FaStTherm: Fast and Stable Full-Chip Transient Thermal Predictor Considering Nonlinear Effects

Tianxiang Zhu[1], Qipan Wang[1,2], Yibo Lin[1,3,4*], Runsheng Wang[1,3,4], Ru Huang[1,3,4]
[1]School of Integrated Circuits, [2]Academy for Advanced Interdisciplinary Studies, Peking University, Beijing,
[3]Institute of Electronic Design Automation, Peking University, Wuxi,
[4]Beijing Advanced Innovation Center for Integrated Circuits
{txzhu, qpwang, yibolin, r.wang, ruhuang}@pku.edu.cn

Abstract

Full-chip transient thermal simulation, which is essential for solving pressing thermal issues, is time-consuming and resource-intensive, especially when nonlinear effects including temperature-dependent leakage power and thermal conductivity are considered. While many deep learning models have been proposed recently to accelerate transient thermal prediction, their long-term stability remains far from satisfactory due to severe error accumulation. In this paper, we focus on the long-term stability of efficient full-chip transient thermal prediction. We propose *FaStTherm*, a deep-learning-based full-chip transient thermal predictor, which learns low-dimensional linear dynamics in the latent space and enables 10,000× speedup compared with commercial simulator COMSOL. We further propose a novel combination of local and global stabilizing techniques to mitigate the error accumulation. Experimental results on a commercial chip design and real-world workloads demonstrate that the prediction error of *FaStTherm* is less than 5% of the full temperature range (5 Kelvin) for more than 15,000 consecutive time steps, which is 42-73× longer than previous studies, showing the excellent long-term stability of our method.

1 Introduction

With the increasing density of transistors in advanced technology nodes, modern integrated circuits are suffering from pressing thermal issues [1, 2]. The recent introduction of advanced packaging technologies, such as 3D stacking, further complicates the circumstances because of the increased thermal resistance [3, 4]. To tackle the increasingly serious thermal issues, thermal-aware design and dynamic thermal management (DTM) are of the essence, both in need of fast and accurate transient thermal simulation of the chip systems, as shown in Fig. 1(a). Transient thermal information will provide an essential guide for package design, thermal-aware sensor placement, testing, and so on [5], While common DTM techniques, such as dynamic voltage and frequency scaling and task mapping, require a fast transient thermal model to estimate the temperature online and perform control ahead of time [6–8]. While conventional finite element method (FEM) or finite difference method (FDM) based numerical simulators can offer accurate transient temperature results, they require long simulation time and a lot of computational resources even for small time horizons because of the large mesh quantities after discretization. To worsen the situation, higher chip temperature in advanced technology nodes and packages

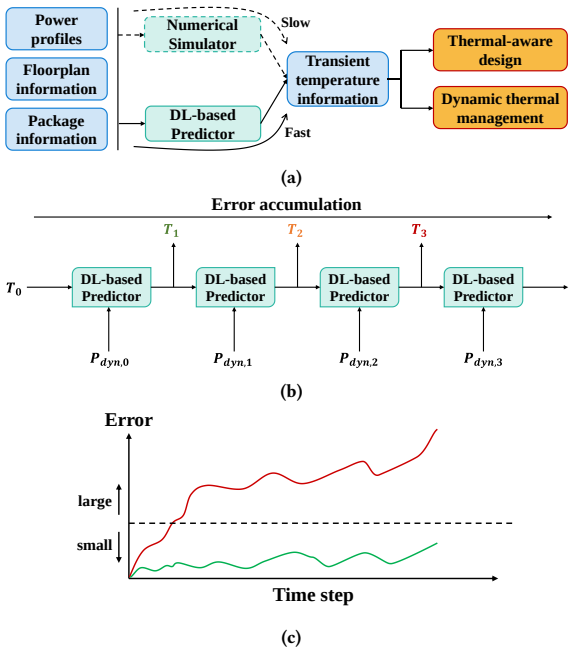


Figure 1: (a) Fast and accurate full-chip transient thermal simulations are needed to solve the pressing thermal issues. While traditional numerical simulators are time-consuming, DL-based predictors are widely studied to accelerate this process. (b) Illustration of error accumulation for DL-based full-chip transient thermal predictors. T_0 is the initial temperature. P_{dyn}s and Ts are the input dynamic powers and the predicted temperatures for each time step, respectively. Prediction results will quickly deviate from ground truth because of the vicious circle of error accumulation. (c) Diagram of two typical error plots which show severe (red) and mild (green) error accumulation, respectively.

results in greater influence of on-chip nonlinear thermal effects, including temperature-dependent leakage power and thermal conductivity. For example, leakage power can account for over 50% of the total power in some designs [9, 10], while temperature dependence of thermal conductivity can lead to an increase of 5 K in peak temperature [11, 12]. These nonlinear effects cannot be neglected during accurate transient thermal simulations [13], which can further increase the simulation time of numerical simulators. Meanwhile, these nonlinear effects hinder the direct utilization of fast analytical methods, such as model order reduction [14], Green's function [4], and separation of variables [15], because they are only applicable for linear systems theoretically.

To accelerate full-chip transient thermal prediction, a number of deep learning (DL) based methods have been proposed recently, with a variety of core models ranging from LSTM [16] to autoencoder [17]. These models achieve great speedup compared with traditional numerical simulators in various scenarios. However, DL-based methods

*Corresponding author

ICCAD '24, October 27–31, 2024, New York, NY, USA

ACM ISBN 979-8-4007-1077-3/24/10…$15.00
https://doi.org/10.1145/3676536.3676738

encounter long-term instability issues, especially when taking nonlinear effects into consideration. Existing methods are only able to output accurate prediction results for a small number of time steps due to severe error accumulation [16–18]. The concept of error accumulation is depicted in Fig. 1(b) and 1(c). For a black box DL model, every pushforward of one single time step brings a small error to the prediction result, which results in a distribution shift of the input for the next step, while the shifted input further incurs a larger error during the next pushforward. This vicious cycle will cause the prediction results to deviate from the ground truth quickly and even become thoroughly distorted. This fatal instability is one of the greatest impediments that keep existing DL-based full-chip transient thermal predictors from real applications.

Recently, some methods to improve the long-term stability of DL -based PDE predictors have been proposed [19–23]. Unfortunately, existing tricks are not enough for the task of full-chip transient thermal prediction. Firstly, these techniques can usually ensure stability for hundreds of time steps. However, for full-chip transient thermal prediction, we expect tasks with much longer traces in practical scenarios, such as real workloads running on a chip. Secondly, previous works mainly focus on general PDEs with no sources or fixed sources, while time-varying power inputs during transient thermal prediction are common in practice. Despite these difficulties, the governing PDE of the heat transfer system is relatively simple, which offers us opportunities to guarantee stability for a longer duration with more dedicated methods.

In this work, we propose a novel deep-learning model for full-chip transient thermal prediction with nonlinear effects including temperature-dependent leakage power and thermal conductivity. The new model, called *FaStTherm*, simultaneously achieves huge speedup compared with the commercial numerical simulator and unprecedented long-term stability compared with existing DL-based transient thermal predictors. The key contributions of this work are as follows:

- We develop a novel deep learning model inspired by Koopman theory, which is forced to learn low-dimensional linear dynamics of the original nonlinear systems in the latent space to realize fast thermal prediction.
- We further propose a novel combination of local and global techniques based on the linear latent space to stabilize the pushforward of our transient thermal predictor and ease the accumulation of error. An ablation study is carried out to verify the effectiveness of the proposed techniques.
- We test our model on a commercial chip design and real-world workloads. Experimental results demonstrate that 1) our model is 10,000× faster than commercial software COMSOL and 2) the prediction error of our model is less than 5% of the full temperature range (5 Kelvin) for more than 15,000 consecutive time steps, which is 42-73× longer than previous studies [16, 24].

The rest of this paper is organized as follows. Section 2 provides a review of relevant work. Section 3 formulates the studied problem. Section 4 elaborates on the whole framework of our model with the proposed techniques for increasing its stability during long-term prediction. Section 5 provides the experimental setup and the data generation flow. Section 6 presents the experimental results and comparisons with other methods, together with the ablation study. Section 7 concludes this paper.

2 Related Work

2.1 Numerical Methods

Traditionally, full-chip transient thermal simulations are performed by numerical solvers. Finite element method (FEM) based general-purpose commercial software, such as COMSOL [25] and Ansys [26], can offer ground truth transient temperature results and support arbitrary nonlinear effects. Meanwhile, some numerical solvers specially designed for IC thermal simulation are based on finite difference method (FDM), such as HotSpot [27] and PACT [28]. These customized solvers usually have limited support for nonlinear effects including temperature-dependent leakage power and thermal conductivity. All of these numerical methods require discretization of the systems and are very time- and resource-consuming because of the large quantities of meshes after discretization. Some fast analytical methods have been proposed to accelerate transient thermal simulation, including model order reduction [14], separation of variables [3], and Green's function [29]. However, these methods are only suitable for linear systems theoretically and require a large number of approximations when dealing with temperature-dependent leakage power and thermal conductivity [13]. Meanwhile, most of them have only been tested for simple scenarios such as constant dynamic powers.

2.2 Deep-Learning-based Methods

Deep-learning-based methods have recently been widely studied to perform fast full-chip transient thermal predictions and they achieve huge speedup compared with traditional numerical simulators. Chhabria *et al.* [16] regard transient thermal prediction as a sequence-to-sequence translation task and use convolutional encoder-decoder networks with LSTM to convert time-varying power maps into transient temperature maps. Ranade *et al.* [17] have proposed a FEM-like discretization-based method together with iteration techniques to improve stability and astringency. Echo state network (ESN) has been used to substitute the thermal model during DTM with consideration of nonlinear leakage power [24]. Besides, Kumar *et al.* [4] combine traditional Green's function method with DeepONet to achieve ultra-high resolution. A different technical route is featured by utilizing performance metrics of processors read from commercial software together with temperature information measured by IR camera to realize transient thermal map estimation [30–32]. This group of techniques can only transform real-time performance information into thermal maps, without the ability to perform thermal prediction into the future, which is outside the scope of this paper.

2.3 Long-Term Stability and Error Accumulation

Although current deep learning models achieve unprecedented speed up for full-chip transient thermal prediction compared with numerical simulators, they are fatally limited by the rapid deviation of the prediction results from the ground truth because of the vicious circle of error accumulation. While the method utilizing Green's function with DeepONet [4] suffers less from this issue because of linear error accumulation, it can only be applied to linear systems as well because of the utilization of Green's function. The lack of long-term stability prevents existing DL-based fast thermal predictors from practical applications and remains a crucial problem.

In the community of DL-based PDE predictors, error accumulation is a common issue for transient PDE prediction and some methods have been proposed recently to alleviate this effect. [33] proposes to train the prediction model with multiple steps instead of a single step to strengthen the robustness of the model during multiple-step prediction. Noise injection to training data and adversarial training are used in [19] and [21] to improve the model's resistivity against small disturbance, while [20] automates this process through a special pushforward training trick. Authors in [22] propose to focus more on the high spatial frequency components in PDE solutions to ease the effect of error accumulation. [23] identifies the intrinsic dimensions of the observed systems with geometric manifold learning algorithms and achieves robust prediction of the underlying dynamics.

Networks enhanced by these methods are usually able to guarantee stability for hundreds of time steps as shown by their results. However, the traces for predicting can be much longer in the context of full-chip transient thermal prediction, for example, when simulating the thermal behavior of real workloads running on the chip. Besides, these methods are usually tested on classical problems such as Navier-Stokes equations and mainly focus on the PDEs themselves with no source term or a fixed source term. Nevertheless, time-varying power inputs as source terms are important in IC thermal prediction. So a new network architecture together with novel techniques to improve its long-term stability is under exploration for the task of full-chip transient thermal prediction.

3 Problem Formulation

3.1 Governing Equation and Nonlinear Effects

The transient temperature distribution of the full chip is governed by the following heat equation [13]:

$$\rho c_p \frac{\partial T}{\partial t} - \nabla \cdot (\kappa \nabla T(\boldsymbol{r})) = q_v\,, \tag{1}$$

where T and q_v denote the temperature and the power dissipation per unit volume, and k, ρ, c_p are material-specific properties representing heat conductivity, mass density, and heat capacity.

In the context of integrated circuits, several parameters are temperature dependent, which makes Eq. 1 a nonlinear PDE. Firstly, the total power dissipation q_v in Eq. 1 consists of dynamic power P_{dyn} and leakage power P_{leak}. Dynamic power is contributed by logic gate switching, whose value doesn't depend on temperature, while leakage power increases exponentially with temperature. The relation can be written as [34]:

$$P_{leak}(T) = P_0 \cdot e^{\beta(T-T_0)}\,, \tag{2}$$

where P_0, β and T_0 are process-related parameters. Secondly, the thermal conductivity κ in Eq. 1 is also a function of temperature, which is given by [11]:

$$\kappa(T) = \kappa_0 (\frac{T}{300})^{-\eta}\,, \tag{3}$$

where κ_0 is the thermal conductivity at 300K and η is a material-specific constant.

The existence of these complex nonlinear (temperature-dependent) effects renders methods such as variable separation unsuitable because they are designed for linear systems.

3.2 Full-Chip Transient Thermal Prediction Considering Nonlinear Effects

Suppose there is an initial temperature distribution of the chip T_0 and a series of time-varying dynamic powers $P_{dyn,0}$, $P_{dyn,1}$, $P_{dyn,2}$, $\cdots$, $P_{dyn,n}$ generated by the active layer of the chip at each time step. Taking temperature-dependent leakage power and thermal conductivity into consideration, the task is to make a fast prediction on subsequent temperatures of the chip and we hope that the prediction results can be accurate enough for as many time steps as possible. The universal form of a full-chip transient thermal predictor can be written as:

$$T_{t+1} = f(T_t,\ P_{dyn,t})\,, \quad t \in \{0\} \cup \mathbb{N}, \tag{4}$$

where t denotes the discrete time step, T is the temperature of the chip, which is specifically the 2-D temperature map of the interested active layer following the convention [35], and P_{dyn} is the dynamic power. The predictor f takes the temperature and the dynamic power at time step t as inputs and outputs the temperature at time step t+1. The temperature-dependent leakage power $P_{leak}(T)$ and thermal conductivity $\kappa(T)$ are encoded in the mapping f, which can be learned by a deep neural network to perform fast thermal prediction.

However, inference error is inevitable for a neural network. During prediction, every pushforward of time step t brings some error to the predicted temperature, which serves as an input for the next step. Because the neural predictor is trained under a certain distribution of data, the input corrupted by error is outside the training distribution and will bring more substantial error to its next step, forming a vicious circle of error accumulation and rendering the prediction results inaccurate after only a few time steps. This is one of the main obstacles that keeps DL-based full-chip transient thermal predictors from practical use.

4 The FaStTherm Framework

In this section, we illuminate the framework of the proposed *FaStTherm* for full-chip transient thermal prediction, together with the local and global techniques proposed to improve its stability. After that, we introduce our automatic training data generation framework.

4.1 Learning Low-Dimensional Linear Dynamics in Latent Space

To achieve fast transient thermal prediction, it is natural to turn to model order reduction of the original system [36, 37]. Different from conventional numerical or statistical methods, a deep autoencoder can realize automatic dimension reduction in its latent space and benefits from the strong representation and generalization ability of deep neural networks, which is selected as the core of our transient thermal predictor, as shown in Fig. 2. Because the input and output temperature are in the form of 2-D maps in the context of IC thermal prediction, we use the popular CNN-based architecture ResNet-18 [38] to build our encoder and decoder. In practice, we substitute the downsampling layers of ResNet-18 with upsampling layers in the decoder and make small adjustments to match the shape of the chip temperature map.

Transient prediction is performed in the low-dimensional latent space of the autoencoder, and the predicted latent vectors, denoted as z_t, will be decoded into temperature maps T_t using the decoder part to get the final results. This process can be presented as:

$$\begin{aligned} z_{t+1} &= q(z_t,\ P_{dyn,t})\,, & T_{t+1} &= g(z_{t+1})\,, \\ z_{t+2} &= q(z_{t+1},\ P_{dyn,t+1})\,, & T_{t+2} &= g(z_{t+2})\,, \\ &\cdots & &\cdots \end{aligned} \tag{5}$$

where g denotes the decoder and q denotes the dynamics of z in the latent space, and the dimension of z is far less than the dimension of T, enabling fast thermal prediction. However, as discussed before, the pushforward of z in the latent space suffers from error accumulation because the mapping q is not error-free, which will lead to a rapid deviation of the latent vectors and then corrupt the prediction results.

Compared with nonlinear systems, linear systems enjoy mature theories of stability and error control, which will bring us the potential for mitigating the accumulation of error [39, 40]. According to Koopman theory [41], a nonlinear system can be mapped into a linear system under a certain set of transformation functions, which can be learned by the encoder h. This means we can force the autoencoder to learn a linear q in the latent space. Eq. 5 can now be rewritten as:

$$\begin{aligned} z_{t+1} &= Az_t + BP_{dyn,t}\,, & T_{t+1} &= g(z_{t+1})\,, \\ z_{t+2} &= Az_{t+1} + BP_{dyn,t+1}\,, & T_{t+2} &= g(z_{t+2})\,, \\ &\cdots & &\cdots \end{aligned} \tag{6}$$

where A and B are linear matrices and can be represented by a trainable fully connected layer without bias and activation function. The inference process of the proposed model is concluded in Fig. 3.

During training, the ground truth temperature maps at time step t, $t+1$ and the input dynamic power at time step t are combined as tuples $(T_t, T_{t+1}, P_{dyn,t})$. Three types of loss functions are included. Firstly, the reconstruction loss L_{re} guarantees the reconstruction ability of the

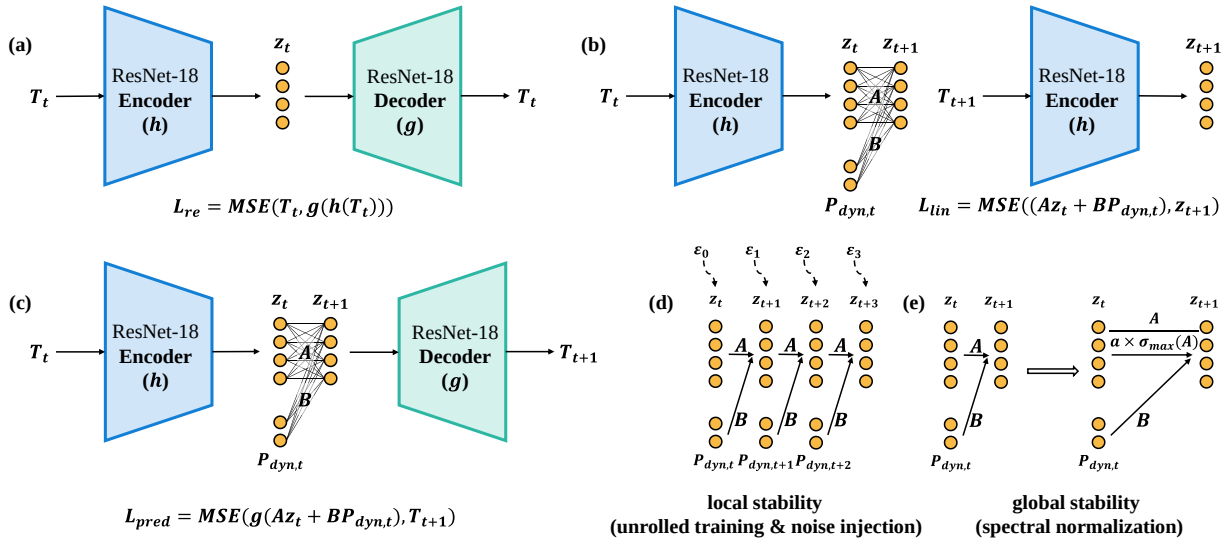


Figure 2: Diagram of the proposed *FaStTherm* for full-chip transient thermal prediction. The model is based on a ResNet-18-based deep autoencoder, which is forced to learn low-dimensional linear dynamics in the latent space. We show the calculation of (a) reconstruction loss, (b) prediction loss in the linear latent space, and (c) prediction loss of the temperature. The techniques for enhancing (d) local stability and (e) global stability are also illustrated.

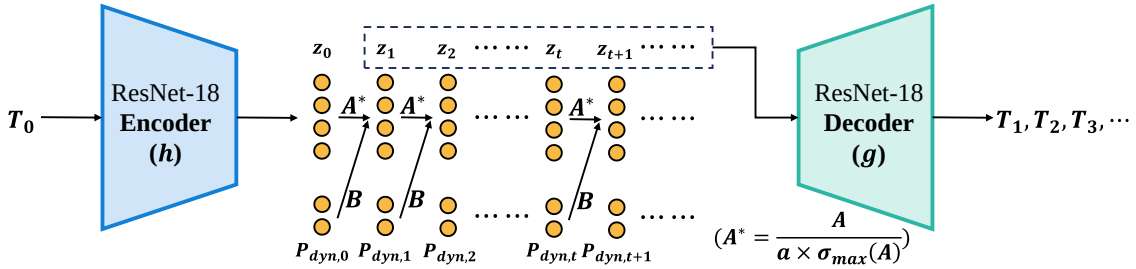


Figure 3: The inference process of the proposed model. The initial temperature map is first encoded into a latent vector by the encoder. Transient prediction is then performed in the latent space step by step and the predicted latent vectors are finally decoded into temperature maps by the decoder. The linear matrix A is spectral normalized as introduced in Section 4.3.

autoencoder. Secondly, the prediction loss in the latent space L_{lin} forces the model to learn the linear dynamics of the latent vectors. Thirdly, the prediction loss of the temperature L_{pred} is added to improve the final prediction performance. We use mean square error (MSE) as the loss function. These losses are concluded below:

$$L_{re} = MSE(T_t,\ g(h(T_t)))\,, \tag{7}$$

$$L_{lin} = MSE((Az_t + BP_{dyn,t}),\ z_{t+1})\,, \tag{8}$$

$$L_{pred} = MSE(g(Az_t + BP_{dyn,t}),\ T_{t+1})\,, \tag{9}$$

where

$$z_t = h(T_t)\,, \quad z_{t+1} = h(T_{t+1})\,, \tag{10}$$

are latent vectors converted from temperature maps by encoder h. The final training loss is the combination of the three types of losses described above:

$$L = \alpha_1 L_{re} + \alpha_2 L_{lin} + \alpha_3 L_{pred}\,, \tag{11}$$

where α_1, α_2 and α_3 are hyperparameters.

The schematic diagrams of the three types of losses depicted above are presented in Fig. 2(a)(b)(c), respectively. In the next two subsections, we will elaborate on the novel techniques we propose to stabilize the pushforward of our transient thermal predictor.

4.2 Unrolled Training and Noise Injection for Local Stability

As discussed above, one of the main reasons for rapid error accumulation is that the previous predicted result corrupted with error and serving as the next input is outside the training distribution and will bring more substantial error to its next step. Then a natural way to increase the stability of the predictor is to train the model with scenarios possibly encountered during the test. We adopt unrolled training and noise injection to imitate the test scenarios to increase the local stability of our predictor, as shown in Fig. 2(d).

For unrolled training, we enforce prediction over m consecutive steps rather than one single step by modifying the training losses L_{lin} and L_{pred} in Eq. 8 and 9 as follows:

$$L_{lin} = \sum_{i=0}^{m-1} MSE((Az_{t+i} + BP_{dyn,t+i}),\ z_{t+i+1})\,, \tag{12}$$

$$L_{pred} = \sum_{i=0}^{m-1} MSE(g(Az_{t+i} + BP_{dyn,t+i}),\ T_{t+i+1})\,. \tag{13}$$

where m is a hyperparameter decided by experiments. Specifically, we choose $m = 4$ for our model to balance the one-step error and multi-step stability.

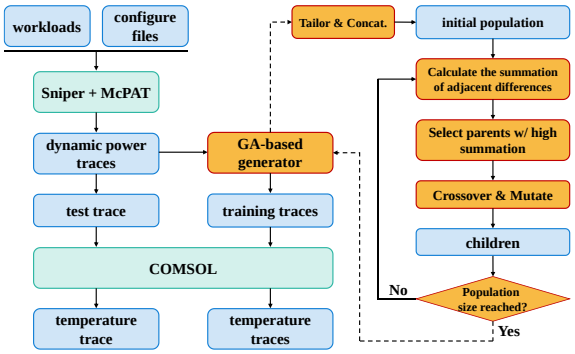


Figure 4: The complete flow for data generation together with the detailed diagram of our GA-based training data generation framework.

For noise injection, we corrupt each z_t with random-walk noise $\mathcal{N}(0, \sigma)$ during the above-mentioned unrolled training, where the standard deviation σ is a constant hyperparameter decided by experiments [19]. This process can be simply fulfilled by replacing z_{t+i} with $z_{t+i} + \varepsilon_i$ in Eq. 12 and 13, where ε_i is a series of random-walk noise following normal distribution. Specifically, we choose $\sigma = 0.001$ in our model to make the noise account for a reasonable part of z_t.

4.3 Spectral Normalization for Global Stability

While many works in the community of neural transient PDE predictors adopt similar local techniques discussed above and successfully ensure stability for hundreds of time steps, they are not enough for the practical use of a full-chip transient thermal predictor because it is likely to encounter much longer power and temperature traces when dealing with real-world workloads. For traces in the order of tens of thousands of time steps or more, methods to reduce one-step or multi-step errors are powerless to guarantee global stability. Under this consideration, we propose a novel method to realize the damping of previously introduced errors to increase the global stability of ultra-long traces, utilizing the linear dynamics learned in the latent space.

Consider an arbitrary transient linear system:

$$x_{t+1} = Ax_t + Bu_t\,, \quad t \in \{0\} \cup \mathbb{N}\,, \tag{14}$$

where x is the state variable and u is the source input (z and P_{dyn} in our model, respectively). If at some certain step m, a noise ε is introduced to the state variable, that is:

$$x_m \longrightarrow x_m + \varepsilon\,, \tag{15}$$

then at time step $n > m$ the state variable corrupted by the noise will be:

$$x_n = Ax_{n-1} + Bu_{n-1} + A^{n-m}\varepsilon\,. \tag{16}$$

According to linear algebra [42], only if the maximum singularity (or the spectral norm) of the dynamics matrix A is less than unity, the term $A^{n-m}\varepsilon$ in Eq. 14 will not diverge but damp to zero as $n \to \infty$, which means error generated at some step will have no influence on the global stability, but only affect its neighbors.

According to the above analysis, we propose to apply spectral normalization to the linear layer in the latent space to restrict its max singularity to less than unity and guarantee the global stability of our predictor, as shown in Fig. 2(e). Specifically, we will rewrite Eq. 6 as:

$$z_{t+1} = \frac{A}{a * \sigma_{max}(A)} z_t + BP_{dyn,t}\,, \quad T_{t+1} = g(z_{t+1})\,, \tag{17}$$

where a is a hyperparameter more than unity and $\sigma_{max}(A)$ is the max singularity of matrix A. Spectral normalization is a common technique used in generative adversarial networks (GAN) to constrain the Lipschitz continuity of the discriminator [43, 44]. For the first time, we apply it to neural PDE predictors to realize error damping against error accumulation, utilizing the linearity of the dynamics learned in the latent space.

With the combination of local and global stabilizing techniques, our transient thermal predictor is able to generate prediction results with sufficiently small errors for much more time steps.

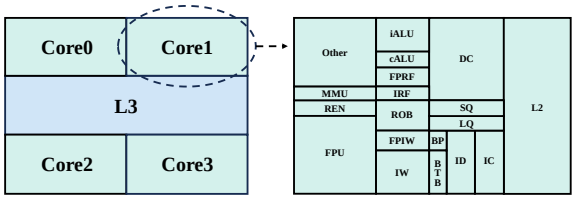


Figure 5: The schematic diagram of the processor's floorplan used for thermal simulation. Dynamic powers simulated by Sniper and McPAT are assigned to corresponding components and locations (eg: Core0-FPU) during thermal simulation.

Training	*swaptions, blackscholes, water.sp, fluidanimate, ocean.cont, fft*
Testing	*bodytrack, canneal, dedup, streamcluster, x264 barnes, cholesky, fmm, lu.cont, lu.ncont, radix, radiosity, ocean.ncont, raytrace, water.nsq*

Table 1: Applications selected for generating training and testing data [46, 47].

4.4 Automatic Training Data Generation

A diversified training set is essential for the generalization performance of a deep learning model. Previous studies use synthetic dynamic powers [16, 45] or dynamic powers of realistic workloads [24, 32] as training data. The former lacks in practical significance for real applications, while the latter is restricted by the regular and limited patterns of real workloads [48]. We avoid these shortcomings by designing a generic algorithm (GA) based flow [48, 49] to generate the diversified training set automatically, as shown in Fig. 4.

To begin with, we choose a set of real-world benchmarks and perform micro-architecture simulations to get their dynamic power traces as the initial popularization. Then we calculate the adjacent differences of the power traces and select the ones with the highest adjacent differences as "parents", which are then crossed over and mutated to create "children". This process is looped until the population size is reached. We then tailor and concatenate the dynamic power traces generated in this process into ones of equal length, and perform thermal simulations to get the corresponding temperature traces. Within the above framework, the training set is automatically generated and the dynamic power traces are diversified in both variational and absolute amplitude.

5 Experimental Setup

In this section, we first present the experimental setup and the data generation flow. After that, we introduce the baselines for comparison and the setup of the ablation study.

5.1 Simulated System and Data Generation Flow

To validate our model for practical applications, we use a commercial Intel quad-core microprocessor with Gainestown architecture as the simulated system. Each core has a 16 KB private L1 instruction cache and a 16 KB private L1 data cache, alongside a 512 KB private L2 cache. All the cores share an 8 MB L3 cache. The diagram of the processor's floorplan used for thermal simulations is shown in Fig. 5, which is extracted from the documents of HotSniper [51], an open-source EDA toolchain integrating Sniper [52] and HotSpot [27] for performance and thermal simulation.

Table 2: Performance comparison between our model and the baselines for the prediction of 15,000 consecutive time steps. Our model is *FaStTherm*. *ESN* [24, 50], *LSTM* [16], and *U-Net* [16, 22] are for comparison study, while *FaStTherm-w/o-global* and *FaStTherm-w/o-local* are for ablation study. "K" refers to "Kelvin".

	Average MAE			Survival time steps (error below 5 K)	Average prediction time
	total 15,000 time steps	the first 500 time steps	the last 500 time steps		
ESN [24, 50]	8.134 K	5.201 K	8.206 K	243	0.546 ms/step
LSTM [16]	8.366 K	4.796 K	8.036 K	351	0.723 ms/step
U-Net [16, 22]	12.168 K	7.717 K	12.337 K	205	1.108 ms/step
FaStTherm-w/o-global	7.320 K	5.705 K	7.332 K	152	0.387 ms/step
FaStTherm-w/o-local	3.513 K	2.719 K	3.993 K	472	0.392 ms/step
FaStTherm (Ours)	2.244 K	2.008 K	2.520 K	> 15,000	0.390 ms/step

Since HotSpot does not support nonlinear leakage power and thermal conductivity for transient thermal simulation, we cannot use HotSniper directly to fulfill the full-chain data generation. Instead, we use Sniper [52] and McPAT [53] to generate dynamic power traces from the simulation of real-world workloads. Then we feed the dynamic power traces into the COMSOL model we build to perform transient thermal simulation with nonlinear leakage power and thermal conductivity. The geometric dimensions of the heat sink and heat spreader, chip thickness, convection coefficient, and other related parameters in the COMSOL model are copied from the HotSpot configuration file in previous work [51] to guarantee the authenticity of the simulation results. Temperature-dependent leakage power and thermal conductivity are set in the COMSOL model, with parameters in Eq. 2 and 3 referring to previous works as well [24, 37, 54]. The ambient temperature is set to 318.15 K. The dimensions of the 2-D temperature maps in our experiments are 88×64, with one pixel being $0.049mm \times 0.049mm$ in size. The complete flow for data generation is shown in Fig. 4.

The interval of time step for transient thermal prediction is another parameter to be decided. Short intervals will result in an excess of time steps while long ones will lead to the omission of details in power traces and then temperature traces. We choose 10ms as the interval of time step in our work according to simulation results, which also serves as the common operating granularity for Linux scheduler and dynamic thermal management [55].

For workloads, we employ 21 applications from PARSEC [46] and SPLASH -2 [47] benchmark suites. These open-source benchmark suites have realistic multithreaded applications, designed to cover a wide range of different domains such as scientific computation, financial analysis, and video encoding. To test the generalization ability of our predictor for practical use, applications used for generating the training set and those for generating the testing set are kept strictly different and separate, as listed in Table 1. For some applications with asymmetrical master and slave threads, we map their master threads to different cores during thermal simulation to generate various temperature maps.

The training set is generated by the GA-based framework introduced in Section 4.4 and composed of 10 pairs of dynamic power traces and the corresponding temperature traces, each with 300 consecutive time steps (30s). The testing set comprises one dynamic power trace and the corresponding temperature trace with 15,000 consecutive time steps (150s), generated by running the testing applications in random order and with random core mappings to simulate a practical scenario. This trace is much longer than the training traces and serves as an arduous trial for the stability of the predictors under test.

5.2 Baselines for Comparison and Ablation Study

We choose three recently proposed full-chip thermal predictors, i.e., *ESN*-based [24, 50], *LSTM*-based [16] and *U-Net*-based [16, 22], as baselines for comparison. For the sake of fairness, we enhance the original *ESN* and *LSTM* with our ResNet-18-based decoder described in Section 4.1 to adapt to the high resolution of temperature maps. The *U-Net* based predictor is originally used for static thermal prediction and is adjusted to be fit for transient thermal prediction by fusing the dynamic power to the deepest features. We adopt this model as one of the baselines because our model, *ESN* and *LSTM* all make predictions in the latent space and then decode the latent vectors into temperature maps. The *U-Net* based predictor directly takes the present temperature map and dynamic power as inputs and predicts the temperature map at the next step, serving as a contrast with other models.

For the ablation study, we test two variants of our model to verify the techniques proposed to increase stability. One is the original model without adopting the global technique and the other without adopting the local technique. We denote the variants as *FaStTherm-w/o-global* and *FaStTherm-w/o-local*, respectively.

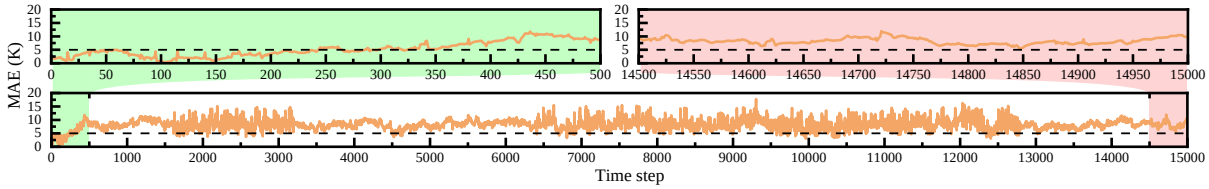
6 Experimental Results and Discussions

In this section, we will present and discuss the experimental results. Thermal simulations are performed on a cloud server with an Intel Xeon Platinum 8350C 2.60GHz processor (64 virtual cores). The thermal simulation of the training dataset costs about 4 hours, while that of the testing dataset with 15,000 time steps costs 16 hours and 27 minutes. All of the deep learning models are implemented within a Pytorch 1.12.1 framework, and their training and testing are performed on a single NVIDIA GeForce RTX2080Ti GPU. The training run-times are not exceeding 3 hours for each of the models.

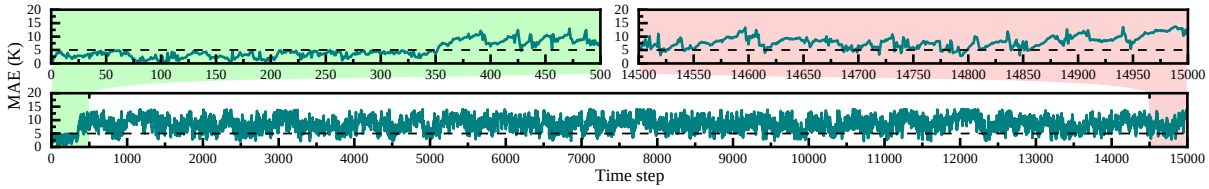
6.1 Main Comparison

We perform the comparison study of our model with the baselines *ESN*, *LSTM*, and *U-Net* on the test trace of 15,000 consecutive time steps in this subsection. Firstly, we inspect the average prediction time per step for each model, as listed in Table 2. Our model shows an outstanding prediction speed among the baselines because transient prediction is performed in the low-dimensional linear latent space in our model. Since the average simulation time using COMSOL is 3.948 s/step for the test trace, the speedup of our model compared with COMSOL is about 10,000×. We also run the simulation with the widely used open-source numerical thermal simulator HotSpot, which does not support nonlinear effects for transient thermal simulations. For the same system without nonlinear effects, the average prediction time of HotSpot is about 500 ms/step, and the average deviation from COMSOL is about 8 K. Special iteration methods have been proposed to include nonlinear effects in the framework of HotSpot, which will lead to substantial extra simulation time [13, 56]. The result shows that our DL-based predictor can realize great speedup compared with HotSpot as well.

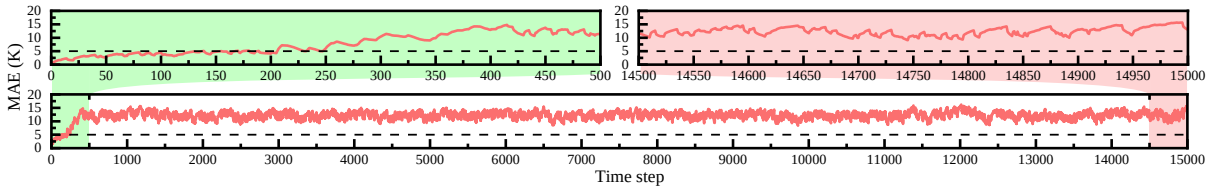
The long-term stability of the baselines and the proposed *FaStTherm* is then thoroughly studied. We calculate the Mean Average Error (MAE) of the predicted temperature map $T_{predict}$ against the ground truth temperature map T_{truth} at each time step t, that is:



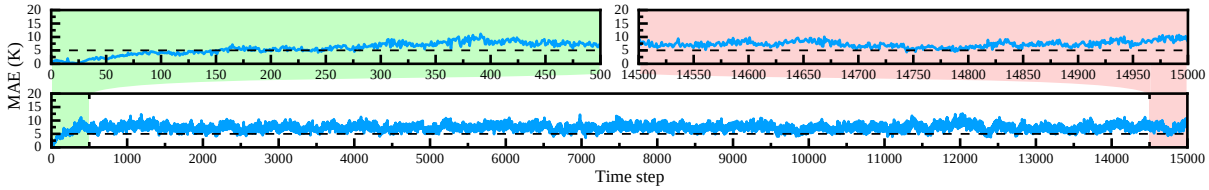
(a) ESN [24, 50]



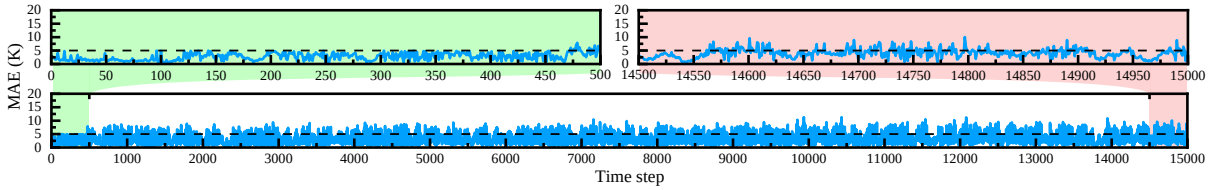
(b) LSTM [16]



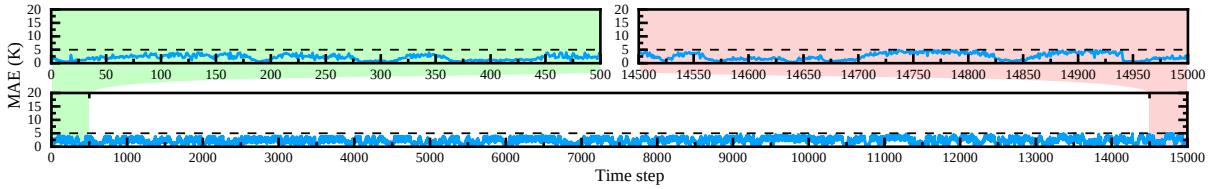
(c) U-Net [16, 22]



(d) FaStTherm-w/o-global



(e) FaStTherm-w/o-local



(f) FaStTherm (Ours)

Figure 6: (a), (b), and (c) show results of the baselines for comparison study while (d) and (e) are for ablation study. (f) shows the result of our model. The horizontal axis stands for time steps and the vertical axis stands for the Mean Absolute Error (MAE) between the predicted temperature map and the ground truth for each time step. For each model, the first and the last 500 time steps are zoomed in to show the effect of error accumulation at the initial stage and the final stage.

$$\mathrm{MAE}_t = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left| T_{predict,ijt} - T_{truth,ijt} \right|, \tag{18}$$

where m and n are the dimensions of the temperature map (88×64 in this work). To study the influence of error accumulation on the predictors' accuracy, the average values of the MAEs across the total 15,000 time steps, the first 500 time steps, and the last 500 time steps are calculated and concluded in Table 2, respectively. It is clearly shown that not only the average MAE of our model is the smallest among the baselines, but also the increase of error from the first 500 time steps to the last 500 time steps is the smallest, indicating the hint of milder error accumulation. To quantify the long-term stability of the transient thermal predictors, we define the survival time steps of a predictor as the time step until which the prediction error exceeds a pre-defined threshold for the first time [20, 22]. In our experiment, the minimum temperature in the test trace is 318.15 K and the maximum temperature is 418.32 K, so the full range of temperature is about 100 K. Regarding 5% of the full temperature range (5 K) as the threshold between small and large errors [32], the survival time steps of the baseline predictors are in the order of hundreds as shown in Table 2. In contrast, the MAEs of the prediction results of our model keep below the 5% threshold until 15,000 time steps, which is 42-73× longer than previous studies.

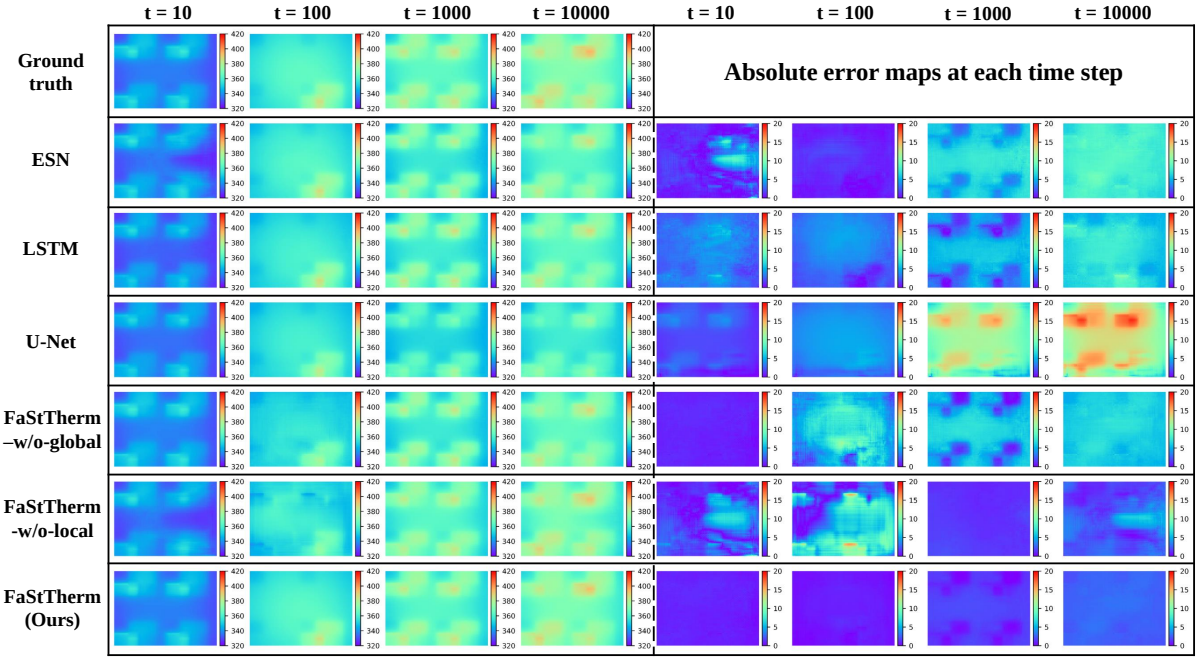


Figure 7: The ground truth temperature maps, the predicted temperature maps (left side), and the absolute error maps (right side) at time step 10, 100, 1000, and 10000 for each predictor. (Unit: Kelvin)

To further investigate the behavior of the baseline predictors and the proposed *FaStTherm*, we plot the calculated MAEs in Fig. 6, where the first 500 time steps and the last 500 time steps are zoomed in for more intuitive visualization of the error accumulation effect. As clearly shown in Fig. 6(a), 6(b), and 6(c), there exists a sharp increase in error (MAE) at the initial stage for *ESN*, and the same goes for *LSTM* and *U-Net*. After that sharp increase, the error remains at a high level until the end. Conversely, our model displays excellent long-term stability during prediction, with no sharp increase at the initial stage and the error remains at a low level for the whole horizon, as shown in Fig. 6(f). As indicated by a detailed inspection of the first 500 time steps, the error of our model goes down when reaching a local maximum, exhibiting a behavior similar to oscillation instead of increasing monotonically. We attribute this behavior mainly to the introduction of the global stabilizing technique, that is, spectral normalization. The spectral normalization of the dynamics matrix prevents the error in the predicted latent vectors from divergence, thus restraining the trend of monotone increase in the final error.

For better visualization of the above results, the ground truth temperature maps, the predicted temperature maps, and the absolute error maps at time steps 10, 100, 1000, and 10000 for each predictor are presented in Fig. 7. While the errors of the baseline predictors show an obvious growth over time, our model remains accurate and only shows a slight trend of increase in error.

6.2 Ablation Study

To verify the effectiveness of the proposed techniques for improving local and global stability, we perform an ablation study and test two variants of our model, *FaStTherm-w/o-global* and *FaStTherm-w/o-local*, as introduced in Section 5.2. The calculated MAEs are plotted in Fig. 6(d) and 6(e) with the first and the last 500 time steps zoomed in as well. The predicted temperature and absolute error maps are also displayed in Fig. 7. For the variant only with the local technique and without the global technique, *FaStTherm-w/o-global*, it is evident from the first 500 time steps in Fig. 6(d) that a sharp increase in the error reappears, which is similar to the behavior of the baseline predictors studied in Section 6.1. This further verifies that the mitigation of error accumulation of our transient thermal predictor mainly benefits from the novel spectral normalization technique for global stabilization.

On the other hand, for the variant with only global technique and without local techniques, *FaStTherm-w/o-local*, there does not exist a sharp increase in the error at the initial stage as expected. However, the local oscillation of the MAEs for *FaStTherm-w/o-local* significantly increases in amplitude due to the lack of local stabilizing techniques, resulting in a decrease in accuracy as indicated by the average values of the MAEs concluded in Table 2 and occasional violations of the 5% threshold as shown in Fig. 6(e). The results manifest that none of the local and global stabilization techniques is dispensable to guarantee the long-term stability of our full-chip transient thermal predictor.

7 Conclusion

In this work, we propose a novel deep learning model, called *FaStTherm*, for fast and stable full-chip transient thermal prediction considering nonlinear effects including temperature-dependent leakage power and thermal conductivity. We focus on the long-term stability of the proposed model and put forward a set of techniques to mitigate the effect of error accumulation. Our model learns low-dimensional linear dynamics in the latent space, which enables 10,000× speedup for transient thermal prediction compared with commercial simulator COMSOL. On this basis, we further propose to use unrolled training and noise injection to enhance local stability and spectral normalization to enhance global stability. Experimental results on a commercial chip design and real-world workloads show that our model exhibits excellent long-term stability. The prediction error (in MAE) of *FaStTherm* stays below 5% of the full temperature range (5 K) for more than 15,000 consecutive time steps, which is 42-73× longer than previous studies. In the future, we will investigate how to further enhance the long-term stability of our full-chip transient thermal predictor, and extend it to more complicated and diversified scenarios.

Acknowledge

This work was supported in part by the National Science Foundations of China (Grant No. 62125401, 62034007), the Natural Science Foundation of Beijing, China (Grant No. Z230002) and the 111 project (B18001).

References

[1] M. Pedram and S. Nazarian, "Thermal modeling, analysis, and management in vlsi circuits: Principles and methods," *Proceedings of the IEEE*, vol. 94, no. 8, pp. 1487–1501, 2006.

[2] L. Zhu and S. K. Lim, "Design automation needs for monolithic 3d ics: Accomplishments and gaps," in *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2023, pp. 1–4.

[3] S. K. Samal, S. Panth, K. Samadi, M. Saedi, Y. Du, and S. K. Lim, "Fast and accurate thermal modeling and optimization for monolithic 3d ics," in *Proceedings of the 51st Annual Design Automation Conference*, 2014, pp. 1–6.

[4] A. Kumar, N. Chang, D. Geb, H. He, S. Pan, J. Wen, S. Asgari, M. Abarham, and C. Ortiz, "Ml-based fast on-chip transient thermal simulation for heterogeneous 2.5 d/3d ic designs," in *2022 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*. IEEE, 2022, pp. 1–8.

[5] H. He, N. Chang, J. Yang, A. Kumar, W. Xia, L. Lin, and R. Ranade, "Solving fine-grained static 3dic thermal with ml thermal solver enhanced with decay curve characterization," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 2023, pp. 1–7.

[6] X. Guo, H. Wang, C. Zhang, H. Tang, and Y. Yuan, "Leakage-aware thermal management for multi-core systems using piecewise linear model based predictive control," in *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, 2019, pp. 64–69.

[7] M. Rapp, M. B. Sikal, H. Khdr, and J. Henkel, "Smartboost: Lightweight ml-driven boosting for thermally-constrained many-core processors," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 265–270.

[8] Y. Shen, S. Niknam, A. Pathania, and A. D. Pimentel, "Thermal management for s-nuca many-cores via synchronous thread rotations," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2023, pp. 1–6.

[9] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach*. Elsevier, 2011.

[10] Q. Xie, X. Lin, Y. Wang, S. Chen, M. J. Dousti, and M. Pedram, "Performance comparisons between 7-nm finfet and conventional bulk cmos standard cell libraries," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 761–765, 2015.

[11] Y. Yang, Z. Gu, C. Zhu, R. P. Dick, and L. Shang, "Isac: Integrated space-and-time-adaptive chip-package thermal analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 1, pp. 86–99, 2006.

[12] L. Chen, J. Lu, W. Jin, and S. X.-D. Tan, "Fast full-chip parametric thermal analysis based on enhanced physics enforced neural networks," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 2023, pp. 1–8.

[13] H. Sultan and S. R. Sarangi, "Varsim: A fast and accurate variability and leakage aware thermal simulator," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.

[14] L. Codecasa, D. D'Amore, and P. Maffezzoni, "An arnoldi based thermal network reduction method for electro-thermal analysis," *IEEE Transactions on Components and Packaging Technologies*, vol. 26, no. 1, pp. 186–192, 2003.

[15] Y. Zhan and S. S. Sapatnekar, "A high efficiency full-chip thermal simulation algorithm," in *ICCAD-2005. IEEE/ACM International Conference on Computer-Aided Design, 2005.* IEEE, 2005, pp. 635–638.

[16] V. A. Chhabria, V. Ahuja, A. Prabhu, N. Patil, P. Jain, and S. S. Sapatnekar, "Thermal and ir drop analysis using convolutional encoder-decoder networks," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, 2021, pp. 690–696.

[17] R. Ranade, C. Hill, H. He, A. Maleki, N. Chang, and J. Pathak, "A composable autoencoder-based iterative algorithm for accelerating numerical simulations," *arXiv preprint arXiv:2110.03780*, 2021.

[18] K. Zhang, A. Guliani, S. Ogrenci-Memik, G. Memik, K. Yoshii, R. Sankaran, and P. Beckman, "Machine learning-based temperature prediction for runtime thermal management across system components," *IEEE Transactions on parallel and distributed systems*, vol. 29, no. 2, pp. 405–419, 2017.

[19] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, "Learning to simulate complex physics with graph networks," in *International conference on machine learning*. PMLR, 2020, pp. 8459–8468.

[20] J. Brandstetter, D. E. Worrall, and M. Welling, "Message passing neural pde solvers," in *International Conference on Learning Representations*, 2021.

[21] A. Mayr, S. Lehner, A. Mayrhofer, C. Kloss, S. Hochreiter, and J. Brandstetter, "Boundary graph neural networks for 3d simulations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, 2023, pp. 9099–9107.

[22] P. Lippe, B. Veeling, P. Perdikaris, R. Turner, and J. Brandstetter, "Pde-refiner: Achieving accurate long rollouts with neural pde solvers," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[23] B. Chen, K. Huang, S. Raghupathi, I. Chandratreya, Q. Du, and H. Lipson, "Automated discovery of fundamental variables hidden in experimental data," *Nature Computational Science*, vol. 2, no. 7, pp. 433–442, 2022.

[24] H. Wang, X. Guo, S. X.-D. Tan, C. Zhang, H. Tang, and Y. Yuan, "Leakage-aware predictive thermal management for multicore systems using echo state network," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 7, pp. 1400–1413, 2019.

[25] https://www.comsol.com/comsol-multiphysics.

[26] https://www.ansys.com/products/electronics/ansys-icepak.

[27] M. R. Stan, K. Skadron, M. Barcella, W. Huang, K. Sankaranarayanan, and S. Velusamy, "Hotspot: A dynamic compact thermal model at the processor-architecture level," *Microelectronics Journal*, vol. 34, no. 12, pp. 1153–1165, 2003.

[28] Z. Yuan, P. Shukla, S. Chetoui, S. Nemtzow, S. Reda, and A. K. Coskun, "Pact: An extensible parallel thermal simulator for emerging integration and cooling technologies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 4, pp. 1048–1061, 2021.

[29] H. Sultan and S. R. Sarangi, "A fast leakage-aware green's-function-based thermal simulator for 3-d chips," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 11, pp. 2342–2355, 2020.

[30] S. Sadiqbatcha, J. Zhang, H. Amrouch, and S. X.-D. Tan, "Real-time full-chip thermal tracking: A post-silicon, machine learning perspective," *IEEE Transactions on Computers*, vol. 71, no. 6, pp. 1411–1424, 2021.

[31] W. Jin, S. Sadiqbatcha, J. Zhang, and S. X.-D. Tan, "Full-chip thermal map estimation for commercial multi-core cpus with generative adversarial learning," in *Proceedings of the 39th International Conference on Computer-Aided Design*, 2020, pp. 1–9.

[32] J. Lu, J. Zhang, and S. X.-D. Tan, "Real-time thermal map estimation for amd multi-core cpus using transformer," in *Proceedings of the 39th International Conference on Computer-Aided Design*, 2023, pp. 1–7.

[33] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.

[34] L.-Y. Lu and L.-Y. Chiou, "Temperature gradient-aware thermal simulator for three-dimensional integrated circuits," *IET Computers & Digital Techniques*, vol. 11, no. 5, pp. 190–196, 2017.

[35] R. Ranade, H. He, J. Pathak, N. Chang, A. Kumar, and J. Wen, "A thermal machine learning solver for chip simulation," in *Proceedings of the 2022 ACM/IEEE Workshop on Machine Learning for CAD*, 2022, pp. 111–117.

[36] H. Wang, J. Ma, S. X.-D. Tan, C. Zhang, H. Tang, K. Huang, and Z. Zhang, "Hierarchical dynamic thermal management method for high-performance many-core microprocessors," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 22, no. 1, pp. 1–21, 2016.

[37] H. Wang, L. Hu, X. Guo, Y. Nie, and H. Tang, "Compact piecewise linear model based temperature control of multicore systems considering leakage power," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7556–7565, 2019.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[39] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[40] V. Zinage and E. Bakolas, "Neural koopman lyapunov control," *Neurocomputing*, vol. 527, pp. 174–183, 2023.

[41] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2015.

[42] F. L. Chernousko, *State estimation for dynamic systems*. crc press, 1993.

[43] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," *arXiv preprint arXiv:1705.10941*, 2017.

[44] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[45] X. Chen, Z. Gong, X. Zhao, W. Zhou, and W. Yao, "A machine learning surrogate modeling benchmark for temperature field reconstruction of heat source systems," *Science China Information Sciences*, vol. 66, no. 5, p. 152203, 2023.

[46] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, 2008, pp. 72–81.

[47] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The splash-2 programs: Characterization and methodological considerations," *ACM SIGARCH computer architecture news*, vol. 23, no. 2, pp. 24–36, 1995.

[48] Z. Hadjilambrou, S. Das, P. N. Whatmough, D. Bull, and Y. Sazeides, "Gest: An automatic framework for generating cpu stress-tests," in *2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2019, pp. 1–10.

[49] Z. Xie, X. Xu, M. Walker, J. Knebel, K. Palaniswamy, N. Hebert, J. Hu, H. Yang, Y. Chen, and S. Das, "Apollo: An automated power modeling framework for runtime power introspection in high-volume commercial microprocessors," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, 2021, pp. 1–14.

[50] C. Sun, M. Song, D. Cai, B. Zhang, S. Hong, and H. Li, "A systematic review of echo state networks from design to application," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, pp. 23–37, 2022.

[51] A. Pathania and J. Henkel, "Hotsniper: Sniper-based toolchain for many-core thermal simulations in open systems," *IEEE Embedded Systems Letters*, vol. 11, no. 2, pp. 54–57, 2018.

[52] T. E. Carlson, W. Heirman, S. Eyerman, I. Hur, and L. Eeckhout, "An evaluation of high-level mechanistic core models," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 11, no. 3, pp. 1–25, 2014.

[53] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "The mcpat framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 10, no. 1, pp. 1–29, 2013.

[54] Q. Wang, T. Zhu, Y. Lin, R. Wang, and R. Huang, "Atsim3d: Towards accurate thermal simulator for heterogeneous 3d ic systems considering nonlinear leakage and conductivity," in *2024 International Symposium of Electronics Design Automation (ISEDA)*, 2024, pp. 1–6.

[55] V. Hanumaiah, S. Vrudhula, and K. S. Chatha, "Performance optimal online dvfs and task migration techniques for thermally constrained multi-core processors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 11, pp. 1677–1690, 2011.

[56] H. Wang, J. Wan, S. X.-D. Tan, C. Zhang, H. Tang, Y. Yuan, K. Huang, and Z. Zhang, "A fast leakage-aware full-chip transient thermal estimation method," *IEEE Transactions on Computers*, vol. 67, no. 5, pp. 617–630, 2017.