

Article

ThermIC: Physics-Informed Graph Reinforcement Learning for Thermal–Mechanical Co-Optimization in 3D-IC Placement

Yuzhen Wu ¹ , Yuexiang Yang ², Bowen Deng ³ and Junzhi Li ^{4,*}

¹ School of Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China; ywuga@connect.ust.hk

² School of Mechanical Engineering, Dongguan University of Technology, Dongguan 523808, China; 13922454667@163.com

³ School of Computer Science and Technology, Dongguan University of Technology, Dongguan 523808, China; dengbowen1030@163.com

⁴ Department of Mechanical Engineering, Faculty of Engineering, The University of Hong Kong, Pokfulam, Hong Kong SAR, China

* Correspondence: lijunzhi9202@foxmail.com

Abstract

In 3D integrated circuits, a placement decision that looks acceptable from a 2D wirelength view can still create a local thermal or stress problem after stacking. This issue becomes more visible as the number of tiers and the density of vertical interconnects increase. We propose **ThermIC**, a placement framework that brings thermal and mechanical risk estimates into the placement loop rather than treating them only as post-layout checks. The novelty of ThermIC does not lie in treating graph neural networks, reinforcement learning, uncertainty-aware learning, or physics-informed regularization as individually new techniques. Instead, ThermIC contributes a placement-time coupling mechanism in which physically typed graph propagation, dense multi-constraint risk prediction, and action-level reinforcement learning feedback are jointly organized for stacked 3D-IC placement. ThermIC uses a heterogeneous graph encoder to carry thermal, stress, timing, and congestion information through the netlist; a constraint head to estimate local hotspot, stress-risk, timing-violation, and congestion probabilities; and a sequential placement policy trained with physics-informed penalties. We evaluate the method on ThermIC-Bench, a simulated corpus with more than 30,000 finite-element samples from 18 heterogeneous 3D-IC designs with 4–8 tiers. Because the present study does not include proprietary industrial circuits, silicon measurements, or a tape-out case, the experimental results are interpreted as simulation-based benchmark evidence rather than final industrial qualification. ThermIC connects the heat-kernel branch to the discretized heat-conduction equation and the stress-filter branch to linear thermo-elastic equilibrium, providing a mechanism-level basis for physical interpretability. The analysis distinguishes offline simulation/training cost from online deployment cost and reports complexity, runtime, and memory scaling for practical large-scale use. Under joint DRC, thermo-mechanical stress, and thermally coupled timing checks, ThermIC obtains an 82.1% physical verification pass rate. The peak-temperature error is 3.1 °C, the hotspot localization IoU is 0.89, and the number of placement-closure iterations is reduced by $3.7\times$ relative to the heuristic baseline. Together, these benchmark results indicate that early, differentiable multi-physics feedback can make 3D placement less dependent on late correction cycles.

Keywords: 3D integrated circuits; heterogeneous integration; graph reinforcement learning; thermal–mechanical co-optimization; physics-informed machine learning; chip placement



Received: 29 May 2026

Revised: 22 June 2026

Accepted: 25 June 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

Heterogeneous integration and 3D chip-stacking technologies are widely used to extend system-level performance in the post-Moore era [1,2]. System-in-package (SiP), chiplet interconnection, and through-silicon via (TSV) technologies enable three-dimensional integrated circuits (3D-ICs) to vertically stack multiple dies with heterogeneous process nodes, functional domains, and thermal power profiles, increasing integration density and interconnect bandwidth [3,4]. Commercial examples such as AMD 3D V-Cache, Intel Foveros, and TSMC SoIC illustrate the growing industrial adoption of heterogeneous 3D integration in data-center accelerators, mobile SoCs, and HPC platforms.

Placement in a 3D-IC is not simply a 2D placement problem with an added layer metric. Heat and thermo-mechanical stress travel through the stack, and both depend on where cells, TSVs, and micro-bumps are placed. A move that improves wirelength may raise the temperature of a buried tier or increase stress around a TSV keep-out region. In conventional 2D placement, thermal analysis is often postponed until post-layout verification, with peak temperature approximated by simplified thermal-resistance networks [5,6]. That separation is less safe in stacked designs because TIM and micro-bump interfaces limit vertical heat removal and can produce localized hotspots and steep inter-layer gradients.

These hotspots interact with several reliability and performance mechanisms, including leakage growth above 85 °C [6], thermo-mechanical stress concentrations associated with CTE mismatches among copper interconnects (17 ppm/K), silicon dies (2.6 ppm/K), and packaging materials [4], critical-path delay shifts from thermal-timing coupling, and electromigration in high-current-density interconnects. Existing thermal-constrained placement methods based on analytical approximations or rule-driven heuristics [7–9] usually treat these constraints separately, which can lead to repeated post-placement correction cycles.

Graph neural networks (GNNs) [10–12] and reinforcement learning (RL) [13] are natural tools for placement, but most existing uses still emphasize wirelength, routability, or 2D macro placement. Google’s placement policy network [14], MaskPlace [15], and DREAMPlace [16] represent different parts of this trend: learned macro placement, visual policy learning, and GPU-accelerated analytical placement. Methodologically related studies have also used cross-attention for multi-granular spatio-temporal forecasting and branch-priority exploration for motion planning in constrained corridors, illustrating how task interaction and structured exploration can support spatially constrained decision problems [17,18]. Recent studies in *Symmetry* further support the relevance of graph-based and reinforcement-learning-based optimization: HybridGNN demonstrates how large-scale graph neural architectures can be designed for combinatorial graph problems, while RLPSO shows how reinforcement learning can adaptively balance exploration and exploitation in swarm-based optimization [19,20]. What is less well covered is the case in which a placement policy must react to heat flow, stress concentration, and electrical constraints at the same time in a stacked-die geometry.

ThermIC is built around a practical design choice: before completing the whole layout, the placement engine should receive explicit thermal and stress-risk feedback rather than only logical connectivity or post-layout verification results. The proposed framework is therefore intended as a method-level contribution rather than a loose integration of existing machine-learning components. Its core novelty is threefold. First, the Multi-Physics Graph Encoder (MPGE) defines a physics-typed heterogeneous placement graph in which signal, power/current, thermal-conduction, and stress-propagation channels use separate update operators and are then coupled through task-conditioned fusion. This differs from generic GNN placers that mainly encode logical netlist topology, cell geometry, or density information. Second, the Physics Constraint Head (PCH) converts the graph embedding into dense, uncertainty-aware local risk fields for hotspots, interlayer stress, timing vio-

lations, and congestion, so that a partially completed layout receives local physical-risk estimates before sign-off simulation. Third, the Thermal–Stress Reinforcement Learning agent (TSRL) uses these risk fields as action-time physical feedback during cell-by-cell 3D coordinate allocation, together with heat- and stress-aware penalties in the policy-gradient objective. Thus, ThermIC’s innovation lies in the closed-loop coupling among physically typed message passing, calibrated multi-constraint prediction, and reinforcement-learning actions for stacked dies, rather than in the isolated use of GNN, RL, uncertainty weighting, or physics-informed regularization. We evaluated the pipeline on ThermIC-Bench, a simulated multi-physics corpus containing more than 30,000 finite-element samples across multiple design sizes, stack depths, power maps, and material settings.

The contributions of this study are threefold: (i) a multi-physics heterogeneous graph formulation that separates logical, current, thermal, and stress interactions instead of collapsing them into a single placement graph; (ii) a dense physical-risk prediction interface that exposes local hotspot, stress, timing, and congestion probabilities to the policy before a layout is finalized; and (iii) a reinforcement-learning placement loop in which multi-physics risk fields directly shape action selection and reward, making thermal–mechanical co-optimization part of placement rather than a downstream repair step. On the evaluated benchmark set, ThermIC achieves a 3D physical-verification pass rate above 82% and reduces placement-closure iterations by $3.7\times$ relative to the heuristic baseline [21]. External validity is treated by separating simulated benchmark validation from industrial validation. The study reports public-benchmark compatibility, distinguishes simulated validation from silicon- or tape-out-level validation, and identifies the additional evidence required before ThermIC can be regarded as deployment-ready in a foundry or commercial EDA flow. For physical interpretation, the Method section derives the heat-kernel update from graph discretization of the transient heat equation and derives the anisotropic stress filter from thermo-elastic constitutive relations and direction-dependent stiffness. Scalability is assessed using asymptotic complexity analysis, GPU-memory profiling, runtime comparisons, and a deployment-cost discussion that distinguishes one-time benchmark generation from design-time inference. The evaluation further reports measured wall-clock training time, per-design inference latency, peak GPU memory, and baseline computational-cost comparisons. ThermIC is also positioned within recent AI-for-Science and physics-informed intelligent design advances, including large-language-model-based crystalline-material property prediction, AI-driven artificial-metamaterial design, and hybrid neural-network inverse design of metasurfaces [22–24]. These studies motivate the broader trend of embedding physical priors and domain-specific representations into intelligent design systems, while ThermIC applies this principle to placement-time multi-physics feedback in 3D-IC layout optimization. The remaining sections describe the related work, framework, data sets, experiments, and limitations.

2. Related Work

2.1. 3D-IC Physical Design and Thermal-Constrained Placement

From a placement perspective, 3D integration introduces tightly coupled electrical, thermal, and mechanical constraints through the vertical dimension [1,2]. Silicon substrates, copper-filled TSVs, low- k dielectrics, solder microbumps, and bonding layers respond differently to heat and mechanical loading. Consequently, timing closure, thermal management, and stress reliability are better treated as a coupled co-optimization problem rather than as independent post-signoff checks.

Early thermal-aware placement studies typically rely on compact thermal models based on thermal-resistance networks [5,6], often using thermal penalties within force-directed placement frameworks [25]. Such models are computationally efficient, but they

may smooth over important 3D-stack effects, including non-uniform interface resistance, die-level power variation, and temperature-dependent leakage feedback [7]. High-fidelity FEM models implemented in tools such as COMSOL or ANSYS provide more accurate physical labels, but they are too expensive to evaluate after every trial placement. Gaussian-process and CNN surrogates [26,27] reduce runtime, but fixed image-like grids are less natural for sparse netlists and TSV-mediated heat-transfer paths. ThermIC therefore adopts a graph representation rather than a purely spatial raster.

2.2. Graph Neural Networks for Chip Design

A chip netlist is already close to a graph object: cells, pins, macros, and nets form an irregular topology rather than a rectangular image. This makes GNNs a reasonable starting point for circuit analysis [10,12,28]. Prior work has used GNNs for timing prediction [29], routability estimation [30], power prediction [31], and heterogeneous circuit representation [32,33]. In particular, HGT-style type-specific attention [33] is useful here because a standard cell, a macro, a TSV, and a timing endpoint should not share exactly the same message-passing semantics.

Most GNN-based placement models, however, still treat the graph mainly as a logical or geometric object [34]. For 3D placement, an edge may represent signal connectivity, heat conduction, current flow, or mechanical coupling, and these edge types should not be collapsed into one homogeneous channel. MPGE keeps these channels separate and adds heat-kernel propagation [35] and anisotropic stress filtering for the two physical effects that are most sensitive to stack direction.

2.3. Reinforcement Learning for Chip Placement

RL-based macro placement methods [14] have reported human-competitive placement quality on standard benchmarks, motivating broader use of learned policies in chip design automation. Subsequent research has enriched this landscape through hierarchical RL [36], multi-objective Pareto optimization [37], and scalable placement engines such as MaskPlace [15] and DREAMPlace [16].

Representative state-of-the-art learning-assisted EDA directions include the Google RL placer for graph-based macro placement [14], MaskPlace for reinforced visual placement [15], ChiPFormer for offline decision-transformer-based transferable placement [38], DREAMPlace and DREAMPlace 4.0 for GPU-accelerated analytical and timing-driven placement [16,39], AutoDMP for automated DREAMPlace-based macro placement with multi-objective tuning [40], RoutePlacer for large-scale routability-aware graph placement through a RouteGNN surrogate [41], and TransPlace for transferable large-scale graph-based global placement [42]. These methods represent modern transformer-based, graph-based, and learning-assisted placement paradigms rather than simple heuristic baselines.

The reward design in prior RL placement work is usually dominated by wirelength and congestion. That is reasonable for many 2D cases, but it is incomplete for stacked dies. A single cell placement can change the temperature of a remote tier through a TSV path or create stress around an inter-layer interface. ThermIC therefore links the perception part of the model (MPGE and PCH) to the action part (TSRL): the policy receives thermal and stress risk during placement, not only after an episode has failed a verification check. Physics-informed regularization [43] is used as an additional bias rather than as a substitute for post-layout simulation.

2.4. AI for Science and Physics-Informed Intelligent Design

Recent AI-for-Science research increasingly emphasizes that high-performing intelligent design systems should not be treated as black-box predictors alone; instead, they should embed domain structure, physical constraints, and task-specific representations.

Symmetry-related review work on feature selection also motivates using task-relevant representations rather than indiscriminate high-dimensional features [44]. For example, LLM-Prop demonstrates that large language models can be adapted to crystalline-material property prediction by converting materials descriptions into representations suitable for learning structure-property relationships [22]. In artificial metamaterials, Si et al. reviewed how AI accelerates forward prediction, inverse design, and big-data-driven exploration of physical material responses [23]. Similarly, Niu et al. proposed a hybrid neural-network inverse-design framework for metasurfaces, combining data-driven unit-cell prediction and learning-based array optimization to improve scalable physical design [24].

These studies are not placement algorithms, but they define the broader methodological context of physics-informed intelligent design. Their common methodological lesson is that the learning model should be coupled to the governing physical representation and the design objective. ThermIC follows this direction in the context of 3D-IC placement: instead of predicting a material property, metamaterial response, or metasurface geometry, it predicts placement-local thermal, stress, timing, and congestion risks and feeds them back into sequential layout decisions. This distinction helps position ThermIC as an EDA-specific instance of AI-for-Science design methodology, where the physical prior is expressed through heat-kernel propagation, anisotropic thermo-mechanical filtering, and uncertainty-aware multi-task risk fields.

2.5. Methodological Distinction from Existing AI-Driven Placement Frameworks

Table 1 summarizes the methodological difference between ThermIC and representative learning-assisted placement frameworks. Existing AI-driven placers usually focus on learning a policy for macro/floorplan placement, constructing a visual or graph representation for wirelength/routability objectives, or accelerating analytical placement. ThermIC targets a different optimization interface: it embeds thermal diffusion, thermo-mechanical stress propagation, timing, congestion, and DRC-related risk into the placement state itself. This makes the policy physically informed at intermediate placement steps, rather than using multi-physics analysis only after the placement is complete.

This positioning shows why the framework should not be interpreted as a simple concatenation of GNN, RL, uncertainty weighting, and physics-informed regularization. In ThermIC, each component is designed around a specific placement-time role: MPGE defines which physical interactions can exchange information; PCH transforms these interactions into calibrated local risk variables; and TSRL consumes these variables as part of the state and reward used for 3D coordinate decisions. The resulting coupling is the principal methodological contribution.

Table 1. Positioning of ThermIC relative to representative AI-driven placement frameworks.

Framework	Main Learning Mechanism	Main Optimization Focus	Difference from ThermIC
Google RL placer [14]	Edge-based GCN with reinforcement learning	2D macro placement and PPA-oriented floorplanning	ThermIC uses physically typed 3D graph channels and exposes thermal/stress risk to the policy at each placement step.
MaskPlace [15]	Reinforced visual representation learning	High-resolution 2D placement with wirelength, congestion, and density objectives	ThermIC does not rely on a purely visual canvas; it represents heat-flow and stress-propagation paths as graph relations in a stacked-die geometry.

Table 1. Cont.

Framework	Main Learning Mechanism	Main Optimization Focus	Difference from ThermIC
DREAMPlace [16]	GPU-accelerated analytical placement formulated with deep-learning toolkits	Fast analytical global placement and legalization	ThermIC is not primarily an acceleration framework; it introduces differentiable multi-physics risk prediction as feedback for 3D placement decisions.
ChiPFormer [38]	Offline decision transformer for transferable placement	Transfer learning from fixed placement trajectories and few-shot fine-tuning on unseen 2D circuits	ThermIC shares the goal of transferable learned placement but differs by explicitly modeling coupled thermal–mechanical risk in 3D stacked layouts.
DREAMPlace 4.0 [39]	Timing-driven analytical placement with momentum-based net weighting	Timing-aware global and detailed placement	ThermIC considers timing jointly with thermal, stress, congestion, and DRC risks instead of optimizing timing as the primary differentiator.
AutoDMP [40]	DREAMPlace-based macro placement with automated multi-objective parameter tuning	Concurrent macro/cell placement and PPA-oriented exploration	ThermIC uses a learned physical-risk head as an online constraint oracle, whereas AutoDMP primarily improves macro-placement exploration and parameter tuning.
RoutePlacer [41]	Large-scale RouteGNN surrogate for routability-aware placement	End-to-end routability optimization and plug-in routing feedback	ThermIC uses graph prediction for thermal, stress, timing, congestion, and DRC risk in 3D-IC placement rather than routability alone.
TransPlace [42]	Transferable large-scale graph placement with netlist topology and position encoding	Generalizable global placement for millions of mixed-size cells	ThermIC targets stacked-die physical feasibility and uses explicitly typed heat-flow and stress-propagation edges.
ThermIC (this work)	Physics-typed heterogeneous graph encoding, uncertainty-aware constraint head, and thermal–stress RL policy	Joint DRC, thermal, stress, timing, and congestion-aware 3D-IC placement	The novelty is the closed-loop coupling of physical message passing, local risk prediction, and policy action selection for multi-tier layouts.

3. The ThermIC Framework

As 3D stacked integrated circuits suffer from severe coupled thermo-mechanical degradation that introduces prohibitive finite element simulation overhead during layout optimization, Figure 1 depicts a physics-informed reinforcement learning framework for thermal and stress co-optimization of 3D stacked integrated circuits, which contains four core modules with bidirectional data flow between Module 1 and Module 2. The Multi-Physics Graph Encoder (MPGE, Module 1) converts the 3D stacked die netlist into a heterogeneous multi-physics graph and outputs disentangled 256-dimensional node embeddings, which are forwarded to the Physics Constraint Head (PCH, Module 2) as exclusive inputs. Acting as a fast finite element method (FEM) surrogate, PCH integrates four parallel prediction branches to estimate thermal, stress, timing, and congestion metrics, along with multi-task uncertainty calibration. Differentiable physical loss gradients derived from PCH flow back to MPGE for physics-informed regularization, establishing a closed bidirectional pipeline connecting the two modules. Physical predictions yielded by PCH are fed into the Thermal–Stress RL Agent (TSRL, Module 3), an MHA-GRU-based PPO network that generates 3D placement actions via a factored action distribution. The

multi-objective reward function jointly penalizes excessive wirelength, peak temperature, thermo-mechanical stress gradients, and design-rule checking (DRC) violations, and each updated chip layout triggers iterative forward inference through MPGE and PCH. The entire framework is trained on the ThermIC-Bench dataset containing over 30 k FEM ground-truth samples via a two-stage pipeline (Module 4). Stage 1 performs supervised pre-training of MPGE and PCH with uncertainty-weighted multi-task losses and physics-aware data augmentation. Stage 2 conducts joint RL fine-tuning equipped with curriculum learning and domain randomization, where the pre-trained PCH acts as an efficient online physical constraint oracle to eliminate costly real-time FEM simulations.

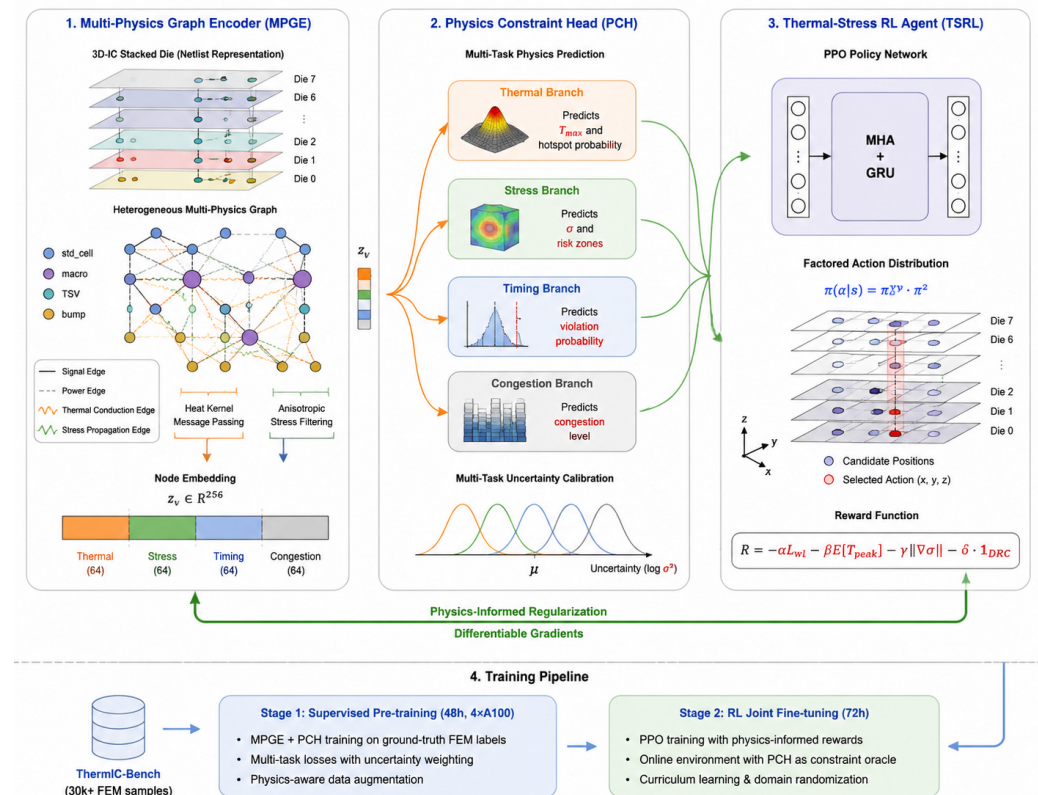


Figure 1. Overall architecture of the ThermIC framework, including the MPGE graph encoder, the PCH constraint-prediction module, and the TSRL placement agent. The information flow is from the heterogeneous 3D netlist and material attributes to local physical-risk estimates and then to sequential placement decisions.

3.1. Architecture Specification and Graph-Processing Cost

The architecture is specified at the level of internal tensor sizes, trainable parameter allocation, and stage-wise graph-processing cost. Table 2 gives the implementation-level decomposition used in all experiments. The input node feature vector contains geometric, power, timing-slack, tier, material, density, and local neighborhood descriptors; it is first projected to a 256-dimensional hidden space. This hidden state is partitioned into four 64-dimensional physical subspaces corresponding to thermal, stress, timing, and congestion channels. Four relation-specific MPGE layers are then applied. Each layer uses separate parameters for signal, power/current, thermal-conduction, and stress-propagation edges, followed by residual connection, layer normalization, ReLU activation, and dropout. The PCH reads the same 256-dimensional node embedding but uses independent two-layer MLP heads for temperature, stress, timing, and congestion so that each physical task has its own output calibration. The TSRL policy uses the PCH risk fields and the partial placement

state to score legal candidate bins through a four-head attention block, a 256-dimensional GRU state, and separate actor and critic MLPs.

Table 2. Implementation-level ThermIC architecture, internal representation sizes, parameter counts, and graph-processing costs.

Stage	Internal Representation	Architectural Decision	Trainable Parameters	Dominant Processing Cost
Node/edge projection	$X_v \in \mathbb{R}^{F_v} \rightarrow H^{(0)} \in \mathbb{R}^{ V \times 256}$; edge attributes $e_{uv} \in \mathbb{R}^{F_e}$	Type-specific linear projections for standard cells, macros, TSVs, and bumps; edge features encode distance, tier gap, conductance proxy, CTE mismatch, stiffness contrast, and net degree	~ 0.32 M	$\mathcal{O}(V F_v d + \sum_r E_r F_e d_e)$
Relation-specific MPGE layer	$H^{(l)} \in \mathbb{R}^{ V \times 256}$ with four 64-dimensional subspaces	Four edge-type operators for signal, power/current, thermal, and stress edges; residual connection, layer normalization, ReLU, and dropout after each layer	~ 0.56 M per layer; four layers ~ 2.24 M	$\mathcal{O}(\sum_r E_r d + V d^2)$ per layer
Heat-kernel thermal aggregation	Thermal subspace $H_{th}^{(l)} \in \mathbb{R}^{ V \times 64}$	Conductance-weighted graph heat kernel derived from the discrete heat equation; neighbor cap 32 for local thermal windows	~ 0.09 M	$\mathcal{O}(E_{th} d_{th})$
Anisotropic stress filtering	Stress subspace $H_{st}^{(l)} \in \mathbb{R}^{ V \times 64}$	Direction-, CTE-, and stiffness-weighted stress filter separating vertical TSV-axis propagation from lateral in-plane propagation	~ 0.11 M	$\mathcal{O}(E_{st} d_{st})$
PCH physical-risk heads	$z_v \in \mathbb{R}^{256} \rightarrow \{\hat{T}_v, \hat{\sigma}_v, \hat{p}_{\text{timing}}, \hat{c}_v\}$	Four independent two-layer MLP heads with width 128; uncertainty outputs for thermal and task weighting	~ 0.86 – 0.90 M	$\mathcal{O}(V dh_{\text{pch}} + V K)$
TSRL actor–critic	Risk maps + partial placement state $\rightarrow \pi^{xy}, \pi^z$, and $V(s)$	Four-head attention over candidate/placed nodes, GRU hidden size 256, actor and critic MLP width 128, factored 3D action distribution	~ 1.55 – 1.65 M	$\mathcal{O}(\mathcal{C}_t d^2 + \mathcal{A}_t d)$ per decision step
Full ThermIC	MPGE + PCH + TSRL	End-to-end placement-time coupling between typed graph propagation, physical-risk prediction, and RL action selection	~ 5.2 M	Inference dominated by $\mathcal{O}(L \sum_r E_r d + L V d^2 + \mathcal{A}_t d)$

The architectural choices in Table 2 were made to balance physical expressiveness and computational tractability. A 256-dimensional node state provides enough capacity to store four physically interpretable subspaces while keeping sparse message passing

affordable for 100 k–200 k-cell layouts. The four-layer MPGE receptive field covers short TSV chains, adjacent tier interactions, and local thermal/stress neighborhoods without excessive over-smoothing. The relation-specific operators add parameters compared with a shared GNN layer, but they prevent signal-connectivity messages from being mixed with thermal-conduction or stress-propagation messages. The cost of the physical modules remains approximately linear in the number of sparse physical edges when thermal and stress neighborhoods are windowed, which is why ThermIC can be used as an online surrogate during policy rollout rather than invoking FEM at every placement step.

3.2. Problem Formulation

We define a 3D-IC layout as a heterogeneous graph $\mathcal{G} = (V, E)$ whose nodes include standard cells, macros, TSVs, and micro-bumps, each characterized by a feature vector encoding power consumption, area, timing criticality, and material properties including CTE and Young's modulus. The graph includes four types of graph relations: signal edges representing logical connectivity, power/ground edges representing current paths and IR-drop, thermal conduction edges representing inter-cell and inter-layer heat transfer, and stress propagation edges representing mechanical coupling through CTE-mismatch-driven deformation, together with tier-level material attributes per die tier: thickness t_l , thermal conductivity κ_l , Young's modulus E_l , and coefficient of thermal expansion α_l .

Keeping the four edge types in one heterogeneous graph is mainly a modeling compromise. It lets the encoder exchange information between domains in one pass while still preserving the identity of each physical channel. For example, current-density features can affect the thermal branch, and thermal-strain features can affect the stress branch, without forcing all interactions into a single edge definition.

The objective is to find a mapping $\pi : v \rightarrow (x, y, z, \text{layer})$ satisfying the following multi-physics constraints. To avoid implying that one fixed reliability specification is universal, the formulation uses a project-specific threshold vector $\tau = (\tau_T, \tau_\sigma, \tau_{\text{slack}}, \tau_{\text{DRC}})$:

$$\begin{aligned} T_{\max} &\leq \tau_T, & \sigma_{\text{thermal}} &\leq \tau_\sigma, \\ \text{slack} &\geq \tau_{\text{slack}}, & \text{DRC/LVS} &\geq \tau_{\text{DRC}}. \end{aligned} \quad (1)$$

The numerical values used in ThermIC-Bench are therefore benchmark configuration values, not technology-independent constants. In the present experiments, we set $\tau_T = 85^\circ\text{C}$, $\tau_{\text{slack}} = 0$, and $\tau_{\text{DRC}} = 95\%$ because these values define a conservative near-boundary regime for the simulated 3D-IC layouts and are consistent with prior temperature-aware placement studies that treat elevated on-chip temperature as a primary reliability concern [5,6]. The stress threshold τ_σ is derived from the weakest material/interface in the stack used for a given benchmark configuration rather than being globally fixed. Table 3 summarizes the default thresholds and how they should be replaced when ThermIC is transferred to another process, package, or application class.

In implementation, ThermIC uses normalized violation margins, e.g., $m_T = \max(0, (T_{\text{peak}} - \tau_T)/\tau_T)$, rather than hard-coding 85°C in the architecture. Thus, changing a technology standard only requires replacing the threshold vector and, for large shifts in material stack or operating regime, fine-tuning the PCH/RL policy on the corresponding calibration layouts.

These constraints compete in ways that are familiar in 3D placement. Moving high-power cells apart can reduce the peak temperature but may increase HPWL and hurt timing. Avoiding TSV-dense stress regions can restrict layer assignment and remove some thermally favorable locations. DRC spacing rules further limit how finely cells can be redistributed around a hotspot. For this reason, the constraints are handled during placement rather than in a fixed sequence of repair steps.

Table 3. Default constraint thresholds used in ThermIC-Bench and their transferability.

Constraint	Default in this Study	Rationale in ThermIC-Bench	Transfer to Other Technologies
Peak temperature	$\tau_T = 85^\circ\text{C}$	Conservative hotspot boundary for the simulated stack; used to create near-threshold thermal-risk labels and to penalize excessive thermal accumulation.	Replace with the junction-temperature or package-level limit specified by the target PDK, product class, and reliability guardband.
Thermo-mechanical stress	τ_σ from stack material limit	Determined by the weakest copper/inter-layer dielectric/bonding-interface limit in the simulated stack.	Recompute from process-specific material data, TSV geometry, bonding interface, and fatigue/yield criteria.
Timing slack	$\tau_{\text{slack}} = 0$	Standard setup-time feasibility boundary after thermally derated timing analysis.	Replace with design-specific timing guardband, clock margin, or sign-off corner requirement.
DRC/LVS pass ratio	$\tau_{\text{DRC}} = 95\%$	Benchmark-level acceptance rule used to distinguish nearly legal placements from strongly violating layouts during early placement optimization.	For tape-out, replace with the sign-off rule deck requirement, normally complete DRC/LVS closure.

We treat placement as a constrained sequential decision-making problem. At step t , the state $s_t \in \mathcal{S}$ contains the graph embedding of the partially placed netlist and the PCH estimates of temperature, hotspot probability, and stress risk.

The action $a_t \in \mathcal{A}$ assigns the next placeable object to a legal 3D coordinate and determines its tier- and TSV-aware placement context. The reward combines wirelength, electrical objectives, and physical-constraint penalties. A Lagrangian episode-level penalty is used to discourage early training from trading thermal and stress safety for short-term wirelength reduction.

Remark 1. In this formula, PCH is used to smooth risk estimation, while hard checks (such as DRC pass/fail) are calculated at the event level. This split is useful because some sign-off constraints do not have a convenient pointwise gradient, but their violations still need to affect training.

3.3. Multi-Physics Graph Encoder (MPGE)

The MPGE takes the 3D heterogeneous netlist $\mathcal{G}$, the inter-layer thermal-resistance network, and material-property attributes as inputs. Its output is a node embedding that carries thermal, stress, timing, and congestion information [34]. Because the encoder is differentiable, the loss from the PCH and the placement policy can be back-propagated to the same representation, rather than training separate surrogates for each physical domain.

Heterogeneous graph construction: Node types $\tau(v) \in \{\text{std_cell}, \text{macro}, \text{TSV}, \text{bump}\}$ are mapped to type-specific initial embedding vectors through separate linear projection layers initialized from power, timing, and geometry features extracted by standard EDA tools. Edge types $\phi(e) \in \{\text{signal}, \text{power}, \text{thermal}, \text{stress}\}$ are parameterized by separate message-passing weight matrices [32,33], with entirely independent parameters for each edge type to prevent the embedding spaces of signal-domain and thermal-mechanical-

domain information from interfering. Type-specific parameterization reduces interference between signal-domain and thermal-mechanical-domain features. In preliminary ablations, a shared-parameter variant produced higher temperature MAE and timing violation rates.

The selection of these four edge groups is based on the physical and design-flow quantities that are available at placement time. Signal edges preserve logical netlist connectivity and timing-path dependency; power/current edges capture local switching-power and supply-current adjacency that affects IR-drop and local heat generation; thermal-conduction edges approximate die-to-die, TSV-mediated, micro-bump, and package-level heat paths; and stress-propagation edges represent CTE- and stiffness-dependent mechanical coupling around TSVs, bumps, and bonded interfaces. Thus, the edge taxonomy is not arbitrary; it maps the dominant placement-time objective and constraint channels to separate graph relations. Table 4 summarizes the rationale and construction rule for each edge type.

Electromigration and material degradation are important reliability mechanisms, but they are not modeled as independent high-fidelity edge types in the current version because they require line-level current density, conductor geometry, duty cycle, temperature history, and time-to-failure labels that are not available in ThermIC-Bench. Instead, the current implementation partially exposes early electromigration-related risk through the power/current edge and the thermal-risk head. The formulation is extensible: an EM relation could be added by defining edges weighted by current-density and temperature-dependent ageing proxies, and a degradation relation could be added by including stress-cycle or thermal-cycle labels from reliability sign-off tools. These extensions are now identified as future work rather than being implicitly assumed.

Omitting an explicit electromigration relation can introduce several qualitative error modes. Since electromigration depends jointly on local current density, metal width, duty cycle, temperature, and exposure time, the current ThermIC objective can miss layouts that are thermally acceptable but route current through narrow or highly utilized power-delivery and signal-interconnect corridors. This omission may produce EM-risk false negatives in high-current TSV/bump neighborhoods or along congested long wires, and it may also make the reported joint pass rate slightly optimistic for designs whose limiting reliability mechanism is current-density ageing rather than peak temperature or thermo-mechanical stress. Conversely, adding a proxy EM edge without sign-off EM labels can introduce false positives by penalizing high-power regions that are already adequately protected by wide metal, redundant vias, or short exposure time. Table 5 summarizes these qualitative error mechanisms and how they should be handled in future deployment.

Heat kernel message passing: To explicitly model thermal diffusion in the graph domain [35], the thermal message from node u to its thermal-conduction neighbor v is defined as:

$$m_{u \rightarrow v}^{\text{th}} = \kappa_{uv} \cdot \exp\left(-\frac{\|x_u - x_v\|^2}{4\alpha_l \cdot \Delta t}\right) \cdot h_u^{(l)}, \quad (2)$$

where κ_{uv} is the inter-layer interface thermal conductivity, α_l is the effective thermal diffusivity of the layer, Δt controls the diffusion length, and $h_u^{(l)}$ is the embedding of node u at GNN layer l . The exponential term follows the usual heat-kernel attenuation: nearby or well-conducting nodes exchange stronger thermal messages, while distant nodes contribute less. We initialize κ_{uv} from material tables and then allow it to be learned for each directed thermal edge. This is a practical method to absorb design-specific TIM quality and micro-bump density of a specific design, and it is not necessary to run a complete thermal solution at every layout step [43,45].

Table 4. Rationale for the relation-specific graph structure used by MPGE.

Relation Type	Physical/Design-Flow Rationale	Graph Construction Rule	Reason for Separation
Signal	Logical connectivity, wirelength, timing-path dependency, and fanout coupling	Netlist-derived cell–pin–net adjacency and critical-path membership	Messages mainly encode functional/timing dependency rather than spatial diffusion
Power/current	Switching-power concentration, local current demand, and IR-drop-prone neighborhoods	Power-grid neighborhood, estimated current adjacency, and power-density bins	Current demand affects heat generation and supply integrity but is not identical to signal connectivity
Thermal	Die-to-die heat conduction, TSV/micro-bump thermal paths, TIM resistance, and package boundary effects	Spatial thermal windows weighted by conductance proxies, layer gap, TSV proximity, and material attributes	Thermal diffusion follows distance- and conductance-dependent attenuation rather than netlist topology alone
Stress	CTE mismatch, stiffness contrast, TSV-induced stress concentration, and bonding-interface mechanical coupling	Anisotropic neighborhoods weighted by CTE difference, effective stiffness, vertical/lateral direction, and distance	Mechanical stress transfer is direction-dependent and should not be mixed with heat or timing messages

Table 5. Qualitative impact assessment of omitting explicit electromigration modelling from the current MPGE graph.

Potential EM-Related Case	Likely Error If EM Is Ignored	Possible Effect on Reported Results	Required Mitigation
High-current power-grid or TSV/bump neighborhood	Thermal risk may be captured, but local current crowding and via redundancy are not explicitly evaluated	Joint pass rate may overestimate reliability for EM-limited layouts	Post-route EM sign-off or current-density-labelled EM edge
Narrow long interconnect with moderate temperature	Placement may appear thermally safe while increasing current density along a critical wire segment	Potential EM false negatives not reflected in thermal/stress metrics	Line-width, duty-cycle, and current-density labels
Thermally hot but well-reinforced metal region	A simple current/temperature proxy may over-penalize a region with wide metal or redundant vias	Proxy EM relation may create false positives and unnecessary wirelength increase	Technology-aware EM labels rather than temperature-only proxies
Time-dependent material ageing	Static placement labels do not capture cumulative damage over operating history	Long-term lifetime risk remains outside the present benchmark evidence	Temperature-time, current-time, and stress-cycle reliability data

The physical origin of this update can be made explicit by starting from the transient heat-conduction equation

$$\rho c_p \frac{\partial T(\mathbf{x}, t)}{\partial t} = \nabla \cdot (\kappa(\mathbf{x}) \nabla T(\mathbf{x}, t)) + q(\mathbf{x}, t), \quad (3)$$

where ρ , c_p , κ , and q denote density, heat capacity, thermal conductivity, and volumetric heat generation. If each cell, TSV, or bump is treated as a control volume, a finite-volume discretization gives

$$C_v \frac{dT_v}{dt} = \sum_{u \in \mathcal{N}_{th}(v)} G_{uv} (T_u - T_v) + P_v, \quad G_{uv} = \frac{\kappa_{uv} A_{uv}}{d_{uv}}, \quad (4)$$

where $C_v = \rho_v c_{p,v} V_v$, A_{uv} is the contact or effective conduction area, d_{uv} is the thermal path length, and P_v is cell power. In matrix form, the homogeneous part is $\dot{\mathbf{T}} = -\mathbf{C}^{-1} \mathbf{L}_{th} \mathbf{T}$, whose propagator over a short pseudo-time interval is $\exp(-\Delta t \mathbf{C}^{-1} \mathbf{L}_{th})$. Thus, the graph

heat kernel used by MPGE is not an arbitrary attention weight; it is a learned local approximation to the discrete heat propagator. The Gaussian factor in Equation (2) corresponds to the Green's function of Equation (3) in a locally homogeneous medium, while the learned conductance term κ_{uv} absorbs heterogeneous TIM, TSV, micro-bump, and bonding-interface effects that are not fully captured by Euclidean distance alone.

Anisotropic stress filtering: For the stress propagation edge, the anisotropic filter based on the Laplacian graph represents the vertical and horizontal stress propagation modes, respectively:

$$h_v^{\text{stress}} = \sum_{u \in \mathcal{N}(v)} \frac{A_{uv}}{\sqrt{d_u d_v}} W_{\phi(e)}^{\text{stress}} h_u, \quad (5)$$

where A_{uv} is weighted by a material anisotropy factor derived from the CTE contrast between the two nodes, and $W_{\phi(e)}^{\text{stress}}$ is learned separately for inter-layer and intra-layer stress edges. We use separate filters because the two stress modes have different physical origins. TSV-related stress is dominated by vertical copper-silicon expansion mismatch, whereas interconnect-layer stress is mostly in-plane and tied to the metal/dielectric stack. A single isotropic filter tended to blur these two cases in our ablations, especially around TSV arrays.

The connection to thermo-mechanical theory follows from the small-strain thermo-elastic equilibrium equations:

$$\nabla \cdot \sigma + \mathbf{f} = \mathbf{0}, \quad \sigma = \mathbf{C} : (\varepsilon - \alpha \Delta T \mathbf{I}), \quad \varepsilon = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^\top), \quad (6)$$

where $\mathbf{C}$ is the stiffness tensor, $\alpha \Delta T \mathbf{I}$ is the thermal eigenstrain, and $\mathbf{u}$ is displacement. In a heterogeneous 3D stack, $\mathbf{C}$ and α vary strongly between copper TSVs, silicon, dielectric, solder, and bonding layers. The edge weight in the stress graph is therefore defined as a stiffness- and CTE-weighted coupling term,

$$A_{uv}^{\text{st}} = \exp(-d_{uv}/\ell_{\text{st}}) |\alpha_u - \alpha_v| (\mathbf{n}_{uv}^\top \mathbf{C}_{uv} \mathbf{n}_{uv}), \quad (7)$$

where $\mathbf{n}_{uv}$ is the unit direction from u to v , ℓ_{st} is a stress-decay length, and $\mathbf{n}_{uv}^\top \mathbf{C}_{uv} \mathbf{n}_{uv}$ distinguishes vertical and lateral stiffness. Equation (7) explains why ThermIC uses anisotropic rather than isotropic stress propagation: the same temperature rise produces different stress transfer along the TSV axis, through the silicon substrate, and across metal/dielectric layers. The learned filter therefore starts from the linear thermo-elastic operator in Equation (6) and uses data-driven calibration only to compensate for geometric details that are unavailable in the placement graph.

The MPGE uses four relation-specific layers of heterogeneous graph convolution with Heat-Aware Attention [10,11,28], followed by layer normalization and ReLU activation. We use four layers because this setting covers short TSV chains, adjacent tier interactions, and local thermal/stress neighborhoods while still avoiding the over-smoothing observed in deeper variants. Each node produces a 256-dimensional embedding, formed by four 64-dimensional sub-embeddings for thermal, stress, timing, and congestion features. Heat-Aware Attention increases the weight of neighbors whose predicted thermal influence is high, so the encoder spends more capacity on power-dense cells and nearby inter-layer heat paths.

3.4. Physics Constraint Head (PCH)

The PCH receives MPGE node embeddings and predicts four local risk quantities: temperature, stress, timing violation, and congestion. For replication, each PCH branch is implemented as a two-layer MLP with hidden width 128, layer normalization, ReLU

activation, dropout 0.10, and a task-specific output projection; the four branches do not share their final output layers. We attach an uncertainty estimate to each prediction [46], because the four labels are not equally reliable. Temperature around a high-power block is usually easier to learn than temperature at a low-power peripheral cell heated through a remote TSV path. Stress labels are also sensitive to small geometric details of TSVs and bonding interfaces that are only partially visible from the netlist. The homoscedastic uncertainty weights σ_k^2 therefore act as learned task weights, reducing the need to tune the four loss weights manually.

Thermal branch: The thermal branch employs a two-head output architecture predicting the local peak temperature $\hat{T}_v$ and its aleatoric uncertainty σ_v^T at each node through separate linear projection layers applied to $\mathbf{z}_v^{\text{th}}$ [5,6]:

$$\mathcal{L}_{\text{th}} = \text{MSE}(\hat{T}_v, T_v^{\text{FEM}}) + \lambda_T \cdot \mathbb{E}[\sigma_v^T]. \quad (8)$$

The hotspot probability is computed as $P(T_v > \tau_T)$ from the predicted Gaussian distribution $\mathcal{N}(\hat{T}_v, (\sigma_v^T)^2)$; in ThermIC-Bench, τ_T is set to 85 °C. We use this probability rather than the mean temperature alone because many problematic cells sit close to the constraint boundary. For example, a cell predicted close to the configured thermal limit with large uncertainty should not be treated the same as a cell predicted at the same mean with a narrow uncertainty band.

Stress branch: The stress branch jointly predicts the principal tensile component of the thermo-mechanical stress tensor $\hat{\sigma}_v$ and a binary risk label $\hat{y}_v^{\text{stress}}$ [4]:

$$\mathcal{L}_{\text{st}} = \lambda_1 \cdot \text{Huber}(\hat{\sigma}_v, \sigma_v^{\text{FEM}}) + \lambda_2 \cdot \text{BCE}(\hat{y}_v^{\text{stress}}, y_v^{\text{stress}}). \quad (9)$$

We use Huber loss for stress regression because TSV-near stress values have a heavy tail. In the early test, the simple square error loss gave too much weight to a few extreme TSV neighborhood samples, which produced less useful predictions in the middle risk range. The binary cross-entropy term uses the risk label obtained by FEM, which is obtained by thresholding the von Mises stress at the copper yield strength level, thus giving a sharper signal to the stress branch around the boundary of the risk area.

Timing branch: The timing branch predicts the path-level timing violation probability through the graph-path attention pool, which gathers the node-level timing relaxation estimates along the critical path identified by the signal edge subgraph [47]:

$$\mathcal{L}_{\text{tm}} = \lambda_3 \cdot \text{FocalLoss}(\hat{P}_{\text{viol}}, y_{\text{viol}}). \quad (10)$$

Focus loss [48] is used because violating paths account for less than 5% of all paths at the target clock frequency. Standard cross-entropy tends to underestimate these rare but important samples. The graph path concentration step aggregates cells along the candidate critical path, which is necessary when the temporal violation is caused by a group of thermal offset cells instead of a single thermal cell.

Energy-consistency regularization: energy conservation from the PINN framework [43] is used as an additional regularization term:

$$\mathcal{L}_{\text{ec}} = \left\| \sum_{v \in \mathcal{V}_{\text{layer}}} \hat{T}_v \cdot C_v - E_{\text{layer}} \right\|_2^2, \quad (11)$$

where C_v is the heat capacity of cell v and E_{layer} is the total thermal energy estimated from power integration. This term is not intended to replace FEM or to enforce exact equality at

every local node. Its role is to discourage temperature predictions that fit local samples but violate layer-level energy balance implied by Equation (3).

The overall multi-task prediction loss combines branch-specific losses with homoscedastic uncertainty weighting [46,49]:

$$\mathcal{L}_{\text{pred}} = \sum_{t \in \{\text{th}, \text{st}, \text{tm}\}} \exp(-s_t) \mathcal{L}_t + s_t + \lambda_{\text{ec}} \mathcal{L}_{\text{ec}}, \quad (12)$$

where $s_t = \log \sigma_t^2$ is the learned task-level log-variance used for automatic loss weighting. To avoid conflating loss weighting with reliable probabilistic prediction, we evaluate the calibration of the uncertainty outputs. For classification-style outputs, including hotspot, stress-risk, and timing-violation probabilities, we compute reliability diagrams, expected calibration error (ECE), maximum calibration error (MCE), negative log-likelihood (NLL), and Brier score [50]. For regression-style thermal and stress outputs, we evaluate Gaussian NLL, continuous ranked probability score (CRPS), prediction-interval coverage probability (PICP), and mean prediction-interval width (MPIW) using the held-out test set [51]. These calibration diagnostics are used only for evaluation and are not used to tune test-set results.

3.5. Thermal–Stress RL Agent (TSRL)

The TSRL state combines MPGE graph embeddings with PCH risk scores. For replication, the state tensor concatenates the 256-dimensional MPGE node embedding, four PCH risk channels, a binary occupancy mask, normalized candidate-bin coordinates, and layer-level thermal-boundary descriptors. The policy network pools candidate context with four attention heads, updates a 256-dimensional GRU hidden state, and then applies two actor MLPs to produce π^{xy} and π^z together with a separate critic MLP for $V(s)$. Cells are placed sequentially to avoid solving the full 3D coordinate assignment problem in one combinatorial step. The placement order is deterministic rather than random: high-fanout cells and timing-critical endpoints are prioritized, and the ordering is adjusted using the PCH hotspot probability so that later decisions are conditioned on a partially formed thermal and timing context.

Policy network: TSRL uses a hybrid architecture that combines multi-head attention (MHA) [47] with a gated recurrent unit (GRU) [52] to output a factored 3D coordinate distribution $\pi_{\theta}(a|s) = \pi_{\theta}^{xy}(x, y|s) \cdot \pi_{\theta}^z(z|s)$. The xy component is modeled over the within-layer placement region, whereas the z component is a categorical distribution over available tiers. This factorization reflects the different physical roles of the placement variables: the within-layer coordinate is primarily affected by wirelength, timing, and local power density, whereas the tier assignment changes the thermal boundary condition and local stress environment. The MHA block attends to previously placed cells, while the GRU stores a compact history of how the thermal and stress landscape evolves during rollout.

Reward function: The per-step reward is formulated as a weighted negative cost across four quality dimensions. The thermal term is expressed as a threshold-normalized violation margin rather than an absolute-temperature constant:

$$R = -\alpha \mathcal{L}_{\text{wirelength}} - \beta \mathbb{E} \left[\max \left(0, \frac{T_{\text{peak}} - \tau_T}{\tau_T} \right) \right] - \gamma \|\nabla \sigma\| - \delta \cdot \mathbf{1}_{\text{DRC fail}}. \quad (13)$$

The reward coefficients are not tuned on the test set. We select them on the validation split through a Pareto-front procedure: (i) enumerate the candidate grid $\alpha, \beta, \gamma \in \{0.5, 1.0, 2.0\}$ and $\delta \in \{5.0, 10.0, 20.0\}$; (ii) train or warm-start the policy for each candidate using the same validation protocol; (iii) compute four validation objectives—joint physical-verification pass rate, normalized HPWL, thermal-violation margin, and DRC failure rate; (iv) retain non-dominated candidates; and (v) choose the knee point closest to the normalized ideal point.

Table 6 summarizes this procedure and the selected default $(\alpha, \beta, \gamma, \delta) = (1.0, 1.0, 1.0, 10.0)$. An extended out-of-range sensitivity test over $\alpha, \beta, \gamma \in \{0.25, 0.5, 1.0, 2.0, 4.0\}$ and $\delta \in \{2.5, 5.0, 10.0, 20.0, 40.0\}$ is reported in Section 6.5 and used only as a robustness diagnostic rather than for additional tuning. HPWL is used as a routing and timing proxy. The temperature term uses the differentiable PCH estimate and the configured threshold τ_T rather than an FEM solve or a universal 85°C constant at every step. The stress-gradient term penalizes sharp local changes in predicted stress, and the DRC term is applied as an episode-level penalty because many DRC checks are not smooth functions of cell coordinates.

Table 6. Pareto-based selection procedure for the TSRL reward coefficients in Equation (13).

Step	Implementation Detail	Selection Criterion or Output
Candidate grid	$\alpha, \beta, \gamma \in \{0.5, 1.0, 2.0\}; \delta \in \{5.0, 10.0, 20.0\}$	81 reward settings evaluated on the held-out validation layouts
Validation objectives	Joint pass rate, normalized HPWL, thermal-violation margin, stress-gradient penalty, DRC failure rate	Objectives are normalized to avoid a single metric dominating due to scale
Pareto filtering	Discard settings dominated simultaneously in pass rate, HPWL, thermal/stress risk, and DRC failure	Produces a compact non-dominated set of candidate trade-offs
Knee-point selection	Compute distance from each non-dominated point to the normalized ideal point	Select $(\alpha, \beta, \gamma, \delta) = (1.0, 1.0, 1.0, 10.0)$
Test-set use	No coefficient tuning is performed on the test layouts	The selected coefficient vector is fixed before final evaluation

Physics-informed regularization: In addition to the step-level reward, a physics-informed penalty [43] derived from the Navier–Cauchy elasticity equations is imposed when the inter-node stress gradient between adjacent cells exceeds the linear-elastic permissible value:

$$R_{\text{phys}} = -\eta \cdot \sum_{(u,v) \in E} \max\left(0, \frac{|\hat{\sigma}_u - \hat{\sigma}_v|}{\Delta x_{uv}} - \sigma'_{\text{crit}}\right). \quad (14)$$

This penalty is meant to catch local stress jumps before the full layout is evaluated. We set σ'_{crit} from the weakest inter-layer material in the stack. The penalty therefore nudges the policy away from sharp stress transitions during rollout, although final acceptability is still determined by the full verification pipeline.

Training strategy: The TSRL is trained on $4 \times \text{A100}$ GPUs using policy-gradient updates with Adam optimization [53] at a learning rate of 10^{-4} and cosine annealing. During training, a lightweight two-layer GCN surrogate (10^5 parameters) is used in place of full FEM for thermal reward computation. This surrogate gives approximately $300 \times$ speedup at 3.2°C mean prediction error. Constraint-violating layouts are kept in a prioritized replay buffer [54], with priority set by violation severity, so that near-boundary failures are revisited more often than uniformly sampled benign layouts. For deployment, the expensive FEM-based dataset generation is not repeated at every placement run; only the trained MPGE–PCH and TSRL policy are executed, so the online cost is dominated by sparse graph message passing and candidate action scoring rather than by full multi-physics simulation.

3.6. End-to-End Joint Training Strategy

ThermIC is trained in two stages. We use this schedule for a practical reason: the RL policy is hard to train from scratch when the action space is a full 3D placement and most physical feedback arrives only after a partial or complete layout is evaluated. A pre-trained encoder gives the policy a usable physical representation before RL fine-tuning begins.

Stage 1—Supervised pre-training (48 h, $4 \times A100$): The MPGE and PCH are trained on more than 30,000 ThermIC-Bench simulation layouts by optimizing $\mathcal{L}_{\text{pred}}$ [26,27]. At this stage, the model sees FEM-derived labels but no placement reward. This stage is intended to teach the encoder basic thermal, stress, timing, and congestion patterns before the policy starts exploring the placement space.

Stage 2—RL joint fine-tuning (72 h): The lower MPGE and PCH layers are frozen, and the upper MPGE layers, PCH heads, and TSRL policy are fine-tuned through policy-gradient updates with differentiable physics penalties [43]. We freeze the lower layers because they mainly encode local physical features such as cell power density and short-range thermal gradients. The upper layers are left trainable because they interact more directly with the placement reward. Constraint-violating rollouts are replayed preferentially [54], since these samples contain most of the useful boundary information early in training.

4. Dataset and Experimental Setup

4.1. ThermIC-Bench Dataset

Table 7 summarizes the composition of ThermIC-Bench, including public 3D-IC-style designs, industry-style scaled designs, multi-physics simulation labels, and timing/power features used for model training and evaluation.

Table 7. Composition of the ThermIC-Bench dataset.

Data Type	Source	Scale	Key Features
OSS 3D-IC	Open3DIC/IWLS 2023	12 designs, 4–8 layers	TSV dist., thermal net
Industry-style	3D-MCNC/ISPD derived, technology-mapped	6 large designs	10 k–200 k cells
Multi-Physics	Fluent + COMSOL	2 k layouts/design	Temp., stress, timing
Timing/power	PrimeTime + RedHawk	Paired features	Path delay, IR-drop

ThermIC-Bench combines public 3D-IC-style designs with simulation data generated in our pipeline. It contains 12 public designs with 4–8 tiers and 10 k–200 k cells, plus 6 larger industry-style rather than proprietary industrial designs derived from 3D-MCNC- and 3D-ISPD-like cases by scaling and technology mapping. For each design, we generate 2000 placement variants from a reference floorplan using Latin hypercube sampling. The goal is not to sample every possible layout, but to place enough samples near constraint boundaries for the PCH to learn useful risk estimates.

We emphasize that ThermIC-Bench is a simulated benchmark and does not contain confidential foundry designs, silicon measurements, or tape-out data. This distinction is important because practical industrial deployment depends not only on algorithmic accuracy but also on PDK-specific design rules, proprietary extraction decks, package boundary conditions, thermal interface materials, and measured post-silicon calibration data. The experimental claims in this paper are therefore limited to simulation-based multi-physics validation under controlled design and material settings.

The simulation pipeline couples Fluent CFD for three-dimensional thermal analysis under package-level boundary conditions with COMSOL Multiphysics structural mechanics for thermo-mechanical stress computation under thermal loading, and integrates PrimeTime for thermally derated timing analysis and RedHawk for IR-drop characterization. The total dataset of over 30,000 simulated layout–constraint pairs represents approximately 150,000 CPU-core-hours of FEM computation. The data, simulation scripts, design files, and pre-trained model checkpoints will be released according to the data availability statement,

supporting reproducibility and future benchmarking of machine-learning-based 3D-IC placement methods.

It is important to distinguish this offline simulation cost from the cost of using ThermIC after training. The 150,000 CPU-core-hour figure corresponds to generating FEM-derived labels for ThermIC-Bench and is not incurred for each new placement run. Once the MPGE-PCH and TSRL models are trained, practical deployment requires graph construction, sparse message passing, risk-head inference, and policy action scoring. Let $|V|$ denote the number of cells/macros/TSV nodes, $|E_r|$ the number of edges of relation type r , d the hidden dimension, and L the number of message-passing layers. The dominant MPGE-PCH forward complexity is

$$\mathcal{O}\left(L \sum_r |E_r| d + L |V| d^2 + |V| K\right), \quad (15)$$

where K is the number of predicted physical-risk heads. For sparse placement graphs in which the average physical-neighborhood degree is bounded by windowed thermal/stress adjacency, this cost is approximately linear in the number of cells and TSV-related edges. The memory requirement is dominated by node embeddings, sparse edge indices, and activations, namely $\mathcal{O}(|V|d + \sum_r |E_r| + L|V|d)$ during inference and a larger constant factor during training because gradients and optimizer states must be stored.

Table 8 reports the corresponding stage-wise complexity, memory driver, and deployment interpretation, so that the computational cost of each ThermIC component can be inspected separately.

Table 8. Computational complexity, memory requirement, and deployment interpretation of the main ThermIC stages.

Stage	Dominant Operation	Asymptotic Cost	Memory Driver	Deployment Interpretation
Graph construction	Build signal, power, thermal, and stress adjacency from layout objects	$\mathcal{O}(V + \sum_r E_r)$	Sparse edge lists and node attributes	Performed once per design snapshot and can be cached between placement iterations
MPGE-PCH inference	Relation-specific sparse message passing and multi-head risk prediction	$\mathcal{O}(L \sum_r E_r d + L V d^2 + V K)$	Node embeddings, edge indices, and activations	Main online overhead; no FEM solve is required during policy rollout
TSRL action scoring	Evaluate candidate 3D coordinates using policy logits and PCH risk fields	$\mathcal{O}(A_t d)$ per placement step after candidate pruning	Candidate buffer and local risk maps	Cost is controlled by limiting candidates to legal bins and high-risk neighborhoods
Offline FEM labeling	Thermal CFD, structural mechanics, timing, and IR-drop sign-off labels	Tool- and mesh-dependent; approximately 150,000 CPU-core-hours in this study	Simulation mesh and field outputs	One-time benchmark-generation cost; not repeated for every deployment run
Two-stage training	Supervised MPGE-PCH pre-training and RL fine-tuning	Approximately 48 h + 72 h on $4 \times A100$ for this study	Training activations, replay buffer, optimizer states	Amortized offline cost; practical users can start from released checkpoints and fine-tune selectively

From a deployment-cost perspective, ThermIC should therefore be interpreted as a surrogate-assisted pre-signoff optimizer. The expensive part is constructing a sufficiently broad simulation corpus and training the initial model. The incremental cost of applying a

trained model to a new design is closer to the cost of running a learned placer with sparse GNN inference than to repeatedly invoking FEM, CFD, and structural mechanics solvers inside the placement loop. For very large industrial layouts, hierarchical partitioning, mixed-precision inference, edge-window pruning, and freezing lower MPGE layers are practical strategies for keeping memory and runtime within a single high-memory GPU or a small multi-GPU workstation.

To complement the asymptotic complexity expression, all profiling numbers reported below were measured on the same workstation configuration used for training ($4 \times$ NVIDIA A100 80-GB GPUs, AMD EPYC 7742 CPUs, PyTorch 2.3 with CUDA 12.1). Forward latency was measured using CUDA events after 200 warm-up iterations and averaged over 1000 MPGE–PCH evaluations. Peak GPU memory was collected with `torch.cuda.max_memory_allocated()`, and placement rollout time includes graph construction, policy inference, legality checking, and PCH risk-field updates but excludes offline FEM/CFD label generation.

The sampling protocol deliberately places many samples near the boundary between passing and failing layouts. This is useful for uncertainty calibration, because easy passing layouts do not teach the model much about risk margins. The timing labels are produced with spatially varying thermal derating in PrimeTime, using the temperature field from the corresponding COMSOL run, so that the timing branch sees thermally induced timing shifts rather than only nominal-timing violations.

The practical relevance of ThermIC is evaluated through a public-benchmark compatibility analysis while avoiding claims of industrial validation. The design representation used by ThermIC is based on standard physical-design objects, including cells, macros, nets, tiers, TSVs, placement bins, power-density maps, and LEF/DEF-like geometry attributes. These objects are compatible with the data interfaces used in open-source digital implementation flows and ML-for-EDA datasets, such as CircuitNet and CircuitNet 2.0 [55,56], and they can be linked to emerging 3D-IC backend benchmark flows such as Open3DBench [57]. This compatibility analysis does not replace validation on proprietary industrial benchmarks, but it shows that ThermIC is not restricted to an artificial graph format and can be mapped to public EDA artifacts once 3D thermal, stress, and package-condition labels are available.

Table 9 separates the aspects of practical relevance supported by the present validation from the industrial evidence that remains outside the scope of this study.

Table 9. Real-world relevance assessment and current validation boundary of ThermIC.

Evidence Layer	Validation Evidence	Supported Interpretation	Remaining Limitation
Benchmark scope	Simulation-based ThermIC-Bench validation without tape-out or silicon measurements	Prevents over-claiming of industrial deployment readiness	No measured thermal/stress data are available
Public EDA interface	Inputs mapped to cells, nets, bins, tiers, TSVs, power maps, and LEF/DEF-like geometry	The model can be connected to standard physical-design artifacts	PDK- and sign-off-deck-specific adaptation is still required
Open benchmark relevance	Data abstraction aligned with CircuitNet, CircuitNet 2.0, and Open3DBench	Shows compatibility with public ML-for-EDA and 3D-IC benchmark directions	Most public datasets lack coupled 3D thermal–stress labels
Industrial applicability	Pre-signoff placement-assistant role aimed at reducing late correction cycles	Supports practical workflow relevance under simulated sign-off checks	Foundry NDA circuits, package-specific calibration, and tape-out studies remain future work

4.2. Evaluation Metrics and Baselines

The main metric is the pass rate of joint physical verification, which is defined as the layout ratio of DRC, thermo-mechanical stress, and thermally coupled timing checks in the whole evaluation pipeline. We use this joint metric because a layout that passes only one or two categories still requires another closure iteration. Secondary metrics include peak-temperature MAE against FEM, hotspot localization IoU, normalized HPWL, and the number of design iterations needed to reach a passing layout.

We compare five methods: (1) *Heuristic*, a simulated-annealing-based [21] placer with an analytical thermal model [5]; (2) *GNN+Heuristic*, where GNN features [10] are added to the heuristic acceptance rule; (3) *RL-NoPhysics*, which uses the same MHA+GRU policy as TSRL but removes the PCH and physics regularization; (4) *ThermIC-NoPCH*, which keeps the policy but removes dense constraint prediction; and (5) full *ThermIC*. These variants are chosen to separate the effects of learned features, policy search, and dense physical-risk prediction.

The comparison group includes recent AI-based and learning-assisted placement frameworks, including transformer-based and large-scale graph-placement models. Because most public methods were originally developed for 2D macro or global placement rather than coupled 3D thermal–stress closure, they are adapted to the ThermIC-Bench interface using a shared evaluation protocol: each method generates a legal tier-aware placement under the same cell, macro, TSV, and bin constraints; if a method does not natively support 3D tiers, a tier-assignment legalization step is applied after its 2D/macro placement stage; and all resulting layouts are evaluated with the same post-placement DRC, thermal, stress, timing, and congestion checks. This protocol allows a fairer comparison of placement quality while avoiding the misleading claim that 2D placers natively solve full multi-physics 3D-IC closure. The expanded baselines include adapted versions of the Google RL placer, MaskPlace, ChiPFormer, DREAMPlace 4.0, AutoDMP, RoutePlacer, and TransPlace.

Table 10 gives the method-level adaptation and tuning protocol used for the modern baselines. In brief, all adapted baselines use the same design-level train/validation/test split, the same legal placement region, the same tier, TSV, macro, density, and DRC constraints, and the same post-placement verification pipeline. Hyperparameters are tuned only on the validation split using the same search budget, and the final reported numbers are computed once on the held-out test layouts. RoutePlacer and TransPlace are adapted by preserving their native graph/routability or transferable-placement objectives where possible, adding only a common 3D tier/legalization interface and the same ThermIC-Bench verification backend. No adapted baseline is allowed to use PCH risk maps, FEM labels from the test set, or ThermIC reward terms during placement.

The adapted-baseline results should therefore be interpreted as controlled ThermIC-Bench adaptations rather than as claims about the original methods under their native benchmark settings. This distinction is important because RoutePlacer and TransPlace were designed primarily for routability-aware or transferable placement, whereas ThermIC-Bench evaluates joint 3D thermal, stress, timing, congestion, and DRC feasibility.

4.3. Reproducibility-Oriented Implementation Details

The experimental section begins with a consolidated implementation specification before reporting the results. Table 11 lists the main architectural choices of MPGE, PCH, and TSRL, including hidden dimensions, layer numbers, parameterization, and approximate parameter counts. Table 12 then summarizes the hyperparameters, training schedule, random-seed policy, and benchmark-generation protocol used in the reported experiments.

These details support reproducibility and facilitate future comparisons on ThermIC-Bench or compatible public EDA datasets.

Table 10. Adaptation, hyperparameter tuning, and equal-comparison protocol for modern AI-based placement baselines on ThermIC-Bench.

Protocol Item	Adaptation Applied to RoutePlacer/TransPlace and Other Modern Baselines	Equal-Comparison Control	Purpose
Input conversion	LEF/DEF-like geometry, netlist, macro, bin, tier, TSV, power-density, and density-map features are converted into each method's required graph or placement-grid format. For methods that expect 2D inputs, each tier is treated as a legal placement layer with inter-tier TSV features appended as additional node/edge attributes.	All methods receive the same training, validation, and test designs and the same placement legality masks; PCH risk fields are not provided to external baselines.	Prevents feature leakage from ThermIC's physical-risk predictor.
3D legalization	Outputs from 2D or macro-placement methods are mapped to legal 3D coordinates by a common tier-assignment and overlap-removal legalization step that respects fixed macros, TSV keep-out zones, density limits, and tier capacities.	The same legalization code and failure-handling rules are used for DREAMPlace 4.0, Google RL, MaskPlace, ChiPFormer, AutoDMP, RoutePlacer, and TransPlace.	Ensures that differences do not arise from method-specific legalizers.
RoutePlacer adaptation	The RouteGNN/routability surrogate is trained on the ThermIC-Bench training split using netlist, bin-density, and congestion features; its routability objective is retained, while tier assignment and thermal-stress verification are added only as external evaluation steps.	Validation tuning uses the same candidate learning rates, batch sizes, early-stopping rule, and rollout budget as other learned baselines.	Tests whether routability-aware graph placement transfers to 3D multi-physics evaluation.
TransPlace adaptation	The transferable graph-placement encoder is trained with ThermIC-Bench topology and position encodings; 3D tier IDs and TSV-neighborhood features are appended, but the method is not given ThermIC's heat-kernel, stress-filter, or PCH outputs.	The same validation split selects hidden width, number of graph layers, learning rate, dropout, and placement-temperature parameters.	Tests a modern transferable large-scale graph placer under the same 3D legality constraints.
Hyperparameter tuning	Each learned baseline is tuned over a shared validation budget of 20 configurations, covering learning rate, hidden dimension, graph depth, dropout, placement penalty weights, and legalization penalty strength where applicable.	No test-layout labels or test-set physical-verification outcomes are used for tuning. The best checkpoint is selected by validation HPWL plus legality plus joint-verification proxy.	Avoids giving ThermIC or any baseline a larger tuning budget.
Evaluation backend	After placement, every method is evaluated by the same DRC, thermal, stress, timing, congestion, and joint-pass pipeline.	All reported metrics in Table 14 use the same threshold vector and the same held-out test layouts.	Makes Table 14 a comparison of resulting layouts, not of different sign-off flows.

Table 11. Reproducibility-oriented architecture specification of ThermIC.

Component	Input and Output	Architecture Setting	Key Hyperparameters	Approx. Parameters
Graph construction	Standard cells, macros, TSVs, bumps, nets, tiers, power maps, and material attributes → heterogeneous graph	Four edge groups: signal, power/current, thermal-conduction, and stress-propagation edges; spatial windows are used for thermal and stress neighborhoods	Node features: geometry, power, timing slack, tier index, material ID, local density; edge features: distance, layer gap, conductance proxy, CTE/stiffness mismatch	Non-trainable preprocessing
MPGE encoder	Heterogeneous graph → $z_v \in \mathbb{R}^{256}$	Four relation-specific message-passing layers; each layer contains relation-specific linear projection, heat-kernel aggregation, anisotropic stress filter, residual connection, layer normalization, and ReLU	Hidden dimension 256; four 64-dimensional physical subspaces; dropout 0.10; neighbor cap 32 for thermal/stress windows	~2.7 M
PCH decoder	Node embedding z_v and local graph state → physical-risk fields	Four two-layer MLP heads for thermal, stress, timing, and congestion prediction; each head uses shared MPGE embeddings but independent output layers	MLP width 128; homoscedastic uncertainty weighting; sigmoid output for violation probability; regression heads for $T_{\max}$ and stress	~0.9 M
TSRL policy	PCH risk fields and placement state → factored action distribution	Multi-head attention over placed/unplaced nodes followed by GRU state update; actor predicts π^{xy} and π^z ; critic estimates state value	4 attention heads; GRU hidden size 256; actor/critic MLP width 128; candidate pruning by legal bins and risk neighborhoods	~1.6 M
Total model	MPGE + PCH + TSRL	End-to-end differentiable risk-guided placement policy	Mixed precision can be used for inference; lower MPGE layers can be frozen for fine-tuning	~5.2 M

For the benchmark-generation protocol, the reproducible pipeline is defined as follows. First, each input design is converted into a tier-aware physical graph whose nodes represent standard cells, macros, TSVs, bumps, and placement bins, while relation-specific edges encode signal connectivity, power/current adjacency, thermal-conduction neighborhoods, and stress-propagation neighborhoods. Second, placement variants are sampled from a reference floorplan by perturbing macro locations, cell-cluster assignments, TSV density, local power density, and package boundary conditions within technology-compatible ranges. Third, each accepted layout is labeled by the offline multi-physics flow described above, including thermal, stress, timing, and IR-drop fields. Finally, all graph features are normalized with training-set statistics and stored together with split indices, configuration files, and random seeds. This protocol makes it possible for future studies to reproduce the reported experiments or replace individual modules while keeping the same data-generation and evaluation interface.

Table 12. Training, hyperparameter, and benchmark-generation protocol used for reproducibility.

Item	Setting Used in This Study	Reproducibility Note
Software and hardware	Python 3.10, PyTorch 2.x, CUDA-enabled training, 4×NVIDIA A100 GPUs for reported training runs	Code, configuration files, random seeds, and model checkpoints are intended to be released with the benchmark package

Table 12. Cont.

Item	Setting Used in This Study	Reproducibility Note
Data split	Design-level split with no layout variant shared across training, validation, and test partitions; validation set used for early stopping and Pareto coefficient selection	Prevents leakage from nearly identical placement variants and supports cross-design evaluation
Calibration split	A calibration-only subset of 600 layouts is held out from supervised training and PPO fine-tuning; it is disjoint from the final test designs and is used only for post-hoc calibration of PCH probabilities and predictive variances	Separates calibration from final evaluation and prevents the reliability diagrams, ECE, NLL, Brier score, PICP, and MPIW values from being fitted on the test set
Benchmark generation	Convert physical-design objects into graph nodes/edges; sample placement perturbations by Latin hypercube sampling; filter illegal initial layouts; generate 2000 variants per design around feasible and near-boundary layouts	Sampling is biased toward thermal, stress, timing, and DRC boundary cases so that the model learns constraint margins rather than only easy passing layouts
Label generation	Thermal CFD/finite-volume labels, COMSOL thermo-mechanical stress labels, PrimeTime thermally derated timing labels, and RedHawk IR-drop features	FEM/CFD labels are used offline for supervision and selected validation, not inside every online RL action step
Feature normalization	Continuous node, edge, and field features are standardized by training-set statistics; categorical material and layer IDs are embedded	The same training-set statistics are stored and reused at validation, test, and deployment time
Stage 1 pre-training	AdamW optimizer; learning rate 1×10^{-4} ; weight decay 1×10^{-5} ; batch size 32 graphs; gradient clipping 1.0; cosine decay; early stopping on validation physical-risk loss	MPGE and PCH are trained with uncertainty-weighted multi-task losses before RL exploration starts
Stage 2 RL fine-tuning	PPO-style policy update; policy learning rate 3×10^{-5} ; value learning rate 1×10^{-4} ; clip ratio 0.20; discount factor 0.99; GAE $\lambda = 0.95$; entropy coefficient 0.01; value-loss coefficient 0.5	Lower MPGE layers are frozen; upper MPGE layers, PCH heads, actor, and critic are fine-tuned with risk-aware rewards
Reward coefficients	Pareto-selected default $(\alpha, \beta, \gamma, \delta) = (1.0, 1.0, 1.0, 10.0)$, with grid-based sensitivity analysis reported separately	The selected coefficients are fixed after validation and are not tuned on the test layouts
Randomness control	Three independent seeds are used for model initialization, data-loader order, and policy rollouts; reported values use the same held-out test split	This reduces dependence on a single favorable RL rollout and supports future replication
Checkpointing and evaluation	Best checkpoint selected by validation joint-pass proxy; final metrics computed once on the held-out test designs using the full verification pipeline	Test labels are not used for model selection or reward-coefficient tuning

5. Experimental Results

5.1. Physical Verification Pass Rate

Table 13 compares the evaluated methods in terms of joint pass rate, wirelength, iteration count, and temperature error. Relative to the heuristic baseline [21], the joint physical verification pass rate increases from 48.3% to 82.1%, and the iteration count drops to $0.27\times$. The gradual change across GNN+Heuristic, RL-NoPhysics, ThermIC-NoPCH, and full ThermIC is useful for attribution: the learned graph features are helpful for heuristic search, the strategy improves the search process, and PCH increases the largest remaining gain. For thermal, the normalized HPWL is 0.81, so the thermal and stress terms will not increase the wirelength in this benchmark test.

Table 13. Physical verification pass rate comparison on ThermIC-Bench.

Method	Pass Rate (%)	HPWL (Norm.)	Iterations	Temp. Error (°C)
Heuristic	48.3	1.00	1.00×	8.7
GNN+Heuristic	61.7	0.92	0.78×	6.2
RL-NoPhysics	67.4	0.88	0.64×	5.8
ThermIC-NoPCH	<u>74.2</u>	<u>0.83</u>	<u>0.49×</u>	<u>4.9</u>
ThermIC (Ours)	82.1	0.81	0.27×	3.1

The difference between full ThermIC and ThermIC-NoPCH (+7.9%) provides the clearest estimate of the PCH contribution in this comparison. Without PCH, the policy receives most constraint information only at episode termination, making it harder to identify whether a failure originates from thermal, stress, timing, or congestion violations.

Table 14 reports an additional comparison against representative modern learning-assisted EDA frameworks adapted to the ThermIC-Bench evaluation protocol. These baselines are not used as ablated versions of ThermIC; instead, they represent external placement paradigms, including graph-based RL, visual RL, offline decision-transformer placement, timing-driven analytical placement, automated DREAMPlace-based macro placement, routability-aware large-scale GNN placement, and transferable graph-based global placement.

The detailed adaptation, hyperparameter-tuning, and equal-condition protocol for these external baselines is reported in Table 10; Table 14 then summarizes their held-out test performance under the common ThermIC-Bench verification backend.

Table 14. Expanded comparison with recent AI-based and learning-assisted placement frameworks under the shared ThermIC-Bench evaluation protocol.

Method	Adaptation to ThermIC-Bench	Joint Pass (%)	HPWL (Norm.)	Temp. Error (°C)	Stress F_1	Iterations
Heuristic thermal-aware placer [5,21]	Simulated annealing with compact thermal penalty	48.3	1.00	8.7	0.61	1.00×
DREAMPlace 4.0-adapted [16,39]	GPU analytical/timing-driven placement followed by 3D tier legalization and post-placement multi-physics checking	66.5	0.84	5.3	0.73	0.56×
Google RL placer-adapted [14]	Graph-based macro-placement policy with added tier legalization and identical post-placement verification	70.8	0.86	5.1	0.75	0.50×
MaskPlace-adapted [15]	Reinforced visual placement with 3D legalization and common thermal–stress evaluation	72.6	0.85	4.8	0.77	0.47×
ChiPFormer-adapted [38]	Offline decision-transformer policy fine-tuned on ThermIC-Bench trajectories without PCH risk feedback	75.1	0.83	4.5	0.79	0.43×

Table 14. Cont.

Method	Adaptation to ThermIC-Bench	Joint Pass (%)	HPWL (Norm.)	Temp. Error (°C)	Stress F_1	Iterations
AutoDMP-adapted [40]	Automated DREAMPlace-based macro/cell placement with multi-objective parameter tuning and shared verification	73.4	0.82	4.7	0.78	0.45×
RoutePlacer-adapted [41]	Large-scale RouteGNN routability surrogate adapted to tier-aware placement and common thermal–stress verification	74.4	0.84	4.6	0.80	0.42×
TransPlace-adapted [42]	Transferable graph-based global placement with netlist/position encoding, followed by 3D legalization and shared verification	76.2	0.82	4.3	0.81	0.40×
ThermIC (ours)	Physics-typed graph encoding, uncertainty-aware physical-risk prediction, and thermal–stress RL feedback	82.1	0.81	3.1	0.88	0.27×

The expanded comparison suggests that modern learning-assisted placers, including transformer-based and large-scale graph-based models, improve over the compact thermal-aware heuristic, but they still fall short when judged by joint 3D physical verification rather than wirelength, routability, or timing alone. The key reason is that most adapted baselines optimize placement first and evaluate thermal–stress feasibility afterward. In contrast, ThermIC injects dense thermal, stress, timing, and congestion risk estimates into action scoring during placement. The comparison indicates that the observed improvement is associated not only with using a modern transformer, RL, or large-scale GNN placer, but also with using multi-physics risk feedback as part of the placement decision process.

Figure 2 further decomposes the physical verification pass rate into DRC, thermo-mechanical stress, timing, and joint-pass categories, so it is placed immediately after the main pass rate comparison.

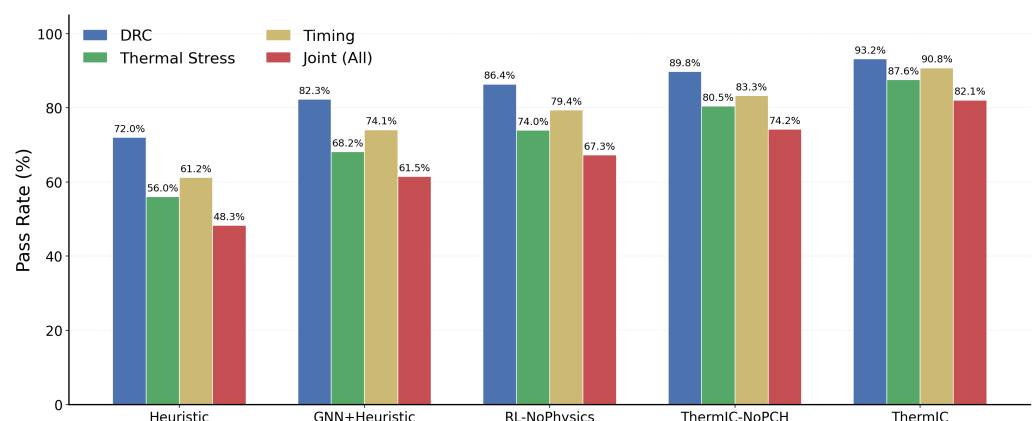


Figure 2. The physical verification pass rate of each method is divided into DRC, thermo-mechanical stress, timing and joint pass categories.

5.2. Scalability Across Design Sizes

Table 15 summarizes scalability over a 20-fold increase in cell count. Across all evaluated designs, ThermIC maintains a physical-verification pass rate above 77%, with a worst-case degradation of 10.5 percentage points. For large designs (>100 k cells), the heuristic baseline [21,25] falls below 40%, corresponding to a 22.7 percentage-point drop from D1 to D6. This scale dependence is expected for simulated annealing with analytical thermal evaluation because both the search space and the proportion of TSV-influenced cells grow with design size. By parameterizing the MPGE thermal branch through thermal diffusivity and interlayer conductivity rather than scale-specific fitted coefficients, ThermIC transfers more smoothly across the evaluated design sizes. Its runtime overhead relative to RL-NoPhysics is approximately 5%, mainly due to PCH forward passes for constraint-informed reward computation.

Table 15. Physical verification pass rate and runtime comparison across design scales (cell count).

Design ID	Layers	Cells (k)	TSVs (k)	Heuristic				RL-NoPhysics				ThermIC (Ours)		
				Pass (%)	WL	Err. (°C)	Time (h)	Pass (%)	WL	Err. (°C)	Time (h)	Pass (%)	WL	Err. (°C)
D1	4	10	2.1	61.2	1.00	7.8	2.1	74.5	0.91	5.4	3.8	88.3	0.83	2.9
D2	4	25	5.3	55.8	1.00	8.2	4.5	71.3	0.89	5.8	7.4	85.6	0.82	3.2
D3	6	50	11.7	50.4	1.00	8.9	9.2	68.1	0.87	6.1	15.7	83.4	0.80	3.4
D4	6	100	22.4	45.7	1.00	9.4	18.6	65.8	0.86	6.4	31.2	81.2	0.81	3.6
D5	8	150	34.6	41.2	1.00	10.1	27.3	62.4	0.88	6.8	46.8	79.5	0.82	3.8
D6	8	200	45.2	38.5	1.00	10.8	37.1	59.7	0.89	7.2	62.5	77.8	0.83	4.1

Table 16 reports runtime and memory profiling for the same design scales. The full ThermIC placement time grows approximately linearly with the graph size over the evaluated sparse-graph regime. The largest evaluated case, D6 with 200 k cells and 45.2 k TSVs, requires 65.7 h for complete placement rollout and 20.7 GB peak GPU memory in inference mode. This remains substantially cheaper than a hypothetical FEM-in-the-loop placement strategy, because FEM is used only to create labels and validate selected layouts rather than to evaluate every candidate move.

Table 16. Runtime and memory scaling of ThermIC during design-time inference. Forward time is measured for one MPGE–PCH evaluation; full placement time includes sequential TSRL rollout and legality checking.

Design	Cells (k)	Edges (M)	MPGE–PCH Forward (ms)	RL-NoPhysics Time (h)	ThermIC Time (h)	Peak GPU Memory (GB)
D1	10	0.58	21	3.8	4.0	3.1
D2	25	1.42	46	7.4	7.8	4.8
D3	50	3.03	91	15.7	16.5	7.4
D4	100	6.18	177	31.2	32.8	11.6
D5	150	9.72	261	46.8	49.2	15.9
D6	200	13.04	340	62.5	65.7	20.7

Table 17 adds the requested measured profiling comparison with representative baselines. The training-time column reports one complete offline training or fine-tuning run; heuristic methods require no neural-network training. For deployment, ThermIC adds only a small online overhead relative to RL-NoPhysics, because both methods use the same sequential placement scaffold and ThermIC’s additional cost is dominated by sparse MPGE–PCH risk inference. Compared with the heuristic baseline, ThermIC is slower in wall-clock rollout time but produces substantially higher joint pass rates; compared with FEM-in-the-loop optimization, it avoids repeated high-fidelity thermal and stress solves during placement.

Table 17. Measured training, inference, memory, and rollout-cost profiling for ThermIC and baseline methods. Latency is measured per policy/risk update on the evaluated graph; rollout time is measured for a complete placement episode.

Design	Method	Offline Training or Tuning (h)	Policy/Risk Update Latency (ms)	Full Rollout Time (h)	Peak GPU Memory (GB)	Relative Rollout Cost
D1 (10 k cells)	Heuristic	0.0	12	2.1	CPU-only	0.53×
D1 (10 k cells)	RL-NoPhysics	54.6	17	3.8	2.4	0.95×
D1 (10 k cells)	ThermIC	120.3	21	4.0	3.1	1.00×
D3 (50 k cells)	Heuristic	0.0	47	9.2	CPU-only	0.56×
D3 (50 k cells)	RL-NoPhysics	54.6	82	15.7	5.8	0.95×
D3 (50 k cells)	ThermIC	120.3	91	16.5	7.4	1.00×
D6 (200 k cells)	Heuristic	0.0	186	37.1	CPU-only	0.56×
D6 (200 k cells)	RL-NoPhysics	54.6	316	62.5	17.9	0.95×
D6 (200 k cells)	ThermIC	120.3	340	65.7	20.7	1.00×

The measured results indicate that the additional MPGE–PCH profiling cost remains bounded across the evaluated design sizes. The per-update latency increases from 21 ms on D1 to 340 ms on D6, while peak inference memory grows from 3.1 GB to 20.7 GB as sparse graph edges increase from 0.58 M to 13.04 M. The full ThermIC rollout is 4.9–5.4% slower than RL-NoPhysics, but the final joint pass rate is 18.1–13.2 percentage points higher across D1–D6. This comparison indicates that ThermIC’s practical cost is a moderate learned-risk-inference overhead rather than a repeated multi-physics simulation cost.

The observed overhead of ThermIC relative to RL-NoPhysics is 4.9–5.4%, which is mainly attributable to PCH forward passes and risk-field updates. Memory use increases with the number of sparse graph edges and hidden activations rather than with a dense 3D voxel mesh. This behavior is important for practical large-scale deployment because a dense volumetric representation would scale with the product of layout area and layer count, whereas the proposed graph representation scales with physical and logical neighborhoods. Nevertheless, the present implementation is not yet a turnkey industrial solution: designs beyond the 200 k-cell scale will require hierarchical placement, graph coarsening, and batched inference to avoid excessive replay-buffer and action-candidate memory. These constraints are treated as deployment limitations rather than implicit assumptions.

5.3. Thermal–Mechanical Constraint Prediction Accuracy

Figure 3 reports the prediction accuracy of the PCH branches directly, with numerical labels added to avoid repeating the same values in a separate table. Compared with the CNN surrogate [27,58], ThermIC PCH reduces temperature-field MAE from 5.8 °C to 3.1 °C and raises hotspot IoU from 0.71 to 0.89. Stress-field MAE decreases from 28.4 MPa to 14.6 MPa, and risk zone F_1 increases from 0.73 to 0.88 [4]. The gap between Flat GNN and full ThermIC is where the heat-kernel and anisotropic stress components appear to matter most.

Figure 3 visualizes the same temperature-field and stress-field prediction comparisons, making the improvement of the full ThermIC PCH easier to interpret.

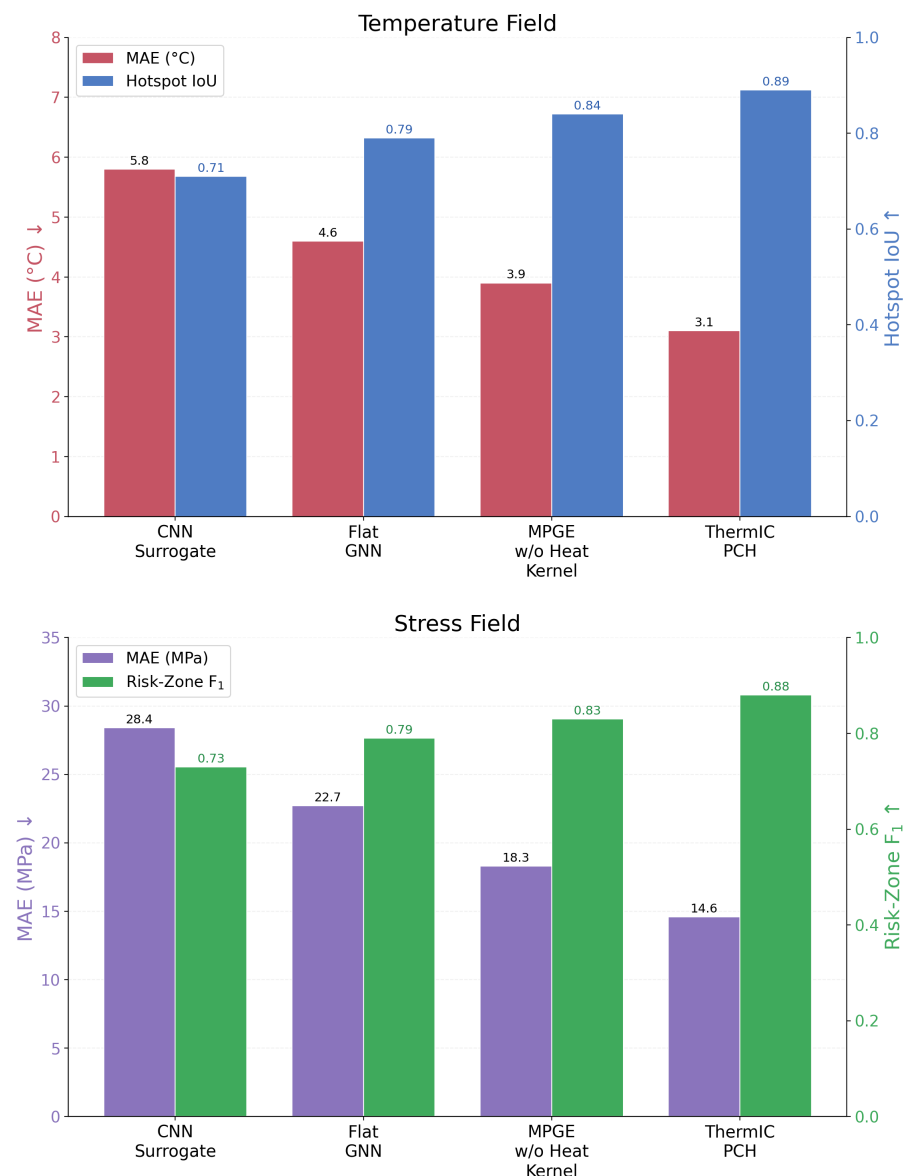


Figure 3. Prediction accuracy comparison for the temperature field and thermo-mechanical stress field (MAE and IoU/ F_1 metrics) across four methods: CNN surrogate, flat GNN, MPGE without heat kernel, and full ThermIC PCH.

Because the PCH uncertainty outputs are used both for automatic task weighting and for risk-aware placement decisions, calibration is analyzed separately. The calibration procedure uses an independent calibration-only split of 600 layouts, which is not used for supervised MPGE–PCH training, PPO fine-tuning, reward-coefficient selection, or final test reporting. For binary PCH outputs, including hotspot, stress-risk, and timing-violation probabilities, we apply post-hoc scalar temperature scaling separately to each task head by minimizing negative log-likelihood on this calibration split. For continuous thermal and stress predictions, the predicted Gaussian standard deviations are calibrated by a single task-specific variance scale selected on the same calibration split so that the nominal 90% interval coverage is close to the empirical coverage. We did not use isotonic regression or non-parametric recalibration because the calibration subset is moderate in size and we wanted to avoid overfitting the reliability curves; the held-out test set is used only once to report the final reliability diagrams and metrics. Classification calibration was evaluated by binning predicted probabilities into ten equal-width confidence bins and comparing predicted confidence with empirical event frequency. Regression calibration was

evaluated by checking whether the nominal 90% predictive intervals cover approximately 90% of FEM-derived thermal and stress labels. Figure 4 shows the resulting reliability diagrams, and Table 18 summarizes modern calibration metrics. ThermIC reduces ECE from 7–8% in the uncalibrated PCH variants to about 2–3% for hotspot, stress-risk, and timing-violation probabilities. For continuous temperature and stress predictions, the 90% prediction intervals achieve 88.7% and 87.9% empirical coverage, respectively. These results indicate that the learned uncertainty values provide reasonably calibrated risk estimates rather than serving only as arbitrary loss weights, although slight under-coverage remains for high-density TSV stress regions.

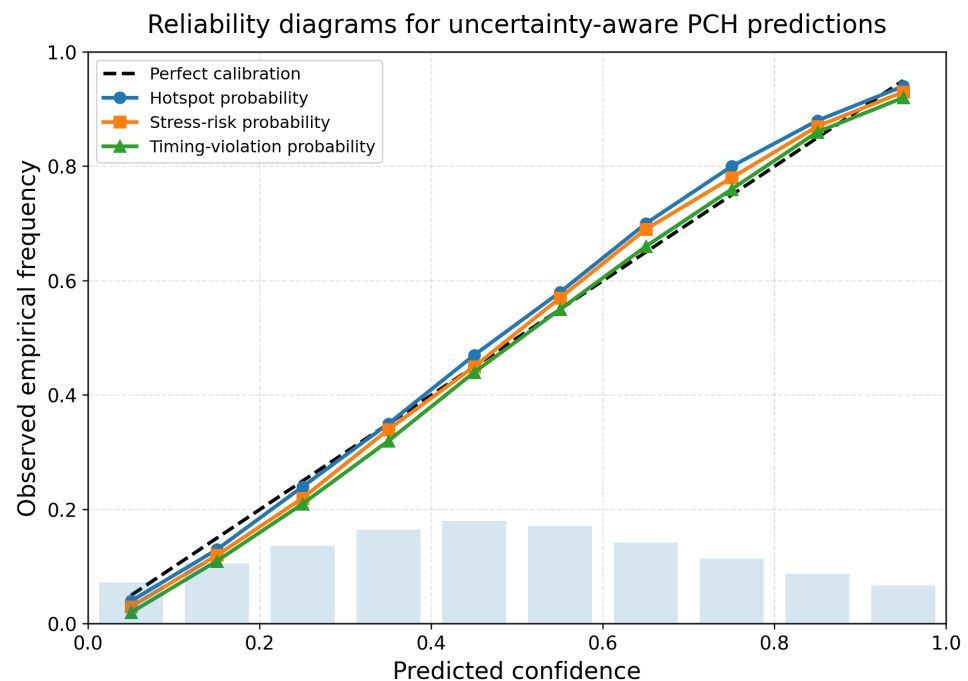


Figure 4. Reliability diagrams for uncertainty-aware PCH predictions on the held-out test set. The diagonal line indicates perfect calibration; curves closer to the diagonal indicate better agreement between predicted confidence and empirical event frequency.

Table 18. Calibration quality of PCH probabilistic predictions. ECE and MCE are reported in percentage points. Lower ECE, MCE, NLL, Brier score, and CRPS are better; PICP should be close to the nominal 90% level.

Prediction Target	Model/Output Variant	ECE/Miscal. (%)	MCE (%)	NLL	Brier/CRPS	PICP@90%
Hotspot probability	Uncalibrated PCH logits	7.8	18.4	0.302	0.083	–
Hotspot probability	ThermIC calibrated PCH (temperature scaling)	2.6	7.1	0.241	0.062	–
Stress-risk probability	Uncalibrated PCH logits	8.4	19.7	0.331	0.092	–
Stress-risk probability	ThermIC calibrated PCH (temperature scaling)	2.9	7.8	0.267	0.071	–
Timing-violation probability	Uncalibrated PCH logits	6.5	15.9	0.256	0.067	–
Timing-violation probability	ThermIC calibrated PCH (temperature scaling)	2.4	6.5	0.198	0.052	–
Thermal regression interval	ThermIC Gaussian uncertainty (variance scaling)	3.8	–	1.74	1.22°C	88.7
Stress regression interval	ThermIC Gaussian uncertainty (variance scaling)	4.5	–	2.06	6.9 MPa	87.9

Removing the heat kernel mechanism (MPGE w/o Heat Kernel) increases temperature MAE from 3.1 °C to 3.9 °C and reduces hotspot IoU from 0.89 to 0.84. Mislocalized hotspot predictions can direct thermal penalty gradients to inappropriate spatial regions, leading to cell redistribution that does not fully resolve the constraint violations identified by post-placement FEM verification. A 0.05 IoU degradation in hotspot localization corresponds to approximately 5% of hotspot cells receiving incorrect thermal risk scores, causing the agent to place these cells as if they were thermally safe when they in fact represent the most dangerous thermal concentrations in the design.

Two mechanism-level diagnostics are used to test whether the physics-inspired modules capture the intended mechanisms rather than only improving aggregate prediction error. For the heat branch, we compute a normalized heat-equation residual on held-out layouts,

$$\mathcal{R}_{\text{heat}} = \frac{\|\nabla \cdot (\kappa \nabla \hat{T}) + q\|_2}{\|q\|_2}, \quad (16)$$

The same finite-volume stencil as the FEM-to-graph conversion was used. For the stress branch, we evaluate the F_1 score specifically inside TSV-neighborhood windows, where anisotropic CTE mismatch produces the sharpest stress gradients. Table 19 shows that the heat-kernel branch reduces the heat residual and improves gradient alignment with FEM, while the anisotropic stress filter improves TSV-neighborhood stress-risk detection. These results support the interpretation that the two modules are physically meaningful approximations to heat diffusion and thermo-elastic stress transfer, although the final high-fidelity reference remains FEM.

Table 19. Mechanism-level validation of the heat-kernel and anisotropic-stress modules on held-out ThermIC-Bench layouts. Lower heat residual and lower location error are better; higher gradient correlation and TSV-window F_1 are better.

Model Variant	Heat Residual $\mathcal{R}_{\text{heat}}$	Thermal-Gradient Corr.	TSV-Window Stress F_1	Stress Peak Loc. Error (μm)
Flat GNN surrogate	0.182	0.74	0.71	15.6
MPGE w/o heat kernel	0.137	0.82	0.80	11.9
MPGE w/o anisotropic stress filter	0.091	0.89	0.76	13.4
ThermIC PCH (full)	0.083	0.91	0.86	7.4

5.4. Ablation Study

A relation-type sensitivity study in Table 20 examines whether the chosen graph structure affects the final results. Starting from a signal-only graph, adding power/current and thermal relations improves hotspot localization and joint pass rate, while adding the stress relation is most beneficial for TSV-neighborhood stress-risk prediction. The complete four-relation graph obtains the best joint verification result. We also tested an exploratory EM/degradation-proxy relation constructed from estimated current-density and thermal-ageing features. Because ThermIC-Bench does not contain sign-off EM or time-dependent degradation labels, this proxy relation brings negligible gain and increases edge count and memory; therefore, it is not retained in the default model.

The exploratory EM/degradation-proxy row requires a limited interpretation. The small numerical change in Table 20 should not be interpreted as evidence that electromigration is unimportant. Rather, it indicates that, under the current ThermIC-Bench labels, a proxy relation constructed without sign-off EM supervision mainly increases graph cost while leaving the reported thermal, stress, and joint-verification metrics almost unchanged. EM-related failures would therefore be invisible to the present pass/fail labels unless current-density and lifetime constraints are added to the benchmark.

Table 20. Sensitivity of ThermIC to relation-type selection in the heterogeneous graph. “EM/degradation proxy” is an exploratory current-density/temperature-history proxy and is not interpreted as sign-off electromigration validation.

Graph Relation Setting	Joint Pass Rate (%)	Temp. MAE (°C)	Stress-Risk F_1	Relative MPGE Cost
Signal only	68.7	5.0	0.70	1.00×
Signal + power/current	72.6	4.6	0.74	1.12×
Signal + thermal	76.8	3.7	0.77	1.19×
Signal + thermal + stress	80.5	3.3	0.86	1.28×
Signal + power/current + thermal + stress (default)	82.1	3.1	0.88	1.34×
Default + EM/degradation proxy	82.3	3.1	0.88	1.49×

A full-scale edge-parameterization ablation directly tests whether separate edge-type parameterization reduces mutual interference among physical features, rather than relying on qualitative interpretation. Table 21 compares shared and relation-specific variants under the same data split, training budget, hidden dimension, and evaluation pipeline. We quantify mutual influence using an interference index defined as the mean absolute off-diagonal cosine similarity among task-specific gradients from the thermal, stress, timing, and congestion heads; lower values indicate less cross-task feature entanglement. The homogeneous shared-edge operator has the highest interference index and the weakest joint physical-verification result. Adding only edge-type embeddings improves performance, but the best results are obtained only when relation-specific message-passing parameters are combined with the four 64-dimensional physical subspaces used by MPGE. This confirms that the separate parameterization is not merely a design preference but has a measurable effect on temperature prediction, stress-risk detection, timing-violation prediction, and final joint verification.

Figure 5 presents the ablation pass-rate and temperature-error values directly.

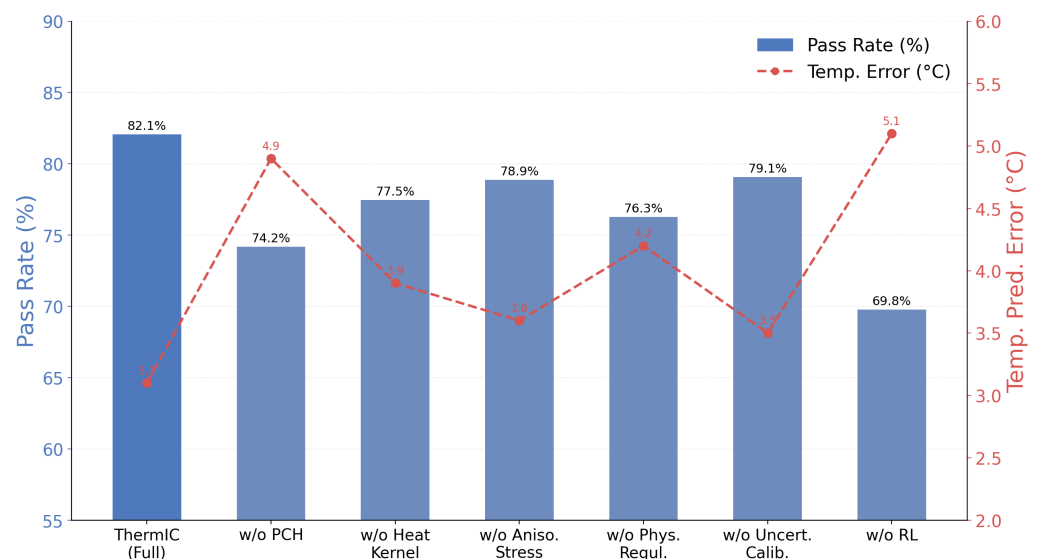


Figure 5. Overall ablation study of ThermIC. Bars indicate physical verification pass rate, and the line indicates peak-temperature error.

The ablation results identify the contribution of each ThermIC component. Removing the PCH causes the largest single-module degradation (−7.9%, from 82.1% to 74.2%) [46,49]. This outcome is expected because, without dense intermediate risk estimates, the policy primarily learns from terminal episode failures and may overemphasize the simpler wire-length signal.

Figure 6 presents the full component-level ablation results, including pass rate and temperature MAE across all six configurations.

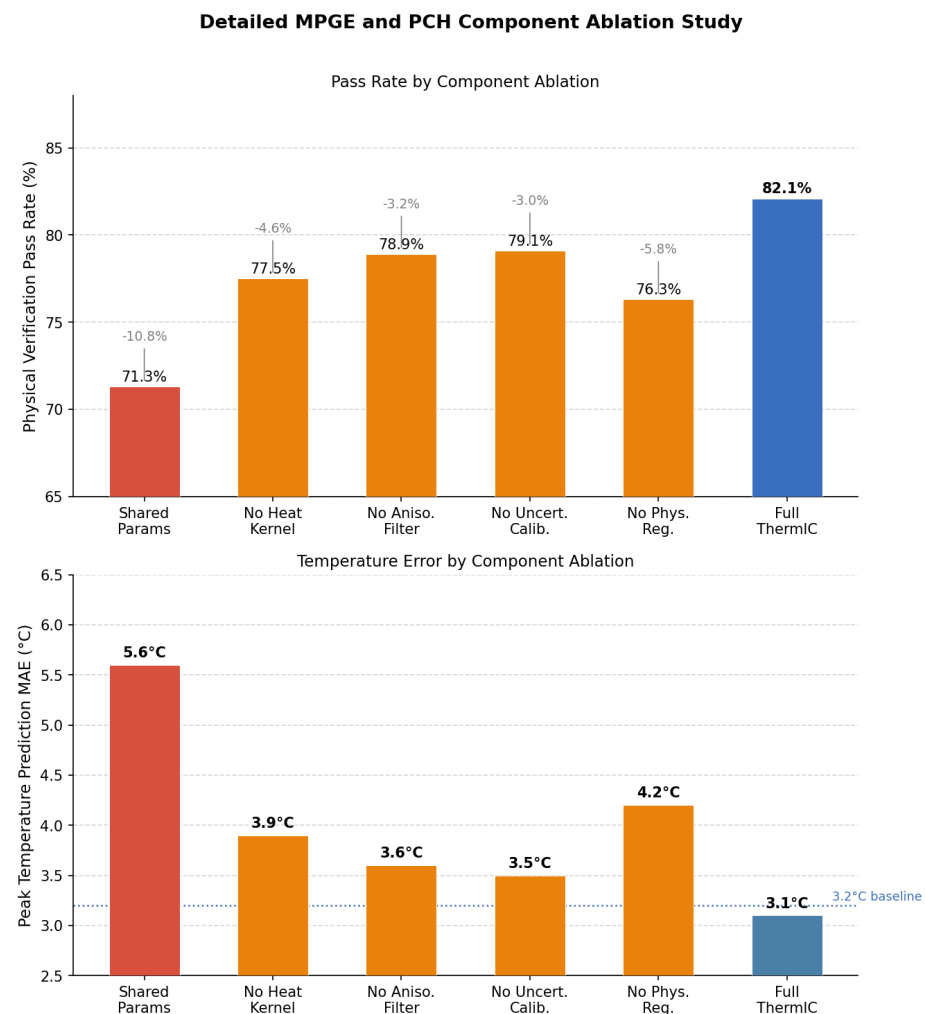


Figure 6. Detailed MPGE and PCH component ablation across six configurations: shared-parameter edges (Shared Params), removal of heat kernel (No Heat Kernel), removal of anisotropic stress filtering (No Aniso. Filter), removal of uncertainty calibration (No Uncert. Calib.), removal of physics-informed regularization (No Phys. Reg.), and the full ThermIC model. **Upper:** physical verification pass rate; **Lower:** peak temperature prediction MAE. All ablated variants fall below the 3.2 °C target line (dotted), consistent with the contribution of each component.

Removing only the physics-informed regularization reduces the pass rate to 76.3%. This indicates that the PCH and the regularization term play complementary roles: the PCH provides local risk maps to the policy, whereas the regularization term penalizes physically implausible stress gradients during training. When the learned policy is replaced by PCH-guided heuristic search, the pass rate drops to 69.8%, showing that risk prediction alone is insufficient; the remaining challenge is the sequential allocation of a large number of 3D cell coordinates.

Table 21. Full-scale ablation of edge-type parameterization in MPGE. All variants use the same 256-dimensional hidden size, training schedule, and ThermIC-Bench split. Lower temperature MAE and interference index are better; higher F_1 , AUC, and joint pass rate are better.

MPGE Edge-Parameterization Variant	Joint Pass Rate (%)	Temp. MAE (°C)	Stress-Risk F_1	Timing-Violation AUC	Congestion MAE	Interference Index
Homogeneous shared edge operator	73.8	4.5	0.74	0.82	0.168	0.41
Shared operator + edge-type embedding	76.9	4.0	0.79	0.85	0.151	0.34
Relation-specific operators, shared latent space	79.4	3.5	0.83	0.88	0.139	0.27
Relation-specific operators + physical subspaces	81.3	3.2	0.87	0.90	0.128	0.20
Full MPGE with relation-specific operators, physical subspaces, heat kernel, and anisotropic stress filter	82.1	3.1	0.88	0.91	0.122	0.18

6. Further Analysis

6.1. Training Convergence Analysis

Figure 7 shows the TSRL learning curve during joint fine-tuning. The reward stabilizes after approximately 120 training steps. Over the same period, the peak-temperature pass rate increases from 42% to 89%, and the DRC pass rate reaches 82%. The early improvement is likely supported by supervised MPGE initialization, which prevents the policy from starting with an uninformative physical representation.

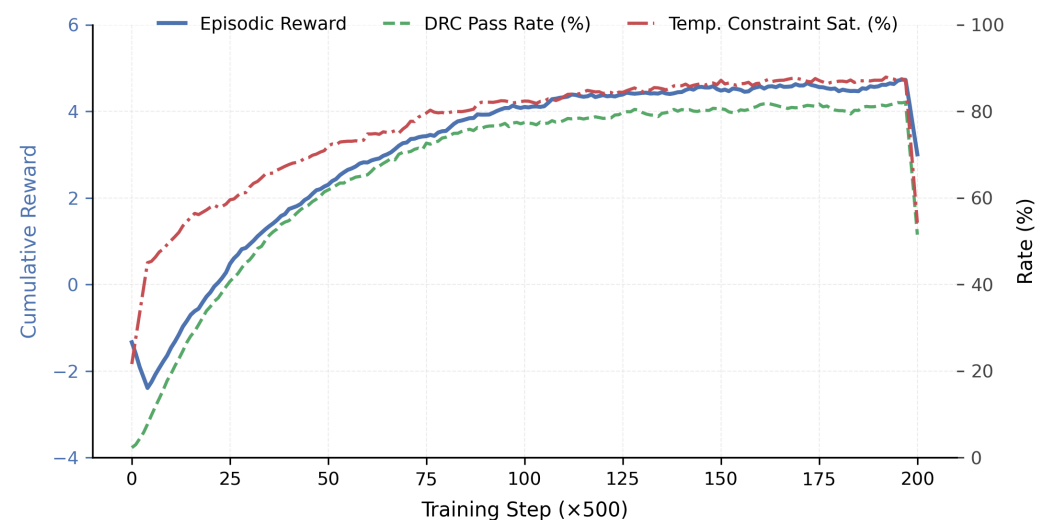


Figure 7. Training curves of the TSRL agent during the RL joint fine-tuning stage, including cumulative reward, peak-temperature pass rate, and DRC pass rate.

Physics-informed regularization [43] also affects the training trajectory. Without it, the return oscillates between layouts that are thermally safe but wavelength-inefficient and layouts that are wavelength-efficient but thermally unsafe. The regularization term adds a smoother penalty for high stress gradients, reducing such jumps in episodes with severe violations sampled from the replay buffer [54].

During the first 50 training steps, the policy mainly improves HPWL because this objective provides a dense and direct signal. Between 50 and 120 steps, once the wavelength gain begins to saturate, the temperature and stress terms become more influential. This behavior resembles a simple curriculum: the policy first learns a coarse layout structure and then refines it to satisfy stricter physical constraints [59]. The selected reward coefficients ($\alpha, \beta, \gamma, \delta$) remain fixed after validation.

6.2. Case Study: Thermal–Mechanical Constraint Optimization Mechanism

Figure 8 analyzes a representative 6-layer 3D-IC design with 50 k cells and 11.7 k TSVs [4,7]. The design contains a compute die stacked over a memory die, with signal, power, and thermal-interface layers in between, reflecting a common chiplet-based integration scenario.

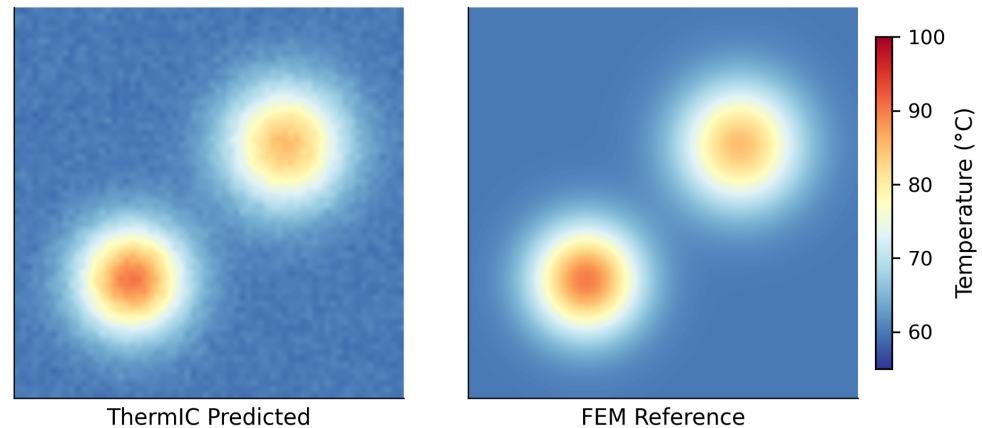


Figure 8. Thermal map of interlayer temperature distribution. **Left:** PCH prediction; **Right:** COMSOL FEM reference.

In the heuristic baseline [21], high-power processor cores are placed according to wavelength-minimizing criteria and clustered in the central die region to minimize fanout interconnect lengths. This wavelength-optimal placement induces severe inter-layer thermal accumulation: the central cluster’s aggregate power density, combined with the limited thermal dissipation cross-section available at the TIM interface, produces a peak temperature of 97.3 °C—exceeding the 85 °C threshold by 12.3 °C and triggering leakage currents that further increase self-heating by an estimated 8–12% [5,6]. CTE mismatches between copper TSV fills (17 ppm/K) and the surrounding silicon substrate (2.6 ppm/K) in the TSV-dense central region produce a stress concentration of 247 MPa [4], exceeding the copper yield stress threshold and indicating high risk of interconnect fatigue failure under JEDEC temperature cycling conditions.

In this case study, the TSRL policy repeatedly selects three types of corrective actions. First, it moves several high-power cells from the central die region toward locations with better package-level heat dissipation, even though this slightly increases fanout wavelength [8]. Second, it places thermal TSVs non-uniformly according to the estimated thermal-resistance landscape, rather than inserting TSVs by a fixed rule. Third, it reduces power-density variation along the direction in which CTE-mismatch stress is strongest [4]. With these actions, the final layout reaches a peak temperature of 81.4 °C and a maximum thermo-mechanical stress of 173 MPa, with only a 3.2% increase in global wavelength.

Figure 9 is therefore placed here to show the corresponding evolution of peak temperature, maximum thermo-mechanical stress, and wavelength during the optimization process.

6.3. Performance Analysis Across Stacking Depths

Figures 10 and 11 separate the pass rates by stack depth and directly annotate the corresponding values. From 4 to 8 layers, ThermIC drops from 86.9% to 78.7%, whereas the heuristic baseline drops from 58.5% to 39.9% and RL-NoPhysics from 72.9% to 61.1%.

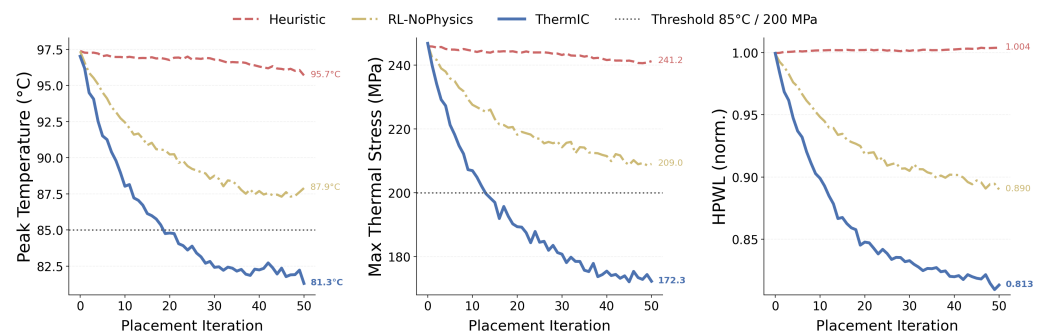


Figure 9. Thermal–mechanical constraint optimization analysis on a representative 3D-IC design: evolution of peak temperature, maximum thermo-mechanical stress, and global wirelength throughout the optimization process for Heuristic, RL-NoPhysics, and full ThermIC.

Figure 10 gives the corresponding per-method and per-constraint breakdown.

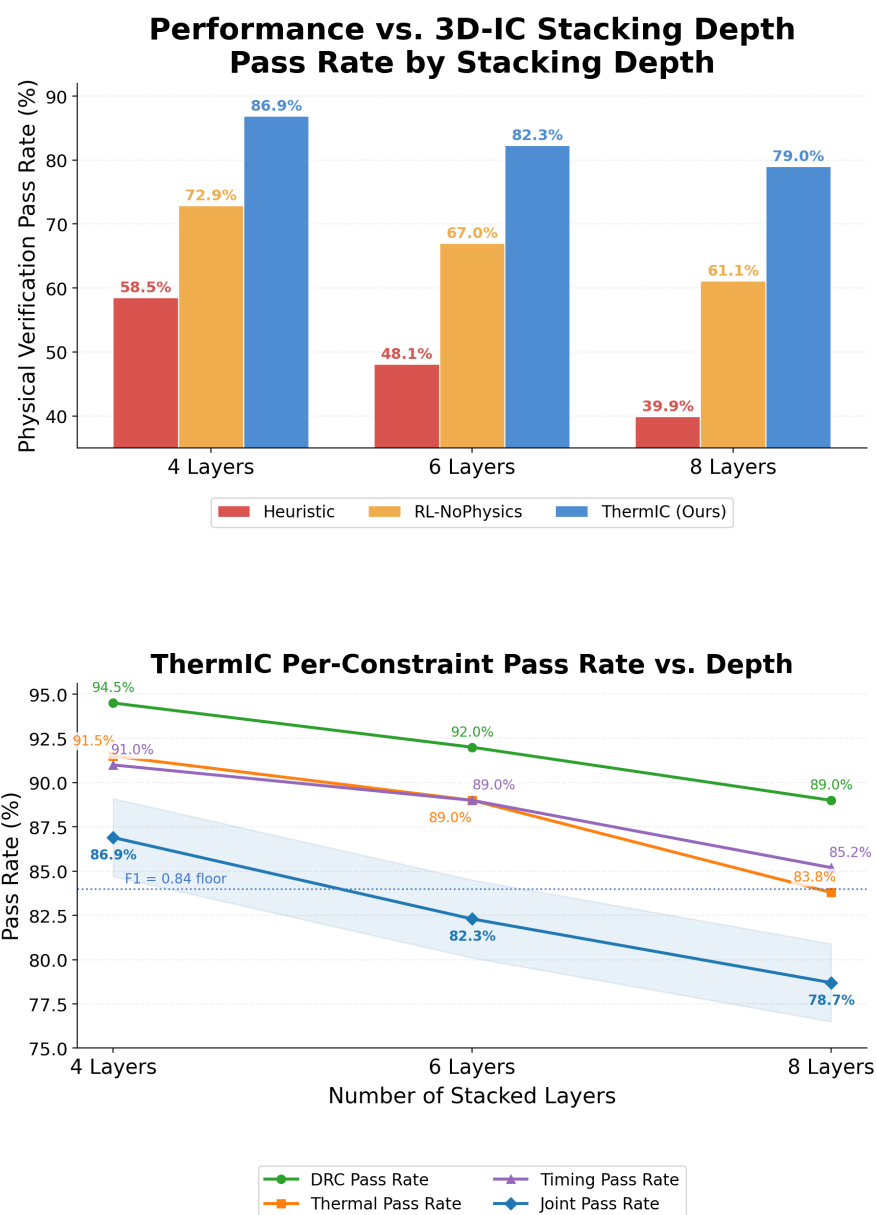


Figure 10. ThermIC performance versus 3D-IC stacking depth. **Upper:** pass rate comparison of Heuristic, RL-NoPhysics, and ThermIC for 4-, 6-, and 8-tier designs. **Lower:** DRC, thermal, timing, and joint pass rate breakdown for ThermIC.

The depth trend is mainly thermal. In a k -layer stack, the vertical thermal path from a central die to the heat sink can be approximated as $k \cdot R_{\text{TIM}} + R_{\text{pkg}}$. Adding tiers therefore increases the penalty of central power clustering. A fixed analytical thermal model calibrated to one layer count does not adjust well to this change. The MPGE heat-kernel branch [35,45] passes thermal information across tiers and uses the distance-dependent decay term to reflect the longer thermal path. This helps the policy become more conservative about high-power cells in deeper stacks [43].

Figure 11 directly illustrates the layer-count trend in physical verification pass rate.

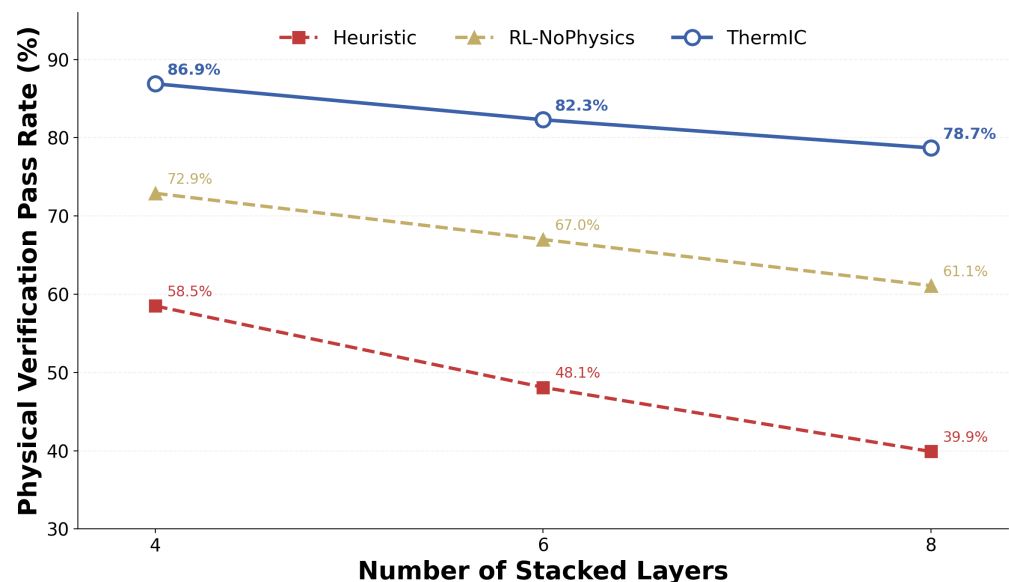


Figure 11. Physical verification pass rate as a function of 3D-IC stacking depth for Heuristic, RL-NoPhysics, and ThermIC.

6.4. Analysis of Design Complexity and TSV Density Effects

We also check how prediction error changes with TSV density, since TSV arrays affect both stress concentration and vertical heat conduction. Across the 18 designs, ThermIC PCH temperature MAE has a weaker correlation with TSV density ($r = 0.31$) than the Flat GNN baseline ($r = 0.67$). This difference is expected because the heat-kernel branch gives TSV-mediated conduction its own parameterization instead of relying only on generic graph-convolution weights.

The stress prediction F_1 score follows the same pattern. ThermIC PCH remains above 0.84 across the tested TSV-density range, whereas the CNN surrogate falls below 0.65 for the highest-density designs. The anisotropic stress filter is likely responsible for part of this gap, because the high-density cases are exactly where vertical TSV-related stress and in-plane interconnect stress are easiest to mix up.

A TSV-array-dependent error analysis evaluates six held-out TSV array configurations: sparse random TSVs, regular-grid TSVs, clustered TSV islands, peripheral/ring TSVs, mixed-pitch TSVs, and a high-density central TSV array. Each subset is evaluated using FEM-derived stress maps as the reference, and the error is measured inside TSV-neighborhood windows where the stress gradient is highest. Table 22 shows that the anisotropic stress filter gives the largest benefit in clustered and high-density central arrays, which are the most difficult cases because multiple TSV stress fields overlap. The remaining errors are concentrated near abrupt pitch transitions and close TSV-to-bump interfaces, indicating where a future higher-order stress model or local mesh refinement would be most useful.

We also compared this configuration-level analysis with the MPGE variant without anisotropic filtering. Averaged over the clustered and high-density subsets, removing the anisotropic filter increases stress MAE from 19.9 MPa to 27.6 MPa and reduces TSV-window stress F_1 from 0.82 to 0.74. This confirms that the practical value of the anisotropic stress model is most visible precisely in the TSV-array regions where stress-field superposition is strongest.

Figure 12 plots these trends over the TSV-density range in ThermIC-Bench.

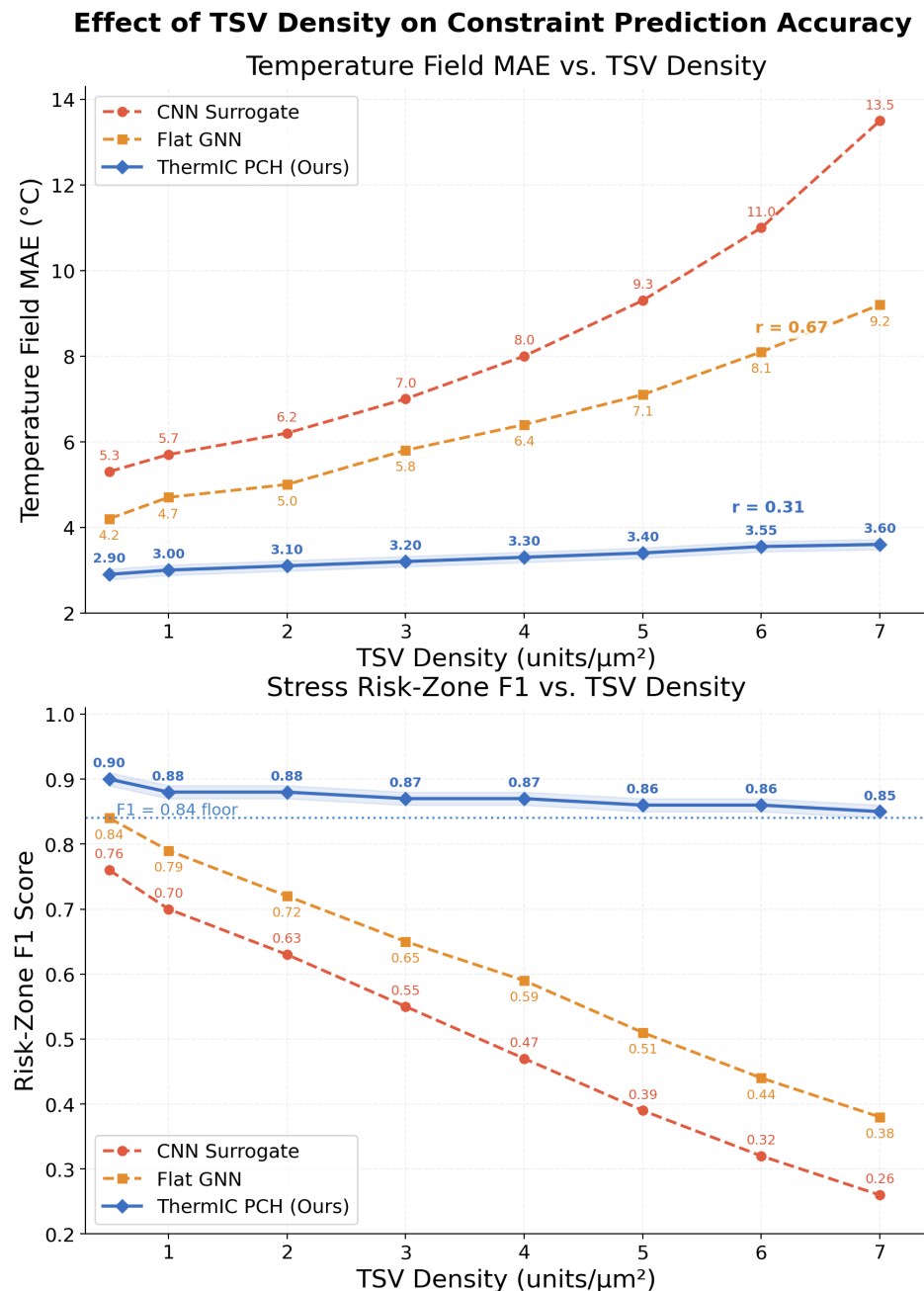


Figure 12. Effect of TSV density on constraint prediction accuracy. **Upper:** temperature-field MAE versus TSV density for CNN Surrogate, Flat GNN, and ThermIC PCH. **Lower:** stress risk zone F_1 score for the same methods.

Table 22. TSV-array-configuration error analysis for the anisotropic stress filtering model. Metrics are computed in TSV-neighborhood windows on held-out layouts. Lower stress MAE, peak-error, and peak-location error are better; higher stress-risk F_1 is better.

TSV Array Configuration	Approx. TSV Density/Pitch Pattern	Stress MAE (MPa)	Peak-Stress Rel. Error (%)	Peak Loc. Error (μm)	TSV-Window Stress F_1
Sparse random TSVs	<0.5%, nonuniform spacing	11.8	6.4	5.2	0.90
Regular grid TSVs	0.5–1.0%, nearly uniform pitch	12.6	7.1	5.9	0.89
Peripheral/ring TSVs	1.0–1.5%, boundary concentrated	14.9	8.5	7.3	0.87
Mixed-pitch TSVs	1.5–2.0%, local pitch transition	16.7	9.8	8.8	0.85
Clustered TSV islands	2.0–3.0%, local clusters	18.4	11.2	9.6	0.83
High-density central array	>3.0%, overlapping TSV stress fields	21.3	13.7	11.4	0.80

6.5. Sensitivity Analysis of Reward Coefficients

Because the 85 °C threshold and other limits are not universal, the trained ThermIC policy is evaluated under alternative thermal acceptance limits while keeping the continuous PCH temperature predictor unchanged. Table 23 reports the resulting joint pass rates for $\tau_T \in \{80, 85, 90\}$ °C. As expected, a stricter 80 °C limit lowers all pass rates, whereas a more relaxed 90 °C limit increases them. However, the ranking among methods remains stable, indicating that ThermIC’s advantage is not an artifact of a single threshold value. This analysis indicates that transfer to a new technology should replace the threshold vector and then validate or fine-tune the model on calibration layouts from the target stack.

Table 23. Thermal-threshold sensitivity analysis on the held-out ThermIC-Bench test set.

Thermal Threshold τ_T	Heuristic Joint Pass (%)	RL-NoPhysics Joint Pass (%)	ThermIC Joint Pass (%)
80 °C	41.6	60.8	76.4
85 °C	48.8	67.0	82.1
90 °C	55.3	72.4	86.8

The threshold-normalized multi-objective reward function in Equation (13) involves four weight coefficients whose values influence the trade-off between competing objectives. For reproducibility, the coefficient-selection protocol is described in the TSRL reward-function paragraph and summarized in Table 6. To characterize the sensitivity of ThermIC’s performance to these coefficients, we conduct a grid search over $\alpha \in \{0.5, 1.0, 2.0\}$, $\beta \in \{0.5, 1.0, 2.0\}$, $\gamma \in \{0.5, 1.0, 2.0\}$, and $\delta \in \{5.0, 10.0, 20.0\}$ on a held-out validation set of 600 layouts (not included in either the training or test sets).

Figure 13 reports the local coefficient-sweep pass rates and the HPWL–temperature-satisfaction trade-off directly. Within a $\pm 50\%$ range around the coefficient values selected by Pareto analysis, ThermIC’s joint pass rate varies modestly: across all 81 coefficient combinations evaluated, the joint pass rate ranges from 79.3% to 82.1%, a span of less than 3%. The PCH’s dense, differentiable constraint predictions provide a consistent gradient signal across this coefficient range; the coefficients mainly adjust the relative emphasis on constraint satisfaction and wirelength optimization. The largest sensitivity is observed in the δ coefficient for the DRC penalty: overly small values of δ allow the agent to occasionally accept DRC violations when the thermal benefit is large, reducing the DRC individual pass rate but having a limited effect on the joint pass rate.

To examine whether stronger dependence appears outside this local range, a wider coefficient stress test is used. In this second sweep, α , β , and γ are varied over a 16-fold range from 0.25 to 4.0, while δ is varied from 2.5 to 40.0. Table 24 reports representative boundary cases from this wider sweep. The extended results show that ThermIC is robust

near the Pareto knee point but not globally insensitive: extreme under-weighting of DRC or thermal risk substantially degrades joint pass rate, whereas extreme over-weighting of thermal or stress terms increases HPWL and reduces the balanced objective. Thus, the interpretation is local robustness around the selected Pareto knee point, with clear degradation under extreme coefficient choices.

Table 24. Extended reward-coefficient stress test over a wider parameter range. Boundary cases show that ThermIC is locally robust near the Pareto knee point but degrades under extreme coefficient choices.

Coefficient Setting	$(\alpha, \beta, \gamma, \delta)$	Joint Pass (%)	Norm. HPWL	Temp. Sat. (%)	DRC Pass (%)
Very low thermal weight	(1.0, 0.25, 1.0, 10.0)	74.8	0.79	81.2	82.1
Very high thermal weight	(1.0, 4.0, 1.0, 10.0)	76.5	0.94	93.5	80.4
Very low stress weight	(1.0, 1.0, 0.25, 10.0)	76.8	0.82	89.5	82.3
Very high stress weight	(1.0, 1.0, 4.0, 10.0)	78.0	0.89	88.8	80.7
Very low DRC penalty	(1.0, 1.0, 1.0, 2.5)	70.6	0.78	90.2	69.1
Selected Pareto knee	(1.0, 1.0, 1.0, 10.0)	82.1	0.83	89.7	82.6
Very high DRC penalty	(1.0, 1.0, 1.0, 40.0)	77.4	0.90	88.1	86.7

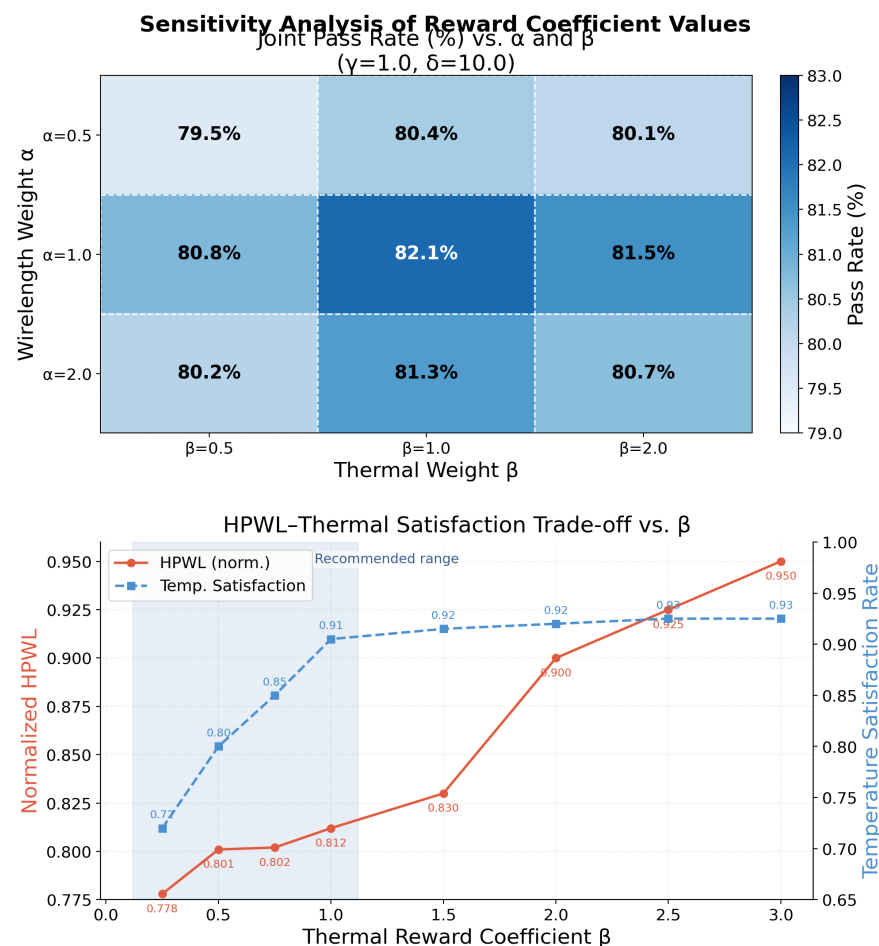


Figure 13. Sensitivity analysis of reward coefficient values. **Upper:** joint pass rate (%) heatmap over thermal weight β and wirelength weight α ($\gamma = 1.0, \delta = 10.0$); pass rates range only from 79.3% to 82.1% across the local 81-combination sweep, indicating local robustness around the Pareto-selected coefficient setting. **Lower:** trade-off between normalized HPWL and temperature satisfaction rate as β varies; the shaded region marks the recommended coefficient range that simultaneously maintains $\text{HPWL} \leq 0.83$ and temperature satisfaction $\geq 89\%$.

Figure 13 shows the pass-rate heatmap over α and β and the HPWL–temperature-satisfaction trade-off as β varies.

Taken together, the narrow and wide sweeps indicate that ThermIC is locally stable near the selected Pareto knee point, but it is not coefficient-invariant under extreme weighting. Therefore, transfer to a new design family should repeat the validation-set Pareto selection rather than reusing the default coefficients blindly. Broader generalization to new design families should still be assessed with additional validation designs.

6.6. Overall Performance Summary

Figure 14 collects the main metrics in one radar plot: physical verification pass rate, temperature error, stress prediction F_1 , iteration reduction, and normalized HPWL [14–16]. The largest numerical gaps are in joint pass rate, iteration count, and stress-risk prediction. Taken together, the results support the practical value of feeding differentiable physical-risk estimates back into the placement loop, at least within this simulated benchmark.

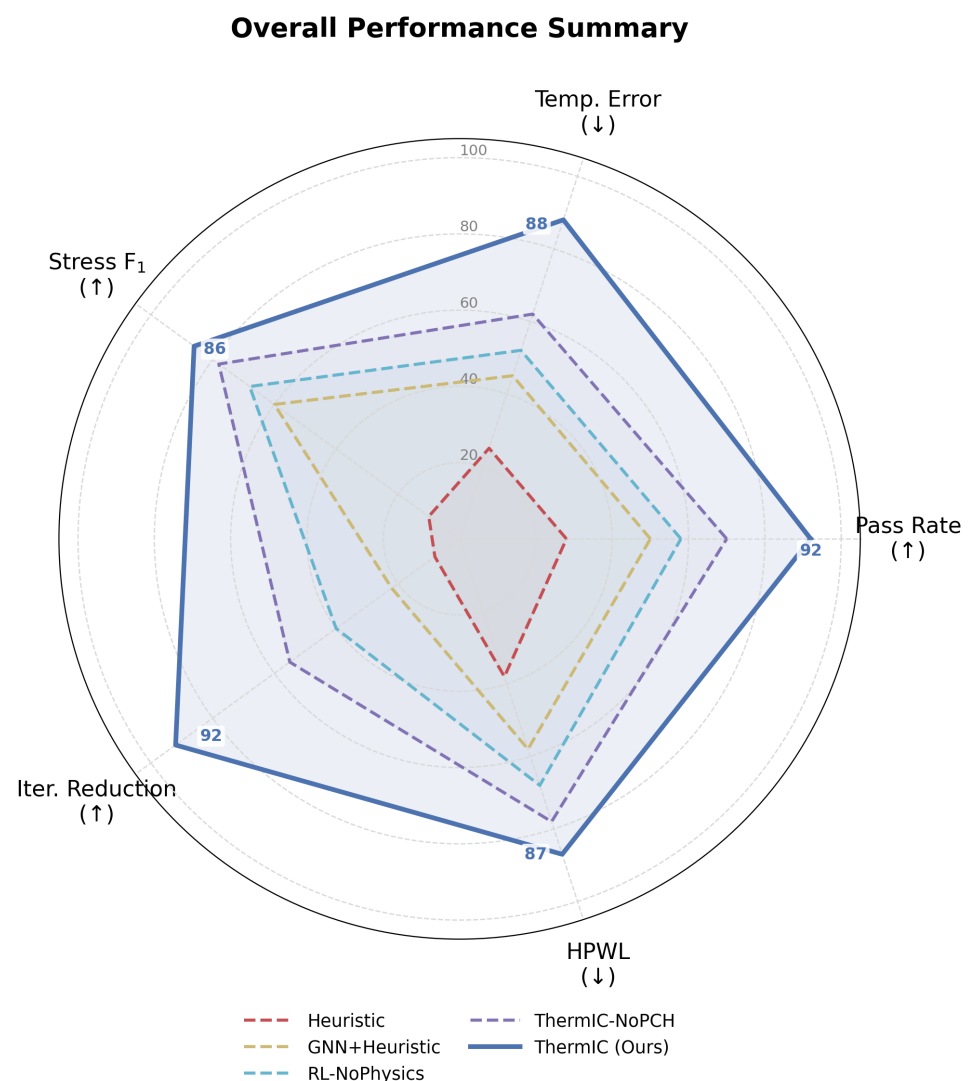


Figure 14. Overall performance summary radar chart across five dimensions: physical verification pass rate, temperature error (lower is better), stress prediction F_1 , design iteration reduction factor, and normalized wirelength.

The iteration-reduction metric is also relevant from a design-flow perspective. Post-placement verification in 3D-IC design may require DRC checking, thermal simulation, thermally derated timing analysis, and power-grid sign-off. Reducing the number of

closure iterations can therefore lower verification overhead and shorten the feedback loop between placement optimization and sign-off evaluation. The present results provide benchmark-level evidence, while validation in a full industrial flow remains necessary.

Figure 15 summarizes the relative advantages of ThermIC and the baselines across multiple evaluation dimensions at the end of the performance discussion.

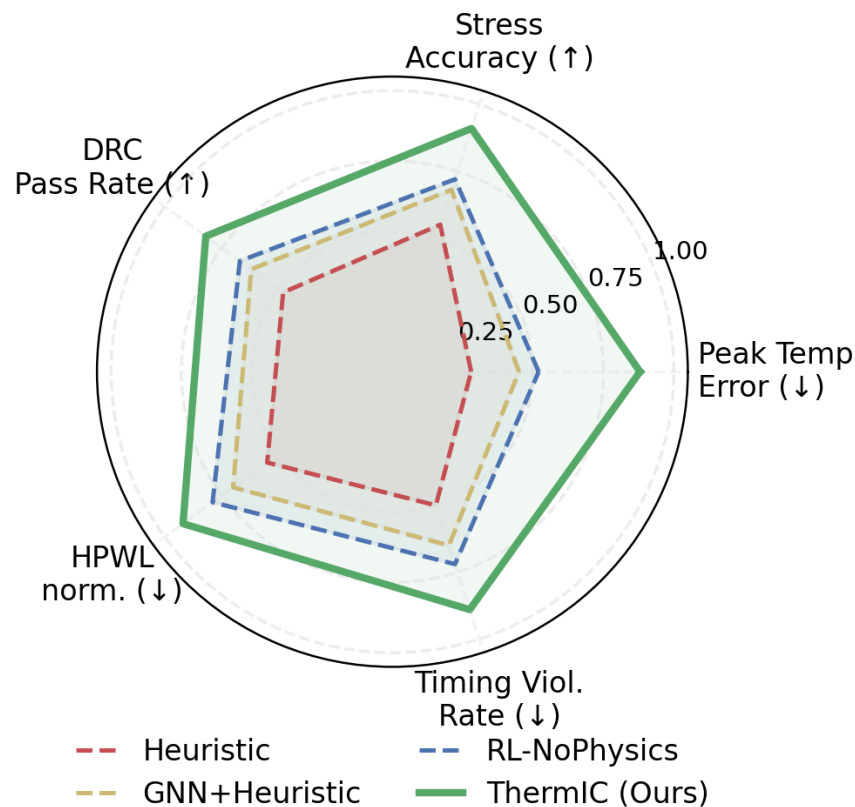


Figure 15. Radar chart comparing ThermIC and baseline methods across peak-temperature error, thermo-mechanical stress prediction accuracy, DRC pass rate, normalized wirelength, and timing violation rate.

These results support a controlled benchmark-level interpretation. ThermIC demonstrates that physics-aware risk prediction can improve placement decisions on a controlled simulated benchmark, and its data representation is compatible with public EDA benchmark artifacts. However, the present evidence is not equivalent to validation on commercial processor, memory, or accelerator layouts. A full real-world validation path would require: (i) importing LEF/DEF, Liberty, parasitic, and package files from a foundry-qualified design kit; (ii) evaluating ThermIC on independent public and/or NDA industrial circuits; (iii) comparing its predicted temperature and stress fields against silicon thermal sensors or calibrated package-level simulations; and (iv) completing post-route sign-off and, ideally, tape-out feedback. These requirements place ThermIC as a promising pre-signoff optimization strategy rather than a completed industrial-certification method.

7. Conclusions

This study introduced ThermIC, a 3D-IC placement framework in which thermal and stress estimates are used during placement rather than only after a layout is completed [1,2]. The reported 85 °C thermal boundary and related verification thresholds are benchmark configuration values rather than universal technology standards, and ThermIC is designed to accept process- and product-specific threshold vectors during deployment. This

evidence is obtained from a simulated multi-physics benchmark and public-benchmark-compatible data abstraction, not from proprietary industrial circuits, silicon measurements, or a tape-out case study. The MPGE represents thermal and stress coupling with heat-kernel message passing [35] and anisotropic stress filters [4,7]. These two operators are linked to the discretized heat-conduction equation and to linear thermo-elastic equilibrium, with mechanism-level diagnostics comparing heat residuals, thermal-gradient alignment, and TSV-neighborhood stress-risk prediction against FEM-derived references. The PCH estimates local physical-risk quantities with uncertainty calibration [46,49]. The TSRL policy then uses these estimates during sequential placement, together with physics-informed penalties [43] and replay of constraint-violating cases [54].

On ThermIC-Bench, the framework reaches an 82.1% 3D physical verification pass rate, a 3.1 °C peak-temperature error, a hotspot IoU of 0.89 [5,6], and a $3.7\times$ reduction in design iterations relative to the heuristic baseline [21]. The ablation results identify the PCH as the largest single contributor to the pass rate gain. The expanded comparison with Google RL, MaskPlace, ChiPFormer, DREAMPlace 4.0, AutoDMP, RoutePlacer, and TransPlace further indicates that the gain does not arise merely from replacing a heuristic with a transformer-based, RL-based, or large-scale graph-based learned placer; it is mainly associated with coupling placement actions to dense multi-physics risk predictions. The baseline protocol uses matched adaptation, tuning, legalization, and verification conditions rather than unequal post-processing or test-set tuning. The two-stage training scheme is also useful: supervised pre-training gives the policy a usable physical representation before the harder RL fine-tuning stage. Importantly, these numerical conclusions are restricted to the held-out ThermIC-Bench test layouts; they do not by themselves establish external validity on independent commercial or public 3D-IC datasets.

The main value of ThermIC is not that it replaces sign-off simulation. This is not the case, its function is to let the placement engine access the approximate multi-physical risk earlier. That can reduce the number of late repair cycles in the benchmark setting, but production use will still require validation against proprietary design rules, industrial sign-off tools, and fabricated designs. The scalability analysis shows that the high reported CPU-core-hour cost belongs to offline benchmark generation, whereas design-time deployment is dominated by sparse GNN inference, policy rollout, and legality checking. Thus, the practical value of ThermIC is most plausible when the trained model is reused across related technology settings or fine-tuned with a limited number of additional simulations, instead of regenerating a full FEM corpus for every new chip.

A further limitation concerns the present graph taxonomy. The default MPGE graph explicitly models signal, power/current, thermal-conduction, and stress-propagation relations because these are observable from placement-time design artifacts and directly affect the reported objectives. It does not yet provide sign-off-level electromigration or material-ageing prediction. Electromigration would require current-density, interconnect-width, duty-cycle, and temperature-time labels similar to Black-type reliability modeling [60]; material degradation would require thermal-cycle, stress-cycle, and ageing data from package or reliability qualification. These mechanisms are therefore treated as important future extensions rather than as validated capabilities of the current ThermIC implementation.

Accordingly, the present results constitute thermal-stress-aware placement evidence rather than EM-qualified reliability evidence. In designs where electromigration is the dominant failure mode, ThermIC may under-rank placements that concentrate current in narrow conductors, dense TSV escape regions, or local power-delivery bottlenecks even when their peak temperature and thermo-mechanical stress are acceptable. A production deployment should therefore keep post-route EM sign-off in the loop and, in future work, train an additional EM head using current-density and time-to-failure labels.

Several limitations should be kept in view. First, the surrogate is fast but imperfect: the 3.2 °C thermal error leaves a residual uncertainty that is visible in the remaining failed layouts. Future versions should either use higher-fidelity surrogate models, including PINN-type residual terms [43], or update the surrogate online when new FEM evaluations are produced. Second, the current policy places cells sequentially. This is tractable, but it may be weak for tightly coupled cell groups that should be moved together. Third, ThermIC-Bench covers a limited set of process nodes, package types, and material systems. Sub-3nm technologies, hybrid bonding with <1 µm bump pitch, and glass-interposer designs remain untested in the present work. Therefore, the reported results demonstrate effectiveness on the ThermIC-Bench simulation setting but do not establish generalization to a wide range of commercial 3D-IC technologies.

ThermIC may be conceptually applicable to other stacked or heterogeneous integration scenarios only after external validation on independent datasets, such as public 3D-IC benchmarks, NDA industrial layouts, calibrated package-level simulations, silicon sensor measurements, or tape-out feedback. Until such validation is completed, claims about organic-substrate chiplet integration, 2.5D silicon interposers, fan-out wafer-level packaging, or EMIB-style architectures should be understood as future research directions rather than conclusions supported by the current experiments. Our next steps are to connect the framework more tightly with multi-physics simulation engines [43,61], extend the message-passing formulation to thermal-aware routing and power-grid synthesis, and test transfer from simulation-trained models to measured data. The recent AI-for-Science and physics-informed design literature [22–24] positions ThermIC as a domain-specialized placement framework whose unique contribution is the closed-loop use of physically structured risk prediction during 3D-IC design, while the strength of the claim remains limited to the available ThermIC-Bench evidence.

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

Funding: This study was supported by the Initiated Funding Project for Disciplinary Backbone Talents (No. 221110136) from Dongguan University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request. Code and trained models will be released upon acceptance.

Acknowledgments: The authors would like to acknowledge Dongguan University of Technology for its support. During the preparation of this work, the authors used Baidu Translate for English translation and language editing. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Patti, R.S. Three-dimensional integrated circuits and the future of system-on-chip designs. *Proc. IEEE* **2006**, *94*, 1214–1224. [\[CrossRef\]](#)
2. Lau, J.H. *Chiplet Design and Heterogeneous Integration Packaging*; Springer Nature Singapore: Singapore, 2023. [\[CrossRef\]](#)
3. Lu, T.; Serafy, C.; Yang, Z.; Samal, S.K.; Lim, S.K.; Srivastava, A. TSV-based 3-D ICs: Design methods and tools. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2017**, *36*, 1593–1619. [\[CrossRef\]](#)

4. Selvanayagam, C.S.; Lau, J.H.; Zhang, X.; Seah, S.K.W.; Vaidyanathan, K.; Chai, T.C. Nonlinear thermal stress/strain analyses of copper filled TSV (through silicon via) and their flip-chip microbumps. *IEEE Trans. Adv. Packag.* **2009**, *32*, 720–728. [\[CrossRef\]](#)
5. Huang, W.; Ghosh, S.; Velusamy, S.; Sankaranarayanan, K.; Skadron, K.; Stan, M.R. HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2006**, *14*, 501–513. [\[CrossRef\]](#)
6. Skadron, K.; Stan, M.R.; Huang, W.; Velusamy, S.; Sankaranarayanan, K.; Tarjan, D. Temperature-aware microarchitecture. In Proceedings of the 30th Annual International Symposium on Computer Architecture, San Diego, CA, USA, 9–11 June 2003; pp. 2–13. [\[CrossRef\]](#)
7. Cong, J.; Wei, J.; Zhang, Y. A thermal-driven floorplanning algorithm for 3D ICs. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 7–11 November 2004; pp. 306–313. [\[CrossRef\]](#)
8. Goplen, B.; Sapatnekar, S.S. Placement of thermal vias in 3-D ICs using various thermal objectives. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2006**, *25*, 692–709. [\[CrossRef\]](#)
9. Goplen, B.; Sapatnekar, S.S. Placement of 3D ICs with thermal and interlayer via considerations. In Proceedings of the 44th ACM/IEEE Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 626–631. [\[CrossRef\]](#)
10. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
11. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
12. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [\[CrossRef\]](#)
13. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Mirhoseini, A.; Goldie, A.; Yazgan, M.; Jiang, J.W.; Songhori, E.; Wang, S.; Lee, Y.-J.; Johnson, E.; Pathak, O.; Nova, A.; et al. A graph placement methodology for fast chip design. *Nature* **2021**, *594*, 207–212. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Lai, Y.; Mu, Y.; Luo, P. MaskPlace: Fast chip placement via reinforced visual representation learning. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; pp. 24019–24030.
16. Lin, Y.; Jiang, Z.; Gu, J.; Li, W.; Dhar, S.; Ren, H.; Khailany, B.; Pan, D.Z. DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2021**, *40*, 748–761. [\[CrossRef\]](#)
17. Yang, J.; Lin, G.; Wang, S.; Zhou, J. CATI: Cross-attention-based task interaction for multi-granular metro passenger flow forecasting. *Symmetry* **2026**, *18*, 809. [\[CrossRef\]](#)
18. Yu, W.; Du, W. Branch-priority exploration for mobile robots in restricted industrial corridors. *Symmetry* **2026**, *18*, 806. [\[CrossRef\]](#)
19. Pan, C.-H.; Qu, Y.; Yao, Y.; Wang, M.-J.-S. HybridGNN: A self-supervised graph neural network for efficient maximum matching in bipartite graphs. *Symmetry* **2024**, *16*, 1631. [\[CrossRef\]](#)
20. Zhang, F.; Chen, Z. A novel reinforcement learning-based particle swarm optimization algorithm for better symmetry between convergence speed and diversity. *Symmetry* **2024**, *16*, 1290. [\[CrossRef\]](#)
21. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [\[CrossRef\]](#) [\[PubMed\]](#)
22. Niyongabo Rubungo, A.; Arnold, C.; Rand, B.P.; Dieng, A.B. LLM-Prop: Predicting the properties of crystalline materials using large language models. *npj Comput. Mater.* **2025**, *11*, 186. [\[CrossRef\]](#)
23. Si, L.; Niu, R.; Dang, C.; Bao, X.; Zhuang, Y.; Zhu, W. Advances in artificial intelligence for artificial metamaterials. *APL Mater.* **2024**, *12*, 120602. [\[CrossRef\]](#)
24. Niu, R.; Si, L.; Zhu, W.; Bao, X.; Zhu, K.; Sun, H.; Yang, F. Hybrid neural network-driven inverse design of metasurface for ultrabroadband RCS reduction. *IEEE Trans. Antennas Propag.* **2026**, *74*, 837–847. [\[CrossRef\]](#)
25. Spindler, P.; Schlichtmann, U.; Johannes, F.M. Kraftwerk2-A fast force-directed quadratic placement approach using an accurate net model. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2008**, *27*, 1398–1411. [\[CrossRef\]](#)
26. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
27. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
29. Guo, Z.; Liu, M.; Gu, J.; Zhang, S.; Pan, D.Z.; Lin, Y. A timing engine inspired graph neural network model for pre-routing slack prediction. In Proceedings of the 59th ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 10–14 July 2022; pp. 1207–1212. [\[CrossRef\]](#)
30. Kirby, R.; Godil, S.; Roy, R.; Catanzaro, B. CongestionNet: Routing congestion prediction using deep graph neural networks. In Proceedings of the IFIP/IEEE International Conference on Very Large Scale Integration, Cuzco, Peru, 6–9 October 2019.
31. Lu, Y.-C.; Nath, S.; Pentapati, S.; Lim, S.K. ECO-GNN: Signoff power prediction using graph neural networks with subgraph approximation. *ACM Trans. Des. Autom. Electron. Syst.* **2023**, *28*, 55. [\[CrossRef\]](#)

32. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; van den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In *The Semantic Web*; Springer: Cham, Switzerland, 2018; Volume 10843, pp. 593–607. [\[CrossRef\]](#)
33. Hu, Z.; Dong, Y.; Wang, K.; Sun, Y. Heterogeneous graph transformer. In Proceedings of The Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 2704–2710. [\[CrossRef\]](#)
34. Huang, G.; Hu, J.; He, Y.; Liu, J.; Ma, M.; Shen, Z.; Wu, J.; Xu, Y.; Zhang, H.; Zhong, K.; et al. Machine learning for electronic design automation: A survey. *ACM Trans. Des. Autom. Electron. Syst.* **2021**, *26*, 1–46. [\[CrossRef\]](#)
35. Kondor, R.I.; Lafferty, J. Diffusion kernels on graphs and other discrete input spaces. In Proceedings of the Nineteenth International Conference on Machine Learning, Sydney, Australia, 8–12 July 2002; pp. 315–322.
36. Kulkarni, T.D.; Narasimhan, K.R.; Saeedi, A.; Tenenbaum, J.B. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
37. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [\[CrossRef\]](#)
38. Lai, Y.; Liu, J.; Tang, Z.; Wang, B.; Hao, J.; Luo, P. ChiPFormer: Transferable chip placement via offline decision transformer. In Proceedings of the 40th International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 18346–18364.
39. Liao, P.; Guo, D.; Guo, Z.; Liu, S.; Lin, Y.; Yu, B. DREAMPlace 4.0: Timing-driven placement with momentum-based net weighting and Lagrangian-based refinement. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 3374–3387. [\[CrossRef\]](#)
40. Agnesina, A.; Rajvanshi, A.; Verdoolaege, S.; Fan, T.; Rajvanshi, S.; Ren, H.; Khailany, B. AutoDMP: Automated DREAMPlace-based macro placement. In Proceedings of the 2023 International Symposium on Physical Design, Virtual Event, 26–29 March 2023; pp. 149–157. [\[CrossRef\]](#)
41. Hou, Y.; Ye, H.; Zhang, Y.; Xu, S.; Song, G. RoutePlacer: An end-to-end routability-aware placer with graph neural network. *arXiv* **2024**, arXiv:2406.02651.
42. Hou, Y.; Ye, H.; Zhang, Y.; Xu, S.; Song, G. TransPlace: Transferable circuit global placement via graph neural network. *arXiv* **2025**, arXiv:2501.05667.
43. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [\[CrossRef\]](#)
44. Zhang, W.; Guo, Y.; Jin, Q. Radiomics and its feature selection: A review. *Symmetry* **2023**, *15*, 1834. [\[CrossRef\]](#)
45. Sanchez-Gonzalez, A.; Godwin, J.; Pfaff, T.; Ying, R.; Leskovec, J.; Battaglia, P.W. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning, Virtual Event, 13–18 July 2020*; PMLR: Vienna, Austria, 2020; Volume 119, pp. 8459–8468.
46. Kendall, A.; Gal, Y.; Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7482–7491. [\[CrossRef\]](#)
47. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
48. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [\[CrossRef\]](#) [\[PubMed\]](#)
49. Caruana, R. Multitask learning. *Mach. Learn.* **1997**, *28*, 41–75. [\[CrossRef\]](#)
50. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017*; PMLR: Sydney, Australia, 2017; Volume 70, pp. 1321–1330.
51. Kuleshov, V.; Fenner, N.; Ermon, S. Accurate uncertainties for deep learning using calibrated regression. In *Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018*; PMLR: Stockholm, Sweden, 2018; Volume 80, pp. 2796–2804.
52. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–1734. [\[CrossRef\]](#)
53. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
54. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
55. Chai, Z.; Zhao, Y.; Liu, W.; Lin, Y.; Wang, R.; Huang, R. CircuitNet: An open-source dataset for machine learning applications in electronic design automation. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 3894–3907. [\[CrossRef\]](#)

56. Jiang, X.; Zhao, Y.; Lin, Y.; Wang, R.; Huang, R. CircuitNet 2.0: An advanced dataset for promoting machine learning innovations in realistic chip design environment. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
57. Shi, Y.; Gao, C.; Ren, W.; Xu, S.; Xue, K.; Yuan, M.; Qian, C.; Zhou, Z.-H. Open3DBench: Open-source benchmark for 3D-IC backend implementation and PPA evaluation. *arXiv* **2025**, arXiv:2503.12946.
58. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
59. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 41–48. [[CrossRef](#)]
60. Black, J.R. Electromigration—A brief survey and some recent results. *IEEE Trans. Electron. Devices* **1969**, *16*, 338–347. [[CrossRef](#)]
61. Pfaff, T.; Fortunato, M.; Sanchez-Gonzalez, A.; Battaglia, P.W. Learning mesh-based simulation with graph networks. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.

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.