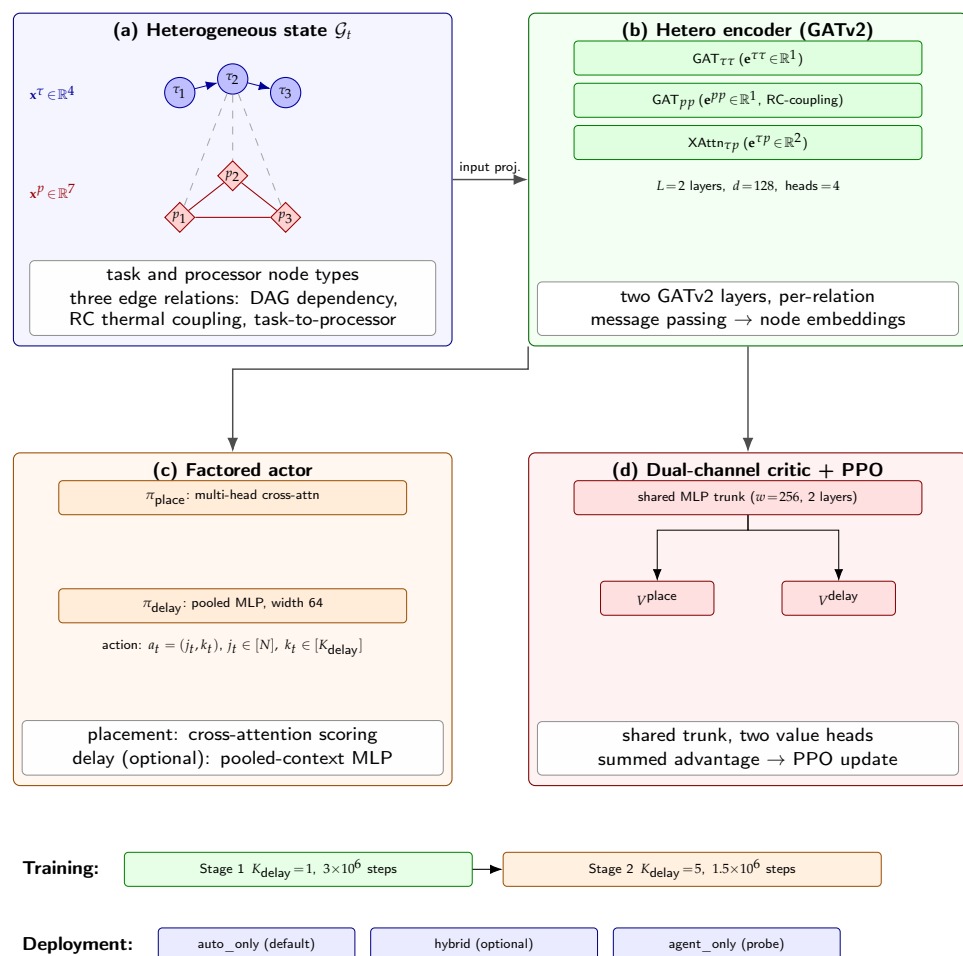


Figure 3. Method framework: Panel (a): the heterogeneous state graph $\mathcal{G}_t$ comprises task nodes (blue circles) and processor nodes (red diamonds) with three edge relations (DAG dependencies, RC coupling, task-to-processor candidates); the lateral processor coupling carries the calibrated conductance G_{ij}^{cross} of Equation (2) as a per-edge attribute. Panel (b): the hetero encoder applies $L = 2$ layers of GATv2 message passing on each relation, producing per-node embeddings of dimension $d = 128$. Panel (c): the factored actor uses cross-attention scoring for the placement head and a pooled-context MLP for the optional delay head (the explicit logit and message-passing forms are given in the text of Section 4). Panel (d): the dual-channel critic predicts v_{place} and v_{delay} from a shared trunk, and the two GAE advantages are summed for the PPO loss (Section 6.1.2). The bottom strip summarizes the two-stage training curriculum (Section 4.5) and the three deployment modes (Section 4.5.4).

4.1.2. Action Space

The action is the product of a placement head and a delay head:

$$\mathcal{A} = \mathcal{A}_{\text{place}} \times \mathcal{A}_{\text{delay}} = \{1, \dots, N\} \times \{0, 1, \dots, K_{\text{delay}} - 1\}. \quad (4)$$

At each decision step the agent emits a tuple $(j, k) \in \mathcal{A}$: j is the target processor for the current task, and k is a discrete pre-execution delay level that pauses dispatch by $k \cdot \Delta_{\text{delay}}$ milliseconds. We use $K_{\text{delay}} = 5$ and $\Delta_{\text{delay}} = 80$ ms throughout, so the maximum agent-issued delay is 320 ms. Figure 4 traces the corresponding forward pass at the tensor level.

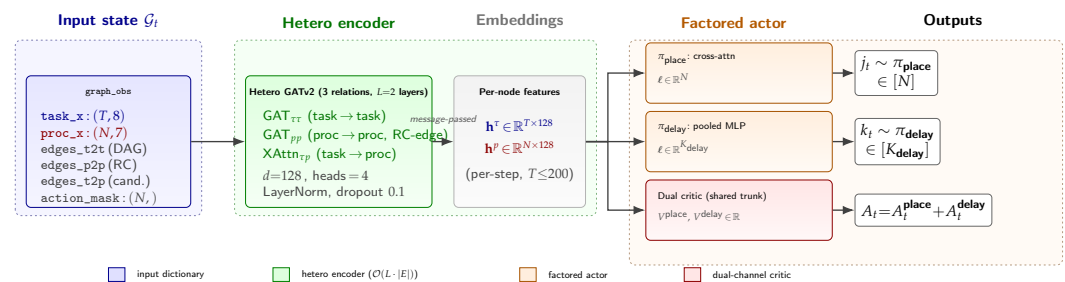


Figure 4. Forward-pass pipeline at one decision step: Concrete tensor-shape view of Section 4, complementing the equation-driven panel layout of Figure 3. The input `graph_obs` dict contains task features $\mathbf{x}^t \in \mathbb{R}^{T \times 8}$, processor features $\mathbf{x}^p \in \mathbb{R}^{N \times 7}$, three typed edge sets (DAG dependencies, RC-coupling, task-to-processor candidates), and an N -dimensional action mask. The hetero encoder applies three GATv2-style relation-specific layers (task–task, proc–proc with the RC-coupling edge attribute, task-to-proc cross-attention) of width $d = 128$ with four heads, yielding per-node embeddings $\mathbf{h}^t \in \mathbb{R}^{T \times 128}$ and $\mathbf{h}^p \in \mathbb{R}^{N \times 128}$. The factored actor emits placement logits $\ell_{\text{place}} \in \mathbb{R}^N$ (masked cross-attention scoring of the current task against every processor candidate) and delay logits $\ell_{\text{delay}} \in \mathbb{R}^{K_{\text{delay}}}$ (pooled MLP over a global context vector). The dual-channel critic predicts V^{place} and V^{delay} from the same global context; their sum drives the clipped PPO surrogate during training. At inference time the action $a_t = (j_t, k_t)$ is obtained by argmax over each head independently (deterministic mode) or by sampling (stochastic mode); the dispatch is then placed on the chosen processor.

Two masking layers are applied before the placement softmax: dependency masking forbids processors with unfinished predecessor tasks, and thermal-safe filtering forbids processors whose pre-task temperature plus the estimated ΔT would exceed the hard wall T_{crit} . The delay head is unmasked.

4.1.3. Reward Function

The reward at step t combines a completion bonus, a makespan-driving term, a soft thermal penalty, a hard violation indicator, and a dense-progress bonus:

$$r_t = r_{\text{base}} - \alpha \cdot \frac{\tau_t}{w_{\text{max}}} - \beta \cdot \max(0, T_{\text{max},t} - T_{\text{pen}}) - \delta \cdot \mathbb{I}[T_{\text{max},t} > T_{\text{crit}}] + r_{\text{bonus},t}, \quad (5)$$

where τ_t is the task completion time including communication delay, $T_{\text{max},t} = \max_j T_j(t)$ is the peak processor temperature after the step, $r_{\text{base}} = 1.0$ rewards each valid task completion, $\alpha = 4.0$ penalizes execution time, $\beta = 0.2$ activates progressively as temperature exceeds $T_{\text{pen}} = 80^\circ\text{C}$, $\delta = 50.0$ imposes a hard penalty whenever any processor exceeds $T_{\text{crit}} = 85^\circ\text{C}$ (thermal crash), and $r_{\text{bonus},t}$ is a sparse DAG-completion shaping term defined as

$$r_{\text{bonus},t} = b_{\text{exit}} \cdot \mathbb{1}[\text{exit task of a DAG completes at } t] + b_{\text{step}} \cdot \mathbb{1}[\text{any task completes at } t], \quad (6)$$

with $b_{\text{exit}} = 2.0$ and $b_{\text{step}} = 0.05$. The hyper-parameter values are stable across training seeds; we did not tune them.

The reward decomposes additively into a *placement-driven* component (the makespan and bonus terms) and a *thermal-driven* component (the soft penalty and the hard violation); this decomposition informs the dual-critic implementation choice of Section 6.1.2, which is adopted on engineering grounds (channel-decomposed advantage estimation matches the channel-decomposed reward) rather than ablated against a single-critic baseline.

4.2. Heterogeneous Graph Encoder

The encoder transforms the per-step heterogeneous graph $\mathcal{G}_t$ into per-node embeddings $\mathbf{h}_i^\tau, \mathbf{h}_j^p \in \mathbb{R}^d$ with $d = 128$, with all module shapes independent of N . Inputs are first mapped from raw feature space to the latent dimension via per-type input projections:

$$\mathbf{h}_i^{\tau,(0)} = \text{LN}(\text{GELU}(\text{Linear}_4(\mathbf{x}_i^\tau))), \quad \mathbf{h}_j^{p,(0)} = \text{LN}(\text{GELU}(\text{Linear}_7(\mathbf{x}_j^p))). \quad (7)$$

The body of the encoder is a stack of $L = 2$ identical layers, each running three message-passing operations *in parallel* on $\mathbf{h}^{\tau,(\ell)}$ and $\mathbf{h}^{p,(\ell)}$:

$$\Delta \mathbf{h}_{\tau\tau,i}^{\tau,(\ell)} = \text{GAT}_{\tau\tau}(\mathbf{h}^{\tau,(\ell)}, \mathbf{e}^{\tau\tau})_i, \quad (8)$$

$$\Delta \mathbf{h}_{pp,j}^{p,(\ell)} = \text{GAT}_{pp}(\mathbf{h}^{p,(\ell)}, \mathbf{e}^{pp})_j, \quad (9)$$

$$(\Delta \mathbf{h}_{\tau p,i}^{\tau,(\ell)}, \Delta \mathbf{h}_{\tau p,j}^{p,(\ell)}) = \text{XAttn}_{\tau p}(\mathbf{h}^{\tau,(\ell)}, \mathbf{h}^{p,(\ell)}, \mathbf{e}^{\tau p})_{ij}. \quad (10)$$

The three update streams reach each node type with disjoint roles ($\tau\tau$ delivers DAG-dependency context to tasks, pp delivers RC-coupling context to processors, τp delivers candidate-affinity context to both). Per-layer outputs combine these deltas additively, with a residual connection and LayerNorm:

$$\mathbf{h}_i^{\tau,(\ell+1)} = \text{LayerNorm}(\mathbf{h}_i^{\tau,(\ell)} + \Delta \mathbf{h}_{\tau\tau,i}^{\tau,(\ell)} + \Delta \mathbf{h}_{\tau p,i}^{\tau,(\ell)}), \quad (11)$$

$$\mathbf{h}_j^{p,(\ell+1)} = \text{LayerNorm}(\mathbf{h}_j^{p,(\ell)} + \Delta \mathbf{h}_{pp,j}^{p,(\ell)} + \Delta \mathbf{h}_{\tau p,j}^{p,(\ell)}). \quad (12)$$

$\text{GAT}_{\tau\tau}$ and GAT_{pp} are *GATv2Conv* layers [47] with edge attributes, 4 attention heads, and concatenation projection back to dimension d ; $\text{XAttn}_{\tau p}$ is a bidirectional *GATv2* update on the bipartite task-processor relation, using the same hyper-parameters. Dropout $p = 0.1$ is applied between layers during training. We deliberately use *GATv2* rather than the original *GAT* [28] because *GATv2*'s dynamic attention is required for edge-attribute-conditioned attention scoring, which is essential here: the placement attention must depend on the thermal-rise estimate carried in $\mathbf{e}^{\tau p}$, not just on the node embeddings.

The two-layer depth is the smallest setting that gives the placement decision two-hop access to both the RC-coupling and the DAG-dependency structure; $L \geq 3$ did not improve training return in our setting and increases inference latency linearly. All module weights depend on d and head count alone, not on N , so the encoder is size-parametric: the same trained weights transfer zero-shot across N (Section 5.4).

4.3. Factored Actor and Dual-Channel Critic

The actor produces two outputs given the encoder embeddings: the placement logits over the N processors and the delay logits over the K_{delay} levels. A dual-channel critic

shares the same encoder and consumes a pooled global context to emit two scalar value heads aligned with the placement and thermal components of the reward.

4.3.1. Placement Head

The placement head is a multi-head scaled dot-product attention in which the query is the current task's embedding and the keys are the processor embeddings. We use $H = 4$ heads of width $d_h = d/H$. Per head $h \in \{1, \dots, H\}$,

$$\ell_{\text{place}}^{(h)} = \frac{(\mathbf{h}_{\text{cur}}^\tau W_Q^{(h)})(\mathbf{h}^p W_K^{(h)})^\top}{\sqrt{d_h}}, \quad W_Q^{(h)}, W_K^{(h)} \in \mathbb{R}^{d \times d_h}, \quad (13)$$

producing N -dimensional per-head logits. Heads are summed (rather than concatenated through an output projection) to give the final placement logits $\ell_{\text{place}} = \sum_{h=1}^H \ell_{\text{place}}^{(h)}$, which then pass through the action mask described in Section 4.1.2 before the softmax:

$$\pi_{\text{place}}(j | s) = \text{softmax}(\ell_{\text{place}} + \mathbf{m})_j, \quad \mathbf{m}_j = \begin{cases} 0 & \text{if } j \text{ is feasible,} \\ -\infty & \text{otherwise.} \end{cases} \quad (14)$$

We use the attention scores *directly as logits* rather than constructing a separate value projection and re-projecting; this “pointer-attention” pattern follows Decima [14] and avoids one redundant linear layer per decision step.

4.3.2. Delay Head

The delay head is a small MLP that takes a pooled-context vector formed from the current-task embedding, the mean processor embedding, and the maximum processor embedding (along the feature dimension):

$$\ell_{\text{delay}} = \text{MLP}(\mathbf{h}_{\text{cur}}^\tau \parallel \overline{\mathbf{h}^p} \parallel \max_j \mathbf{h}_j^p), \quad \pi_{\text{delay}}(k | s) = \text{softmax}(\ell_{\text{delay}})_k. \quad (15)$$

The max-pool is the architectural detail that surfaces hot-spot information to the delay head: a single processor near T_{pen} is more relevant for delay decisions than the average processor, which the mean-pool would dilute. The MLP has hidden width 64.

4.3.3. Joint Policy

The two heads define a factorised policy:

$$\pi_\theta(a | s) = \pi_{\text{place}}(j | s) \pi_{\text{delay}}(k | s), \quad a = (j, k). \quad (16)$$

For inference modes that disable the delay action (Section 4.5.4), the delay head is deterministically clipped to $k = 0$.

4.3.4. Dual-Channel Critic

The critic shares the encoder with the actor and ingests a pooled global context formed from the same task, mean-processor, and max-processor channels used by the delay head:

$$\mathbf{c}_t = \overline{\mathbf{h}_t^\tau} \parallel \overline{\mathbf{h}_t^p} \parallel \max_j \mathbf{h}_{j,t}^p. \quad (17)$$

A two-layer MLP trunk ϕ_θ of hidden width 256 then feeds two scalar linear heads, one for the credit-assignment target of each action head:

$$V_t^{\text{place}} = w_{\text{place}}^\top \phi_\theta(\mathbf{c}_t), \quad V_t^{\text{delay}} = w_{\text{delay}}^\top \phi_\theta(\mathbf{c}_t). \quad (18)$$

V_t^{place} regresses against the discounted return under the placement-driven reward channel (makespan + DAG-completion bonus); V_t^{delay} regresses against the discounted return under the thermal-driven reward channel (soft penalty above T_{pen} and hard penalty plus episode truncation above T_{crit} , Section 4.1.3). The shared trunk amortizes the encoder forward across the two channels; only the final linear heads $w_{\text{place}}, w_{\text{delay}} \in \mathbb{R}^{256}$ are channel-specific. The two channels generate two GAE advantages whose sum drives the PPO update in Section 4.5. Algorithm 1 summarizes the resulting forward pass at one decision step.

Algorithm 1: Forward pass at one decision step. The sampling notation $\sim$ corresponds to training-time stochastic rollouts; at evaluation the heads are deterministic, with $j_t = \arg \max_j \pi_{\text{place}}(j | s_t)$ and $k_t = \arg \max_k \pi_{\text{delay}}(k | s_t)$.

Input: Encoder E_θ , actor heads $(\pi_{\text{place}}, \pi_{\text{delay}})$, dual critic $(V^{\text{place}}, V^{\text{delay}})$, state $s_t = \mathcal{G}_t$, action mask $\mathbf{m}_t$, current task index i^* , deployment mode $\mu \in \{\text{auto_only}, \text{hybrid}, \text{agent_only}\}$

Output: Action $a_t = (j_t, k_t)$, log-probability $\log \pi_\theta(a_t | s_t)$, values $(V_t^{\text{place}}, V_t^{\text{delay}})$

```

// Step 1: encoder forward -- produce per-node embeddings
1  $(\mathbf{h}^\tau, \mathbf{h}^p) \leftarrow E_\theta(\mathcal{G}_t)$  via input proj. Equation (7) and  $L = 2$  message-passing layers
   Equations (8)–(12)

// Step 2: placement head -- cross-attention scoring
2  $\ell_{\text{place}} \leftarrow$  multi-head scaled dot-product (Equation (13)) between query  $\mathbf{h}_{i^*}^\tau$  and
   processor keys  $\mathbf{h}^p$ 
3  $\ell_{\text{place}} \leftarrow \ell_{\text{place}} + \mathbf{m}_t$  // apply dependency + thermal-safe mask
4  $j_t \sim \pi_{\text{place}}(\cdot | s_t) = \text{softmax}(\ell_{\text{place}})$ 

// Step 3: delay head -- pooled-context MLP
5 if  $\mu = \text{auto\_only}$  then
6    $k_t \leftarrow 0$  // delay disabled in mode auto_only
7 end
8 else
9    $\ell_{\text{delay}} \leftarrow \text{MLP}(\mathbf{h}_{i^*}^\tau \| \overline{\mathbf{h}^p} \| \max_j \mathbf{h}_j^p)$  (Equation (15))
10   $k_t \sim \pi_{\text{delay}}(\cdot | s_t) = \text{softmax}(\ell_{\text{delay}})$ 
11 end

// Step 4: dual critic -- predict the two value channels
12  $\mathbf{c}_t \leftarrow (\overline{\mathbf{h}^\tau} \| \overline{\mathbf{h}^p} \| \max_j \mathbf{h}_j^p)$  (Equation (17))
13  $(V_t^{\text{place}}, V_t^{\text{delay}}) \leftarrow w_{\text{place}}^\top \phi_\theta(\mathbf{c}_t), w_{\text{delay}}^\top \phi_\theta(\mathbf{c}_t)$  (Equation (18))
14  $a_t \leftarrow (j_t, k_t)$ 
15  $\log \pi_\theta(a_t | s_t) \leftarrow \log \pi_{\text{place}}(j_t) + \log \pi_{\text{delay}}(k_t)$ 
16 return  $a_t, \log \pi_\theta(a_t | s_t), V_t^{\text{place}}, V_t^{\text{delay}}$ 

```

4.4. Symmetry Properties of the Policy

The encoder and actor of Sections 4.2 and 4.3 are constructed so that the policy respects the symmetry structure of the scheduling problem. We make three properties explicit, because they are the structural reason a single trained policy transfers across module sizes without retraining.

4.4.1. Permutation Equivariance of Placement

Let σ be any permutation of the N processor nodes, acting jointly on the processor features and on every incident edge of the proc–proc (RC-coupling) and task–proc relations. The heterogeneous encoder is built entirely from per-node feature projections, LayerNorm, GELU, residual connections, and GATv2 message passing [28,47], each of which commutes with node relabeling, so the encoder is permutation-equivariant by construction [27]. The placement head scores each processor by a dot product between the current-task query and that processor’s key (Section 4.3), and the action mask is itself indexed by processor; relabeling the processors by σ thus permutes the placement logits by the same σ . Equivalently, the probability the policy assigns to dispatching onto a given physical engine is independent of the order in which engines are enumerated.

4.4.2. Permutation Invariance of Delay

The anticipatory-delay head reads only mean- and max-pooled processor context (Section 4.3), both symmetric functions of the processor set. The delay distribution is hence permutation-invariant: how long to wait before dispatch does not depend on processor labeling, matching the physical semantics of the delay action.

4.4.3. Size Invariance

No parameter tensor in the encoder or either head has a dimension that scales with N : shapes depend only on the embedding width d , the head count, and the fixed feature dimensions (Section 4.2). A single trained parameter set therefore induces a valid, label-consistent forward pass at every N . Equivariance makes that forward pass well-defined across sizes; whether it also yields strong safety at unseen N is an empirical question, answered by the zero-shot transfer across $N \in \{9, 13, 17, 24, 33\}$ (Section 5.4).

4.4.4. Symmetry Breaking and Heterogeneity

A homogeneous-graph policy such as Decima [14] treats every node as one type, implicitly assuming the full symmetric group S_{T+N} acts on the joint task–processor set. The CPO problem violates that assumption: tasks and processors are not interchangeable, and the ASIC and optical engines differ in both role and power (the ASIC dissipates $3.75 \times$ the dynamic power of any single engine, Section 3.1). Our encoder instead respects only the product group $S_T \times S_N$, being equivariant to relabelings within each node type by typing task and processor nodes separately. The physical coupling structure, namely the star-plus-ring topology of Section 3.2, is not part of the architecture’s symmetry group; it is supplied through the processor–processor edge set and its RC-coupling attributes, which the equivariant encoder exploits without hard-coding any processor’s identity (the ASIC/engine distinction enters as an input feature, not a positional index). The six-test ablation chain of Section 5.3 is best read not as a symmetry ordering but as isolating this thermal-physics inductive bias: the controlled single-axis comparison Ours-NoRCEdge (Section 5.6) identifies the RC-coupling edge attribute, rather than equivariance alone, as the design element that drives the safety margin.

4.5. PPO Training and Two-Stage Curriculum

We train with the standard clipped PPO objective [46] (clip $\epsilon = 0.2$, value coefficient $c_v = 0.5$, entropy bonus $c_e = 0.01$, GAE $\lambda = 0.95$) and the joint advantage $A_t = A_t^{\text{place}} + A_t^{\text{delay}}$ when both heads are active. Training proceeds in two stages: *Stage 1* (placement only, K_{delay} clipped to 0, 3×10^6 steps, learning rate 3×10^{-4} , reward-curriculum transitions cold $\rightarrow$ warm $\rightarrow$ hot at $\{4 \times 10^5, 1.5 \times 10^6\}$ steps) produces the deployable Ours-auto_only checkpoint; *Stage 2* (full action space, 1.5×10^6 steps, learning rate 1×10^{-4} , warm-started

from the Stage 1 checkpoint with a freshly initialized delay head) produces the optional Ours-hybrid checkpoint. The two-stage decomposition matches the structure of the action space: the delay head conditions on a context that is only well-defined once the placement head has learned the size-parametric thermal-aware policy. During Stage 1 the delay-head parameters and the V^{delay} critic head receive no gradient (the delay action is clipped to $k_t = 0$, so $\log \pi_{\text{delay}}$ contributes zero to the surrogate and A_t^{delay} is dropped from the joint advantage); both are initialized fresh at the start of Stage 2.

4.5.1. Common Training Hyper-Parameters

All PPO-based policies (Ours, Ours-NoThermal, D2, HGATE-PPO) share Adam ($\epsilon = 10^{-5}$, linear LR decay to zero), num_envs = 16, rollout_length = 256 (4096 transitions per update), ppo_epochs = 4, num_minibatches = 8 (minibatch 512), PPO clip $\epsilon = 0.2$, $c_v = 0.5$, $c_e = 0.01$, GAE $\lambda = 0.95$, max_grad_norm = 0.5, encoder hidden width $d_h = 128$, critic trunk width 256, dropout 0.1, and seed 42 for the rollout RNG. Decima-vanilla and Decima-thermal use REINFORCE with a moving-average baseline (Algorithm 1 of Mao et al. [14]), Adam ($\epsilon = 10^{-5}$, lr 5×10^{-4} , no scheduler), and one gradient update per episode; the only difference between the two is the reward signal. The HGATE-PPO trainer [15] performs single-env per-step PPO updates per the author code convention rather than batched rollouts; we flag this when reading Component 3. Per-policy training budgets and learning rates are listed in Table 3. All learned baselines are reported on a single training seed (42), so the paired McNemar comparison characterizes sampling variance over the 500 paired evaluation episodes per cell *conditional on* this seed, not over independent re-runs of the training pipeline. Component 6 ($p = 0.625$, n.s.) is the comparison most affected by this caveat; the other five components have effect sizes that remain significant in absolute magnitude.

Table 3. Per-policy training budgets and learning rates: “Steps” is total environment steps. Common PPO hyper-parameters (batch composition, clip, GAE, optimizer) are shared across all PPO-trained rows; only the algorithm-axis differences are reported here. Training budgets are not uniform across the ablation chain (2.0–5.0 M steps): each baseline is trained to its return-curve plateau under its own optimizer regime (Section 4.5.1), partially mitigating architecture-vs-compute confounding but not eliminating it.

Policy	Steps	LR	Notes
Ours Stage 1	3.0 M	3×10^{-4}	placement only ($K_{\text{delay}} = 0$)
Ours Stage 2	1.5 M	1×10^{-4}	warm-started; delay head re-init
Ours-NoThermal	2.0 M	3×10^{-4}	strips T_{norm} , $\hat{T}$, leakage, headroom
Ours-NoRCEdge	3.0 M	3×10^{-4}	RC edge attribute ablated
D2	5.0 M	2×10^{-4}	homog. GCN trunk + Ours actor head
HGATE-PPO	5.0 M	3×10^{-4}	single-env per-step PPO [15]
Decima-vanilla	5.0 M	5×10^{-4}	REINFORCE + moving-avg baseline
Decima-thermal	5.0 M	5×10^{-4}	REINFORCE + moving-avg baseline

4.5.2. Training and Evaluation Compute Budget

The total training compute consumed was approximately 43 GPU-hours on a single NVIDIA V100 or A100: ~ 6 h for Stage-1 Ours (3×10^6 steps), ~ 3 h for Stage-2 hybrid (1.5×10^6 steps), ~ 4 h for Ours-NoThermal (2×10^6 steps), ~ 6 h for Ours-NoRCEdge (3×10^6 steps), ~ 9 h for D2 (5×10^6 steps), ~ 9 h for HGATE-PPO (5×10^6 steps in single-env mode), and ~ 3 h each for Decima-vanilla and Decima-thermal under REINFORCE-with-moving-average-baseline. Evaluation was CPU-only (Apple M3 Pro, Apple Inc., Cupertino, CA, USA; single-thread for paired determinism); the 120,000-episode main matrix plus the horizon stress test, the 2200-episode auto-cool ablation, and the 1000-episode hot+extreme

RC-edge isolation consumed approximately 48 CPU-hours end-to-end. All training and evaluation seeds, configs, and checkpoint hashes were recorded with the released code. Figure 5 shows the Stage-1 and Stage-2 training-return curves for Ours-auto_only and Ours-hybrid.

4.5.3. Training-Time Strategies

Three minor strategies improve convergence beyond standard PPO: (i) a curriculum on the initial-temperature distribution that ramps from cold ($T_0 \in [25, 40]^\circ\text{C}$) through warm ($[40, 65]^\circ\text{C}$) to hot ($[60, 75]^\circ\text{C}$) at the Stage-1 boundaries; (ii) both the dependency mask and the thermal-safe candidate mask are applied during training rather than only at inference; and (iii) best checkpoints are selected on hot-stage validation return, evaluated every 5×10^4 training steps over 30 deterministic-argmax episodes drawn from a validation seed pool that is disjoint from the evaluation seed_base = 100,000 pool of Section 5.1, with a rolling-window-50 maximum used to debounce noise in the hot-stage validation return curve.

4.5.4. Three Inference Modes

The trained policy is deployed in three configurations: auto_only (Stage-1 checkpoint, delay clipped to zero)—the primary deployed configuration of this paper—hybrid (Stage-2 checkpoint, full action space, adding the optional anticipatory delay), and agent_only (Stage-2 checkpoint with the env auto-cool fallback disabled, used in Section 5.6 to probe policy-intrinsic safety). All configurations evaluate under deterministic argmax. As Section 5.6 shows, once the module’s reactive auto-cool budget is calibrated the anticipatory-delay extension is safety-neutral—hybrid matches auto_only under deterministic inference—so we report auto_only as the headline configuration and treat the delay head as an optional component.

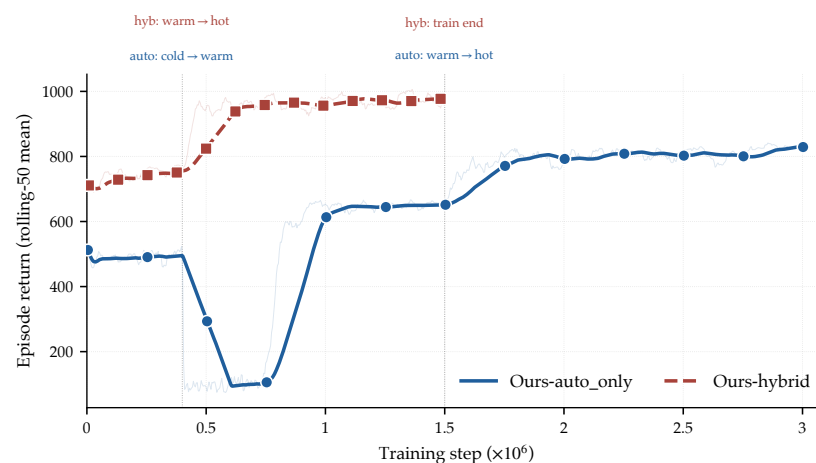


Figure 5. PPO training reward curves at $N = 17$. Episode return is reported as a rolling-50 mean over the raw training-time samples (faint thin lines). The two vertical dotted guides at 4×10^5 and 1.5×10^6 steps mark curriculum transitions that have *different* meanings for the two runs (the in-plot labels are color-coded by run accordingly). Ours-auto_only (solid blue, circle markers) trains from scratch under the Stage-1 curriculum cold $\rightarrow$ warm $\rightarrow$ hot, so the two guides delimit those three stages and the temporary return drop near 7×10^5 steps coincides with the cold $\rightarrow$ warm initial-temperature shift; the policy recovers and converges to ~ 830 by 3 M steps. Ours-hybrid (dashed red, square markers) is warm-started from the Stage-1 best .pt and runs Stage-2’s two-stage curriculum warm $\rightarrow$ hot over 1.5 M steps (the cold stage is skipped because the warm-start already passed it), so for this run 4×10^5 marks the warm $\rightarrow$ hot transition and 1.5×10^6 marks training termination; Ours-hybrid converges smoothly to ~ 980 inside its budget.

5. Experiments

We report the empirical evaluation of the proposed scheduler against 11 other scheduler configurations on the Alibaba 2021 microservice trace, together with a Lagrangian constrained-PPO baseline evaluated separately in Section 5.8. Section 5.1 states the experimental setup. Section 5.2 presents the main hot-ambient results. Section 5.3 decomposes the design contributions along a six-test ablation chain. Section 5.4 reports the size, ambient, and horizon generalization envelope. Section 5.5 examines the optional anticipatory-delay extension; Section 5.7 reports robustness to thermal-calibration error; and Section 5.8 compares against a Lagrangian constrained-PPO baseline.

5.1. Evaluation Setup

All the schedulers were evaluated on the Alibaba 2021 production microservice trace. The evaluation matrix spanned five processor counts $N \in \{9, 13, 17, 24, 33\}$ and four ambient regimes (cold $[25, 40]^\circ\text{C}$, warm $[40, 65]^\circ\text{C}$, hot $[60, 75]^\circ\text{C}$, extreme $[70, 80]^\circ\text{C}$), with 500 paired episodes per cell ($n = 500$, $\text{seed_base} = 100,000$). Episode k refers to the same DAG, the same initial-temperature vector, and the same inter-arrival time draw across every scheduler. The thermal guardband is $T_{\text{pen}} = 80^\circ\text{C}$ and the hard truncation threshold is $T_{\text{crit}} = 85^\circ\text{C}$.

The schedulers compared in this section are: HEFT, Thermal-HEFT, Round Robin, Throttled-HEFT (hybrid mode, the production-default; the *agent_only* variant is omitted from the headline tables for compactness and trends with the hybrid numbers within sample-size noise), Decima-vanilla and Decima-thermal (Mao et al. [14] reproductions trained from scratch with makespan-only and thermal-aware rewards respectively; *reproduction* refers to the homogeneous-GCN trunk and the REINFORCE-with-moving-average-baseline training procedure, not to the original Spark workload), HGATE-PPO [15] (Wu et al. 2025 reproduction), D2 (a composition baseline combining Decima's homogeneous GCN trunk with our cross-attention placement actor of Section 4.3), Ours-NoThermal (the proposed architecture trained without per-processor thermal observation features while retaining the thermal-aware reward), Ours-auto_only (Stage-1 placement-only checkpoint), and Ours-hybrid (Stage-2 checkpoint with the anticipatory delay action active). The evaluation matrix consisted of 5 size cells ($N \in \{9, 13, 17, 24, 33\}$) $\times$ 4 ambient cells (cold, warm, hot, extreme) $\times$ 12 scheduler configurations = 240 cells in the size-by-ambient sweep, plus a horizon stress test that ran an extended $H = 200$ schedule on the schedulers reported in Section 5.4 (the $H = 20$ columns of the horizon table are inherited from the main sweep). At 500 paired episodes per cell the size-by-ambient sweep totaled 120,000 paired episodes; the paired auto-cool ablation of Section 5.6 added an additional $4 \times 500 + 200 = 2200$ episodes at $\text{max_cooling_steps}=0$ (four schedulers at the $n = 500$ pool, plus Ours-NoThermal on its $n = 200$ sub-pool).

5.1.1. Hardware and Runtime

Our evaluations ran on a CPU to eliminate GPU non-determinism in the paired comparison. Training of the learned baselines used a single NVIDIA V100 or A100 card under the per-policy budget and schedule of Section 4.5.1. Components 3, 4 and 5 of the ablation chain crossed an optimizer-regime boundary as well as an architectural one (HGATE-PPO used single-env PPO per the author code convention; Decima-vanilla and Decima-thermal used REINFORCE with one gradient update per episode), and were interpreted with that caveat. Component 2 (Decima-vanilla $\rightarrow$ Decima-thermal) was within-REINFORCE and optimizer-clean.

5.1.2. Metrics

We report five operational metrics, all aggregated over the 500 paired episodes per cell: (i) total makespan in milliseconds; (ii) peak processor temperature in °C; (iii) violation rate, defined as the fraction of episodes in which the peak temperature crosses T_{pen} ; (iv) DAG completion rate, defined as the fraction of the 20 DAGs dispatched per episode that completed before episode truncation occurred (the denominator was the dispatched-DAG count, not the episode count); (v) episode return, the cumulative reward under the thermal-aware reward shaping of Section 4.1.3.

5.1.3. Statistical Methodology

Significance is reported via the paired McNemar exact test on the binary safety indicator (an episode was unsafe if and only if $T_{\text{peak}} \geq T_{\text{pen}}$), applied to the six-test ablation chain in Section 5.3. Pairing protocol: for every cell (N, T_{ambient}) each episode $i \in \{0, \dots, 499\}$ was assigned the integer seed $\text{seed}(i) = \text{seed_base} + i$ with $\text{seed_base} = 100,000$ across the entire main matrix. Each scheduler reset the environment under $\text{seed}(i)$ via `env.reset(seed=seed(i))` before episode i , which seeded the env's single internal RNG that drove the DAG sampling, the initial-temperature draw, and the inter-arrival schedule from one common state. Identical seeds therefore produced identical physical trajectories across schedulers paired on episode i . The auto-cool ablation (Section 5.6) re-used the same $\text{seed_base}=100,000$ pool: the four full-pool schedulers (HEFT, Decima-thermal, Ours-NoRCEdge, Ours-auto_only) replayed episodes $i \in \{0, \dots, 499\}$, and Ours-NoThermal at $\text{max_cooling_steps}=0$ replayed the first 200 episodes $i \in \{0, \dots, 199\}$ of the same pool. The McNemar discordant counts n_{01} and n_{10} thus indexed the same physical conditions under different policies. Family-wise error was controlled at $\alpha = 0.05$ via the Holm–Bonferroni step-down procedure across the six tests of Section 5.3. The auto-cool ablation tests of Section 5.6 are reported as uncorrected exploratory comparisons outside this family. Markers * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ applied to Holm-adjusted p -values inside the six-test family and to raw p -values otherwise. We adopted argmax (deterministic) inference at evaluation time for reproducibility; sampling-based evaluation may surface delay-head behavior that argmax masks at the standard $H = 20$ horizon, a caveat we acknowledge in Section 6.3.

5.2. Main Results

Table 4 reports the headline cross-scheduler comparison at $N = 17$ in the hot ambient regime. The Ours-auto_only configuration achieved a violation rate of 0.002 (one unsafe episode out of 500), full DAG completion on every paired episode, and the lowest peak temperature among all the evaluated schedulers. The optional anticipatory-delay extension (Ours-hybrid) is analyzed separately in Section 5.5: under deterministic inference its delay head selected zero delay, so it coincided with Ours-auto_only and is not tabulated as a separate row here. Removing the per-processor thermal observation features (Ours-NoThermal) while retaining the thermal-aware reward raised the violation rate marginally to 0.006 while leaving DAG completion at 1.000.

Table 4. Main results on the hot-ambient validation set ($T_{\text{ambient}} \in [60, 75]^\circ\text{C}$, $N = 17$ processors, $n = 500$ paired episodes per row, $\text{seed_base} = 100,000$). The Ours-auto_only row, in bold, is the primary deployed configuration and the terminus of the ablation chain of Section 5.3; Ours-NoThermal is its thermal-observation ablation. The optional anticipatory-delay extension (Ours-hybrid) coincides with Ours-auto_only under deterministic inference and is analyzed separately in Section 5.5. The headline view above contains eight of the ten displayed schedulers; Round Robin and Decima-vanilla are present in the ablation chain (Table 5) and the full size-by-ambient sweep (Tables 6 and 7) but are omitted from this headline view for compactness, and neither would change the ranking observable here. The Throttled-HEFT agent_only variant was evaluated under the same protocol; it tracks the hybrid numbers within sample-size noise (Section 5.1) and is omitted from all displayed tables. Arrows in the header indicate the preferred direction ($\downarrow$ lower is better, $\uparrow$ higher is better).

Scheduler	Makespan (ms) $\downarrow$	Peak T ($^\circ\text{C}$) $\downarrow$	Viol. Rate $\downarrow$	Completion $\uparrow$	Episode Return $\uparrow$
HEFT	1243 $\pm$ 854	91.11 $\pm$ 3.98	0.986	0.260	+60.81
Thermal-HEFT	1697 $\pm$ 1101	89.67 $\pm$ 4.32	0.968	0.337	+86.93
Throttled-HEFT	1278 $\pm$ 864	91.08 $\pm$ 3.99	0.984	0.261	+60.67
Decima-thermal	3958 $\pm$ 2456	77.96 $\pm$ 10.29	0.382	0.741	+243.45

Table 4. Cont.

Scheduler	Makespan (ms) $\downarrow$	Peak T ($^\circ\text{C}$) $\downarrow$	Viol. Rate $\downarrow$	Completion $\uparrow$	Episode Return $\uparrow$
HGATE-PPO	2023 $\pm$ 1423	89.31 $\pm$ 4.63	0.968	0.401	+113.49
D2	3664 $\pm$ 2613	78.54 $\pm$ 11.05	0.478	0.697	+227.52
Ours-NoThermal	5317 $\pm$ 2042	68.62 $\pm$ 3.81	0.006	1.000	+339.00
Ours-auto_only	5317 $\pm$ 2042	68.33 $\pm$ 4.07	0.002	1.000	+339.19

The makespan column requires interpretation against the completion column. For HEFT, Round Robin, Throttled-HEFT, and HGATE-PPO the mean makespan reflects time to the first thermal truncation rather than time to DAG completion: in those rows, fewer than half of the dispatched DAGs finished before the episode ended. For the Ours family and Ours-NoThermal, every paired episode completed, so the makespan was end-to-end. The thermally-aware methods appear slower in this column because their episodes completed the full horizon, and the cross-family comparison on makespan alone is not meaningful. We treat violation rate and DAG completion as the primary correctness metrics throughout. Figure 6 plots the safety–makespan trade-off across the eight schedulers and traces the resulting Pareto frontier.

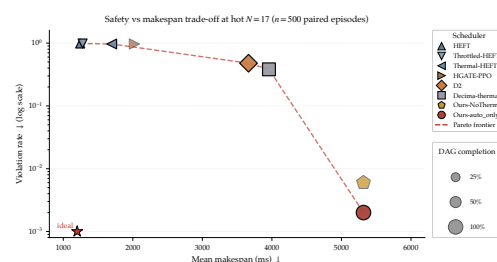


Figure 6. Safety vs makespan trade-off for the eight schedulers of Table 4 at hot $T_{\text{ambient}} \in [60, 75]^\circ\text{C}$ and $N = 17$ processors ($n = 500$ paired episodes per row). Marker area is an illustrative encoding of DAG completion rate (exact values in Table 4); markers on the Pareto frontier (dashed red line) have a heavier black border. The Pareto frontier traces HEFT $\rightarrow$ Thermal-HEFT $\rightarrow$ D2 $\rightarrow$ Decima-thermal $\rightarrow$ Ours-NoThermal $\rightarrow$ Ours-auto_only, with each transition trading a step up in makespan for a step down in violation rate. The Ours family occupies the safety-optimal corner of the trade-off at the cost of $\sim 4\times$ the classical-heuristic makespan, but the completion-rate caveat of the paragraph above qualifies this comparison: classical heuristics do not finish most DAGs in this regime, so their faster makespan number is not measured on the same workload.

5.3. Six-Test Ablation Chain

Table 4 decomposes the contribution of the design choices that distinguish our policy from heuristic and prior-RL baselines along six paired comparisons. We refer to each row of the chain as “Component N ” for prose convenience; the *chain* contains six tests, each isolating one or more axes. The chain spans four design factors: the RL backbone, the thermal-aware reward, heterogeneous edge typing, and the CPO-specific RC-coupling edge attribute. It is not a clean ordering by symmetry structure. Heterogeneous typing alone (HGATE-PPO, Component 3) did not improve safety over the homogeneous-GCN baselines under our reproduction, and the largest single jump (Component 5) simultaneously changed the optimizer regime from REINFORCE to PPO. The cleanest attribution came instead from the optimizer-matched single-axis comparison Ours-NoRCEdge (Section 5.6), which isolated the RC-coupling edge attribute: the thermal-physics inductive bias that distinguishes our encoder from generic heterogeneous-GAT prior work. Table 5 reports the Holm–Bonferroni-corrected McNemar p -values for each link; Figure 7 plots the per-configuration violation rates underlying the chain.

Table 5. Six-test ablation chain at $N = 17$ in the hot regime ($n = 500$ paired episodes, family-wise $\alpha = 0.05$ via Holm–Bonferroni). Component 5 jointly introduces the four architectural axes of the proposed encoder/actor as a bundle (see Section 5.6 for the single-axis RC-edge isolation); Component 6 then tests the further contribution of per-processor thermal observation features.

Test (Link)	Viol. Rate Change	Holm p	Sig
1. HEFT $\rightarrow$ Decima-vanilla	0.986 $\rightarrow$ 1.000	0.031	*
2. Decima-vanilla $\rightarrow$ Decima-thermal	1.000 $\rightarrow$ 0.382	$<10^{-4}$	***
3. Decima-thermal $\rightarrow$ HGATE-PPO	0.382 $\rightarrow$ 0.968	$<10^{-4}$	***
4. Decima-thermal $\rightarrow$ D2	0.382 $\rightarrow$ 0.478	0.029	*
5. Decima-thermal $\rightarrow$ Ours-NoThermal	0.382 $\rightarrow$ 0.006	$<10^{-4}$	***
6. Ours-NoThermal $\rightarrow$ Ours-auto_only	0.006 $\rightarrow$ 0.002	0.625	n.s.

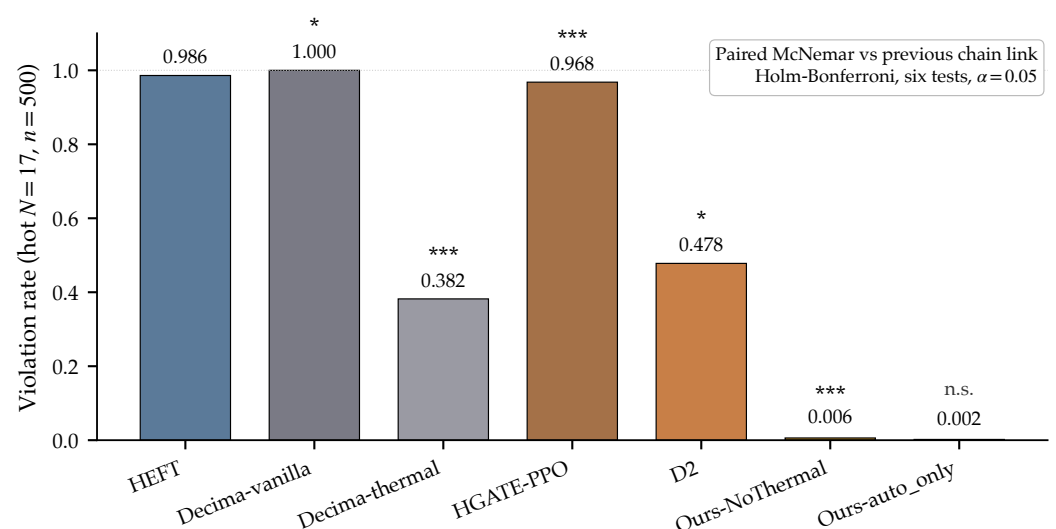


Figure 7. Violation rate of the seven scheduler configurations in the hot $N = 17$ regime ($n = 500$ paired episodes per bar). The six components of Section 5.3 read against this view: Components 2 and 5 are the two large paired-significant safety improvements, Components 3 and 4 are paired-significant regressions from Decima-thermal under partial-architecture substitutions (HGATE-PPO and D2 respectively), and Component 6 (Ours-NoThermal $\rightarrow$ Ours-auto_only) is the only chain link that is not statistically detectable at $n=500$. Stars: paired McNemar exact test vs. the previous link under Holm–Bonferroni control (* $p < 0.05$, *** $p < 0.001$). The RC-coupling-edge-attribute contribution within the Component 5 four-axis bundle is isolated separately in Section 5.6.

5.3.1. Component 1: RL Backbone Alone Does Not Help

The Decima-vanilla configuration (homogeneous GCN with REINFORCE training under a makespan-only reward) recorded a violation rate of 1.000, statistically worse than HEFT's 0.986 ($p = 0.031$). A plain RL backbone without thermal information is not a useful substitute for the heuristic in this regime.

5.3.2. Component 2: Thermal Reward Shaping Is the First Component to Move Safety

Adding the thermal-aware reward shaping of Section 4.1.3 to the same homogeneous GCN policy (Decima-thermal) lowered the violation rate from 1.000 to 0.382 ($p < 10^{-4}$). DAG completion rose from 0.32 to 0.74 and peak temperature dropped by 12.7 °C. The thermal reward channel is what makes a homogeneous-GCN policy non-trivially safer than the heuristic.

5.3.3. Component 3: The Published HGATE-PPO Recipe Does Not Close the Gap in Our Regime

HGATE-PPO [15] replaced the homogeneous GCN with a heterogeneous GATv2 encoder over typed (task, processor) nodes; it performed placement scoring via GAT-based attention over candidate destinations and contained no CPO-specific thermal-coupling edges. Under the hyper-parameters and training budget of Wu et al. [15], the violation rate rose from 0.382 (Decima-thermal) to 0.968 ($p < 10^{-4}$); HGATE-PPO policy entropy collapsed early in the curriculum, and the rolling-mean best checkpoint never improved beyond the cold-stage policy. We therefore report this row as a recipe-level outcome of the published reproduction rather than an architectural verdict on heterogeneous edge typing in general; a dedicated hyper-parameter sweep for HGATE-PPO on this workload is left to future work.

5.3.4. Component 4: Cross-Attention Placement on a Homogeneous Trunk Also Fails to Close the Gap

D2 substituted our cross-attention placement actor of Section 4.3 for the MLP scorer while retaining Decima's homogeneous GCN trunk and the thermal-aware reward. The violation rate increased from 0.382 (Decima-thermal) to 0.478 ($p = 0.029$). This was a paired-significant regression: the cross-attention placement actor was not by itself a sufficient architectural fix to the homogeneous trunk.

5.3.5. Component 5: The Proposed Architecture Delivers Near-Optimal Safety Even Without Per-Processor Thermal Observations, Provided the Training Reward Remains Thermal-Aware

Ours-NoThermal carried the same heterogeneous GAT encoder, the RC-coupling edge attribute, and the cross-attention placement actor as Ours-auto_only, and shared the identical thermal-aware reward shaping of Section 4.1.3; the only training-time change was that the per-processor thermal observation features and the `est_temp_rise` task-to-processor edge attribute were stripped from the policy input. The violation rate dropped from 0.382 (Decima-thermal) to 0.006 ($p < 10^{-4}$), a 64-fold reduction; DAG completion reached 1.000. The architecture together with the thermal-aware reward signal was sufficient to recover almost all of the safety gain of the full configuration, even without per-processor thermal observations in the policy input. Component 5 attributed this gain to the *joint* introduction of four architectural axes on top of Decima-thermal: the heterogeneous GATv2 trunk, the typed task/proc node structure, the cross-attention placement actor, and the RC-coupling edge attribute on proc-proc edges. The dual-channel critic of Section 4.3.4 was also introduced in this bundle as an unablated implementation choice grounded in the additive reward decomposition. We discuss in the caveat paragraph below what this

chain did and did not isolate; the RC-coupling edge attribute exposed the cross-processor thermal-coupling structure of (2) directly on the message-passing graph, and the heterogeneous edge typing distinguished that physical coupling from the DAG dependency and task-to-processor candidate edges that shared the graph. The Component 5 jump was the largest single safety improvement in the chain.

5.3.6. Component 6: Adding Per-Processor Thermal Observation Features Yields Only a Marginal Improvement

On top of the Ours-NoThermal configuration, Ours-auto_only restored the per-processor temperature features (T_{norm} , dT/dt , leakage, headroom) on the proc node feature vector and the est_temp_rise attribute on task-to-processor edges; the thermal-aware reward shaping was unchanged between the two configurations. The violation rate moved from 0.006 to 0.002, but this difference was not statistically significant after Holm correction ($p = 0.625$). Three Ours-NoThermal episodes per 500 were unsafe where Ours-auto_only was safe; one Ours-auto_only episode per 500 was unsafe where Ours-NoThermal was safe. The architectural choice of how thermal physics entered the graph encoder appeared to be the dominant contribution; explicit per-processor temperature features in the observation were a second-order refinement, with the caveat that all learned baselines were trained on a single seed (Section 4.5.1).

5.3.7. Aggregated Reading

Five of the six chain components were paired-significant under Holm–Bonferroni: Component 1 (HEFT $\rightarrow$ Decima-vanilla) was a small regression ($p = 0.031$); Component 2 (Decima-vanilla $\rightarrow$ Decima-thermal) was the largest single improvement ($p < 10^{-4}$); Components 3 (Decima-thermal $\rightarrow$ HGATE-PPO) and 4 (Decima-thermal $\rightarrow$ D2) were paired-significant regressions away from the Ours-style architecture, indicating that neither hetero-typing on a generic GATv2 trunk alone nor a cross-attention actor on a homogeneous GCN trunk alone is sufficient; Component 5 (Decima-thermal $\rightarrow$ Ours-NoThermal) was the largest paired-significant improvement and the dominant finding of the chain. Only Component 6 (Ours-NoThermal $\rightarrow$ Ours-auto_only) was not detectable at this sample size. Components 2 and 5 together attributed most of the safety claim to two factors: thermal-aware reward shaping, and the joint proposed architecture (heterogeneous trunk + cross-attention actor + RC-coupling edge attribute) trained under the same reward. The specifically-RC-edge contribution within the Component-5 four-axis bundle is isolated separately in the auto-cool ablation of Section 5.6.

5.3.8. What This Chain Does and Does Not Isolate

Component 3 (Decima-thermal $\rightarrow$ HGATE-PPO) and Component 4 (Decima-thermal $\rightarrow$ D2) demonstrated that, on top of the same thermal-aware reward, heterogeneous edge typing on a generic GATv2 trunk alone and the cross-attention placement actor on a homogeneous GCN trunk alone are each *individually insufficient* to recover the Ours-NoThermal violation rate. Component 5 (Decima-thermal $\rightarrow$ Ours-NoThermal) then jointly introduced the heterogeneous GATv2 trunk, the typed task/proc node structure, the cross-attention placement actor, and the RC-coupling edge attribute on proc–proc edges. The chain therefore attributed the safety gap to the *combination* of these axes rather than to any single one in isolation. The marginal contribution of the RC-coupling edge attribute itself is isolated separately in Section 5.6 via an Ours-NoRCEdge training run (same architecture, RC-coupling attribute ablated on proc–proc edges only); Section 6.1 discusses the underlying inductive-bias rationale.

5.4. Generalization Envelope

The $N = 17$ hot-ambient slice reported above is one cell of the full 5×4 size-by-ambient evaluation matrix. This subsection reports the Ours family behavior across the remaining 19 cells and under a tenfold horizon stress test.

5.4.1. Ambient Envelope

Table 6 reports the violation rate at $N = 17$ across the four ambient regimes. Every Ours-family configuration (Ours-NoThermal and Ours-auto_only) maintained a violation rate at or below 0.006 in all four regimes. Every heuristic and prior-RL baseline recorded a violation rate above 0.94 in all four regimes except Decima-thermal, which alternated between conservative behavior at extreme ambient (violation rate 0.042) and partial failure at the intermediate regimes (0.374 to 0.536); the conservative behavior at extreme ambient reflected a 4% truncation rate, not a 96% safety rate, since 2.7% of the Decima-thermal episodes at extreme ambient also failed to complete the dispatched DAGs. Figure 8 visualizes the same envelope as a side-by-side heatmap for HEFT and the deployed Ours-auto_only policy.

Table 6. Violation rate across the four ambient regimes at $N = 17$, $n = 500$ paired episodes per cell. All Ours-family configurations maintain a violation rate at or below 0.006 across the entire 25 to 80 °C ambient envelope; Ours-auto_only (bold) is the primary deployed configuration.

Scheduler	Cold [25, 40]	Warm [40, 65]	Hot [60, 75]	Extreme [70, 80]
HEFT	0.972	0.978	0.986	0.992
Round Robin	0.974	0.980	0.986	0.992
Thermal-HEFT	0.950	0.958	0.968	0.976
Throttled-HEFT (hybrid)	0.972	0.978	0.984	0.992
Decima-vanilla	0.996	0.998	1.000	1.000
Decima-thermal	0.374	0.536	0.382	0.042
HGATE-PPO	0.970	0.974	0.968	0.984
D2	0.000	0.042	0.478	0.656
Ours-NoThermal	0.000	0.000	0.006	0.006
Ours-auto_only	0.002	0.002	0.002	0.006

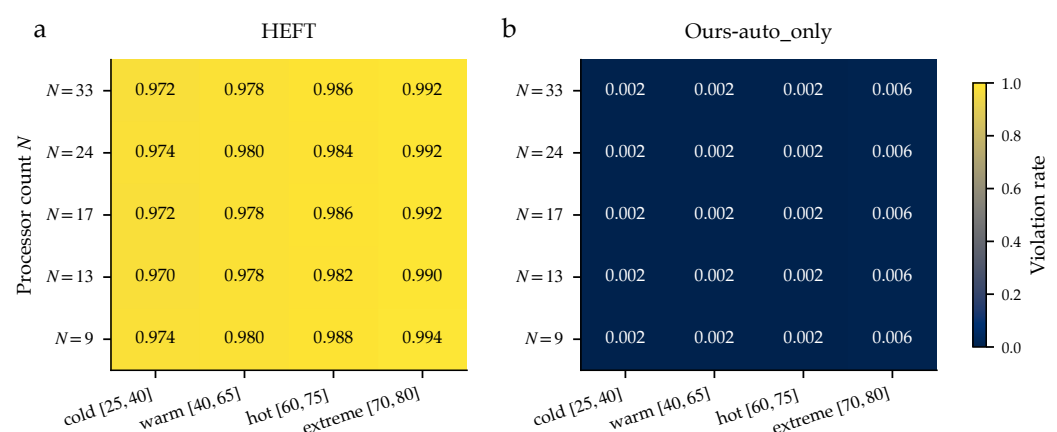


Figure 8. Violation rate across the full 5×4 size-by-ambient envelope: HEFT (a) versus the deployed Ours-auto_only policy (b); $n = 500$ paired episodes per cell, identical seed_base across the two panels. HEFT's violation rate stays above 0.97 at every cell; Ours-auto_only's stays at 0.002 across the cold, warm, and hot cells and rises to 0.006 in the extreme band. Thirty residual unsafe episodes appear in the right panel (1 in each cold, warm, and hot cell and 3 in each extreme cell, summing to 30/10,000); their concentration in the high-ambient band is the empirical content of the bounded safety claim of Section 6.3.

The D2 row shows a distinct pattern. The D2 violation rate was 0.000 at cold ambient, rose through 0.042 (warm) and 0.478 (hot), and reached 0.656 at extreme ambient. The cross-attention placement actor on a homogeneous GCN trunk was competitive at low thermal stress but degraded sharply as the ambient envelope tightened. The Ours family did not exhibit this degradation, which we attribute to the RC-coupling edge attribute supplying the encoder with the cross-processor thermal-coupling structure that the homogeneous GCN trunk was forced to infer indirectly.

5.4.2. Processor Count Scaling

Table 7 reports the violation rate across $N \in \{9, 13, 17, 24, 33\}$ in the hot regime. Ours-auto_only maintained a violation rate of 0.002 at every N ; Ours-NoThermal stayed at 0.006 across $N \in \{9, 13, 17, 24\}$ and dropped to 0.002 at $N = 33$. Therefore, a single trained policy generalized across the full size range without size-specific retraining: this was the size-parametric property claimed in Section 1, evaluated on the operational metric directly.

Table 7. Violation rate across processor count $N \in \{9, 13, 17, 24, 33\}$ in the hot-ambient regime, $n = 500$ paired episodes per cell.

Scheduler	$N = 9$	$N = 13$	$N = 17$	$N = 24$	$N = 33$
HEFT	0.988	0.982	0.986	0.984	0.986
Round Robin	0.986	0.986	0.986	0.986	0.984
Thermal-HEFT	0.972	0.966	0.968	0.966	0.966
Throttled-HEFT (hybrid)	0.988	0.982	0.984	0.984	0.984
Decima-vanilla	1.000	1.000	1.000	1.000	1.000
Decima-thermal	0.456	0.412	0.382	0.324	0.272
HGATE-PPO	0.976	0.978	0.968	0.952	0.988
D2	0.450	0.448	0.478	0.518	0.544
Ours-NoThermal	0.006	0.006	0.006	0.006	0.002
Ours-auto_only	0.002	0.002	0.002	0.002	0.002

5.4.3. Bounded-Claim Envelope

Aggregating the full 5×4 size-by-ambient envelope ($N \in \{9, 13, 17, 24, 33\} \times \text{ambient} \in \{\text{cold, warm, hot, extreme}\} \times 500$ paired episodes per cell = 10,000 paired episodes per scheduler) gives the abstract's headline number. Ours-auto_only recorded 30 unsafe episodes in total, decomposed as 5/2500 at cold, 5/2500 at warm, 5/2500 at hot, and 15/2500 at extreme; Ours-NoThermal recorded 31 unsafe episodes (2/2/13/14). The 30 Ours-auto_only residual failures spanned all four ambient bands and were most frequent in the extreme band, where the per-cell rate reached 0.006; HEFT, by contrast, recorded 9821 unsafe episodes across the same envelope, giving the $327\times$ reduction reported in the abstract and Section 7.

5.4.4. Horizon Stress Test

We additionally evaluated the schedulers of Table 8 at $N = 17$ in the hot regime under a tenfold horizon scaling (dags_per_episode raised from 20 to 200), with 500 paired episodes per cell. Table 8 summarizes the resulting violation and completion rates.

Table 8. Horizon stress test at $N = 17$, hot ambient, $n = 500$ paired episodes per cell. Standard horizon ($H = 20$ DAGs per episode) versus long horizon ($H = 200$). Non-thermal schedulers collapse to a 1.000 violation rate at $H = 200$, while Ours-auto_only degrades only to 0.032 at full DAG completion. The learned baselines (Decima-thermal, HGATE-PPO, D2) and the Ours-NoThermal ablation are characterized in Tables 4 and 7 and are omitted here. Arrows indicate the preferred direction ($\downarrow$ lower is better, $\uparrow$ higher is better).

Scheduler	Viol. Rate $\downarrow$		Completion $\uparrow$	
	$H = 20$	$H = 200$	$H = 20$	$H = 200$
HEFT	0.986	1.000	0.260	0.027
Thermal-HEFT	0.968	1.000	0.337	0.037
Throttled-HEFT (hybrid)	0.984	1.000	0.261	0.027
Ours-auto_only	0.002	0.032	1.000	1.000

The non-thermal schedulers all crossed to a 1.000 violation rate at $H = 200$, with DAG completion below 0.05 in each case: the longer episode horizon exhausted the thermal budget for any policy without thermal awareness. Ours-auto_only, by contrast, degraded only modestly: its violation rate rose from 0.002 to 0.032 while DAG completion stayed at 1.000. The anticipatory delay action of Section 4.1.2 therefore yielded no measurable safety benefit even at the long horizon, consistent with the auto-cool ablation of Section 5.6. Figure 9 complements these violation-rate counts by showing the full per-episode peak-temperature and makespan distributions over the hot $N = 17$ paired set.

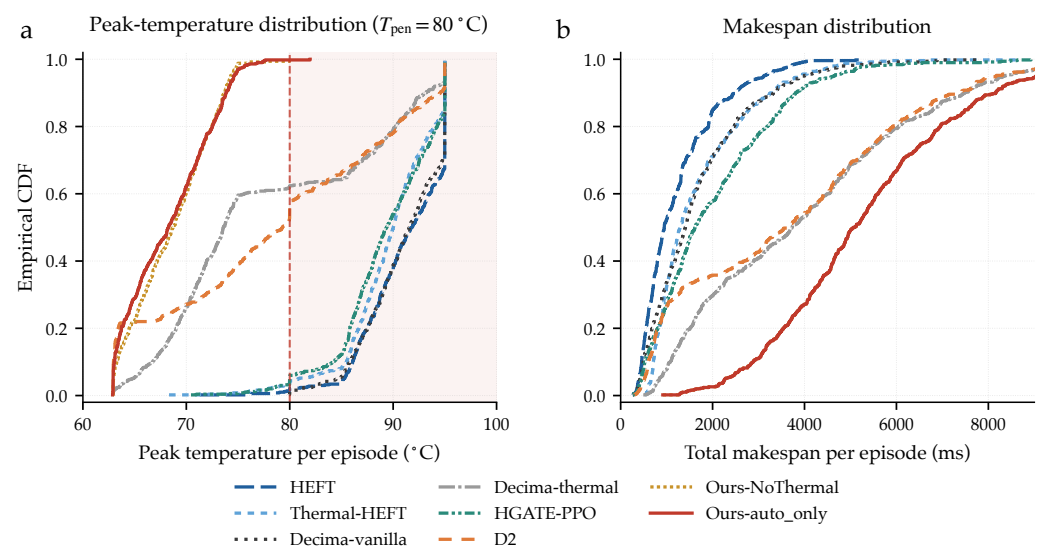


Figure 9. Empirical cumulative distribution function (CDF) over the hot $N = 17$ paired evaluation set ($n = 500$ paired episodes per scheduler): (a) Per-episode peak processor temperature; the dashed vertical red line marks $T_{\text{pen}} = 80^\circ\text{C}$ and the shaded region $T_{\text{peak}} \geq T_{\text{pen}}$ is the unsafe band. Ours-auto_only (red) and Ours-NoThermal (gold) climb to 1.0 well before the threshold, while HEFT, Thermal-HEFT, Decima-vanilla, and HGATE-PPO reach 1.0 only deep inside the unsafe band; Decima-thermal and D2 are bimodal. Curves are distinguished by both color and line style; Ours-hybrid is omitted as it is paired-indistinguishable from Ours-auto_only under deterministic inference. (b) Per-episode total makespan. HEFT, Thermal-HEFT, and Decima-vanilla have nearly the entire mass below 2000 ms because their episodes truncate early at the first thermal violation; the Ours-auto_only distribution is shifted right because every paired episode completes the full 20-DAG horizon. The two panels together quantify the safety-makespan trade-off as distributions rather than as single mean values. The plot is at the standard $H = 20$ horizon; the $H = 200$ long-horizon contrast is reported in Section 5.5 and Table 8.

5.5. The Optional Anticipatory-Delay Extension

The six-test ablation chain of Section 5.3 measured Ours-auto_only under `action_mode = auto_only`, exercising only the placement head. The Ours-hybrid configuration additionally enabled the anticipatory pre-execution delay action of Section 4.1.2 by warm-starting from the Stage-1 placement-only checkpoint and continuing training with the delay head un-clipped under the Stage-2 schedule of Section 4.5. We evaluate whether this extra degree of freedom improved safety over placement-only scheduling.

Under deterministic (argmax) inference it did not. Across every operating point that we evaluated—the main hot $N = 17$ comparison (Table 4), the full 5×4 size-by-ambient envelope (Tables 6 and 7), and the $10 \times$ long-horizon stress test (Table 8, $H = 200$)—Ours-hybrid reproduced Ours-auto_only to within sampling noise, and at $H = 200$ the two coincided at the displayed precision (0.032 violation rate, 16/500 unsafe episodes, 1,000 DAG completion for both). (The $H=200$ figures came from a dedicated long-horizon re-evaluation of the Stage-2 checkpoints used throughout this section; short-horizon logs from earlier evaluation runs were superseded.) The trained delay head selected zero delay at essentially every greedy decision, so the deployed behavior collapsed onto the placement-only policy.

The mechanism is that the placement-only policy is already safe without anticipatory delay. The single-axis auto-cool ablation of Section 5.6 shows that Ours-auto_only remained safe at the hot $N = 17$ cell even when the environment’s reactive auto-cool budget was removed entirely, so there was no safety gap for an anticipatory delay to close at the operating points studied. We have therefore reported Ours-auto_only as the primary deployed configuration and retained the delay head and its dual-channel critic as an optional, safety-neutral component of the architecture rather than as a source of the safety result. This was a deliberate revision: an earlier reading attributed a long-horizon advantage to the delay head, but under the calibrated reactive-cooling model that advantage does not reproduce.

Taken together, the six-test ablation chain and the 5×4 generalization envelope support the central thesis of the paper: that a heterogeneous-graph DRL scheduler with a thermally-shaped reward and an RC-coupling edge attribute on the encoder can deliver thermal-failure rates more than two orders of magnitude below classical scheduling on the CPO interposer (30/10,000, Wilson 95% CI [21, 43]/10,000, vs 9821/10,000, a $327 \times$ reduction) while remaining size-parametric and generalizing zero-shot across the operationally relevant range of system sizes. Component 6 of the ablation chain ($p = 0.625$, n.s.) further showed that, on top of those two factors, explicit per-processor temperature features in the policy observation are a second-order refinement at the standard $H = 20$ horizon. The anticipatory-delay action is a separate axis, and under deterministic inference it is paired-indistinguishable from Ours-auto_only at both $H = 20$ and $H = 200$ (Section 5.5); the deployed safety result therefore rests on placement and the RC-coupling edge, not on anticipatory delay.

5.6. Auto-Cool Ablation: Env-Mechanism vs Policy-Intrinsic Safety

The environment of Section 4.1 includes a built-in lookahead auto-cool mechanism that, before committing each placement, simulates the predicted execution peak and inserts idle cooling steps on the target processor when the simulation predicts a violation, capped at `max_cooling_steps` ms per dispatch (default 100 ms). This is an env-side safety floor that operates independently of the scheduling policy. Reading the violation rates of Sections 5.2–5.4 without isolating this mechanism would conflate *policy-intrinsic* safety with *env-mechanism-dependent* safety. To disentangle the two, we re-ran the hot ambient $N = 17$ paired evaluation with the env auto-cool budget set to zero (`max_cooling_steps = 0`),

forcing every scheduler to rely on its placement decisions alone. The `seed_base` matched the standard hot validation set so the comparison was paired episode-by-episode. Table 9 reports the resulting violation rates and peak-temperature shifts; Figure 10 visualizes the per-episode peak distributions under both `max_cooling_steps` settings.

The auto-cool ablation revealed three readings absent from the standard evaluation. First, removing the budget left HEFT and Decima-thermal unchanged (0.986 and 0.382 respectively): their placement decisions produced predicted peaks so far above T_{pen} that the 100 ms budget could not rescue them, so the env mechanism is not a universal safety floor. Second, both deployed Ours configurations were policy-intrinsically safe at hot $N = 17$ under the operational definition of Section 3.4: Ours-auto_only's violation rate was identical at both settings (0.002 $\rightarrow$ 0.002, the same single paired-identical unsafe episode). Ours-hybrid recorded 0.002 $\rightarrow$ 0.000 with no new violations introduced under `cool = 0` (within-policy exact McNemar across `cool = 100` vs. `cool = 0` on Ours-hybrid: 1 discordant pair, $p = 1.000$). Neither configuration relied on env rescue for its hot $N = 17$ headline number. The third reading isolated which axis of the proposed architecture delivers this independence.

Table 9. Auto-cool ablation at $N = 17$ in the hot regime ($T_{\text{ambient}} \in [60, 75]^\circ\text{C}$). The `max_cooling_steps = 100` column is inherited from the $n = 500$ paired pool of Table 4; the `max_cooling_steps = 0` column is the new $n = 500$ paired evaluation with the env auto-cool budget disabled (Ours-NoThermal `cool = 0` uses the $n = 200$ sub-pool sharing seeds with the matching $n = 500$ `cool = 100` slice, hence the non-integer $0.010 = 2/200$ vs $0.006 = 3/500$ pairing); Δ columns report the change relative to the `max_cooling_steps = 100` baseline. The McNemar column is the paired exact test on the binary safety indicator against Ours-auto_only as anchor at `max_cooling_steps = 0`. These tests are reported as uncorrected exploratory comparisons outside the Holm–Bonferroni family of the six ablation-chain tests in Section 5.3; significance markers *** $p < 0.001$ are on raw p -values.

Scheduler	Viol Cool = 100	Viol Cool = 0	Δ Viol	Peak T (Cool = 0)	McNemar vs. Auto_Only
HEFT	0.986	0.986	0.000	91.46 $^\circ\text{C}$	$p < 10^{-99}$ ***
Decima-thermal	0.382	0.382	0.000	78.18 $^\circ\text{C}$	$p < 10^{-50}$ ***
Ours-NoThermal	0.006	0.010	+0.004	68.66 $^\circ\text{C}$	$p = 1.000$, n.s.
Ours-NoRCEdge	0.000	0.056	+0.056	74.79 $^\circ\text{C}$	$p < 10^{-6}$ ***
Ours-auto_only	0.002	0.002	0.000	68.34 $^\circ\text{C}$	—
Ours-hybrid	0.002	0.000	−0.002	69.20 $^\circ\text{C}$	$p = 1.000$, n.s.

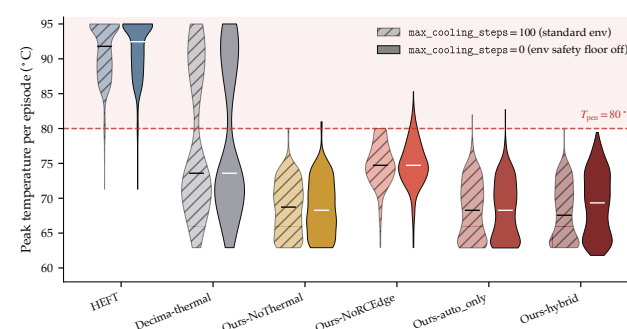


Figure 10. Per-episode peak temperature distribution for each of the six schedulers in the auto-cool ablation. Light violins: `max_cooling_steps = 100` (standard env). Dark violins: `max_cooling_steps = 0` (env safety floor off). Median marked with a horizontal line. The dashed red line marks $T_{\text{pen}} = 80^\circ\text{C}$. HEFT and Decima-thermal are unaffected by removing the auto-cool budget (the env safety floor cannot rescue them under standard conditions either). Ours-auto_only's distribution is also unaffected (intrinsic safety). Ours-NoRCEdge shifts visibly toward T_{pen} once the safety floor is removed, exposing the 6.1 $^\circ\text{C}$ thermal-margin gap to Ours-auto_only that the `max_cooling_steps = 100` budget had been absorbing in Table 4.

Reading 3: The RC-Coupling Edge Attribute Is the Dominant Axis We Have Identified for Auto-Cool-Independent Safety

The single-axis ablation Ours-auto_only $\rightarrow$ Ours-NoRCEdge (removing the RC-coupling edge attribute from proc-proc edges and re-training from scratch) was paired-indistinguishable from auto_only under the standard env (0.002 for auto_only vs 0.000 for NoRCEdge, $p = 1.000$, n.s.; the single unsafe episode of auto_only happened to be safe under NoRCEdge, which the auto-cool budget rescued) but degraded sharply once the env safety floor is removed: the violation rate rose from 0.000 to 0.056, with 28 episodes paired-discordant in NoRCEdge's favor of being unsafe and only 1 in the opposite direction (paired McNemar $p < 10^{-6}$). The peak-temperature distribution of Ours-NoRCEdge sat a median 6.1 °C closer to T_{pen} than Ours-auto_only's at $\text{max_cooling_steps} = 100$ (Wilcoxon $p < 10^{-74}$ on $n = 500$ paired episodes); the auto-cool ablation revealed that this 6 °C margin was exactly the safety reserve that env auto-cool consumed when RC-edge was absent. Removing per-processor thermal observations instead of the RC-edge attribute (Ours-auto_only $\rightarrow$ Ours-NoThermal) was paired-not-significant at both cooling settings ($p = 1.000$), confirming that thermal observation features are second-order to the RC-coupling edge attribute on the auto-cool-independence axis.

We interpret these three readings as follows. The RC-coupling edge attribute is the architectural element that converts a trained policy from *env-mechanism-dependent* safety (safe-at-the-margin-with-env-rescue) to *policy-intrinsic* safety (safe-without-rescue). Both modes look identical under the standard env in headline violation rate, but only the latter deploys without the simulator-side cooling-budget guarantee.

5.7. Robustness Against Thermal-Calibration Error

A deployed module's true RC constants are never known exactly, so we tested how sensitive the safety result was to the calibrated parameters of Table 1. We used a *decoupled* calibration-error sweep: the policy kept observing the nominal calibration it was trained on, while the underlying thermal plant was perturbed by $\delta \in \{\pm 10, \pm 20, \pm 30\}\%$ on one parameter at a time. This isolated genuine model-mismatch robustness (how the policy behaves when reality differs from its internal model) from mere retraining on a different calibration. We perturbed the three heat-flow parameters independently—the lateral coupling G^{cross} , the cold-plate conductance G^{env} , and the leakage coefficient $P^{\text{leak},0}$ —and evaluated the headline Ours-auto_only at the hot $N = 17$ operating point with 200 paired episodes per cell.

Table 10 and Figure 11 show three regularities. First, the policy was insensitive to coupling-strength error: the violation rate was flat at 0.005 across the full $\pm 30\%$ range of G^{cross} , so a mis-estimated chip-to-chip coupling did not compromise safety. Second, the heat-balance parameters G^{env} and $P^{\text{leak},0}$ had a tolerance of about $\pm 10\%$: the rate stayed at or below 0.045 for $G^{\text{env}} \geq -10\%$ and $P^{\text{leak},0} \leq +10\%$ then climbed steeply outside that band. Third, the failures were *one-sided*—the policy degraded only when the real plant ran *hotter* than calibrated (less cold-plate cooling, or more leakage), never when it ran cooler. A deployment therefore only needed to calibrate the heat-balance parameters to within roughly 10% on the conservative side, which was comfortably within the accuracy of the published CPO thermal characterizations the model is built on (Section 3.2). This validated the lumped RC model for the operational scheduling task: the safety result did not hinge on the exact calibrated values.

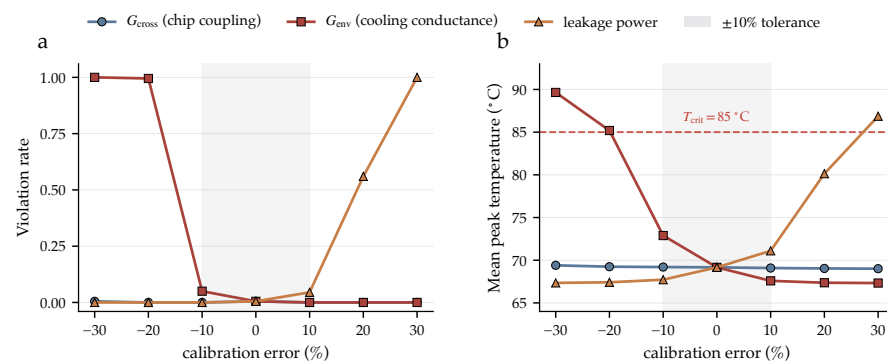


Figure 11. Decoupled calibration-error sensitivity of Ours-auto_only at hot $N = 17$ (200 paired episodes per cell): (a) violation rate and (b) mean peak temperature versus per-parameter calibration error, for the chip-to-chip coupling G^{cross} , the cold-plate conductance G^{env} , and the leakage power. The policy observes the nominal calibration while the plant is perturbed; the shaded band marks the $\pm 10\%$ tolerance and the dashed line in (b) marks $T_{\text{crit}} = 85^\circ\text{C}$.

Table 10. Decoupled calibration-error sensitivity of Ours-auto_only at hot $N = 17$ (200 paired episodes per cell). The thermal plant is perturbed by δ on one parameter while the policy observes the nominal calibration; entries are the violation rate.

δ	−30%	−20%	−10%	0%	+10%	+20%	+30%
G^{cross}	0.005	0.005	0.005	0.005	0.005	0.005	0.005
G^{env}	1.000	1.000	0.045	0.005	0.000	0.000	0.000
$p^{\text{leak},0}$	0.000	0.000	0.000	0.005	0.025	0.450	1.000

5.8. Comparison with a Lagrangian Constrained-PPO Baseline

The proposed policy enforces the thermal limit through a shaped reward (Section 4.1.3) rather than as an explicit constraint, which raises the question of whether a formally constrained formulation—a constrained MDP solved by a Lagrangian/RCPO method [39–41]—would give a stronger safety guarantee. We trained such a baseline, Ours-Lagrangian, on the identical architecture, environment, and reward components, replacing the fixed thermal-penalty weights with a Lagrangian dual variable λ updated by dual ascent on the violation constraint (warm-started from the same Stage-1 checkpoint, 1.5×10^6 steps; λ converges to ≈ 0.34). This isolated the effect of the *objective* (shaped reward vs. formal constraint) on the *same* architecture.

Table 11 compares the two at the hot operating point across all module sizes. The constrained formulation did *not* deliver stronger safety: both configurations held the violation rate at or below 0.002 and DAG completion at 1.000 at every N , at a comparable makespan. The two differed only in second-order trade-offs. Ours-auto_only ran 2–4 °C cooler in peak temperature, retaining more thermal margin, whereas the Lagrangian baseline was marginally more robust at the edge of the calibration band of Section 5.7 (at $G^{\text{env}} = -10\%$ it recorded a 0.005 violation rate versus 0.045 for Ours-auto_only). Because the safety headline was reached under both a shaped-reward and a formally-constrained objective on the *same* heterogeneous-graph architecture, the safety result is best read as a property of that architecture—the RC-coupling edge and the typed encoder—rather than as an artifact of reward-weight tuning; a constrained MDP does not provide a stronger guarantee in this regime.

Table 11. Ours-auto_only (shaped reward) vs. Ours-Lagrangian (constrained PPO) at hot $T_{\text{ambient}} \in [60, 75]^\circ\text{C}$, 500 episodes per cell evaluated under the same seed pool ($\text{seed_base} = 100,000$); the Ours-auto_only metrics are reproduced from the size-by-ambient envelope evaluation (Table 7). Both reach the safety headline at every size; both complete 100% of DAGs at a comparable makespan (≈ 5317 ms for both at $N = 17$). The shaped-reward policy retains a larger thermal margin (lower peak temperature).

Metric, Scheduler	$N = 9$	$N = 13$	$N = 17$	$N = 24$	$N = 33$
Viol. rate, Ours-auto_only	0.002	0.002	0.002	0.002	0.002
Viol. rate, Ours-Lagrangian	0.000	0.000	0.000	0.002	0.002
Peak T , Ours-auto_only	68.43	68.35	68.33	68.30	68.22
Peak T , Ours-Lagrangian	72.03	71.64	71.41	70.91	70.54

6. Discussion

The six-test ablation chain of Section 5.3 isolated which design decision drives the safety gap. This section briefly revisits the architectural commitments behind those links (Section 6.1), reports deployment cost (Section 6.2), and bounds the safety claim explicitly (Section 6.3).

6.1. Design Choices Revisited

The six-test ablation chain attributes the safety contribution of the following design choices: (i) a thermal-shaped reward; (ii) heterogeneous edge typing of DAG dependencies vs. RC coupling vs. task–processor candidates; and (iii) the RC-coupling edge attribute supplied directly to the encoder. Three further architectural choices are not isolated by the §5 chain individually.

6.1.1. Cross-Attention for Placement, Pooled MLP for Delay

The two policy heads use structurally different primitives. Placement is a matching problem (which processor for this task); multi-head attention is the appropriate primitive. Delay is a regression problem (how much delay across the system); a small MLP on a $(\mathbf{h}^p, \max_j \mathbf{h}_j^p)$ pool captures the max-temperature signal that dominates the delay decision. A single attention-everywhere architecture would have to learn both behaviors from gradient alone.

6.1.2. Dual-Channel Critic for Stage 2

The Stage 2 hybrid training of Section 4.5 uses two scalar value heads V^{place} and V^{delay} rather than one, mirroring the additive decomposition of the reward (5) into a placement-driven part (makespan + bonus) and a thermal-driven part (soft + hard penalties).

6.1.3. RC-Coupling Edge Attribute vs. Raw Temperature Concatenation

The RC-coupling edge relation $\mathcal{E}_{pp}$ carries the calibrated cross-conductance G_{ij}^{cross} of the discretized thermal network (2) as an edge attribute, rather than concatenating per-processor temperatures into the processor node feature. The two routes are not equivalent: the governing dynamics on processor i are $C_i dT_i/dt = P_i^{\text{total}} - G_i^{\text{env}}(T_i - T_{\text{amb}}) - \sum_{j \neq i} G_{ij}^{\text{cross}}(T_i - T_j)$, in which the cross-coupling term is a bilinear function of the conductance *and* the inter-processor temperature gradient. A node-only concatenation exposes $\{T_j\}$ to the encoder but leaves the bilinear coupling implicit, requiring it to be relearned through message passing on every forward pass; the edge-attribute route surfaces the physical coupling structure directly on the edges that carry it. The six-test ablation chain of Section 5.3 does not isolate the RC-coupling edge attribute on its own: Component 3 (Decima-thermal $\rightarrow$ HGATE-PPO) introduces heterogeneous edge typing on top of a

homogeneous-GCN baseline and is paired-worse, and Component 5 (Decima-thermal $\rightarrow$ Ours-NoThermal) jointly introduces the heterogeneous trunk, the cross-attention placement actor, and the RC-coupling attribute. The chain supports the inductive-bias argument at the granularity of the joint architecture, not at the granularity of the RC-edge attribute alone. To isolate this attribute, we trained a separate Ours-NoRCEdge policy (same heterogeneous trunk, same cross-attention actor, RC-coupling attribute ablated on proc-proc edges only, identical training schedule to Ours-auto_only) and evaluated it under the auto-cool ablation of Section 5.6. The result resolved the isolation cleanly: at the standard env configuration the two policies were paired-indistinguishable on safety (0.002 vs. 0.000, $p = 1.000$, n.s.), but the median peak temperature of Ours-NoRCEdge was 6.1 °C closer to T_{pen} on 89% of paired episodes (Wilcoxon $p < 10^{-74}$); when the env auto-cool budget was disabled, the violation rate of Ours-NoRCEdge climbed to 0.056 (28 of 500 paired episodes were unsafe where Ours-auto_only was safe; only 1 in the opposite direction; paired McNemar $p < 10^{-6}$). Removing per-processor thermal observations instead of the RC-edge attribute (Ours-auto_only $\rightarrow$ Ours-NoThermal) was paired-not-significant under both cooling settings ($p = 1.000$ each). The single-axis ablation isolated RC-edge as the largest contributor among the four axes we examined; we do not exclude other architectural choices contributing to auto-cool-independent safety. This isolates the RC-coupling edge attribute as the specific contributor to *auto-cool-independent* safety: the attribute moves the deployed policy out of the regime where env-side cooling-budget rescue is needed to keep the headline violation rate at zero, into the regime where the policy's own placement distribution carries the safety margin. The two routes (RC-edge attribute vs. raw temperature concatenation) incur identical encoder forward cost; the difference is one of inductive bias, not parameter count.

6.2. Inference Latency and Deployment Cost

We measured per-decision inference latency on a commodity CPU (Apple M3 Pro, 11-core, single-thread PyTorch 2.3.0 with `torch.set_num_threads(1)`) using the deployed checkpoint at $N = 17$ and the three deployment modes. Times were in milliseconds, computed over 1000 steady-state trials preceded by 200 warm-up forward passes per mode under PyTorch 2.3.0 with neither `torch.compile` nor TorchScript enabled (eager-mode forward pass on the same checkpoint used for the evaluation tables). Each trial was one act-step decision on a freshly sampled `graph_obs` drawn from the standard $H = 20$ workload at $T_{\text{ambient}} \in [60, 75]$ °C. Table 12 reports the median, p_{95} , p_{99} , and max latency for each mode:

Table 12. Per-decision inference latency on commodity CPU. $N = 17$, single thread, batch size 1. The three modes shared the encoder and the placement head; differences arose from whether the delay head was sampled and the small overhead of the dual critic. Latency was well below the dominant thermal time constants of the CPO interposer (Section 3.2), so the scheduler is computationally cheap relative to the phenomena it controls.

Mode	Median (ms)	p95 (ms)	p99 (ms)	Max (ms)
auto_only	1.68	1.79	1.92	2.20
hybrid	1.70	1.78	1.85	3.06
agent_only	1.74	1.78	1.81	2.22

6.2.1. Three Modes Within $\sim 60 \mu\text{s}$ of Each Other

The three deployment modes share the encoder and the placement head; they differ only in whether the delay head's softmax is sampled and whether the dual critic's delay-channel value is computed downstream. The measured difference is $\sim 60 \mu\text{s}$ (1.74–1.68), or 4% of the total. This confirms that the delay-head and dual-critic components are cheap

relative to the encoder, which is consistent with the parameter-count breakdown: the encoder accounts for the large majority of model parameters and computation. From a deployment engineering standpoint, this means switching between the deployment modes of Section 4.5.4 incurs no measurable latency penalty.

6.2.2. Latency Is $65\text{--}141\times$ Below the Dominant Thermal Time Constants

The thermal time constants of the CPO interposer (Section 3.2) are $\tau_{OE} \approx 110$ ms and $\tau_{ASIC} \approx 244$ ms. A 1.70 ms scheduling decision commits to placement well within the time horizon over which the thermal field can change appreciably, with $\sim 65\times\text{--}141\times$ headroom for the per-decision compute relative to the per-decision thermal evolution. The scheduler is, in deployment terms, computationally cheap relative to the phenomena it controls.

6.2.3. Comparison to Flat-Vector MLP Baselines

A flat-vector MLP scheduler at the same task scale ($N = 17$ processors, ~ 5 ready tasks per decision step) would have inference latency in the $50\text{--}100\ \mu\text{s}$ range; the proposed heterogeneous-graph encoder is approximately $25\times$ slower per decision. For applications where the absolute scheduling rate is the binding constraint, a flat-vector encoder would be the correct choice. For the CPO regime studied here, where the binding constraint is *thermal safety* rather than scheduling throughput, the $25\times$ latency cost is well worth the $327\times$ reduction in unsafe-episode rate ($30/10,000$ vs $9821/10,000$, see Section 5.4) that the architectural complexity enables.

6.2.4. Concurrent Deployment in Dense AI-Factory Settings

Beyond a single CPO module, a hyperscale AI-factory deployment may host hundreds of concurrent microservice DAGs across many modules, each requiring its own scheduling decision. The 1.70 ms median CPU latency reported above is the cost of a single decision; the aggregate decision rate scales linearly with the number of modules. In settings where the management plane has access to a small embedded GPU or NPU on the control board, the heterogeneous-graph encoder can be batched across modules, amortizing the constant-factor overhead per forward pass. We have not benchmarked batched inference in this work. The architectural property that enables batching is the size-parametric encoder: graphs of different module sizes $N \in \{9, 13, 17, 24, 33\}$ can be packed into a single batch through standard heterogeneous-graph batching primitives. We treat batched accelerator deployment as a deployment engineering question rather than a research one, and a direct next step for scaling the proposed scheduler to chassis-wide and rack-wide settings.

6.3. Limitations and Future Work

- **Boundary of the safety claim.** Across the full 5×4 size-by-ambient envelope (10,000 paired episodes per scheduler), the proposed Ours-auto_only scheduler recorded 30 residual unsafe episodes. These spanned all four ambient bands—5 in cold, 5 in warm, 5 in hot, and 15 in extreme (cf. Section 5.1 for band definitions)—and were spread across all five evaluated sizes $N \in \{9, 13, 17, 24, 33\}$, with one unsafe episode in each cold, warm, and hot cell and three in each extreme cell. The residual rate was thus low and roughly size-independent but rose with ambient temperature, concentrating in the extreme band where the elevated starting temperature left the least placement margin to T_{pen} . We have reported these envelope residuals rather than aggregating them away. Operation outside the trained ambient range is out of scope.
- **Env-side auto-cool floor and its disclosure.** The evaluation env of Section 4.1 included a lookahead auto-cool mechanism with a 100 ms cap that may insert idle cooling steps before each dispatch (`max_cooling_steps`). This is an env-side safety floor that operates independently of the scheduler. Section 5.6 isolated its effect by

re-running the hot $N = 17$ paired evaluation at `max_cooling_steps = 0` and found three regimes: HEFT and Decima-thermal were not rescued by the auto-cool floor (their predicted peaks exceeded the 100 ms budget); both deployed Ours configurations (Ours-auto_only and Ours-hybrid) were policy-intrinsically safe at this cell, statistically unchanged by removing the floor (Ours-auto_only $0.002 \rightarrow 0.002$; Ours-hybrid $0.002 \rightarrow 0.000$ with within-policy exact McNemar $\text{cool}=100$ vs $\text{cool}=0$ on Ours-hybrid, 1 discordant pair, $p = 1.000$); and Ours-NoRCEdge was the single configuration that depended on the floor for its standard headline rate (28 paired episodes were rescued when the budget was available, $p < 10^{-6}$ when removed). The envelope-wide headline rates of Sections 5.2–5.4 were reported under the standard env auto-cool budget; the policy-intrinsic property has been verified at hot $N = 17$ for both Ours-auto_only and Ours-hybrid, and we leave its extension to non-hot cells of the envelope as future work.

- **Sim-to-real gap.** The RC thermal network is calibrated against published CPO measurements [1] but our evaluation ran entirely in simulation. Sustained workload pressure on a deployed module exhibits long-timescale ambient drift and component-level wear that our finite-horizon MDP does not see. Online fine-tuning on logged operational data is a direct next step.
- **Active DVFS composition.** The bounded-claim experiments treat the DVFS controller as inactive. The scheduler operates at a strictly higher level (DAG-level placement) than DVFS (per-clock-cycle frequency adjustment) and the two are operationally compositional, but a study of the interaction under regimes where DVFS itself becomes the bottleneck is left to future work.
- **Single-DAG-source and accelerator deployment.** We evaluated exclusively on the Alibaba 2021 microservice trace and on commodity CPU. Generalization across DAG distributions (scientific workflows, AI/ML training graphs) and batched accelerator inference for chassis-wide multi-module orchestration are future directions; the size-parametric encoder is the architectural property that makes batched accelerator inference straightforward.
- **Statistical-rigor limitations.** Several of our methodological choices reflected a trade-off between fidelity and the available compute budget. Specifically: (i) all the learned baselines were trained with a single fixed seed; per-baseline training-seed variance was not reported and constitutes a notable rigor limitation. (ii) The training budgets across the ablation chain were not uniform ($2\text{--}5 \times 10^6$ environment steps depending on variant; see Table 3); we reported each baseline at its return-curve plateau. (iii) Confidence intervals on per-cell violation rates were not reported; the public repository enables independent re-computation. (iv) The HGATE-PPO comparison was a recipe-level reproduction under our budget; we have flagged training instability in Section 5.3 and a dedicated hyper-parameter sweep is future work. (v) Among the four architectural axes introduced jointly in Component 5, only the RC-coupling edge attribute was individually isolated (Section 5.6); the other three were evaluated only as part of the bundle. These limitations bound the strength of the architectural-attribution claims but do not affect the headline empirical contrast under the standard evaluation protocol.
- **Toward optical-computing-network scheduling.** The present scheduler targets thermal-aware dispatch within a single CPO module, but the same heterogeneous-graph, symmetry-respecting formulation is a natural fit for the emerging *optical-computing-communication integrated network* paradigm [9], in which computation, optical switching, and shared photonic resources are co-optimized across the fabric. Such an extension would enrich the heterogeneous graph with additional typed nodes and

edges—optical-link and wavelength-channel nodes, laser-source nodes, and photonic compute tiles alongside the present ASIC and OE processors—and would broaden the factored actor with heads for communication-path and wavelength selection alongside the current placement and delay heads, so that compute placement, routing, and thermal management are emitted jointly. The permutation-equivariance and size-invariance the present design exploits carry over directly to these added relations, making the architecture a principled starting point for this tri-objective (compute, optical switching, thermal) problem, for which, to our knowledge, no unified scheduler yet exists. Recent reinforcement-learning approaches to optical-network optimization [48] and reconfigurable optical switching for data centers [49] suggest the learning machinery transfers, with the CPO-specific RC-coupling edge supplying the thermal ingredient such methods currently lack.

7. Conclusions

We have presented a thermal-aware DAG scheduler for co-packaged-optics data centers, combining a heterogeneous-graph encoder with an RC-coupling edge attribute, a cross-attention factored actor, and PPO with a two-stage curriculum. A six-test ablation chain characterized the contribution of each design factor: RL backbone alone (HEFT $\rightarrow$ Decima-vanilla), thermal reward shaping (Decima-vanilla $\rightarrow$ Decima-thermal), heterogeneous edge typing alone (Decima-thermal $\rightarrow$ HGATE-PPO), cross-attention actor on a homogeneous trunk (Decima-thermal $\rightarrow$ D2), the full proposed architecture (Decima-thermal $\rightarrow$ Ours-NoThermal), and per-processor thermal observation features (Ours-NoThermal $\rightarrow$ Ours-auto_only), with paired McNemar significance and Holm–Bonferroni family-wise control across the six tests. The proposed scheduler incurred 30 unsafe episodes per 10,000 across the full 5×4 size-by-ambient envelope—a low rate that held across all five sizes $N \in \{9, 13, 17, 24, 33\}$ and rose with ambient temperature, concentrating in the extreme band—while every comparable thermally blind or no-RC-edge baseline retained a non-trivial violation rate. Per-decision inference latency on a commodity CPU was 1.70 ms, well below the ~ 110 ms (OE) and ~ 244 ms (ASIC) thermal time constants of the interposer. We reported the safety claim with explicit boundary conditions (30 residual unsafe episodes across the 10,000-paired evaluation envelope per scheduler, spanning all four ambient bands and most frequent in the extreme band (cf. Section 5.1), spread across $N \in \{9, 13, 17, 24, 33\}$). The single trained policy attained this envelope-wide safety at every evaluated size without per-size retraining: by construction it is permutation-equivariant in placement, permutation-invariant in delay, and invariant to the processor count, which makes a single policy well-defined at every size (Section 4.4); the zero-shot generalization we observed is the empirical payoff. Three extensions remain open: online adaptation under ambient drift, multi-module chassis-scale coordination, and principled composition with active DVFS controllers.

Author Contributions: Conceptualization, Z.Q. and T.D.; methodology, Z.Q.; software, Z.Q.; validation, Z.Q., F.F., B.X. and T.D.; formal analysis, Z.Q.; investigation, Z.Q.; resources, T.D., F.F. and B.X.; data curation, Z.Q.; writing—original draft preparation, Z.Q.; writing—review and editing, Z.Q., L.P., F.F., H.Z., W.Q., B.X. and T.D.; visualization, Z.Q. and L.P.; supervision, T.D.; project administration, T.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Central Government Guided Local Science and Technology Development Fund Projects under Grant 2025BEA006.

Data Availability Statement: The source code supporting the results in this paper is publicly available at <https://github.com/JackQqiu/cpo-thermal-drl> (accessed on 9 July 2026); the trained model checkpoints and evaluation logs are available from the corresponding author on reasonable

request. The Alibaba 2021 microservice trace used as workload input is available from the Alibaba cluster-trace-v2021 repository (<https://github.com/alibaba/clusterdata> (accessed on 9 July 2026)) under its original license.

Acknowledgments: During the preparation of this manuscript, the authors used Claude (a large language model developed by Anthropic; Claude Opus/Sonnet family, 2025–2026 versions) for English-language editing, including correcting grammar and usage, improving sentence clarity and conciseness, rephrasing non-idiomatic wording into standard academic English, and keeping terminology and notation consistent. The methodology, experiments, data, and results are the authors' own work. The authors have reviewed and edited the AI-assisted output and take full responsibility for the content of this publication.

Conflicts of Interest: F.F. is affiliated with Fiberhome Telecommunication Technologies Co., Ltd., an optical-communications equipment manufacturer, and B.X. is affiliated with Hubei Chutianyun Co., Ltd., a cloud-computing company; the research reported in this paper was conducted independently of these companies' commercial product roadmaps, and neither company funded or directed this study. The remaining authors declare no conflicts of interest.

References

1. Wu, S.; Wen, S.; He, H.; Feng, J.; Chen, C.; Xue, H. Simulation and experimental investigation of liquid-cooling thermal management for high-bandwidth co-packaged optics. *Front. Optoelectron.* **2025**, *18*, 11. <https://doi.org/10.1007/s12200-025-00156-4>.
2. Mahajan, R.; Li, X.; Fryman, J.; Zhang, Z.; Nekkanty, S.; Tadayon, P.; Jaussi, J.; Shumarayev, S.; Agrawal, A.; Jadhav, S.; et al. Co-Packaged Photonics for High Performance Computing: Status, Challenges and Opportunities. *J. Light. Technol.* **2022**, *40*, 379–392. <https://doi.org/10.1109/JLT.2021.3104725>.
3. Tan, M.; Xu, J.; Liu, S.; Feng, J.; Zhang, H.; Yao, C.; Chen, S.; Guo, H.; Han, G.; Wen, Z.; et al. Co-packaged optics (CPO): Status, challenges, and solutions. *Front. Optoelectron.* **2023**, *16*, 1. <https://doi.org/10.1007/s12200-022-00055-y>.
4. Shekhar, S.; Bogaerts, W.; Chrostowski, L.; Bowers, J.E.; Hochberg, M.; Soref, R.; Shastri, B.J. Roadmapping the next generation of silicon photonics. *Nat. Commun.* **2024**, *15*, 751. <https://doi.org/10.1038/s41467-024-44750-0>.
5. Tian, W.; Hou, H.; Dang, H.; Cao, X.; Li, D.; Chen, S.; Ma, B. Progress in Research on Co-Packaged Optics. *Micromachines* **2024**, *15*, 1211. <https://doi.org/10.3390/mi15101211>.
6. Broadcom Inc. *Broadcom Delivers Industry's First 51.2-Tbps Co-Packaged Optics Ethernet Switch Platform for Scalable AI Systems*; Press Release; Tomahawk 5-Bailly: One 51.2 Tbps Switch ASIC Co-Packaged with Eight 6.4 Tbps Silicon-Photonic Optical Engines; Broadcom Inc.: Palo Alto, CA, USA, 2024.
7. Broadcom Inc. *Broadcom Announces Tomahawk 6—Davisson, the Industry's First 102.4-Tbps Ethernet Switch with Co-Packaged Optics*; Press Release; Third-Generation CPO: 120 × 120 mm Package with 16 Optical Engines; 3.5 W Per 800 Gb/s Port; Broadcom Inc.: Palo Alto, CA, USA, 2025.
8. Xu, Z.; Zhou, T.; Ma, M.; Deng, C.; Dai, Q.; Fang, L. Large-scale photonic chiplet Taichi empowers 160-TOPS/W artificial general intelligence. *Science* **2024**, *384*, 202–209. <https://doi.org/10.1126/science.adl1203>.
9. Hai, D.T.; Woungang, I. The Dual Horizon: A Rendezvous of Computing and Communication Services at the Optical Layer in Optical Computing-Communication Integrated Network. *arXiv* **2025**, arXiv:2505.18753. <http://arxiv.org/abs/2505.18753>.
10. Padmaraju, K.; Bergman, K. Resolving the thermal challenges for silicon microring resonator devices. *Nanophotonics* **2014**, *3*, 269–281. <https://doi.org/10.1515/nanoph-2013-0013>.
11. Luo, S.; Xu, H.; Lu, C.; Ye, K.; Xu, G.; Zhang, L.; Ding, Y.; He, J.; Xu, C. Characterizing Microservice Dependency and Performance: Alibaba Trace Analysis. In Proceedings of the ACM SoCC '21, Seattle, WA, USA, 1–4 November 2021; pp. 412–426. <https://doi.org/10.1145/3472883.3487003>.
12. Gan, Y.; Zhang, Y.; Cheng, D.; Shetty, A.; Rath, P.; Katarki, N.; Bruno, A.; Hu, J.; Ritchken, B.; Jackson, B.; et al. An Open-Source Benchmark Suite for Microservices and Their Hardware-Software Implications for Cloud and Edge Systems. In Proceedings of the ASPLOS '19, Providence, RI, USA, 13–17 April 2019; pp. 3–18. <https://doi.org/10.1145/3297858.3304013>.
13. Topcuoglu, H.; Hariri, S.; Wu, M.Y. Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. *IEEE Trans. Parallel Distrib. Syst.* **2002**, *13*, 260–274. <https://doi.org/10.1109/71.993206>.
14. Mao, H.; Schwarzkopf, M.; Venkatakrisnan, S.B.; Meng, Z.; Alizadeh, M. Learning Scheduling Algorithms for Data Processing Clusters. In Proceedings of the ACM SIGCOMM 2019, Beijing, China, 19–23 August 2019; pp. 270–288. <https://doi.org/10.1145/3341302.3342080>.

15. Wu, J.M.; Zou, Y.; Zhang, X.; Liu, J.; Sun, W.; Du, G. Dependency-Aware Task Offloading Strategy via Heterogeneous Graph Neural Network and Deep Reinforcement Learning. *IEEE Internet Things J.* **2025**, *12*, 22915–22933. <https://doi.org/10.1109/JIOT.2025.3549441>.
16. Durillo, J.J.; Fard, H.M.; Prodan, R. MOHEFT: A Multi-Objective List-Based Method for Workflow Scheduling. In Proceedings of the IEEE CloudCom 2012, Taipei, Taiwan, 3–6 December 2012; pp. 185–192. <https://doi.org/10.1109/CloudCom.2012.6427573>.
17. Arabnejad, H.; Barbosa, J.G. List Scheduling Algorithm for Heterogeneous Systems by an Optimistic Cost Table. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 682–694. <https://doi.org/10.1109/TPDS.2013.57>.
18. Bittencourt, L.F.; Sakellariou, R.; Madeira, E.R.M. DAG Scheduling Using a Lookahead Variant of the Heterogeneous Earliest Finish Time Algorithm. In Proceedings of the 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Pisa, Italy, 17–19 February 2010; pp. 27–34. <https://doi.org/10.1109/PDP.2010.56>.
19. Yi, D.; Zhou, X.; Wen, Y.; Tan, R. Efficient Compute-Intensive Job Allocation in Data Centers via Deep Reinforcement Learning. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 1474–1485. <https://doi.org/10.1109/TPDS.2020.2968427>.
20. Chaudhry, M.T.; Ling, T.C.; Manzoor, A.; Hussain, S.A.; Kim, J. Thermal-Aware Scheduling in Green Data Centers. *ACM Comput. Surv.* **2015**, *47*, 39. <https://doi.org/10.1145/2678278>.
21. Chandrasiri, S.; Meedeniya, D. Energy-Efficient Dynamic Workflow Scheduling in Cloud Environments Using Deep Learning. *Sensors* **2025**, *25*, 1428. <https://doi.org/10.3390/s25051428>.
22. Wan, L.; Cui, X.; Zhao, H.; Li, C.; Wang, Z. Flexible job shop scheduling via deep reinforcement learning with meta-path-based heterogeneous graph neural network. *Knowl. Based Syst.* **2024**, *296*, 111940.
23. Tang, H.; Dong, J. Solving Flexible Job-Shop Scheduling Problem with Heterogeneous Graph Neural Network Based on Relation and Deep Reinforcement Learning. *Machines* **2024**, *12*, 584. <https://doi.org/10.3390/machines12080584>.
24. Sun, B.; Theile, M.; Qin, Z.; Bernardini, D.; Roy, D.; Bastoni, A.; Caccamo, M. Edge Generation Scheduling for DAG Tasks Using Deep Reinforcement Learning. *IEEE Trans. Comput.* **2024**, *73*, 1034–1047.
25. Zhao, T.; Chen, T.; Sun, Y.; Xu, Y. IM-GNN: Microservice Orchestration Recommendation via Interface-Matched Dependency Graphs and Graph Neural Networks. *Symmetry* **2025**, *17*, 525. <https://doi.org/10.3390/sym17040525>.
26. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 7–12 December 2015.
27. Bronstein, M.M.; Bruna, J.; Cohen, T.; Velicković, P. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv* **2021**, arXiv:2104.13478. <http://arxiv.org/abs/2104.13478>.
28. Velicković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
29. Zhadan, A.; Allahverdyan, A.; Kondratov, I.; Mikheev, V.; Petrosian, O.; Romanovskii, A.; Kharin, V. Multi-Agent Reinforcement Learning-Based Adaptive Heterogeneous DAG Scheduling. *ACM Trans. Intell. Syst. Technol.* **2023**, *14*, 87. <https://doi.org/10.1145/3610300>.
30. Hu, Y.; Huang, J.; Yu, Y.; Hu, M.; Peng, K.; Zhang, T.; Chen, C.; Xu, Z. Energy-aware Service Mesh Deployment and Online Request Routing in Edge: A Hierarchical Deep Reinforcement Learning Approach. *IEEE Trans. Mob. Comput.* **2026**. <https://doi.org/10.1109/TMC.2026.3674203>.
31. Xing, S.; Wang, Y.; Liu, W. Self-Adapting CPU Scheduling for Mixed Database Workloads via Hierarchical Deep Reinforcement Learning. *Symmetry* **2025**, *17*, 1109. <https://doi.org/10.3390/sym17071109>.
32. Sun, Y.; Bian, Y.; Li, H.; Tan, F.; Liu, L. Flexible Offloading and Task Scheduling for IoT Applications in Dynamic Multi-Access Edge Computing Environments. *Symmetry* **2023**, *15*, 2196. <https://doi.org/10.3390/sym15122196>.
33. Peng, K.; Li, T.; Ren, M.; Hu, Y.; Wang, L.; Hu, M.; Jiang, H. Symmetric Orchestration Under Service Mesh Paradigm: Empowering Massive Online Applications in Edge Clouds. *IEEE Trans. Mob. Comput.* **2026**, *25*, 4300–4316. <https://doi.org/10.1109/TMC.2025.3624628>.
34. Peng, K.; Huang, J.; Wang, Z.; Hu, Y.; Hu, M.; Xu, Z.; Chen, Z. Collaborative Orchestration of Microservices and AI Services in Edges: A Dual-Time-Scale Reinforcement Learning Approach. *IEEE Trans. Serv. Comput.* **2026**. <https://doi.org/10.1109/TSC.2026.3680248>.
35. Peng, K.; Liu, X.; Han, D.; Hu, Y.; Hu, M.; Cai, C.; Xiong, Z. Hybrid Orchestration of AI Services and Microservices in Cloud-Edge Collaboration. *IEEE Trans. Mob. Comput.* **2026**. <https://doi.org/10.1109/TMC.2026.3654236>.
36. Stojkovic, J.; Zhang, C.; Goiri, Í.; Choukse, E.; Qiu, H.; Fonseca, R.; Torrellas, J.; Bianchini, R. TAPAS: Thermal- and Power-Aware Scheduling for LLM Inference in Cloud Platforms. In Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Rotterdam, The Netherlands, 30 March–3 April 2025; pp. 1266–1281. <https://doi.org/10.1145/3676641.3716025>.
37. Ran, Y.; Yin, H.; Sun, T.; Zhou, X.; Luo, J.; Chen, S. D3T: Dual-Timescale Optimization of Task Scheduling and Thermal Management for Energy Efficient Geo-Distributed Data Centers. *IEEE Trans. Parallel Distrib. Syst.* **2026**, *37*, 230–246. <https://doi.org/10.1109/TPDS.2025.3631654>.

38. Lu, R.; Wang, D. A Thermal-Aware Workload Scheduler for High-Performance LLM Inference in Cooling-Regulated Datacenters. *Acm. Sigenergy Energy Inform. Rev.* **2025**, *5*, 98–104. <https://doi.org/10.1145/3757892.3757906>.
39. Achiam, J.; Held, D.; Tamar, A.; Abbeel, P. Constrained Policy Optimization. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017.
40. Tessler, C.; Mankowitz, D.J.; Mannor, S. Reward Constrained Policy Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
41. Stooke, A.; Achiam, J.; Abbeel, P. Responsive Safety in Reinforcement Learning by PID Lagrangian Methods. In Proceedings of the 37th International Conference on Machine Learning (ICML), Virtual, 13–18 July 2020.
42. Chang, N.; Kumar, A. Harnessing ML and Agentic AI for Co-packaged Optics Co-optimization in 3D Heterogeneous Integration. In *Proceedings of the IEEE Electronic Design Process Symposium (EDPS) Keynote*; Ansys, Inc.: Canonsburg, PA, USA, 2025.
43. Alam, A.; Li, Q.; Yin, L.; Lin, L.; Xia, W. Accurate Thermal Simulation of 3DIC Package with Co-Packaged Optics. In Proceedings of the 62nd ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 22–25 June 2025;
44. Hsia, H.K.; Tsai, C.H.; Ting, K.C.; Kuo, F.W.; Lin, C.C.; Wang, C.T.; Hou, S.Y.; Chiou, W.C.; Yu, D.C.H. Heterogeneous Integration of a Compact Universal Photonic Engine (COUPE) for Silicon Photonics Applications in HPC. In Proceedings of the IEEE Electronic Components and Technology Conference (ECTC), Denver, Colorado, USA, 28–31 May 2024.
45. Youn, J.; Pond, J.; Chang, N.; et al. Electronic-Photonic IC Co-Design with Signal/Power Integrity and Thermal Simulation for Silicon Photonic 3D IC. In Proceedings of the DesignCon, San Jose, CA, USA, 16–18 August 2021.
46. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. <https://doi.org/10.48550/arXiv.1707.06347>.
47. Brody, S.; Alon, U.; Yahav, E. How Attentive are Graph Attention Networks? In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
48. Li, S.; Lin, X.; Liu, Y.; Li, G.; Li, J. OpticGAI: Generative AI-aided Deep Reinforcement Learning for Optical Networks Optimization. In Proceedings of the 1st SIGCOMM Workshop on Hot Topics in Optical Technologies and Applications in Networking (HotOptics '24). ACM, Sydney, Australia, 4–8 August 2024; pp. 1–6. <https://doi.org/10.1145/3672201.3674119>.
49. Patronas, G.; Terzenidis, N.; Kashinkunti, P.; Zahavi, E.; Syrivelis, D.; Capps, L.; Wertheimer, Z.A.; Argyris, N.; Fevgas, A.; Thompson, C.; et al. Optical switching for data centers and advanced computing systems [Invited]. *J. Opt. Commun. Netw.* **2025**, *17*, A87–A95. <https://doi.org/10.1364/JOCN.534317>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.