MIX-3D: AI-based Architecture-Circuit Co-design Methodology for Mixed-Node, Mixed-Area 3D ICs

Min Gyu Park[1], Doyun Kim[2], Sung Kyu Lim[1]
[1]Georgia Institute of Technology, Atlanta, USA; [2]Samsung Advanced Institute of Technology, Suwon, South Korea
mgpark@gatech.edu

***Abstract*—3D Integrated Circuits (ICs) significantly enhance chip performance but require substantial engineering due to their expanded design space. To tackle this, we introduce the MIX-3D framework, an advanced optimizer utilizing Variational Autoencoders for robust extrapolation, identifying energy-efficient and thermal-aware design configurations for mixed-node, mixed-area 3D ICs. It enables architecture-circuit co-design for F2F 2-tier Logic-on-Memory 3D ICs, providing real-time predictions of both back-end and front-end metrics. Additionally, transfer learning reduces dataset construction time by 64%, a common challenge in supervised learning. Experimental results demonstrate that MIX-3D delivers 12% improvements in energy efficiency and 62% less power compared to equal-area 3D ICs. Furthermore, our thermal-aware design reduces chip temperature by 38%.**

I. Introduction

Design Space Exploration (DSE) frameworks are essential for optimizing chip design, given specifications such as energy-efficiency and thermal-aware design. Traditional 2D AI accelerator DSE frameworks [17], [11], [21] rely on analytical approaches mainly focused on front-end metrics, often neglecting back-end constraints. To address these limitations, regression-based DSE methods [4], [16] have been developed, using physical design and architectural simulation datasets to predict both front-end and back-end metrics. Although this methodology allows for identifying the optimal configuration, it necessitates time-intensive dataset creation for each architecture.

Meanwhile, 3D integration has emerged as a key solution to the slowing of Moore's Law by offering expanded memory capacity and shorter interconnects. Notably, Heterogeneous 3D ICs, demonstrated in chips like Lakefield [5] and Ponte Vecchio [6], enable low power and cost-effective designs through diverse technology (tech) nodes across tiers. However, unlike 2D ICs, 3D designs introduce additional parameters—like per-tier tech nodes, die areas, metal layer counts, and bonding mechanisms—vastly expanding the design space. Particularly, variable die areas increase the number of design points exponentially with the number of memory components in the memory pool, making it challenging to find an optimal 3D IC configuration.

In this paper, we introduce MIX-3D, a novel architecture-circuit co-design framework for 3D AI accelerators using a Variational Autoencoder (VAE). Our key contributions are outlined as follows:

- We present the first framework for mixed-node, mixed-area 3D ICs. It uncovers a design that outperforms the energy-efficient configuration in [16] across all metrics.
- For the first time, chip temperature, IR-drop, and cost metrics are all together integrated, facilitating practical solution exploration.
- We employ VAE models that exhibit exceptional extrapolation capabilities, whereas conventional regression models face challenges in balancing overfitting and underfitting.
- By applying transfer learning to the VAE, we achieve equivalent accuracy while reducing the dataset generation time from 653 hours to 237 hours, a 64% reduction compared to the time required when using conventional regression models.

979-8-3315-2710-5/25/$31.00 ©2025 IEEE

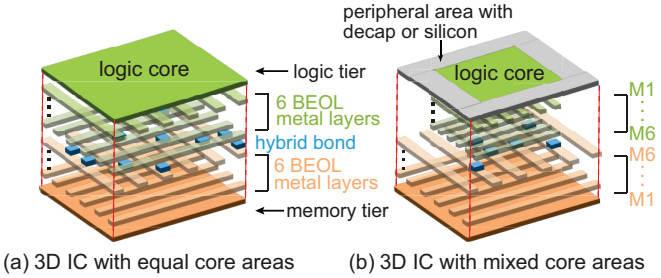

Fig. 1: Face-to-Face 2-tier 3D ICs with tier area balancing.

II. Related Work

Front-end metrics influence system performance, while back-end metrics affect physical design, necessitating simultaneous consideration due to trade-offs. Regression-based DSE framework for 2D ICs was introduced in [4], using interpolation on power, performance, runtime, and energy metrics. 3DNN-Xplorer [16] extended this to a 3D framework for a 2-tier 3D accelerator, adding extrapolation capabilities. Despite its effectiveness in predicting both front-end and back-end metrics for large-scale designs, the methodology still uses conventional regression models which require comprehensive dataset construction for each additional architecture that needs to be investigated. Additionally, the 3DNN-Xplorer framework imposes equal core area constraints, as illustrated in Fig. 1-(a). This severely limits the design space of 3D ICs, particularly in AI accelerators where on-chip memory capacity heavily impacts front-end metrics. As a result, they were forced into sub-optimal tech node combinations to increase the on-chip memory while sacrificing other metrics.

For the first time, we fully explore the mixed-node, mixed-area scenarios in Face-to-Face (F2F) 2-tier 3D accelerators. This approach reveals the optimal design that [16] failed to uncover, demonstrating improvements across all evaluated metrics. With mixed core allocation, we incorporate decap and silicon into the peripheral of the logic core, where silicon facilitates heat dissipation as well as uniform pressure during packaging.

III. Motivation

A. Why Is 3D Chip Architectural DSE Harder?

Table I compares architectural design space complexity for 2D and 3D ICs. In terms of technology complexity, 2D ICs exhibit a complexity of $\mathcal{O}(m)$, while 3D ICs expand to $\mathcal{O}(m^k)$ technology combinations across tiers. Once the technology is selected for each tier, the memory complexity in 2D ICs is $\mathcal{O}(n^p)$, with independent memory type assignment per component, where n denotes the number

TABLE I: Architectural design space complexity of 2D and 3D ICs. n : #mem. types, p : #mem. components, m : #tech, k : #tiers in 3D ICs.

	2D ICs	**3D ICs equal-area**	**3D ICs mixed-area**
Tech complexity	$\mathcal{O}(m)$	$\mathcal{O}(m^k)$	$\mathcal{O}(m^k)$
Memory complexity	$\mathcal{O}(n^p)$	$\mathcal{O}(2^k)$	$\mathcal{O}(n^p * 2^k)$
#Design points of 128×128 Sys. Array	151	39	473

of memory types and p represents the number of memory components (e.g., $p = 1$ for MAERI, $p = 3$ for the Systolic Array). For 3D ICs with equal core areas, area uniformity restricts the number of feasible memory configurations to a constant set, yielding a complexity of $\mathcal{O}(2^k)$. This complexity is derived from $kC1+kC2+\cdots+kC(k-1)$, which accounts for the allocation of memory across the k available tiers. Finally, 3D ICs with mixed core areas, free from area uniformity constraints, have a memory complexity of $\mathcal{O}(n^p * 2^k)$.

The difference in design space complexity is evident in the number of feasible design points in our work. For a 128×128 Systolic Array, there are 151 design points in 2D ICs, 39 design points in 3D ICs with equal-area, and 473 design points in 3D ICs with mixed-area configurations. The equal-area case considers only scenarios where the area difference between logic and memory is within 5%, while mixed-area configurations account for cases where the memory area is larger than the logic area. This highlights how the equal-area constraint significantly limits the design space for 3D ICs, while mixed-area 3D ICs explore a broader space than the 2D ICs.

B. Why Deep Generative Models?

Deep Generative Models (DGMs) offer a powerful approach for generating new data points. Among various types of DGMs, VAE [9] stands out in learning the data distribution through the maximization of the Evidence Lower Bound (ELBO), as described in Equation 1.

$$ELBO = \log p_\theta(\mathbf{x}|\mathbf{z}^*) - D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\right) \tag{1}$$

Given that regression tasks can be addressed by learning distributions from data, VAEs are highly applicable in this context. Conventional methods, such as polynomial regression and tree-based models, utilized in prior works, are susceptible to overfitting, especially at higher polynomial degrees, and exhibit inherent limitations [14] in extrapolation. In contrast, the neural network-based VAE has strengths in extrapolation, making it effective in predicting metrics of large-scale design. Additionally, the neural network structure enables transfer learning, significantly reducing the dataset construction time.

IV. Our Architecture/Circuit Co-design Setup

A. Benchmark Architectures

Our framework supports two distinct architectures, MAERI [12] and Systolic Array [10], as depicted in Fig. 2. The Systolic Array, a specialized DNN accelerator with fixed interconnects, offers compactness but often under-utilizes Processing Elements (PEs) in certain workloads. Conversely, MAERI's flexible, programmable interconnects ensure high PE utilization across diverse workloads. These interconnect differences drive distinct connectivity complexity profiles—$\mathcal{O}(n)$ for Systolic Array and $\mathcal{O}(nlogn)$ for MAERI—leading to significant differences in routing resource demand that impact congestion, power, and performance. By selecting these fundamentally different architectures, we aimed to demonstrate the effective application of Variational Autoencoder-based transfer learning across a broad range of designs.

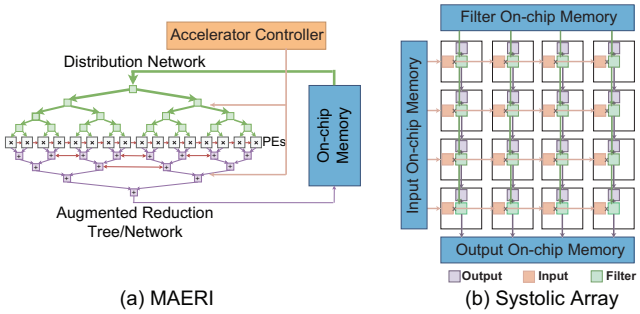

Fig. 2: Benchmark architectures used in this work. (a) MAERI with flexible interconnect [12], (b) Systolic Array [10] with rigid interconnect.

B. Architectural Simulation

Cycle-accurate simulators are used to evaluate runtime, with STONNE[15] for MAERI and SCALE-Sim[18] for the Systolic Array. Our framework predicts runtime at the layer level and integrates results, enabling extension to workloads beyond those covered in this study. We report results for ResNet-50, MobileNet-v1, and Transformer, demonstrating the framework's robustness across Convolution (Conv), Depthwise-Separable Convolution, MLP, and Attention layers. Users can further extend the framework to other workloads; for example, GNN computations align with MLP operations on the hardware, which ensures accurate predictions with current framework.

Given the challenge of acquiring component-wise power for energy calculation, we employ Equation 2 to estimate on-chip energy. This approach calculates on-chip energy by deriving power and effective frequency from Cadence Tempus, alongside total cycles and on-chip memory access count from cycle-accurate simulators.

$$\begin{aligned} \mathrm{E}_{logic} &= \mathrm{P}_{logic} \times \frac{1}{\mathrm{F}_{eff}} \times \text{Total Cycles} \\ \mathrm{E}_{mem} &= \mathrm{P}_{mem} \times \frac{1}{\mathrm{F}_{eff}} \times \text{\#On-chip memory Access} \end{aligned} \tag{2}$$

Furthermore, we use Algorithm 1 from [16], incorporating DRAM access counts to factor in DRAM latency and energy consumption on final runtime and total energy.

C. Physical Design of 3D ICs

Logic RTL for various processing element (PE) numbers and bandwidth (BW) configurations is generated using the Bluespec Verilog (BSV) compiler [8] for MAERI and parameterized RTL code [19] for the Systolic Array. Subsequently, netlists are synthesized with Synopsys Design Compiler.

3D IC physical design follows the Macro-3D [2] flow, a pioneering methodology capable of generating commercial-grade Logic-on-Memory 3D IC layouts. Using Cadence Innovus with 16nm and 28nm nodes, we explore four 3D integration options: homogeneous 28nm (Hom28), homogeneous 16nm (Hom16), heterogeneous 28nm for logic & 16nm for memory (Het28(L)16(M)) and heterogeneous 16nm for logic & 28nm for memory (Het16(L)28(M)). 3D ICs employ F2F hybrid bonding with a $1\mu m$ pitch and six metal layers per tier as a 3D back-end of line (BEOL). Since hybrid bond pitch is constrained by foundry fabrication limits, this study focuses on a single specification.

Post-physical design, we extract back-end metrics including power, performance, peak dynamic IR-drop, and maximum chip temperature. Power and timing analyses are conducted via static timing analysis (STA) with Cadence Tempus, dynamic IR-drop with Cadence Voltus, and thermal analysis with Hotspot [22] simulator.

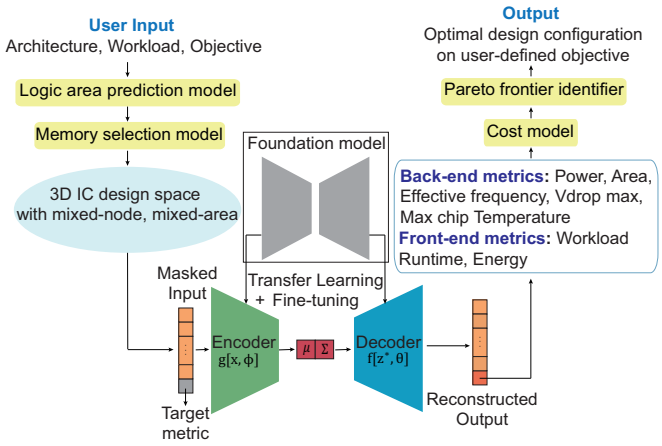

Fig. 3: Overview of our MIX-3D framework. Users can select benchmark architecture, target workload, and objective.

TABLE II: Contents of our database and common features.

Categ.	Feature	Values	Description
Arch.-related	PEs	16, 32, 64, 128, 256	# of PE in MAERI
	Memory BW	PE/4, PE/2	Global buffer Bandwidth in MAERI.
	(ROW, COL)	(8, 4), ···, (32, 32)	# of PEs in Systolic Array
Physical design-related	Logic node	16, 28nm	Tech on logic tier
	Memory node	16, 28nm	Tech on mem. tier
	Frequency	0.1 - 4 GHz	0.1 GHz increment
	Logic core area	Depends on design	Area to achieve 70% density
	Mem. tier area	{1×, 1.2×, ... 3.2×}	w.r.t. logic core area

V. Our MIX-3D Co-design Framework

A. Overview

Fig. 3 illustrates the MIX-3D framework, consisting of two models for design point generation and Variational Autoencoders (VAEs) for predicting five metrics. Back-end VAEs are trained on physical design data, while the front-end model uses cycle-accurate simulation results. Users can also leverage transfer learning with a pre-trained VAE on the foundation architecture, significantly reducing dataset construction time.

B. AI Training Data Generation Approach

To construct a database, it is essential to generate a sufficient volume of data that enables the model to accurately capture trends. For MAERI, parameter sweeps are conducted on processing element (PE) size and bandwidth (BW), while for the Systolic Array, variations in the number of rows (ROW) and columns (COL) are explored to capture trends across different design scales. Additionally, for the power and performance models, the target frequency is swept to account for frequency-dependent trends. The specific parameter values used in our dataset construction are detailed in Table II.

In modeling IR drop and thermal behavior, we adjust the memory tier area to assess the impact of peripheral decap (or silicon for thermal) as shown in Fig. 4. To meet large memory capacity needs and leverage peripheral area benefits on the logic tier, the memory tier area is swept over a range greater than the logic core area.

TABLE III: Number of data in our database and the wall time for construction. It takes a total of **1,888 hours** without transfer learning.

Metric	Design	Train dataset		Test dataset	
		#Data	Wall	#Data	Wall
Power & Perf.	MAERI	960	425h	640	450h
	Sys. Array	800	65h	480	196h
IR drop & Thermal	MAERI	384	112h	96	142h
	Sys. Array	288	22h	48	60h
Workload Runtime	MAERI	2,472	206h	1,648	127h
	Sys. Array	515	52h	309	31h

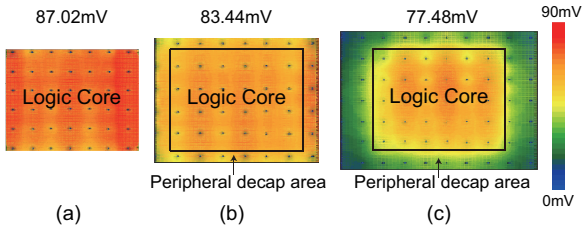

Fig. 4: Dynamic IR drop map of the logic (= noisier) tier. Heterogeneous 3D Sys. Array with 16nm (logic) + 28nm (memory) and 32 × 32 size is used. Logic and memory tier footprint is identical in (a).

Power Delivery Network (PDN) width and pitch are standardized across architectures to ensure consistent routing resources, while Redistribution Layers (RDL) mitigate dynamic IR drop, as suggested in [7], [13]. RDL effects are simulated by adjusting P/G pad pitch to achieve IR drops of 95mV and 85mV for equal-area 3D designs on 28nm and 16nm technologies, respectively.

Table III details the data generation specifics: 2,080 physical designs and 4,120 architectural simulations for MAERI, alongside 1,616 and 824 for the Systolic Array. This work required a total of 1,888 hours executed on six servers, each powered by a 2.10-GHz Intel® Xeon® Gold 6230 processors with 64 cores.

We use two Train-Test Split methods. For datasets on power, performance, and runtime, test datasets include MAERI with 128 and 256 PEs and Systolic Arrays of 16×32, 32×16, and 32×32. For fixed-frequency datasets, such as dynamic IR drop and thermal data, test datasets include MAERI with 256 PEs and a 32×32 Systolic Array to ensure sufficient data in the train datasets.

C. Logic vs. Memory Size Decision

To facilitate comprehensive exploration of the mixed-node, mixed-area 3D IC design space, MIX-3D handles logic and memory tiers separately. Based on the user-defined architectural space, it generates all feasible mixed-node, mixed-area 3D IC configurations.

a) Logic Area Prediction: The synthesized logic area indicates the post-physical design logic core area and defines the minimum area of the memory tier. Due to the scalable nature of logic area with respect to PEs, linear regression is applied with a single input and output feature. Separate regression models are developed for designs across different logic tech nodes.

b) Memory Area and Capacity Decision: Following logic area estimation, the framework explores on-chip memory configurations from a predefined memory pool, ensuring the memory tier area meets or exceeds that of the logic tier. Our memory pool includes 16-bit and 32-bit bandwidths at various depths for single- and dual-port memories. The memory selection model assigns specific types to each on-chip component, allowing the Systolic Array to select up to three memory types, thereby expanding design options.

TABLE IV: Input and output features of our ML models. Masked VAEs utilize identical input and output dimensions.

Model	Arch.	Input & Output
Linear Regression Model		
Logic area	MAERI	In = PE, Out = area after synthesis
	S-array	In = ROW×COL, Out = area after syn.
Variational Encoder		
Runtime	Both	In = Out = PE, BW, workload layer parameters, total cycle
Performance	Both	In = Out = PE, BW, logic & mem. tech, target freq, effective freq.
Logic pwr	Both	In = Out = PE, BW, logic & mem. tech, target freq., logic power
Mem. pwr	Both	In = Out = PE, BW, logic & mem. tech, target freq. baseline mem. pwr., mem. power
IR drop	Both	In = Out = PE, BW, logic & mem. tech, logic core area, peripheral area, dynamic IR drop max
Thermal	Both	In = Out = PE, BW, logic & mem. tech, power, power density, max. chip temp.

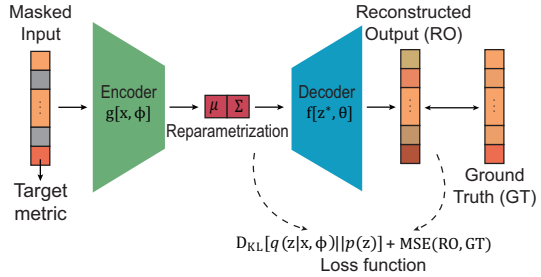

Fig. 5: The Masked VAE used in this work. The gray-shaded areas within the input represent randomly masked features during the training phase.

D. Masked Variational Autoencoder

The Variational Autoencoder (VAE) predicts key metrics, including power, performance, IR drop, maximum chip temperature, and runtime. Highly correlated design features are selected and concatenated with the respective target metric, resulting in a total of N dimensions of input and output, as shown in Table IV. This enables the VAE to learn the N-dimensional distribution, transforming the regression task into a generative task for predicting unseen design points.

Among these, a special case is the prediction of memory power consumption, which is influenced by memory type, connected logic size, and operating frequency. While common features handle logic size and frequency, memory type necessitates a new feature. Therefore, baseline memory power is obtained from physical designs with memory paired to inactive logic and integrated into our framework.

During inference for unseen designs, target metric values cannot be used as input. To address this, we implement target metric masking, as shown in Fig. 3. In this method, the VAE predicts the target metric based on the provided design features. Since the VAE has learned the N-dimensional data distribution from the train datasets, it can leverage this knowledge to estimate the value of the Nth coordinate when provided with the values of the preceding N-1 coordinates.

To ensure accurate predictions during inference, random masking is applied to the input during training, as depicted in Fig. 5. Masking can be introduced anywhere, provided that no more than half of input dimensions are masked. Consequently, we refer to this model as the masked VAE. This approach enhances the model's generalization capability by reducing the risk of relying on specific features and effectively augments the limited training data available.

TABLE V: Mean/Max APE of ML models trained from scratch.

Metric		MAERI	Sys. Array
Logic area		1.2% / 3.1%	0.8% / 1.1%
Runtime	ResNet-50	2.1% / 7.2%	4.6% / 9.7%
	MobileNet v1	1.7% / 7.8%	3.6% / 8.1%
	Transformer	2.5% / 7.4%	2.8% / 7.8%
Performance		1.1% / 5.2%	0.6% / 4.8%
Logic power		3.2% / 7.3%	2.2% / 6.9%
Memory power		2.8% / 6.5%	3.3% / 7.3%
Dynamic IR drop		3.7% / 7.1%	3.4% / 7.5%
Temperature		3.9% / 6.4%	2.9% / 5.1%

Algorithm 1 Masked VAE for regression problems. We use $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The feature dimension n is adjusted according to different target metrics.

Input: $\mathbf{x} = [\mathbf{F}, y]$: concatenation of regression input feature $\boldsymbol{F} \in \mathbb{R}^n$ and output $y \in \mathbb{R}$, $\mu_{\mathbf{z}}, \mathbf{\Sigma_z}$: trainable mean and covariance matrix of multivariate Gaussian posterior $q(\mathbf{z}|\mathbf{x})$, $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$: random variable for the reparameterization trick, θ : parameters of VAE, α : learning rate, β_1, β_2: Adam parameters.

Output: $\hat{\mathbf{x}} = [\hat{\mathbf{F}}, \hat{y}]$: reconstructed vectors from latent $\mathbf{z}$.

Training Phase:
for epoch in epochs **do**
  $\mathbf{masked_x} = \mathbf{x} \odot [0, \cdots 1]$ ▷ Random masking applied to $\mathbf{x}$
  $[\mu_{\mathbf{z}}, \mathbf{\Sigma_z}] = Encoder(\mathbf{masked_x})$
  $\mathbf{z} = \mu_{\mathbf{z}} + \mathbf{\Sigma_z}^{\frac{1}{2}} \odot \varepsilon.$ ▷ Reparameterization trick
  $\hat{\mathbf{x}} = Decoder(\mathbf{z})$
  $Loss = D_{KL}(\mathcal{N}(\mu_{\mathbf{z}}, \mathbf{\Sigma_z})||\mathcal{N}(\mathbf{0}, \mathbf{I})) + \parallel \mathbf{x} - \hat{\mathbf{x}} \parallel^2$
  $\{\theta, \mu_{\mathbf{z}}, \mathbf{\Sigma_z}\} \leftarrow Adam(\alpha, \{\theta, \mu_{\mathbf{z}}, \mathbf{\Sigma_z}\}, \beta_1, \beta_2, \nabla Loss)$
end for
Inference Phase:
  $\mathbf{masked_x} = \mathbf{x} \odot [1, \cdots 1, 0]$ ▷ Target metric masking
  $[\mu_{\mathbf{z}}, \mathbf{\Sigma_z}] = Encoder(\mathbf{masked_x})$
  $\mathbf{z} = \mu_{\mathbf{z}} + \mathbf{\Sigma_z}^{\frac{1}{2}} \odot \varepsilon.$
  $[\hat{\mathbf{F}}, \hat{y}] = Decoder(\mathbf{z})$ ▷ $\hat{y}$ represents the predicted metric.

The overall method is outlined in Algorithm 1. The encoder and decoder consist of Fully Connected (FC) layers with ReLU. Starting with an input expansion to 256 dimensions, the encoder reduces them sequentially to 128, 64, 32, and 4, while the decoder follows a reversed structure. We use mean and maximum absolute percentage error (Mean APE, Max APE) to evaluate each model, as summarized in Table V. Comparable error rates with [16] validate its effectiveness in learning distributions and accurately performing regression.

E. Transfer Learning and Foundation Model

One key advantage of the masked VAE is its support for transfer learning. Unlike conventional regression models, the neural network-based structure allows knowledge transfer across different architectures. Through evaluation of transfer learning between MAERI and Systolic Array architectures, we establish a criteria for selecting the foundation model. (1) Metric Prediction Complexity: The foundation model should handle challenging metric prediction tasks to ensure broad generalization. (2) Data Collection Efficiency: It should support rapid data collection from smaller design sizes for robust training. Systolic Array satisfies these criteria and is selected as the foundation

TABLE VI: Mean APE of each model with different sizes of train dataset.

Logic power prediction	**Trainset 1**	**Trainset 2**	**Trainset 3**
#PE in trainset	16, 32, 64	16, 32	16
#Data points	960	640	320
Trainset build time	425h	177h (-58%)	70h (-84%)
Poly(deg.=5) regression	3.1%	28.4%	57.7%
VAE w/o transfer	3.2%	17.4%	30.9%
VAE w transfer	3.1%	3.9%	9.0%

TABLE VII: Design space explored for MAERI & Systolic Array.

	MAERI	**Sys. Array**
Arch. config.	PE = 128, $\cdots$, 2048 BW = PE/4, PE/2	row x col = 16x16, $\cdots$ 128x128
3D (logic+mem)	28+28nm, 16+16nm, 28+16nm, 16+28nm	
Workload	ResNet-50, Transformer	
Objective	Energy-efficiency, Area-efficiency	
# Design points	142	8,672
Exploration time	5.67s	393.63s

architecture. The rigid structure results in smaller sizes, facilitating faster data collection. Additionally, predicting runtime with varying PE utilization presents a more challenging problem.

The strength of transfer learning is highlighted in Table VI. Beyond the original train dataset (trainset 1) with #PE = 16, 32, and 64, we reduce the size by limiting the number of #PE for the logic power prediction on MAERI benchmark. To ensure a fair comparison, all models are evaluated using the same number of test data.

Polynomial regression with degree 5, which achieves a mean error rate of 3.1%, exhibits a significant degradation in accuracy with reduced trainset. This steep drop stems from the inherent characteristics of regression, which demands at least three data points per feature to enable effective extrapolation. In contrast, the VAE model with transfer learning sustains consistent error rates even with a trainset 2. By drawing on trend related to the number of #PE from its foundation model, the VAE adeptly adjusts learned distributions to new domains using just two feature values, resulting in a remarkable 58% reduction in the power trainset construction time.

As a result, the transfer learning with foundation model of Systolic Array significantly streamlines overall trainset generation time for MAERI from 653 hours to 237 hours—a 64% drop. These findings underscore the value of a robust foundation model: once established, transfer learning facilitates rapid and efficient adaptation to diverse architectures, markedly enhancing scalability.

F. Cost Model

Design evaluations incorporate a cost perspective utilizing the analytical framework outlined in [1]. It delivers notable accuracy by accommodating precise material costs and yields validated by the foundry, along with a range of chip parameters. It supports both Wafer-to-Wafer (W2W) and Collective Die-to-Wafer (Co-D2W) hybrid bonding methodologies. W2W bonding is applied to equal-area designs and mixed-area designs with decap, leveraging consistent tier areas to integrate peripheral decap on the logic tier. In contrast, mixed-area designs that incorporate peripheral silicon rely on Co-D2W bonding to effectively join tiers with differing core dimensions. This tool can be seamlessly tailored to any other fabrication process by effortlessly incorporating parameters specific to alternative fabrication techniques.

TABLE VIII: Energy-efficient 3D Systolic Array on ResNet-50.

	Equal area	**Mixed area**
Logic spec.	28nm, 128x128	16nm, 128x128
Mem. spec.	16nm, $3MB$	28nm, $2.25MB$
E-eff ($TOPS/W$)	12.3	13.8 (+12.2%)
Max. frequency (GHz)	1.59	1.79 (+12.6%)
Power (W)	12.4	4.7 (-62.1%)
Max. temp. ($^{\circ}C$)	80.9	75.1 (-7.2%)
IR drop w. decap (mV)	95	75.3 (-20.7%)
W2W cost (w. decap)	1	0.647 (-35.3%)
Co-D2W cost (w. silicon)	-	0.454 (-54.3%)

TABLE IX: Area-efficient 3D MAERI under thermal constraint.

	Equal area	**Mixed area**
Logic spec.	16nm, 2048PE, 1024BW	
Mem. spec.	28nm, $0.5MB$	16nm, $4MB$
A-eff ($TOPS/W$)	0.35	0.20 (-42.9%)
Max. frequency (GHz)	1.94	2.07 (+6.7%)
Power (W)	18.5	17.6 (-4.9%)
Max. temp. ($^{\circ}C$)	116.1	71.7 (-38.2%)
IR drop w. decap (mV)	85	78.2 (-8.0%)
ResNet-50 Runtime (ms)	10.4	5.8 (-44.2%)
ResNet-50 Energy (mJ)	5.4	2.9 (-46.3%)

VI. Results and Discussion

A. Experimental Setup

We employ the design space outlined in Table VII. Based on the architectural configuration and 3D technology candidates, MIX-3D framework identifies 142 design points for MAERI and 8,672 for Systolic Array in a mixed-node, mixed-area setting through systematic exploration of on-chip memory combinations. This encompasses all possible designs within the user-defined pool, enabling the selection of the optimal design by evaluating back-end and front-end metrics and identifying the best one based on the objectives.

For each design point, MIX-3D evaluates target frequencies from 0.5GHz to 4GHz, returning maximum effective frequencies within a 5% region of interest. These maximum frequencies are then utilized to predict subsequent back-end and front-end metrics. Leveraging the rapid inference capabilities of the VAE model, the comprehensive design space exploration requires 5.7 seconds for MAERI and 393 seconds for the Systolic Array.

B. Benefits of Mixed-Area 3D ICs

This section presents the best design under ResNet-50 workload. Due to the space limit, one representative per objective is reported.

1) Improvement on Energy-Efficient 3D ICs: Table VIII illustrates the benefits of mixed-area allocation in enhancing energy efficiency and other key metrics. By employing 16nm logic, the effective frequency increased by 12.6% and power is lowered by 62.1%. Furthermore, optimizing the logic core and memory tier areas to 2.1 and 4.2 mm^2, respectively—compared to 6.25 mm^2 in the equal-area scenario—contributed to overall cost reduction. Besides that, ResNet-50 runtime and energy consumption showed moderate improvements, achieving 4.69ms and 2.84mJ, respectively. These overall improvements demonstrate that the equal-area constraint leads to sub-optimal designs by prioritizing configurations that use older tech nodes for the logic tier and advanced nodes for the memory tier to maximize memory capacity. In contrast, our mixed-area 3D ICs

TABLE X: Best 3D MAERI and Sys. Array for Transformer workload.

Best design	MAERI		Sys. Array	
	E-eff Mixed	A-eff Thermal	E-eff Mixed	A-eff Thermal
E-eff. ($TOPS/W$)	0.59	0.53	7.36	1.31
A-eff. ($TOPS/mm^2$)	0.023	0.028	1.22	9.49
Runtime (ms)	40.5	38.2	17.1	56.3
Energy (mJ)	22.7	24.8	6.36	27.1

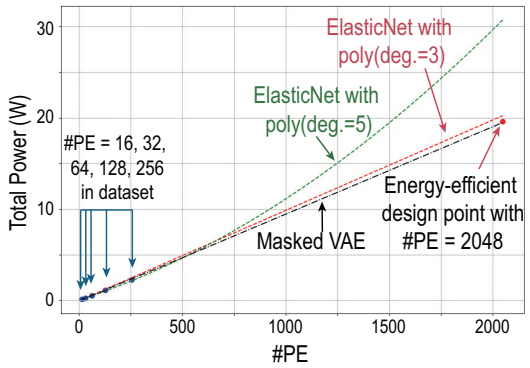

Fig. 6: Power prediction trend for MAERI. The masked VAE achieves a 0.7% error rate for an industrial-scale energy-efficient design, whereas the ElasticNet model with polynomial degree 5 exhibits overfitting.

leverage advanced technology nodes for the logic tier while providing ample memory capacity by utilizing a larger area for the memory tier.

Table VIII also reveals a trade-off between the inclusion of peripheral decap and silicon. Peripheral decap improves both IR drop and thermal, whereas silicon only contributes to temperature reduction. However, incorporating decap requires W2W bonding, which incurs higher costs than including silicon. Thus, designers can strategically select an appropriate method based on these trade-offs.

2) Area-efficient 3D ICs Under Thermal Constraint: MIX-3D enables consideration of heat dissipation, a critical challenge in 3D ICs. Table IX shows prior area-efficient design reaching impractical chip temperatures of 116°C. In contrast, our design using 16nm tech and expanding the memory tier area from 6.7 mm^2 to 12.4 mm^2 achieves a 38% temperature reduction. Although this increases costs by 1.3x and reduces area efficiency, it significantly improves frequency, power density, IR drop, runtime, and energy consumption.

C. Best Configuration for Transformer Workload

Transformer impacts metrics due to two key factors: the use of FC layers, which have lower data reuse than Conv layers, and complex Key-Value caching in Attention layers, which demands larger on-chip memory. Insufficient memory leads to more DRAM accesses, resulting in longer runtime and higher energy consumption as shown in the Table X. Furthermore, the multi-head attention mechanism alters the energy-efficient design on Systolic Arrays. The small hidden dimension of 64 per head reduces PE utilization, especially in large Systolic Arrays, impacting $TOPS$ and shifting the optimal design from 128×128 to 128×64. This trend is consistently appeared in modern LLMs like GPT-4 and LLaMA, where the hidden dimension per head typically remains within the 64-128 range [3], [20].

D. Strength of the masked VAE on extrapolation

Fig. 6 compares the performance of the conventional regression model from [16] with the masked VAE in predicting the total power

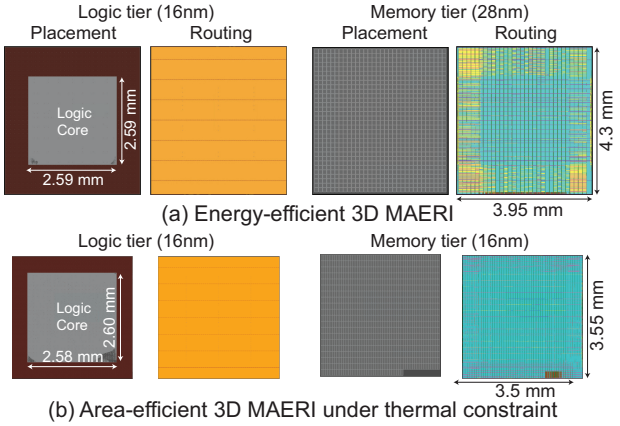

Fig. 7: Physical design of the best 3D MAERI configurations.

for MAERI at a fixed target frequency of 2.0 GHz. Previous work reported successful power prediction with polynomial regression (degree 5), achieving a mean error rate of 3.1% on the test dataset. However, this approach exhibits overfitting issues, particularly overestimating power consumption for larger design sizes, resulting in 56.6% error when predicting power for the energy-efficient design. In contrast, the masked VAE achieves a robust performance, maintaining a mean error rate of 3.2% on the test dataset and significantly outperforming the regression model in extrapolation, with only a 0.7% error for the prediction of energy-efficient design. This underscores the robustness of the masked VAE compared to conventional regression techniques in handling extrapolation without overfitting issues.

E. Layout-based Validation

Physical designs for energy- and area-efficient configurations are executed to verify the accuracy of MIX-3D framework, with final layouts shown in Fig. 7 for MAERI (those for Systolic Array are omitted due to the space limit). MIX-3D framework achieves max absolute percentage errors of 2.0% for effective frequency, 3.3% for power, 5.8% for maximum chip temperature, 5.1% for worst dynamic IR drop, and 4.5% for runtime. Considering the extensive design time of approximately 1,488 hours for large-scale configurations, the framework is essential for efficiently identifying optimal designs.

VII. Conclusion

This study introduced the MIX-3D framework, leveraging masked VAEs to identify optimal design configurations for 3D ICs. The masked VAE successfully transformed regression tasks into generative tasks, reducing overfitting compared to conventional regression models. Additionally, transfer learning reduces dataset construction time by 64%. With the masked VAE, MIX-3D revealed that mixed die area allocation significantly improved energy efficiency and chip temperature, underscoring its importance in 3D IC design. Based on the generalizability of the framework on different architectures and workloads, our work is expected to evolve into a scalable resource, expanding in scope after its initial deployment.

Acknowledgment

This work was supported by Semiconductor Research Corporation under the JUMP 2.0 Center Program (CHIMES 3136.002) and Samsung Advanced Institute of Technology (SAIT) under the AI for Semiconductors Program.

REFERENCES

[1] A. Agnesina, M. Brunion, J. Kim, A. Garcia-Ortiz, D. Milojevic, F. Catthoor, G. Mirabelli, M. Komalan, and S. K. Lim. Power, Performance, Area, and Cost Analysis of Face-to-Face Bonded 3D ICs. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 2023.

[2] L. Bamberg, A. García-Ortiz, L. Zhu, S. Pentapati, S. K. Lim, et al. Macro-3D: A Physical Design Methodology for Face-to-Face-Stacked Heterogeneous 3D ICs. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 37–42. IEEE, 2020.

[3] T. B. Brown. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*, 2020.

[4] H. Esmaeilzadeh, S. Ghodrati, A. B. Kahng, J. K. Kim, S. Kinzer, S. Kundu, R. Mahapatra, S. D. Manasi, S. S. Sapatnekar, Z. Wang, et al. Physically Accurate Learning-based Performance Prediction of Hardware-accelerated ML Algorithms. In *Proceedings of the 2022 ACM/IEEE Workshop on Machine Learning for CAD*, pages 119–126, 2022.

[5] W. Gomes, S. Khushu, D. B. Ingerly, P. N. Stover, N. I. Chowdhury, F. O'Mahony, A. Balankutty, N. Dolev, M. G. Dixon, L. Jiang, et al. 8.1 Lakefield and Mobility Compute: A 3D Stacked 10nm and 22FFL Hybrid Processor System in 12× 12mm 2, 1mm Package-on-Package. In *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 144–146. IEEE, 2020.

[6] W. Gomes, A. Koker, P. Stover, D. Ingerly, S. Siers, S. Venkataraman, C. Pelto, T. Shah, A. Rao, F. O'Mahony, et al. Ponte Vecchio: A Multi-Tile 3D Stacked Processor for Exascale Computing. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 65, pages 42–44. IEEE, 2022.

[7] M. B. Healy and S. K. Lim. Distributed TSV Topology for 3-D Power-Supply Networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(11):2066–2079, 2011.

[8] Hyoukjun Kwon, Ananda Samajdar, and Tushar Krishna. MAERI Project. https://github.com/maeri-project/MAERI_bsv, 2019. GitHub repository, accessed Nov 12, 2024.

[9] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] H.-T. Kung. *Why systolic architecture?* Design Research Center, Carnegie-Mellon University, 1982.

[11] H. Kwon, P. Chatarasi, V. Sarkar, T. Krishna, M. Pellauer, and A. Parashar. MAESTRO: A Data-Centric Approach to Understand Reuse, Performance, and Hardware Cost of DNN Mappings. *IEEE micro*, 40(3):20–29, 2020.

[12] H. Kwon, A. Samajdar, and T. Krishna. MAERI: Enabling Flexible Dataflow Mapping over DNN Accelerators via Reconfigurable Interconnects. *ACM SIGPLAN Notices*, 53(2):461–475, 2018.

[13] M. Li, P. Periasamy, K.-N. Tu, and S. S. Iyer. Optimized Power Delivery for 3D IC Technology Using Grind Side Redistribution Layers. In *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*, pages 2449–2454. IEEE, 2016.

[14] A. Malistov and A. Trushin. Gradient Boosted Trees with Extrapolation. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 783–789. IEEE, 2019.

[15] F. Muñoz-Martínez, J. L. Abellán, M. E. Acacio, and T. Krishna. STONNE: Enabling Cycle-Level Microarchitectural Simulation for DNN Inference Accelerators. In *2021 IEEE International Symposium on Workload Characterization (IISWC)*, pages 201–213. IEEE, 2021.

[16] G. Murali, M. G. Park, and S. K. Lim. 3DNN-Xplorer: A Machine Learning Framework for Design Space Exploration of Heterogeneous 3-D DNN Accelerators. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2024.

[17] A. Parashar, P. Raina, Y. S. Shao, Y.-H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. Emer. Timeloop: A Systematic Approach to DNN Accelerator Evaluation. In *2019 IEEE international symposium on performance analysis of systems and software (ISPASS)*, pages 304–315. IEEE, 2019.

[18] A. Samajdar, Y. Zhu, P. Whatmough, M. Mattina, and T. Krishna. SCALE-Sim: Systolic CNN Accelerator Simulator. *arXiv preprint arXiv:1811.02883*, 2018.

[19] Samajdar, Ananda and Joseph, Jan Moritz and Zhu, Yuhao and Whatmough, Paul and Mattina, Matthew and Krishna, Tushar. SCALE-Sim v2. https://github.com/scalesim-project/scale-sim-v2, 2020. GitHub repository, accessed Nov 12, 2024.

[20] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*, 2023.

[21] X. Yang, M. Gao, Q. Liu, J. Setter, J. Pu, A. Nayak, S. Bell, K. Cao, H. Ha, P. Raina, et al. Interstellar: Using Halide's Scheduling Language to Analyze DNN Accelerators. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 369–383, 2020.

[22] R. Zhang, M. R. Stan, and K. Skadron. Hotspot 6.0: Validation, Acceleration and Extension. *University of Virginia, Tech. Rep*, 2015.