DiffCool: Label-Free Synthesis of Chip-Tailored Heat Sinks via Thermal-Aware Diffusion

Siyuan Liang[1], **Zixiao Wang**[2], **Chenghan Wang**[1], **Shanyi Li**[1], **Yushen Zhang**[3], **Leilei Jin**[1], **Zhen Zhuang**[*1], **Ulf Schlichtmann**[3], **Bei Yu**[1], **Tsung-Yi Ho**[1]
[1]The Chinese University of Hong Kong,
[2]The Hong Kong University of Science and Technology (Guangzhou),
[3]Technical University of Munich

Abstract

Recent advances in semiconductor technology have significantly improved integrated circuit performance; however, escalating transistor densities and operating frequencies induce unprecedented heat fluxes, necessitating spatially adaptive thermal management. Conventional heat sinks, constrained by fixed geometries, fail to adapt to chip-specific thermal profiles. Furthermore, the lack of optimal design datasets and stringent manufacturing constraints severely limit the applicability of generative AI in this domain. To overcome these limitations, we introduce **DiffCool**, a physics-guided framework that reformulates heat sink synthesis as a constrained discrete denoising process. By embedding manufacturing-aware state transitions into the diffusion dynamics and optimizing through a thermal-aware loss coupled with a compact differentiable simulator, our model directly synthesizes structurally sound, high-performance topologies from bare-chip heat maps without relying on labeled data. Experiments on an open-source processor demonstrate that DiffCool synthesizes production-ready designs in under one second, achieving reductions of 27.9% in peak temperature, 24.6% in average temperature, and 57.4% in temperature gradient compared to conventional baselines. This work establishes a scalable, label-free paradigm that seamlessly integrates generative modeling with physical constraints for automated thermal design.

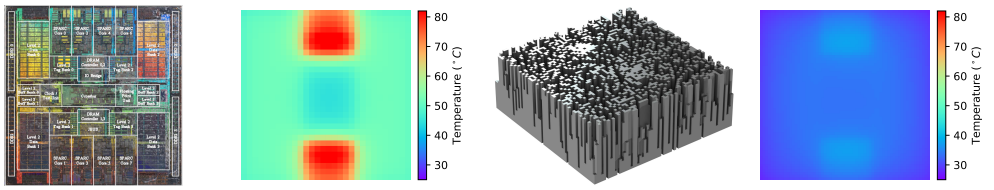
1 Introduction

The relentless advancement of semiconductor technology continues to deliver unprecedented transistor density and performance [1]. However, this progress is fundamentally constrained by a thermal management crisis, where skyrocketing heat fluxes directly threaten chip reliability and operational efficiency [2]. While liquid cooling offers superior thermal performance, its cost and complexity limit adoption in mass-produced electronics [3]. Consequently, passive heat sinks remain the preferred thermal solution, offering the best balance of cost, reliability, and performance [4].

Conventional heat sink design is dominated by flat-plate fin (FPFHS) [5] and pin-fin (PFHS) [6] architectures. Extensive research has pursued manual geometric optimizations through modified fin shapes [7], oblique arrays [8], non-uniform pin features [9], and various perforation patterns [10]. Despite these efforts, a fundamental limitation persists: these designs lack customization for the unique, spatially-variant thermal profiles of modern chips. The design process involves an inherent trade-off between metal fill (enhancing vertical conduction) and void space (maximizing horizontal convection) [9]. Manual navigation of this trade-off through iterative simulations is inefficient and often converges to suboptimal solutions [11, 12], creating a critical need for automated, customized design frameworks.

*Corresponding Author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Die shot of an UltraSPARC T1 [17].
(b) Bare-chip heat map under the average workload.
(c) Customized heat sink synthesized by DiffCool.
(d) Heat map of the chip with the customized heat sink.

Figure 1: Automated heat sink customization. Mirroring the thermal profile, the synthesized design transits from dense fins over hotspots for conduction to sparse pins in cooler areas for convection.

Generative AI presents a transformative opportunity for this domain, capable of exploring vast design spaces to discover non-intuitive, high-performance topologies [13, 14] while compressing design cycles from weeks to minutes [15, 16]. However, direct application to heat sink synthesis faces two pivotal challenges: (1) the absence of labeled datasets containing optimal heat sink designs makes supervised learning inapplicable, and (2) AI-generated structures frequently violate manufacturability constraints, producing fragile overhangs incompatible with mass production.

To address these challenges, we introduce **DiffCool**, a physics-guided, label-free discrete diffusion framework for automated heat sink synthesis. As shown in Figure 1, our system takes a chip thermal map and synthesizes a custom, high-performance design within one second. The core of our method is a constrained, layer-by-layer categorical denoising process. It begins with a solid metal base and progressively removes material to form an optimal structure. This bottom-up formulation inherently guarantees manufacturability by preventing unsupported overhangs and enforcing minimum feature sizes compatible with standard molding processes [18]. For label-free optimization, we introduce a differentiable thermal-aware loss function that directly evaluates heat dissipation performance, enabling the model to learn optimal metal-void trade-offs through gradient backpropagation without requiring optimal reference designs.

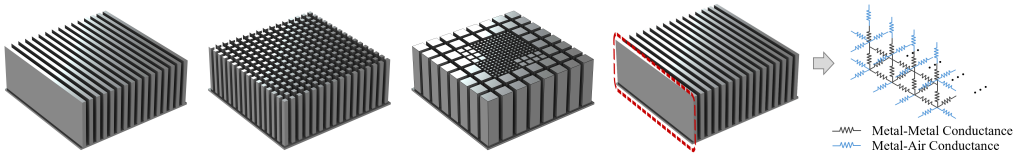
The main contributions of this work are summarized as follows:

- **A label-free discrete diffusion framework:** We reformulate heat sink synthesis as a constrained categorical denoising process, eliminating dependency on optimal datasets through direct physics-based gradient propagation.
- **Manufacturing-aware transition design:** We construct an absorbing-state transition matrix and a bottom-up denoising protocol that inherently eliminates unsupported overhangs, with minimum feature size enforced during output, ensuring direct compatibility with high-volume production.
- **Thermal-aware composite loss:** We introduce a differentiable objective coupling vertical conduction, horizontal convection, and peak temperature reduction, enabling the model to balance competing heat transfer mechanisms without labeled supervision.
- **Rapid synthesis and empirical validation:** We demonstrate an end-to-end pipeline that translates thermal maps to optimized heat sinks in under 1 second, achieving 27.9% lower peak temperature, 24.6% lower average temperature, and 57.4% reduced temperature gradient compared to conventional baselines across diverse workload patterns.

2 Preliminaries and Related Works

2.1 Conventional Heat Sink Design

Conventional passive heat sinks are categorized by their heat dissipation topology (Figure 2). Flat-plate fin heat sinks (FPFHS) use parallel channels to enhance horizontal convection but require manual geometric tuning [5]. Pin-fin heat sinks (PFHS) employ dense pin arrays to reduce vertical thermal resistance via short conductive paths [6]. Although non-uniform variants (NU-PFHS) apply gradient-based topology optimization [9], these architectures fundamentally lack spatial adaptability to localized chip thermal profiles.



(a) FPFHS [5]. (b) PFHS [6]. (c) NU-PFHS [9]. (d) Equivalent thermal conductance of a fin.

Figure 2: Conventional heat sink designs. FPFHS, PFHS, and NU-PFHS lack sufficient spatial adaptability to localized thermal hotspots, resulting in suboptimal conduction-convection trade-offs.

This limitation arises from a material distribution trade-off between conduction and convection. High metal density improves vertical conduction but restricts airflow, lowering convective coefficients and causing thermal stagnation. Conversely, low metal density expands convective surface area but increases vertical thermal resistance, impeding heat extraction. Resolving this multi-objective optimization problem analytically is intractable. Manual iterative simulation of this high-dimensional design space is computationally prohibitive and typically yields suboptimal geometries, necessitating automated, globally optimized synthesis frameworks.

2.2 Thermal Simulation for Optimization

Efficient thermal simulation is essential for rapid design exploration. Numerical approaches balance high-fidelity methods like Finite Element Analysis (FEA) against compact modeling. Although FEA accurately resolves full thermal fields and serves as a validation benchmark [19, 20], its computational cost prohibits the iterative evaluations required for generative synthesis.

Compact modeling offers a tractable alternative by abstracting systems into low-order conductance networks [21]. The transient thermal behavior of a discretized heat sink obeys:

$$\mathbf{C}\frac{\partial \mathbf{T}}{\partial t} + \mathbf{K}\mathbf{T} = \mathbf{P}, \tag{1}$$

where $\mathbf{T}$ is nodal temperature, $\mathbf{P}$ denotes localized heat sources, $\mathbf{K}$ is the sparse symmetric conductance matrix, and $\mathbf{C}$ is the capacitance matrix. Figure 2d illustrates the equivalent conductance of an FPFHS fin. Under steady-state conditions, this simplifies to $\mathbf{K}\mathbf{T} = \mathbf{P}$, efficiently solvable via conjugate gradient algorithms [22].

Recent advances combine AI with compact models to yield differentiable thermal simulators [23]. These surrogates achieve millisecond inference while preserving physical fidelity, supplying exact gradients for backpropagation. This capability is vital for our framework, enabling direct physics-driven optimization during generation without invoking high-fidelity solvers at each step.

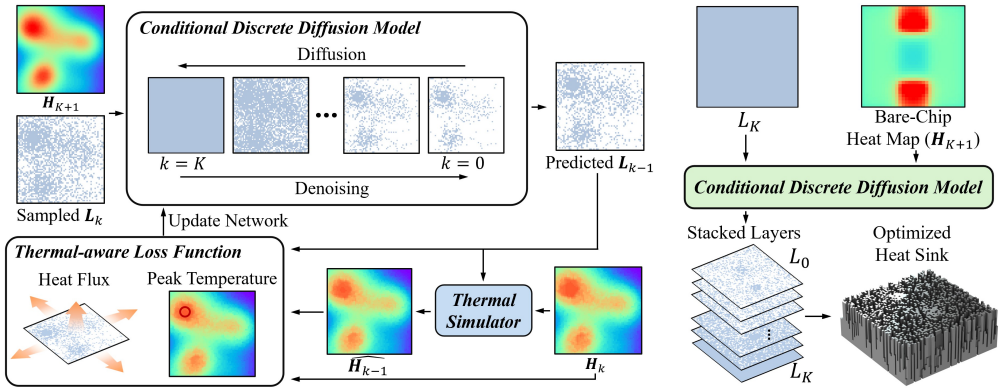
3 DiffCool

Conventional heat sink designs struggle with modern electronics' growing thermal challenges, limited by fixed geometries and manual optimization. These shortcomings motivate DiffCool, an AI framework leveraging generative design to automatically create high-performance, manufacturable cooling solutions. We detail how physical constraints and conditional diffusion models enable automated, performance-optimized heat sink synthesis.

3.1 Overview of DiffCool

Figure 3 presents the overall flow of DiffCool, built around a conditional discrete diffusion model.

We formulate heat sink synthesis as a denoising process within a diffusion framework. Synthesis proceeds layer-by-layer, starting from a solid metal substrate and evolving upward through progressive material removal. We impose a manufacturing-aware constraint: at each denoising step, material can only be removed relative to the supporting layer beneath. This bottom-up approach inherently eliminates unsupported overhangs. We also enforce a minimum feature size meeting standard molding precision requirements [18]. By guaranteeing structural feasibility and manufacturable feature sizes,



(a) The training process. (b) The inference process.

Figure 3: Overall flow of DiffCool. The framework integrates a manufacturing-aware discrete diffusion process with a differentiable thermal simulator, enabling label-free, sub-second synthesis of customized heat sinks from bare-chip thermal profiles.

our method ensures direct compatibility with high-volume processes like die casting or injection molding, avoiding expensive specialized fabrication.

The absence of analytical solutions precludes directly generating optimal heat sink datasets, creating a fundamental barrier for supervised learning [24, 25]. We instead develop an iterative framework where the diffusion model progressively optimizes its synthesis strategy, starting from randomly generated heat sinks providing an initial design distribution.

Specifically, as shown in Figure 3a, at each iteration, we sample a diffusion timestep k and task the neural network with predicting the next metal layer $\boldsymbol{L}_{k-1}$ from sampled layer $\boldsymbol{L}_k$. To guide optimization, we introduce a thermal-aware loss function evaluating predicted layers using key thermal metrics, including heat flux distribution and peak temperatures. This loss incorporates three inputs: (1) predicted layer $\boldsymbol{L}_{k-1}$, (2) pre-collected heat map $\boldsymbol{H}_k$ from underlying layer k, and (3) simulated heat map $\boldsymbol{H}_{k-1}$ above the predicted layer. Through backpropagation, the network learns to balance competing thermal transport mechanisms, maximizing vertical conduction via metal fills while optimizing horizontal convection through void patterns.

In practical deployment, a chip operates under multiple workload conditions. To accommodate this, the input thermal map H_{K+1} can be constructed as a time-weighted representative heat map $H_{\text{rep}} = \sum_i w_i H_i$, where H_i is the heat map under workload i and w_i is its operating time fraction. This unified representation allows DiffCool to generate a single passive heat sink optimized for the expected workload distribution without modifying the network architecture.

Upon training completion, the model synthesizes complete heat sink designs layer-by-layer. As shown in Figure 3b, given the initial bare-chip heat map, the model sequentially generates and stacks optimized layers to produce a final design with optimal heat dissipation.

3.2 Conditional Discrete Diffusion Model

Unlike conventional diffusion models for computer vision, we synthesize discrete physical heat sink structures, where each entry belongs to a discrete state. We make key modifications to enhance the diffusion model for direct discrete layer synthesis.

To align with the reformulated synthesis problem, we modify the diffusion process. At the k-th step of K, each entry $x_k \in \{0\,(\text{void}), 1\,(\text{metal})\}$ in layer $\boldsymbol{L}_k$ undergoes a transition:

$$\boldsymbol{q}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) := \text{Cat}\left(\boldsymbol{x}_k; \boldsymbol{p} = \boldsymbol{x}_{k-1}\boldsymbol{Q}_k\right), \tag{2}$$

where $\boldsymbol{x}_k$ is the one-hot encoding of x_k, $\text{Cat}(\boldsymbol{x}; \boldsymbol{p})$ denotes a categorical distribution, and $\boldsymbol{Q}_k$ is a transition matrix with entries $[\boldsymbol{Q}_k]_{ij} = q(x_k = j|x_{k-1} = i)$, capturing state transition probabilities at step k. The diffusion process applies $\boldsymbol{Q}_k$ independently to each entry in $\boldsymbol{L}_{k-1}$, factorizing the joint distribution $\boldsymbol{q}$. During denoising, the neural network predicts the reverse transition $\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k, \boldsymbol{H}_{K+1})$

to synthesize the heat sink from bare-chip heat map $\boldsymbol{H}_{K+1}$ and initialized bottom layer $\boldsymbol{L}_K = \mathbf{1}$ (fully metal).

The choice of $\boldsymbol{Q}_k$ ensures that as $k \to K$, the stationary distribution of any entry $\boldsymbol{x}_k$ converges to metal (state 1):

$$\boldsymbol{q}(\boldsymbol{x}_k|\boldsymbol{x}_0) \to [0, 1], \text{ where } k \to K. \tag{3}$$

We design a transition matrix $\boldsymbol{Q}_k$ with absorbing state 1:

$$\boldsymbol{Q}_k = \begin{bmatrix} 1-\beta_k & \beta_k \\ 0 & 1 \end{bmatrix}, \tag{4}$$

where $\beta_k \in (0, 1)$ controls flipping probability. To ensure all entries reach state 1 at final step K, we use an increasing noise schedule:

$$\beta_k = \frac{(k-1)(1-\beta_1)}{K-1} + \beta_1,\ k = 1, ..., K, \tag{5}$$

where β_1 is a hyperparameter.

To enable gradient backpropagation through our discrete diffusion model, we implement Gumbel-Softmax relaxation [26] at the final output layer. This provides a differentiable approximation of categorical sampling, transforming logits π into softened one-hot vectors:

$$y_i = \text{softmax}(\frac{\log \pi_i + g_i}{\tau}), \quad g_i \sim \text{Gumbel}(0, 1), \tag{6}$$

where $i \in \{0, 1\}$ indexes void/metal categories, y_i represents the i-th output element, π_i denotes predicted class probabilities, and τ controls softmax temperature, balancing exploration and exploitation.

We emphasize that DiffCool guarantees topology-level manufacturability, specifically preventing unsupported overhangs and enforcing minimum feature sizes. Process-specific details, such as draft angles, fillets, and parting lines, depend on the selected foundry and tooling configuration. These can be incorporated during subsequent CAD conversion and tooling preparation without altering the principal heat-dissipation topology.

3.3 Training and Inference of Diffusion Model

Training. Training requires no optimal heat sink dataset, starting solely from random designs and their corresponding layer-wise heat maps. Synthesized designs gradually evolve toward optimality.

Given top layer $\boldsymbol{L}_0$, we randomly sample timestep k from a uniform distribution on $[1, K]$ to obtain noisy sample $\boldsymbol{L}_k$. Analytically, $\boldsymbol{x}_k$ obeys:

$$\boldsymbol{q}(\boldsymbol{x}_k|\boldsymbol{x}_0) = \text{Cat}(\boldsymbol{x}_k; \boldsymbol{p} = \boldsymbol{x}_0\overline{\boldsymbol{Q}}_k), \tag{7}$$

where $\overline{\boldsymbol{Q}}_k = \boldsymbol{Q}_1\boldsymbol{Q}_2 \cdots \boldsymbol{Q}_k$. This enables direct $\boldsymbol{L}_k$ sampling without k sequential diffusion steps.

The model processes noisy input $\boldsymbol{L}_k$, bare-chip heat map $\boldsymbol{H}_{K+1}$, and timestep k to predict denoising distribution $\boldsymbol{L}_{k-1}$. Output logits parameterize $\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_k, \boldsymbol{H}_{K+1})$, yielding reverse transition probability:

$$\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k, \boldsymbol{H}_{K+1}) = \sum_{\tilde{\boldsymbol{x}}_\mathbf{0}} \boldsymbol{q}(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k, \tilde{\boldsymbol{x}}_\mathbf{0})\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_k, \boldsymbol{H}_{K+1}), \tag{8}$$

where $\tilde{\boldsymbol{x}}_\mathbf{0}$ visits every possible state of $\boldsymbol{x}_0$, and $\boldsymbol{q}(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k, \boldsymbol{x}_0)$ has closed form via Equation (2) and Bayes' theorem [27]:

$$\boldsymbol{q}(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k, \boldsymbol{x}_0) = \text{Cat}(\boldsymbol{x}_{k-1}; \boldsymbol{p} = \frac{\boldsymbol{x}_k\boldsymbol{Q}_k^\top \odot \boldsymbol{x}_0\overline{\boldsymbol{Q}}_{k-1}}{\boldsymbol{x}_0\overline{\boldsymbol{Q}}_k\boldsymbol{x}_k^\top}), \tag{9}$$

where $\odot$ denotes element-wise multiplication.

Unlike standard diffusion models reconstructing $\boldsymbol{L}_0$, our framework introduces a novel thermal-aware optimization objective. This loss incorporates heat flux distribution and peak temperature minimization to guide generation toward thermally optimal designs. Complete formulation appears in Section 3.4.

Inference. Once trained, given initial heat map $\boldsymbol{H}_{K+1}$ of the bare chip, we synthesize each layer during denoising:

$$\boldsymbol{p}_{\boldsymbol{\theta}}(\hat{\boldsymbol{L}}_0|\boldsymbol{L}_k, \boldsymbol{H}_{K+1}) = \boldsymbol{p}_{\boldsymbol{\theta}}(\hat{\boldsymbol{L}}_0|\boldsymbol{L}_1, \boldsymbol{H}_{K+1}) \prod_{k=2}^{K} \boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{L}_{k-1}|\boldsymbol{L}_k, \boldsymbol{H}_{K+1}), \tag{10}$$

where $\boldsymbol{L}_k$ is the estimated metal layer at step k, and $\hat{\boldsymbol{L}}_0$ is the newly sampled top layer.

All synthesized layers $\boldsymbol{L}_k$ are collected and stacked sequentially ($\boldsymbol{L}_0$ top, $\boldsymbol{L}_K$ bottom), then placed on a pure metal substrate to form the final heat sink.

3.4 Thermal-aware Loss Function

To ensure superior thermal performance, we design a thermal-aware loss guiding denoising toward physically favorable structures. To balance overall performance and hotspot management, this loss optimizes two critical metrics: maximizing heat flux for enhanced dissipation efficiency, and maximizing peak temperature reduction.

Computing these metrics requires thermal simulation at each optimization step. For efficiency, we employ a differentiable thermal simulator deriving layered structure behavior:

$$\widehat{\boldsymbol{H}_{k-1}} = \text{Sim}(\boldsymbol{L}_{k-1}, \boldsymbol{H}_k), \tag{11}$$

where $\text{Sim}(\cdot)$ predicts heat map $\widehat{\boldsymbol{H}_{k-1}}$ atop layer $\boldsymbol{L}_{k-1}$ given layer configuration and heat map $\boldsymbol{H}_k$ beneath. Here, $\boldsymbol{H}_k$ is pre-collected during random dataset generation. This compact thermal model-based simulator balances speed and accuracy, delivering real-time feedback and precise gradients for effective diffusion model training.

Total heat flux $\boldsymbol{F}$ through the layer comprises vertical conductive transfer $\boldsymbol{F}_v$ and horizontal convective dissipation $\boldsymbol{F}_h$ from exposed surfaces to ambient:

$$\begin{aligned} \boldsymbol{F} &= \boldsymbol{F}_v + \boldsymbol{F}_h, \\ \boldsymbol{F}_v &= G_{\text{amb}} \sum_{(x,y)} \left([\boldsymbol{H}_{k-1}]_{xy} - T_{\text{amb}} \right), \\ \boldsymbol{F}_h &= G_{\text{amb}} \sum_{(x,y)\in\Gamma} \left([\boldsymbol{H}_{k-0.5}]_{xy} - T_{\text{amb}} \right), \end{aligned} \tag{12}$$

where G_{amb} is ambient thermal conductance, (x, y) are cell coordinates, T_{amb} is ambient temperature, $\boldsymbol{H}_{k-0.5}$ is the temperature distribution inside layer k, and Γ is the boundary. We set $G_{\text{amb}} = 10^{-6}$ W/K and $T_{\text{amb}} = 25°\text{C}$ [28].

As each layer is relatively thin, the vertical temperature gradient is approximately linear. Thus, $\boldsymbol{H}_{k-0.5} \approx \frac{1}{2}(\boldsymbol{H}_k + \boldsymbol{H}_{k-1})$.

While heat flux models global dissipation behavior, we also target peak temperature mitigation for uniform heat distribution. Peak temperature reduction $\triangle T_{\text{max}}$ is:

$$\triangle T_{\text{max}} = \max(\boldsymbol{H}_k) - \max(\boldsymbol{H}_{k-1}), \tag{13}$$

where $\max(\cdot)$ extracts peak temperature from heat maps. To balance both thermal metrics, we normalize each quantity and formulate the composite loss:

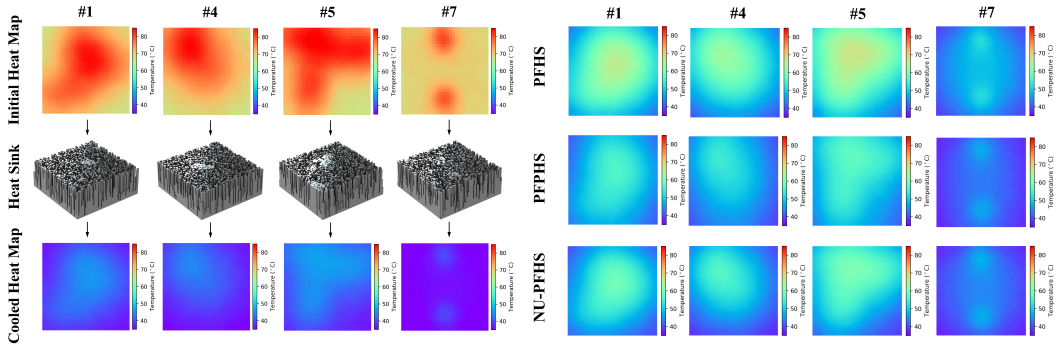
$$\text{loss} = -\frac{1}{N} \frac{\boldsymbol{F}}{G_{\text{amb}} \cdot \max(\boldsymbol{H}_k)} - \frac{\triangle T_{\text{max}}}{\max(\boldsymbol{H}_k)}, \tag{14}$$

where N is the number of cells in the layer. The normalization by $\max(\boldsymbol{H}_k)$ aligns the magnitude of global heat flux and local peak reduction to the same scale, preventing gradient dominance by either term during early training stages when temperature fields are highly non-uniform.

4 Experiments

4.1 Experiment Setups

Diffusion Model Configurations. We employ a U-Net architecture [29] with five downsampling residual blocks, symmetric transposed-convolution upsampling, and skip connections. Feature



(a) Performance of DiffCool-synthesized heat sinks.

(b) Performance of conventional heat sinks.

Figure 4: Selective performance evaluation across test cases. (a) DiffCool achieves peak temperature reductions up to 47.9%, average temperature reductions up to 39.8%, and temperature gradient reductions up to 64.1% versus the bare chip. (b) Conventional heat sinks exhibit significantly inferior performance compared to the DiffCool-synthesized designs.

maps reduce from 32×32 to 2×2 with channels $[16, 32, 64, 512, 512]$. Transformer encoders at the bottleneck and upsampling stages capture global context, while sinusoidal embeddings [30] inject temporal information. A parallel convolutional tower fuses conditioning images at multiple scales. Outputs (B, C, H, W) are discretized via Gumbel-Softmax with initial noise $\beta_1 = 10^{-2}$ and temperature $\tau = 1.0$.

Datasets. We generate 1,000 synthetic heat maps with Gaussian-distributed sources, each paired with random, manufacturable heat sink designs. Per-layer heat maps are recorded to supervise training. Starting from random initialization, the model progressively learns to synthesize thermally optimal structures.

Training Details. We train for 1,000 iterations (batch size 32, learning rate 10^{-3}) using Adam [31]. An AI-based surrogate thermal simulator [23] accelerates evaluation. We set dropout to 0.5, apply gradient clipping at 0.1, and use $K = 25$ diffusion timesteps with cubic cells for geometric plausibility. Training requires $\sim$5 hours on a Linux server (Intel Xeon Gold 6226R, 256 GB RAM, dual NVIDIA RTX 3090 GPUs).

Test Cases. We evaluate an UltraSPARC T1 chip (3 cm $\times$ 3 cm) [17] across seven realistic bare-chip heat maps (Table 1). Scenarios cover single large (#1, #2) and small (#3, #4) hotspots, plus continuous (#5) and isolated (#6, #7) multi-hotspot patterns. Initial profiles average $T_{\max} = 83.90°$C, $\bar{T} = 64.14°$C, and $\nabla T = 23.00°$C. All thermal evaluations use COMSOL Multiphysics as the industry gold standard [19].

Table 1: Statistics of the bare-chip heat maps in the test cases.

#	1	2	3	4	5	6	7	Average
$T_{\max}(°C)$	84.33	85.74	54.67	54.52	85.43	81.76	80.56	83.90
$\bar{T}(°C)$	64.45	64.68	63.98	64.18	64.82	63.49	63.40	64.14
$\nabla T(°C)$	23.41	24.87	23.78	23.93	24.55	20.85	19.64	23.00

4.2 Performance Evaluation

Heat Dissipation Performance. Figure 4a demonstrates DiffCool's ability to synthesize customized heat sinks addressing diverse thermal challenges. With a surface area of 6.4 cm $\times$ 6.4 cm [32], the resulting thermal profiles reveal an average $T_{\max}$ reduction of 47.9% (to $43.65°C$), $\bar{T}$ reduction of 39.8% (to $39.27°C$), and ∇T reduction of 64.1% (to $8.27°C$). These enhancements translate to reduced computational throttling and improved reliability [33, 34]. Benchmarking against PFHS [6], FPFHS [5], and NU-PFHS [9] (Figure 4b) confirms DiffCool's superiority. As shown in Table 2,

Table 2: Evaluation of heat dissipation performance across different heat sink designs. Both absolute values and relative performance (compared to PFHS) are reported for all evaluation metrics.

#		1	2	3	4	5	6	7	Average
PFHS	$T_{\max}(^\circ C)$	65.96 (1.00)	66.23 (1.00)	60.99 (1.00)	63.05 (1.00)	67.46 (1.00)	52.16 (1.00)	51.61 (1.00)	61.06 (1.00)
	$\bar{T}(^\circ C)$	55.46 (1.00)	56.40 (1.00)	50.67 (1.00)	52.70 (1.00)	57.73 (1.00)	46.13 (1.00)	45.50 (1.00)	52.09 (1.00)
	$\nabla T(^\circ C)$	23.11 (1.00)	24.03 (1.00)	19.94 (1.00)	22.83 (1.00)	25.55 (1.00)	12.61 (1.00)	12.54 (1.00)	20.09 (1.00)
FPFHS	$T_{\max}(^\circ C)$	55.71 (0.84)	56.03 (0.85)	52.09 (0.85)	53.77 (0.85)	57.06 (0.85)	47.22 (0.91)	47.03 (0.91)	52.70 (0.87)
	$\bar{T}(^\circ C)$	49.05 (0.88)	49.99 (0.89)	45.32 (0.89)	46.95 (0.89)	51.05 (0.88)	41.89 (0.91)	41.38 (0.91)	45.52 (0.89)
	$\nabla T(^\circ C)$	14.54 (0.63)	14.37 (0.60)	12.17 (0.61)	14.68 (0.64)	15.59 (0.61)	9.20 (0.73)	9.58 (0.76)	12.88 (0.65)
NU-PFHS	$T_{\max}(^\circ C)$	58.75 (0.89)	59.14 (0.89)	55.29 (0.90)	56.14 (0.89)	60.11 (0.89)	48.19 (0.92)	47.61 (0.92)	55.03 (0.90)
	$\bar{T}(^\circ C)$	49.72 (0.89)	50.55 (0.90)	45.97 (0.91)	47.02 (0.89)	51.66 (0.89)	42.42 (0.92)	41.81 (0.92)	47.02 (0.90)
	$\nabla T(^\circ C)$	18.82 (0.81)	20.08 (0.84)	16.98 (0.85)	18.93 (0.83)	21.46 (0.84)	10.67 (0.85)	10.47 (0.84)	16.77 (0.84)
DiffCool	$T_{\max}(^\circ C)$	**45.33 (0.69)**	**45.59 (0.69)**	**43.66 (0.72)**	**44.20 (0.70)**	**46.18 (0.68)**	**40.61 (0.78)**	**39.99 (0.77)**	**43.65 (0.72)**
	$\bar{T}(^\circ C)$	**40.97 (0.74)**	**41.64 (0.74)**	**38.48 (0.76)**	**39.47 (0.75)**	**42.28 (0.73)**	**36.27 (0.79)**	**35.75 (0.79)**	**39.27 (0.75)**
	$\nabla T(^\circ C)$	**9.06 (0.39)**	**9.12 (0.38)**	**8.44 (0.42)**	**9.12 (0.40)**	**9.75 (0.38)**	**6.33 (0.50)**	**6.08 (0.48)**	**8.27 (0.42)**
	Time(s)	0.65	0.61	0.70	0.69	0.63	0.70	0.66	0.66

Table 3: Comparison of metal fill ratio (in percentage) for different heat sink designs.

#	1	2	3	4	5	6	7	Average
PFHS	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
FPFHS	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00
NU-PFHS	50.39	50.39	51.36	51.36	50.39	53.32	52.83	51.43
DiffCool	54.17	55.16	53.52	53.91	54.43	54.97	55.08	54.47

DiffCool achieves 28.3% lower $T_{\max}$ and 56.9% better ∇T than PFHS, synthesizing designs in under 1 second per design on a single RTX 3090 GPU.

Comparison with Topology Optimization. We compare DiffCool against a surrogate-guided filtered density topology optimization (TO) baseline [35, 36, 37, 38]. Both methods use the same 64×64 heat map, surrogate, and constraints (feature-size and bottom-up support). The TO baseline performs layer-wise local optimization without cross-layer look-ahead. At nearly identical metal fills (54.00% for TO, 54.47% for DiffCool), DiffCool reduces average $T_{\max}$ by $32.40^\circ C$ (42.6%) and $\bar{T}$ by $32.65^\circ C$ (45.4%) (Table 4). Although TO yields a slightly lower ∇T ($7.82^\circ C$ vs. $8.27^\circ C$), its uniformly hotter temperature field indicates less effective cooling, demonstrating that DiffCool's gains arise from learned multilayer topology rather than additional material.

Table 4: Comparison with topology optimization (TO) baseline. Both methods use nearly identical metal fills ($\sim$54%).

Case	Topology Optimization			DiffCool		
	$T_{\max}\ (^\circ C)$	$\bar{T}\ (^\circ C)$	$\nabla T\ (^\circ C)$	$T_{\max}\ (^\circ C)$	$\bar{T}\ (^\circ C)$	$\nabla T\ (^\circ C)$
1	81.53	77.55	8.39	45.33	40.97	9.06
2	83.10	79.46	8.67	45.59	41.64	9.12
3	74.27	69.45	7.93	43.66	38.48	8.44
4	77.04	72.60	8.75	44.20	39.47	9.12
5	85.70	82.14	9.10	46.18	42.28	9.75
6	66.20	61.87	6.13	40.61	36.27	6.33
7	64.54	60.38	5.80	39.99	35.75	6.08
Avg	76.05	71.92	7.82	43.65	39.27	8.27

Material-Budget-Controlled Evaluation. To isolate topological advantages from metal quantity, we constrained DiffCool to a 25.00% fill, matching FPFHS. Despite removing over half the metal, controlled DiffCool outperforms PFHS on all metrics (Table 5). Compared with FPFHS at the identical fill, its $T_{\max}$ is only $2.17^\circ C$ higher, while ∇T is $1.94^\circ C$ (15.0%) lower, confirming that topological structure dictates heat dissipation and DiffCool retains strong performance under strict material budgets.

Metal Utilization Efficiency. Table 3 and Table 2 reveal no direct correlation between metal usage and cooling effectiveness. PFHS and NU-PFHS employ comparable metal volumes (50.00% vs. 51.43%)

Table 5: Material-budget-controlled evaluation. DiffCool is constrained to 25.00% fill to match FPFHS.

Method	Metal Fill	$T_{\max}$ (°C)	$\bar{T}$ (°C)	∇T (°C)
PFHS	50.00%	61.06	52.09	20.09
FPFHS	25.00%	52.70	45.52	12.88
NU-PFHS	51.43%	55.03	47.02	16.77
DiffCool (controlled)	25.00%	54.87	49.77	10.94
DiffCool (original)	54.47%	43.65	39.27	8.27

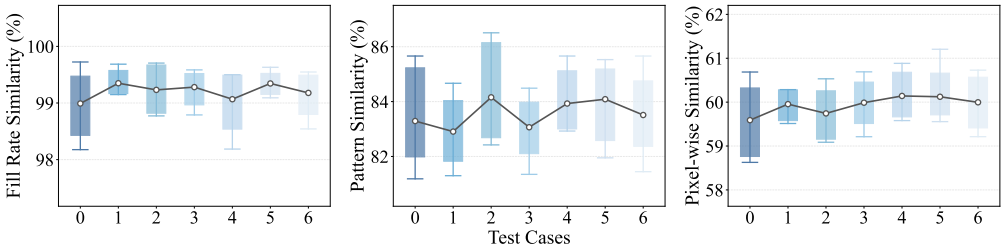


Figure 5: Output similarity of DiffCool across multiple runs.

yet exhibit notable performance differences. DiffCool achieves superior thermal performance with a similar fill ratio to NU-PFHS (54.47% vs. 51.43%), confirming that topological structure, rather than sheer quantity, dictates heat dissipation.

Customization Ability for Hotspots. As shown in Figure 4a, DiffCool synthesizes topologies adapting precisely to spatial thermal profiles. For continuous hotspots, it forms dense, interconnected metal fills for efficient conduction. For isolated hotspots, it creates clustered features to mitigate local peaks while preserving voids for convection. This dynamic balancing significantly outperforms conventional uniform architectures.

4.3 Reproducibility and Ablation Study

Reproducibility. We evaluate reproducibility by generating heat sinks five times per test case. Consistency is quantified via: (1) *fill rate similarity*; (2) *block-wise pattern match* (partitioning the design into 4×4 blocks, counting a match if the relative fill difference is below 20%); and (3) *pixel-wise accuracy*. As shown in Figure 5, fill rate similarity averages 99.21%, block-wise match reaches 83.56%, and pixel-wise accuracy is 59.93%. This validates that thermal-aware loss and manufacturing constraints guide the model toward structurally consistent solutions.

Ablation Study. We compare three training objectives: heat-flux-driven, hotspot-driven, and our composite loss. As shown in Table 6, the heat-flux objective minimizes $T_{\max}$ and $\bar{T}$ but yields larger ∇T due to insufficient localized cooling. The hotspot objective reduces ∇T but increases global temperatures. Our composite loss balances these, achieving ∇T reductions comparable to the hotspot-driven approach with only marginal increases in $T_{\max}$ and $\bar{T}$, yielding the most balanced thermal performance.

Table 6: Ablation study of thermal-aware loss components. Both absolute values and relative performance (normalized to the heat-flux-driven baseline) are reported.

#		1	2	3	4	5	6	7	Average
Heat-Flux -Driven	$T_{\max}(°C)$	**45.19 (1.00)**	**45.54 (1.00)**	**43.64 (1.00)**	**44.10 (1.00)**	**46.09 (1.00)**	**40.50 (1.00)**	**39.99 (1.00)**	**43.58 (1.00)**
	$\bar{T}(°C)$	**40.73 (1.00)**	**41.48 (1.00)**	**38.43 (1.00)**	**39.35 (1.00)**	**42.13 (1.00)**	**36.15 (1.00)**	**35.75 (1.00)**	**39.15 (1.00)**
	$\nabla T(°C)$	9.20 (1.00)	9.60 (1.00)	8.51 (1.00)	9.35 (1.00)	9.99 (1.00)	6.43 (1.00)	6.17 (1.00)	8.46 (1.00)
Hotspot -Driven	$T_{\max}(°C)$	45.60 (1.01)	46.13 (1.01)	43.84 (1.00)	44.55 (1.01)	46.43 (1.01)	40.89 (1.01)	40.79 (1.02)	44.03 (1.01)
	$\bar{T}(°C)$	41.55 (1.02)	42.42 (1.02)	38.89 (1.01)	40.04 (1.02)	42.80 (1.02)	36.55 (1.01)	36.62 (1.02)	39.84 (1.02)
	$\nabla T(°C)$	**8.29 (0.90)**	**8.45 (0.88)**	**7.82 (0.92)**	**8.58 (0.92)**	**8.95 (0.90)**	**6.19 (0.96)**	**5.89 (0.95)**	**7.74 (0.91)**
DiffCool	$T_{\max}(°C)$	45.33 (1.00)	45.59 (1.00)	43.66 (1.00)	44.20 (1.00)	46.18 (1.00)	40.61 (1.00)	39.99 (1.00)	43.65 (1.00)
	$\bar{T}(°C)$	40.97 (1.01)	41.64 (1.00)	38.48 (1.00)	39.47 (1.00)	42.28 (1.00)	36.27 (1.00)	35.75 (1.00)	39.27 (1.00)
	$\nabla T(°C)$	9.06 (0.98)	9.12 (0.95)	8.44 (0.99)	9.12 (0.98)	9.75 (0.98)	6.33 (0.98)	6.08 (0.99)	8.27 (0.98)

Surrogate Validation. We evaluated 16 independently constructed designs under four unseen power maps using both the surrogate and COMSOL. As shown in Table 7, the surrogate consistently selects the same best design as COMSOL. Rankings are identical for Maps 1 and 3; minor swaps for Maps 2 and 4 involve candidates differing by merely $0.01°C$ and $0.06°C$. On unseen structures, the mean absolute errors between the surrogate and COMSOL predictions are $0.253°C$ for $T_{\max}$ and $0.218°C$ for $\bar{T}$, with worst-case errors strictly bounded at $0.313°C$ and $0.230°C$, respectively. Crucially, all headline results are independently re-evaluated using high-fidelity COMSOL simulations.

Table 7: Ranking consistency between COMSOL and the surrogate model across four unseen power maps. Candidates are ordered from lowest to highest $T_{\max}$.

Power Map	COMSOL Ranking	Surrogate Ranking	Best Matched
1	Case 4, Case 3, Case 2, Case 1	Case 4, Case 3, Case 2, Case 1	Yes
2	Case 4, Case 2, Case 1, Case 3	Case 4, Case 2, Case 3, Case 1	Yes
3	Case 1, Case 4, Case 3, Case 2	Case 1, Case 4, Case 3, Case 2	Yes
4	Case 4, Case 1, Case 3, Case 2	Case 4, Case 3, Case 1, Case 2	Yes

5 Conclusion and Limitations

We presented DiffCool, a physics-guided, label-free framework for automated heat sink synthesis. Integrating manufacturing-aware denoising with a thermal-aware loss enables sub-second generation of optimized designs from thermal profiles. Experiments achieve 27.9% lower peak temperature, 24.6% lower average temperature, and 57.4% improved thermal gradient versus conventional designs, confirming generative AI effectively bridges thermal performance and mass production requirements.

Limitations. First, while the compact-model-based differentiable simulator enables efficient gradient propagation during training, its simplified convection approximation may exhibit quantitative deviations from high-fidelity conjugate heat transfer simulations under complex or non-uniform airflow regimes not fully captured by the current model. Second, the framework exclusively models passive natural convection into still air, leaving active forced-convection scenarios (e.g., fan-assisted cooling systems) unvalidated. Finally, our evaluation relies on fixed grid resolutions and synthetically generated thermal profiles; generalizing the methodology to arbitrary, high-resolution industrial layouts or co-optimizing it alongside electrical routing constraints remains an open challenge. While COMSOL provides reliable relative performance estimates under matched physical assumptions, absolute temperatures in hardware may differ due to manufactured dimension deviations, surface roughness affecting convection, thermal interface contact resistance, and ambient airflow variations. Physical fabrication and measurement remain necessary to quantify these absolute sim-to-real differences, which constitutes our future work.

6 Acknowledgement

This research was partially conducted by ACCESS-AI Chip Center for Emerging Smart Systems, supported by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government. This work is jointly supported by the Research Grants Council of Hong Kong SAR No. CUHK14211324 and No. T46-415/25-R.

References

[1] B. Qin, J. Jiang, L. Wang, Q. Guo, C. Zhang, L. Xu, X. Ni, P. Yin, L.-M. Peng, E. Wang, F. Ding, C. Qiu, C. Liu, and K. Liu, "Two-dimensional indium selenide wafers for integrated electronics," *Science*, vol. 389, no. 6757, pp. 299–302, 2025.

[2] H. E. Ahmed, B. H. Salman, A. S. Kherbeet, and M. Ahmed, "Optimization of thermal design of heat sinks: A review," *International Journal of Heat and Mass Transfer*, vol. 118, pp. 129–153, 2018.

[3] I. Mudawar, "Two-phase microchannel heat sinks: theory, applications, and limitations," *Journal of Electrical Packaging*, vol. 133, no. 4, p. 041002, 2011.

[4] N. A. C. Sidik, M. N. A. W. Muhamad, W. M. A. A. Japar, and Z. A. Rasid, "An overview of passive techniques for heat transfer augmentation in microchannel heat sink," *International Communications in Heat and Mass Transfer*, vol. 88, pp. 74–83, 2017.

[5] H.-H. Wu, Y.-Y. Hsiao, H.-S. Huang, P.-H. Tang, and S.-L. Chen, "A practical plate-fin heat sink model," *Applied Thermal Engineering*, vol. 31, no. 5, pp. 984–992, 2011.

[6] H. Shafeie, O. Abouali, K. Jafarpur, and G. Ahmadi, "Numerical study of heat transfer performance of single-phase heat sinks with micro pin-fin structures," *Applied Thermal Engineering*, vol. 58, no. 1-2, pp. 68–76, 2013.

[7] D.-K. Kim, "Thermal optimization of branched-fin heat sinks subject to a parallel flow," *International Journal of Heat and Mass Transfer*, vol. 77, pp. 278–287, 2014.

[8] S.-C. Lin, F.-S. Chuang, and C.-A. Chou, "Experimental study of the heat sink assembly with oblique straight fins," *Experimental Thermal and Fluid Science*, vol. 29, no. 5, pp. 591–600, 2005.

[9] C.-H. Huang and Y.-H. Chen, "An impingement heat sink module design problem in determining simultaneously the optimal non-uniform fin widths and heights," *International Journal of Heat and Mass Transfer*, vol. 73, pp. 627–633, 2014.

[10] A. Al-Damook, N. Kapur, J. Summers, and H. Thompson, "Computational design and optimisation of pin fin heat sinks with rectangular perforations," *Applied Thermal Engineering*, vol. 105, pp. 691–703, 2016.

[11] W. Yuan, J. Zhao, C. Tso, T. Wu, W. Liu, and T. Ming, "Numerical simulation of the thermal hydraulic performance of a plate pin fin heat sink," *Applied Thermal Engineering*, vol. 48, pp. 81–88, 2012.

[12] M. Gierczak, P. Markowski, and A. Dziedzic, "Modeling, simulation and analysis of temperature distribution in a heat sink," in *39th International Spring Seminar on Electronics Technology*, pp. 122–127, IEEE, 2016.

[13] J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, and S. Fidler, "Get3d: A generative model of high quality 3d textured shapes learned from images," *Advances in neural information processing systems*, vol. 35, pp. 31841–31854, 2022.

[14] Z. Wang, Y. Shen, W. Zhao, Y. Bai, G. Chen, F. Farnia, and B. Yu, "Diffpattern: Layout pattern generation via discrete diffusion," in *Proceedings of the 60th ACM/IEEE Design Automation Conference*, pp. 1–6, 2023.

[15] Z. Yang, Z. Zhuang, B. Yu, T.-Y. Ho, M. D. Wong, and S. K. Lim, "Ml-based fine-grained modeling of dc current crowding in power delivery tsvs for face-to-face 3d ics," in *Proceedings of the 2025 International Symposium on Physical Design*, pp. 233–241, 2025.

[16] S. Liang, M. Lian, M. Li, T.-M. Tseng, U. Schlichtmann, and T.-Y. Ho, "ARMM: Adaptive reliability quantification model of microfluidic designs and its graph-transformer-based implementation," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 1–9, IEEE, 2023.

[17] A. S. Leon, K. W. Tam, J. L. Shin, D. Weisner, and F. Schumacher, "A power-efficient high-throughput 32-thread sparc processor," *IEEE Journal of Solid-state circuits*, vol. 42, no. 1, pp. 7–16, 2006.

[18] Y. Chuang *et al.*, "Micro precision casting based on investment casting for micro structures with high aspect ratio," *Transactions of Nonferrous Metals Society of China*, vol. 19, pp. s521–s525, 2009.

[19] C. Multiphysics, "Introduction to comsol multiphysics®," *COMSOL Multiphysics, Burlington, MA, accessed Feb*, vol. 9, no. 2018, p. 32, 1998.

[20] C. Wang, Q. Xu, C. Nie, H. Cao, J. Liu, and Z. Li, "An efficient thermal model of chiplet heterogeneous integration system for steady-state temperature prediction," *Microelectronics Reliability*, vol. 146, p. 115006, 2023.

[21] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusam, "Compact thermal modeling for temperature-aware design," in *Proceedings of the 41st annual design automation conference*, pp. 878–883, 2004.

[22] J. R. Shewchuk *et al.*, "An introduction to the conjugate gradient method without the agonizing pain," 1994.

[23] M. Edalatifar, M. B. Tavakoli, M. Ghalambaz, and F. Setoudeh, "Using deep learning to learn physics of conduction heat transfer," *Journal of Thermal Analysis and Calorimetry*, vol. 146, no. 3, pp. 1435–1452, 2021.

[24] Y. Zhang, Z. Gao, K. He, C. Li, and R. Mao, "From patches to wsis: A systematic review of deep multiple instance learning in computational pathology," *Information Fusion*, p. 103027, 2025.

[25] P. Sun, L. Huo, X. Chen, and S. Liang, "Rotor angle stability prediction using temporal and topological embedding deep neural network based on grid-informed adjacency matrix," *Journal of Modern Power Systems and Clean Energy*, vol. 12, no. 3, pp. 695–706, 2023.

[26] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[27] D. Berrar, "Bayes' theorem and naive bayes classifier," *Encyclopedia of bioinformatics and computational biology: ABC of bioinformatics*, vol. 403, no. 412, 2018.

[28] C. Wang, Q. Xu, C. Nie, H. Cao, J. Liu, D. Zhang, and Z. Li, "A multiscale anisotropic thermal model of chiplet heterogeneous integration system," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 32, no. 1, pp. 178–189, 2023.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] F. Tavakkoli, S. Ebrahimi, S. Wang, and K. Vafai, "Analysis of critical thermal issues in 3d integrated circuits," *International Journal of Heat and Mass Transfer*, vol. 97, pp. 337–352, 2016.

[33] S.-C. Lin and K. Banerjee, "Cool chips: Opportunities and implications for power and thermal management," *IEEE Transactions on Electron Devices*, vol. 55, no. 1, pp. 245–255, 2007.

[34] T. Braun, K.-F. Becker, M. Koch, V. Bader, R. Aschenbrenner, and H. Reichl, "High-temperature reliability of flip chip assemblies," *Microelectronics Reliability*, vol. 46, no. 1, pp. 144–154, 2006.

[35] T. Zeng, H. Wang, M. Yang, and J. Alexandersen, "Topology optimization of heat sinks for instantaneous chip cooling using a transient pseudo-3d thermofluid model," *International Journal of Heat and Mass Transfer*, vol. 154, p. 119681, 2020.

[36] B. Rogié and C. S. Andreasen, "Design complexity tradeoffs in topology optimization of forced convection laminar flow heat sinks: B. rogié, cs andreasen," *Structural and Multidisciplinary Optimization*, vol. 66, no. 1, p. 6, 2023.

[37] S. Sun, B. Rankouhi, D. J. Thoma, M. J. Cheadle, G. D. Maples, M. H. Anderson, G. Nellis, and X. Qian, "Topology optimization, additive manufacturing and thermohydraulic testing of heat sinks," *International Journal of Heat and Mass Transfer*, vol. 224, p. 125281, 2024.

[38] Y. Hua, L. Luo, S. Le Corre, and Y. Fan, "Machine-learning topology optimization with stochastic gradient descent optimizer for heat conduction problems," *International Journal of Heat and Mass Transfer*, vol. 223, p. 125226, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We make sure that the main claims made in the abstract and introduction accurately reflect the paper's contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations at the end of the main paper. Specifically, the limitations of the compact simulator, the consideration of only passive dissipation into still air, and the generalizability are discussed.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: We don't have theoretical result in this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We has introduced the details of implementation and training settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our source code and dataset will be released at https://github.com/Kryie24/DiffCool

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the reproducibility using candle charts.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We introduce these details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We make sure to preserve anonymity.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets used in the paper, such as datasets and models, are properly credited with clear documentation of their sources and licenses, ensuring full compliance with existing terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer:[N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer:[N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Detailed Heat Dissipation Performances

To complement the quantitative metrics in Table 2 from the main document, Figure 6 visualizes the structural differences between conventional heat sinks and DiffCool's synthesized designs. While standard pin fin heat sink (PFHS) and folded flat plate fin heat sink (FPFHS) configurations maintain identical geometries across all thermal scenarios, the non-uniform pin fin heat sink (NU-PFHS) exhibits limited adaptation to heatmap variations. The detailed performance of DiffCool over all the 7 test cases can be observed from Figure 7. Despite these minor customizations, the cooled thermal profiles in Figure 8 demonstrate NU-PFHS's persistent shortcomings: hotspot regions show much higher residual temperatures compared to DiffCool's solutions, with particularly pronounced deficits in multi-hotspot configurations (test cases #5-7).

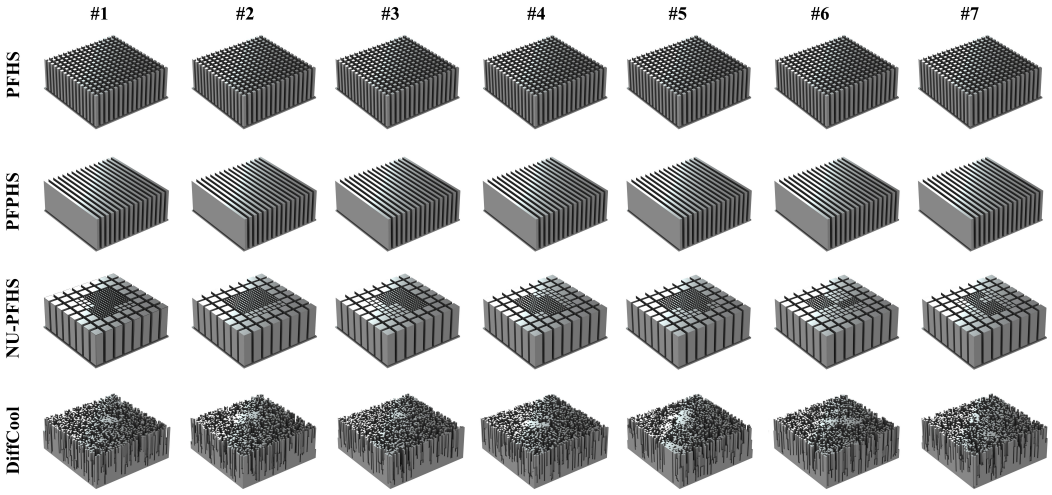


Figure 6: Different heat sinks for the 7 test cases.

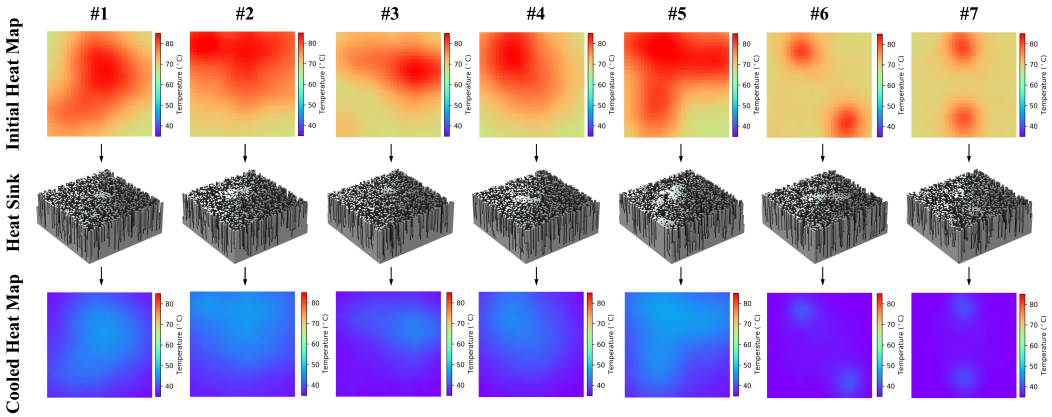


Figure 7: Performance of DiffColl over all the 7 test cases.

B Supplementary Results of Customization Ability for Hotspots

In the main text, due to page limitations, we only show the 3D heat sink morphology (Figure 4a) to give readers a general impression of DiffCool's ability to customize structures for different hotspot patterns. More concrete results, including the synthesis of dense interconnected metal fills for continuous hotspots and clustered metal features with distributed voids for isolated hotspots, are presented in Figure 9 for a better understanding. The figure recaps DiffCool's two hotspot-adaptive

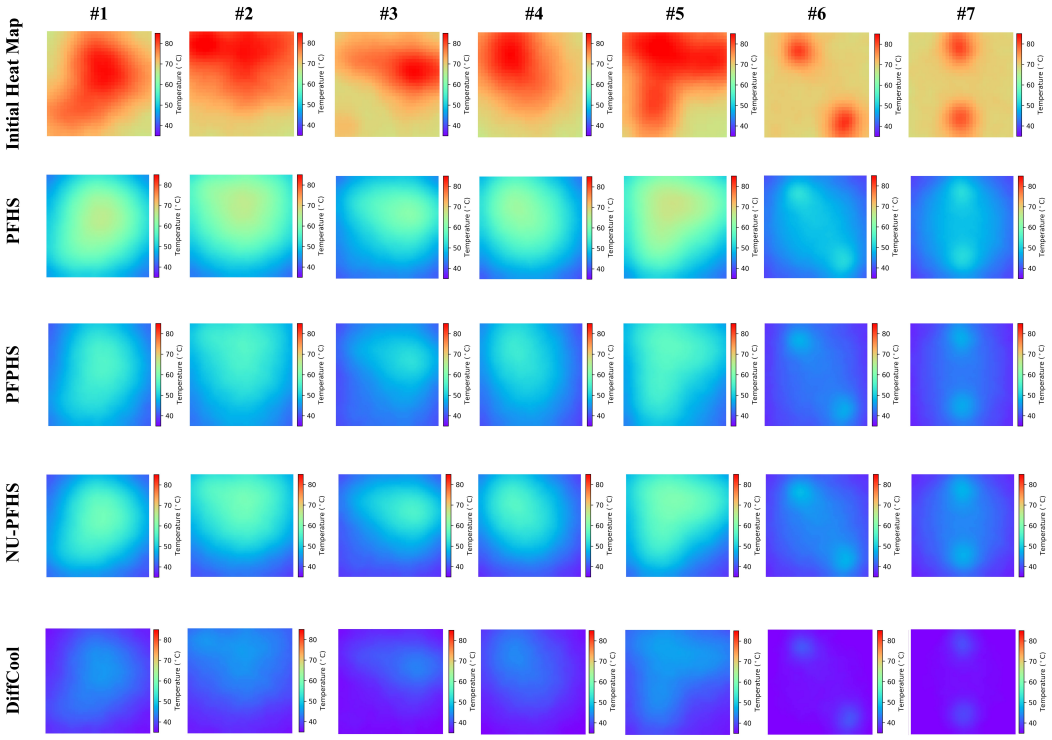
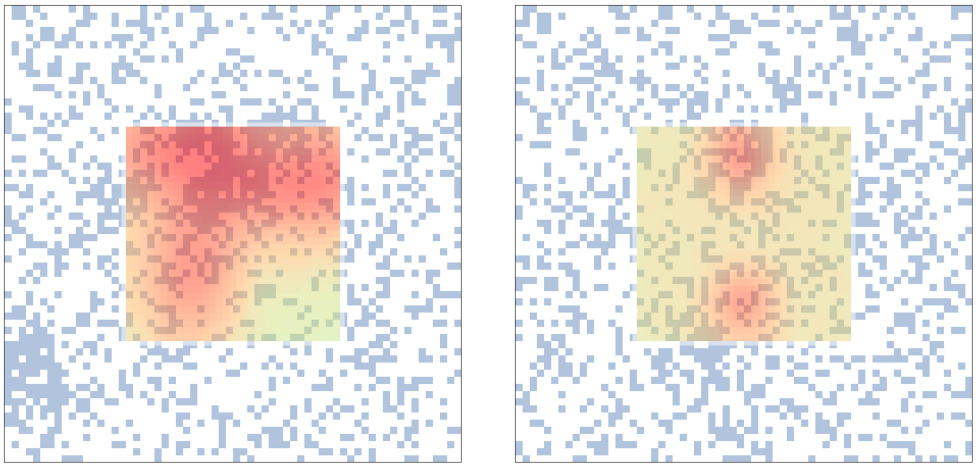


Figure 8: Detailed cooled heat maps comparison of utilizing different heat sinks.

mechanisms: dense conductive pathways for continuous hotspots, and clustered features with voids for isolated hotspots, highlighting its ability to outperform conventional uniform heat sink designs.



(a) Test case #5.

(b) Test case #7.

Figure 9: Customized heat sink structure aligned with bare-chip hotspots. (a) Continuous hotspot: dense metal fills match large hot regions. (b) Isolated hotspots: clustered metal fills target discrete hotspots while maintaining convective voids.

C Training Data Generation and Learning Effects

Since optimal heat sink designs cannot be analytically derived, we generate our training dataset by creating randomized yet manufacturable heat sink configurations. As illustrated in Figure 10, initial heat maps are constructed using randomly positioned Gaussian-distributed heat sources with varying intensities and distributions. Corresponding heat sink designs are generated through a constrained random process shown in Figure 11, where metal fills are probabilistically removed from the previous layer, with the metal removal probability decreasing as local temperature increases. This creates physically plausible but suboptimal designs that satisfy basic manufacturing constraints.

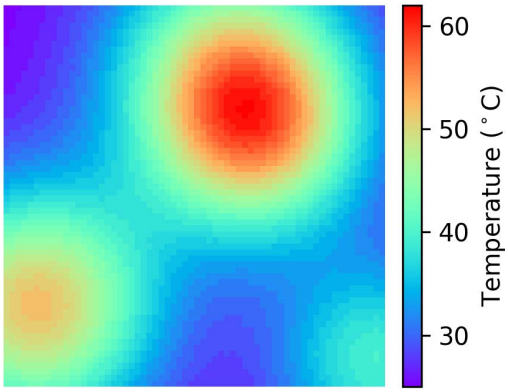


Figure 10: A demo heat map from the training dataset.

The diffusion model learns to refine these initial designs through training, progressively developing superior synthesis strategies. Figure 12 demonstrates this evolution, contrasting the dataset's random configurations with DiffCool's optimized outputs. The dramatic performance improvement, evidenced by the 27.9% lower peak temperature, 24.6% lower average temperature, and 57.4% improved thermal gradient in our main experiments, confirms the model's successful learning of underlying thermal-structural relationships, validating our training paradigm's effectiveness despite starting from suboptimal examples.

D Selection of Hyperparameters

We conduct a series of studies to investigate the impact of various hyperparameters.

D.1 Learning Rate

Figure 13(a) shows the normalized loss after model convergence as we vary the learning rate from $1e{-}5$ to $1e{-}2$. The results indicate that the model achieves the lowest loss with a learning rate of $1e{-}3$. Consequently, we adopt $1e{-}3$ as the learning rate in our experiments.

D.2 Batch Size

As shown in Figure 13(b), we empirically observe that a batch size of 32 yields the optimal normalized loss. Under this setting, the training process takes approximately 5 hours to converge.

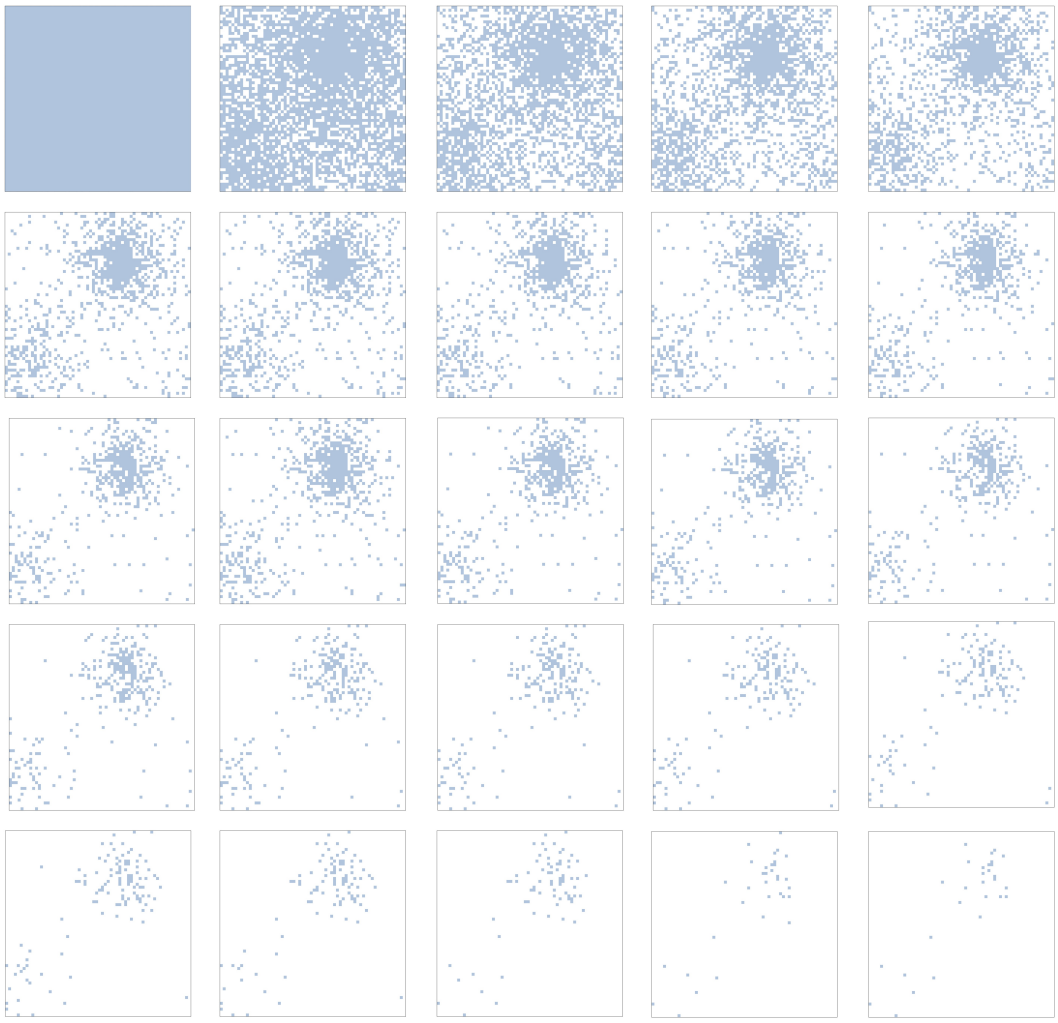


Figure 11: Layers of the initial randomly generated heat sink.

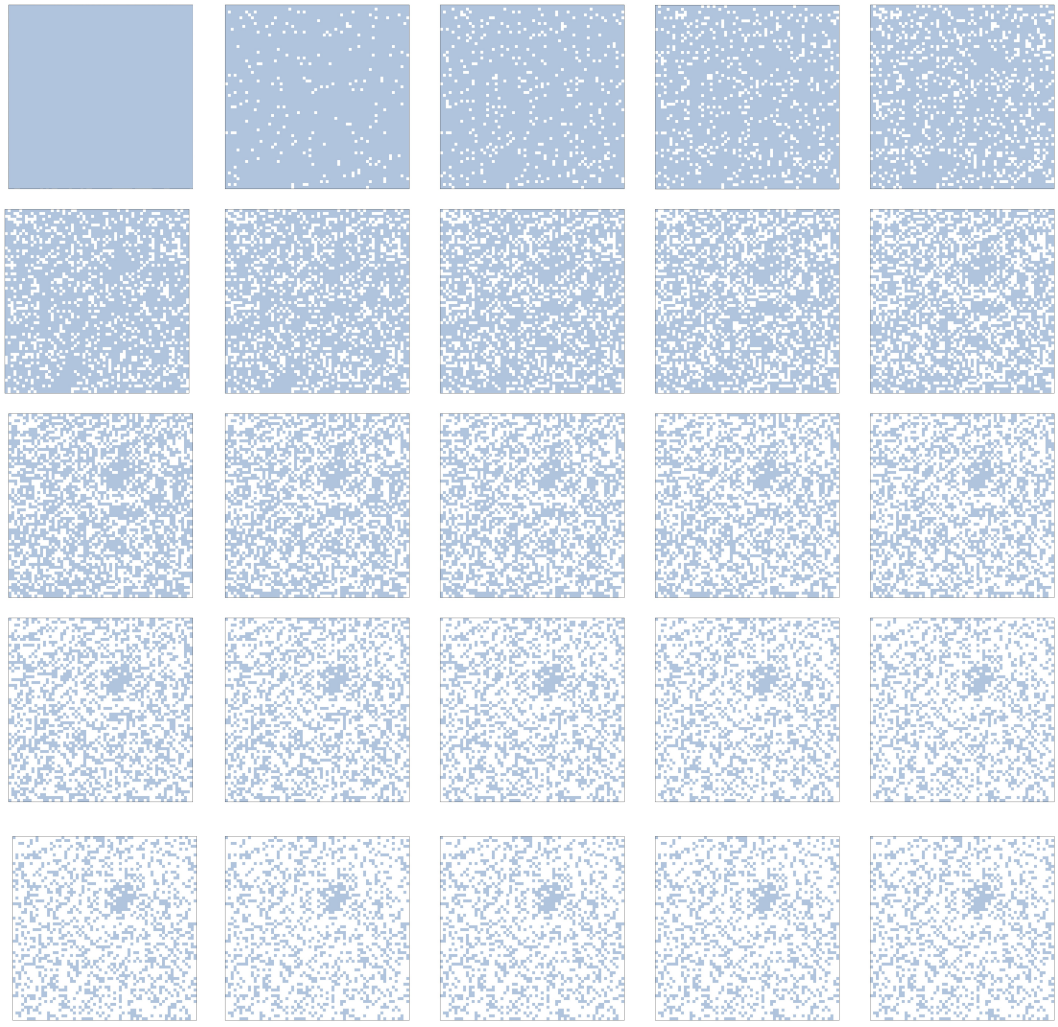
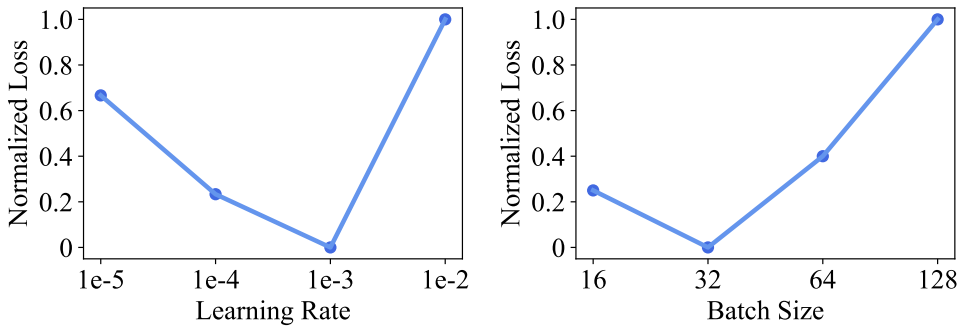


Figure 12: Layers of the DiffCool-synthesized heat sink.



(a) Learning rate sensitivity.

(b) Batch size sensitivity.

Figure 13: Hyperparameter effects on model performance.