FasterRW: An Efficient Prior-Assisted Random Walk Solver for Thermal Analysis

Zixiao Wang‡, Tianshu Hou†, Chenghan Wang, Zhen Zhuang, Leilei Jin,
Tsung-Yi Ho, Farzan Farnia, Bei Yu†

***Abstract*—Thermal simulation is becoming increasingly critical in modern IC design and manufacturing. Feynman–Kac random walk methods can estimate localized temperatures without solving the global field, but in practical settings without Dirichlet boundaries they require very long paths to control the truncation residual. This paper presents FasterRW, a continuation-value framework for accelerating Feynman–Kac thermal random walks. We prove a stopping-time residual identity showing that the expected tail after truncation equals the current attenuation weight multiplied by the temperature at the stopping location. This identity enables a prior-corrected single-point estimator: a cheap, noisy prior replaces the unknown continuation value, making the truncation bias scale with prior error rather than absolute temperature and yielding an error-controlled threshold-selection rule. For multi-point queries, FasterRW further converts cross-query pass-through tails into pseudo-samples and refines the resulting observations in prior-error coordinates through kriging. Compared with prior state-of-the-art methods, FasterRW achieves up to $28\times$ acceleration while maintaining accuracy. Our code is open-sourced at https://github.com/ShiningSord/FastRW.**

***Index Terms*—Thermal analysis, random walk, Feynman–Kac formula, integrated circuits.**

I. Introduction

Heterogeneous Integration (HI) technologies, such as 3D Integrated Circuits (3D-ICs) in semiconductor packaging, are essential to meet the demands of advanced microelectronics, enabling continued system-level scaling [1]. However, higher power density in advanced ICs intensifies thermal issues, leading to performance degradation and reduced lifetime [2]. The heat conduction process, governed by computationally expensive partial differential equations (PDEs), poses significant challenges for thermal management, as exploring the design space may require thousands of simulations to identify optimal parameters [3]–[5].

To mitigate this problem, analytical methods, often leveraging Green's function, have been proposed to provide closed-form expressions for fast temperature estimation on simple geometries [6], [7]. Numerical methods, such as finite element methods (FEMs), finite difference methods (FDMs), and finite volume methods (FVMs), offer accurate thermal simulation at the cost of fine-grained spatial discretization [8]–[11]. Compact thermal models (CTMs) provide an efficient alternative by representing the thermal behavior based on equivalent RC networks and thereby transforming the problem into SPICE-like simulations, alleviating computational overhead [12]–[14].

†Corresponding authors. ‡Zixiao Wang is with The Hong Kong University of Science and Technology (Guangzhou). The other authors are with The Chinese University of Hong Kong.

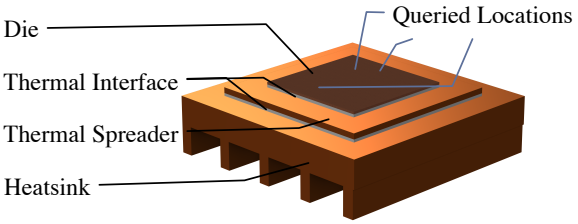


Fig. 1 An illustration of a pyramid-shaped IC geometry. In certain scenarios, only the temperatures at a few queried locations are needed.

Recently, learning-based thermal predictors have also been explored for fast thermal prediction, including convolutional encoder-decoder models, graph-based predictors, operator-learning methods, and package-level neural simulators [15]–[20]. After training, these models can provide rapid inference and are attractive for repeated thermal estimation. However, their accuracy and reliability depend heavily on the coverage of the training data, and their behavior under unseen power maps, package geometries, material stacks, or boundary conditions is often harder to certify. In contrast, analytical, numerical, and stochastic solvers are directly derived from the governing PDEs and boundary conditions, which remains important for accuracy-critical thermal analysis.

Nevertheless, high-resolution temperature data across the entire chip is not always necessary for thermal design, as localized hotspot estimation is typically more critical [21] (see Fig. 1). The stochastic random walk method excels at obtaining localized solutions through parallelized computation and has shown success in thermal analysis and other EDA applications involving Dirichlet boundary conditions (BCs) [22]–[24]. The study in [25] introduces a hybrid random walk approach capable of handling both adiabatic (Neumann) and convective (Robin) BCs for modeling realistic thermal dissipation. This method was later enhanced in [26]–[29] by incorporating the Feynman–Kac formula to improve accuracy. Unlike heuristic treatments, Feynman–Kac-based methods compute the boundary contribution by counting the local time spent near Neumann and Robin boundaries. While effective, these methods require a large number of steps to converge in practical scenarios without Dirichlet boundaries, significantly limiting their efficiency [30].

To overcome this limitation, we revisit finite-step Feynman–Kac random walks from the viewpoint of continuation-value estimation. We show that, at any stopping time before ab-

sorption, the expected remaining tail is exactly the current Feynman–Kac attenuation weight multiplied by the temperature at the stopping location. This identity changes the role of truncation: instead of choosing an empirical cutoff and accepting an uncontrolled residual, one can estimate the continuation value at the stopped location and make the induced bias explicit.

Building on this observation, we propose FasterRW, an efficient random-walk solver for localized thermal analysis. For a single query, FasterRW replaces the unknown continuation value with a cheap prior temperature estimate, so the truncation bias is governed by the prior error rather than by the absolute temperature magnitude. This yields an error-controlled threshold-selection rule and permits substantially shorter walks. For multiple queried locations, the same Markov structure exposes additional information: when a path launched from one query passes through another query voxel, its remaining tail can be reused as a pseudo-sample for the visited query. FasterRW further refines these direct and reused observations in prior-error coordinates, where the residual of a coarse prior is smoother than the temperature field itself. Our main contributions are summarized as follows:

- A stopping-time residual identity for Feynman–Kac-based thermal estimation, showing that the expected truncation tail equals a weighted continuation value at the stopping location.
- A prior-corrected single-point estimator with an explicit bias bound and a threshold-selection rule that links the allowable truncation threshold to the prior accuracy.
- A multi-point estimation framework that converts cross-query pass-through tails into pseudo-samples and fuses the resulting observations in prior-error coordinates through kriging refinement.
- Comprehensive experiments on commonly used 3D thermal benchmarks validating the residual identity, the prior-accuracy/threshold trade-off, the multi-query acceleration, and the end-to-end efficiency of FasterRW.

II. Preliminaries

In this section, we present the mathematical and algorithmic background necessary for our method. We begin by formulating the steady-state 3D thermal conduction problem with mixed boundary conditions. We then review the application of the Feynman–Kac formula to such problems, which connects the deterministic PDE formulation to stochastic process representations. Finally, we summarize two classes of random-walk-based numerical methods, Walk-on-Grids (WOG) and Walk-on-Spheres (WOS), and their respective roles in simulating heat transfer in source and source-free regions.

A. Problem Formulation

We consider the steady-state 3D thermal conduction problem governed by the Poisson equation with mixed boundary conditions as in Equation (1), where $T(\boldsymbol{x})$ is the temperature field over the spatial domain $\Omega \subset \mathbb{R}^3$ and ∇^2 denotes the Laplace operator. The source term is defined as $f(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{2k(\boldsymbol{x})}$, with $p(\boldsymbol{x})$ the volumetric heat generation density and $k(\boldsymbol{x})$ the thermal conductivity. The boundary $\partial\Omega$ is decomposed into three disjoint subsets: $\partial\Omega_D$ (Dirichlet), $\partial\Omega_N$ (Neumann), and $\partial\Omega_R$ (Robin), with $\boldsymbol{n}$ denoting the outward unit normal vector. The boundary conditions are specified as follows:

$$\begin{aligned} -\frac{1}{2}\nabla^2 T(\boldsymbol{x}) &= f(\boldsymbol{x}), && \boldsymbol{x} \in \Omega, \\ T(\boldsymbol{x}) &= \phi_D(\boldsymbol{x}), && \boldsymbol{x} \in \partial\Omega_D, \\ \frac{\partial T}{\partial \boldsymbol{n}} &= \phi_N(\boldsymbol{x}), && \boldsymbol{x} \in \partial\Omega_N, \\ \frac{\partial T}{\partial \boldsymbol{n}} + c(\boldsymbol{x})T(\boldsymbol{x}) &= \phi_R(\boldsymbol{x}), && \boldsymbol{x} \in \partial\Omega_R, \end{aligned} \tag{1}$$

- **Dirichlet:** $\phi_D(\boldsymbol{x}) = T_D(\boldsymbol{x})$, where T_D is the prescribed temperature.
- **Neumann:** $\phi_N(\boldsymbol{x}) = -q_n(\boldsymbol{x})/k(\boldsymbol{x})$, where $q_n(\boldsymbol{x})$ is the prescribed normal heat flux density. An adiabatic boundary corresponds to $q_n(\boldsymbol{x}) = 0$.
- **Robin:** $\phi_R(\boldsymbol{x}) = c(\boldsymbol{x})T_R(\boldsymbol{x})$ with $c(\boldsymbol{x}) = h(\boldsymbol{x})/k(\boldsymbol{x})$, where $h(\boldsymbol{x})$ is the convection coefficient and $T_R(\boldsymbol{x})$ is the reference temperature at the Robin boundary.

The domain Ω may contain multiple materials with different conductivities $k(\boldsymbol{x})$; across material interfaces, both temperature and normal heat flux are continuous. We further distinguish between heat-source regions ($p(\boldsymbol{x}) \neq 0$) and source-free regions ($p(\boldsymbol{x}) = 0$). With the above notation, the problem of 3D thermal estimation can be formulated as follows.

Problem 1. *Given the Poisson Equation (1) with mixed boundary conditions and a set of query locations $\{\boldsymbol{x}_q\}$ in Ω, estimate the temperature $T(\boldsymbol{x}_q)$ at each query location.*

B. Feynman–Kac Method

The Feynman–Kac formula establishes a connection between certain partial differential equations (PDEs) and expectations over stochastic processes [31]. In recent years, it has been applied to thermal simulation problems [26]–[28], enabling the solution of the Poisson equation in Equation (1) through Monte Carlo simulation of stochastic particle paths.

Let X_t denote a standard reflecting Brownian motion (SRBM) in the domain Ω, which is reflected at Neumann and Robin boundaries and absorbed at Dirichlet boundaries. The Feynman–Kac representation of the temperature field is

$$\begin{aligned} T(\boldsymbol{x}) = \mathbb{E}&\left\{\int_0^\infty \hat{e}_c(t) f(X_t)\,\mathrm{d}t\right\} \\ &+ \mathbb{E}\left\{\hat{e}_c(t_D)\,\phi_D(X_{t_D})\right\} \\ &+ \mathbb{E}\left\{\int_0^\infty \hat{e}_c(t)\,\phi_N(X_t)\,\mathrm{d}L(t)\right\} \\ &+ \mathbb{E}\left\{\int_0^\infty \hat{e}_c(t)\,\phi_R(X_t)\,\mathrm{d}L(t)\right\}, \end{aligned} \tag{2}$$

where

$$\hat{e}_c(t) = \exp\left(-\int_0^t c(X_s)\,\mathrm{d}L(s)\right), \tag{3}$$

$\mathbb{E}\{\cdot\}$ denotes the expectation over all SRBM paths, $L(t)$ is the boundary local time (which increases when the process is

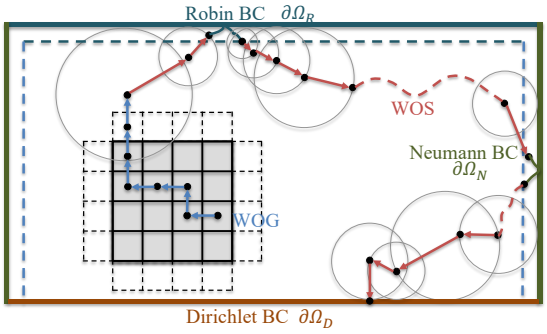


Fig. 2 An illustration of the random walk path in FasterRW.

reflected at Neumann or Robin boundaries) [28], [32], and t_D is the first hitting time of the Dirichlet boundary. We use the convention

$$L(t) = \lim_{\rho \to 0} \frac{1}{2\rho} \int_0^t \mathbf{1}_{D_\rho}(X_s)\, \mathrm{d}s, \tag{4}$$

where D_ρ denotes the ρ-width neighborhood of the reflecting boundary inside Ω; this convention absorbs the usual factor $1/2$ in the boundary local-time integrals. The coefficient $c(\boldsymbol{x})$ is given by $c(\boldsymbol{x}) = h(\boldsymbol{x})/k(\boldsymbol{x})$ on the Robin boundary and is zero elsewhere. In implementation, the growth of $L(t)$ is evaluated through the random walk behavior inside a thin boundary neighborhood, following the boundary-strip allocation in prior PIRW implementations [26]–[28].

In practice, the infinite integrals in Equation (2) are truncated. Simulation paths terminate in one of two cases:

1) the path hits the Dirichlet boundary, or
2) the Feynman–Kac functional $\hat{e}_c(t)$ decreases below a predefined threshold as Robin-boundary local time accumulates.

For problems where Dirichlet boundaries are present, paths usually terminate at a Dirichlet boundary before the functional $\hat{e}_c(t)$ becomes small enough. For problems without Dirichlet boundary conditions, which are common in practice, Feynman–Kac-based methods achieve higher accuracy than heuristic approaches [25]. However, in such cases, a large number of simulation steps may be required to sufficiently reduce $\hat{e}_c(t)$ and control the truncation error, resulting in high computational cost. Furthermore, the choice of path length heavily depends on empirical experience, and **there is currently no clear guideline on when a path should be terminated**. One of the key contributions of this work is to provide a quantitative analysis of the truncation error, together with a novel approach that reduces the number of required simulation steps and paths while preserving accuracy.

C. Random Walk Methods for Thermal Simulation

In the context of thermal simulation, the design of random-walk paths can be broadly categorized into two classes: *Walk-on-Grids* and *Walk-on-Spheres*.

Walk-on-Grids. In WOG methods, the heat-source region is discretized into a uniform voxel grid, with each voxel assigned constant temperature and power density values. Brownian motion is approximated by random jumps between adjacent grid nodes. For two neighboring nodes a and b, the probability of transition from a to b, denoted $P(a,b)$, is given by

$$P(a,b) = \frac{k(a,b)}{\sum_{x \in \mathrm{adj}(a)} k(a,x)}, \quad k(a,b) = \frac{2h_x h_y}{\frac{h_{za}}{k_a} + \frac{h_{zb}}{k_b}}, \tag{5}$$

where $\mathrm{adj}(a)$ denotes the set of nodes adjacent to a; k_a and k_b are the thermal conductivities at a and b, respectively; h_x and h_y are the side lengths of the shared interface; and h_{za} and h_{zb} are the edge lengths (thicknesses) of the two voxels along the direction normal to the shared interface (i.e., perpendicular to that interface).

Walk-on-Spheres. WOS methods are particularly efficient in homogeneous, source-free regions. In such regions, the temperature at a point $\boldsymbol{x}$ equals the average temperature over any sphere centered at $\boldsymbol{x}$:

$$T(\boldsymbol{x}) = \frac{1}{4\pi} \int_0^{2\pi} \int_0^{\pi} T(r,\omega,\theta)\, \sin\theta\, \mathrm{d}\theta\, \mathrm{d}\omega, \tag{6}$$

where r is the sphere radius, ω and θ are the azimuthal and polar angles, respectively, and $T(r,\omega,\theta)$ is the boundary temperature on the spherical surface. Each WOS step moves from the current center $\boldsymbol{x}$ to a uniformly sampled point $\boldsymbol{y}$ on the sphere surface. The next step repeats the process with $\boldsymbol{y}$ as the new center. For efficiency, the radius r is chosen as large as possible without crossing a boundary.

In this work, we adopt WOG for simulating heat transfer in the heat-source region and WOS for the source-free region, following established practice in the literature [25], [26]. An illustration of the random walk process can be found in Fig. 2.

III. FasterRW Method

The method is organized around one question: after a reflected Feynman–Kac random-walk path is stopped early, what information is needed to replace its remaining tail? As illustrated in Fig. 3, we first show that the expected tail has an exact continuation-value form at any stopping time in Section III-A. This residual identity then leads to three estimators: a prior-corrected single-point estimator that controls truncation bias (Section III-B), a tail-reuse pseudo-sample for multi-point queries (Section III-C), and a prior-error-space kriging refinement that exploits spatial coherence across queried points (Section III-D).

A. Truncation as Continuation-Value Estimation

This subsection establishes the mathematical basis of FasterRW. Instead of treating early stopping as an empirical approximation, we show that the expected remaining Feynman–Kac tail at any stopping time is exactly the attenuation weight times the steady-state temperature at the stopping location. This converts path truncation into a continuation-value estimation problem: controlling the truncation error amounts to estimating the temperature at the random stopping location accurately enough.

As discussed in Section II, problems without Dirichlet boundaries are the most common in practice, and are precisely the cases where Feynman–Kac-based methods demonstrate

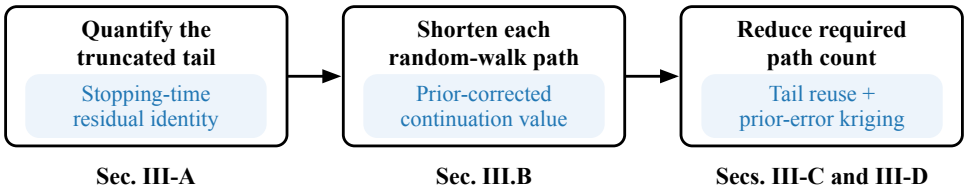


Fig. 3 FasterRW method overview.

significant accuracy advantages. We focus on the reflecting-dominated regime where the SRBM path does not encounter a Dirichlet boundary before the Feynman–Kac functional $\hat{e}_c(t)$ becomes sufficiently small. If a Dirichlet boundary is reached earlier, the path terminates by absorption and no truncation error is incurred.

In the Feynman–Kac representation (Equation (2)), for a given positive threshold $\Lambda \in (0,1)$, we define t_Λ as the first time at which $\hat{e}_c(t) \le \Lambda$. The single-path contribution can then be decomposed into two parts:

$$T_i(\boldsymbol{x}) = \mathbb{P}_0^{t_\Lambda} + \mathbb{P}_{t_\Lambda}^{\infty}, \tag{7}$$

where $T_i(\boldsymbol{x})$ denotes one realization of the temperature at the starting point $\boldsymbol{x}$, $\mathbb{P}_0^{t_\Lambda}$ aggregates the source and boundary contributions collected before t_Λ, and $\mathbb{P}_{t_\Lambda}^{\infty}$ represents the remaining contribution after t_Λ. If the path is truncated at t_Λ, the resulting truncation error is exactly $\mathbb{P}_{t_\Lambda}^{\infty}$.

To quantify this error, we formalize the following residual identity.

Theorem 1 (Stopping-time truncation residual identity)**.** *Let $\{X_t\}_{t\ge 0}$ be the reflected process defined in Section II with natural filtration $\{\mathcal{F}_t\}_{t\ge 0}$, and let τ be a stopping time such that $\tau < t_D$ on the event of interest. Then the truncation tail satisfies*

$$\mathbb{E}[\mathbb{P}_\tau^{\infty} \mid \mathcal{F}_\tau] = \hat{e}_c(\tau)\, T(X_\tau)\,. \tag{8}$$

This identity is the central object used by FasterRW. It says that **the unsimulated tail is not an arbitrary leftover term: in expectation, it is a continuation value at the stopping location, scaled by the current attenuation weight.**

Corollary 1. *Given the truncation location X_t and the corresponding functional $\hat{e}_c(t)$, the expected truncation error equals the weighted temperature $\hat{e}_c(t)\,T(X_t)$ at the truncation location. In particular, for the threshold rule $\tau = t_\Lambda$,*

$$\mathbb{E}\big[\mathbb{P}_{t_\Lambda}^{\infty} \mid \mathcal{F}_{t_\Lambda}\big] = \hat{e}_c(t_\Lambda)\, T(X_{t_\Lambda}) \le \Lambda\, T(X_{t_\Lambda}). \tag{9}$$

The proof follows from the strong Markov property of the reflected process and is provided in Section A. An intuitive way to interpret Equation (8) is that the truncation tail behaves like the contribution induced by imposing a virtual Dirichlet boundary condition at the truncation position X_{t_Λ}. Unlike previous works, which select Λ empirically, this identity provides a principled guideline: to maintain the truncation error within an acceptable range, the threshold should be chosen such that the expected residual scale, $\Lambda\,\mathbb{E}[T(X_{t_\Lambda})]$, remains sufficiently small.

However, in problems with only Neumann or Robin boundaries, $\hat{e}_c(t)$ decreases only upon reflections, meaning that a large number of SRBM steps are required before termination. For example, in one of our test cases, reducing $\hat{e}_c(t)$ to 10% of its initial value required roughly 1.5×10^6 random walk steps. The next subsection uses the continuation-value view to stop such paths earlier while keeping the induced bias explicit.

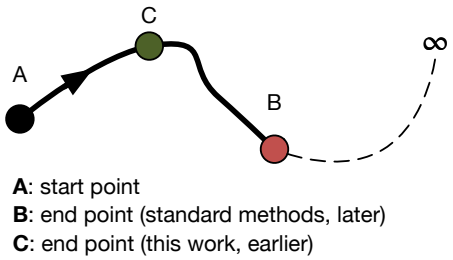


Fig. 4 A random walk path in a single-point simulation. Standard methods compute the integration from A to B. In contrast, FasterRW computes the integration from A to C and estimates the remaining contribution from C to infinity.

B. Prior-Corrected Single-Point Estimation

This subsection uses the residual identity to shorten a single random-walk path. The exact continuation value is not available during simulation, so we replace it with a cheap prior estimate at the stopping location. The key effect is to make the remaining bias depend on the prior's error, rather than on the much larger absolute temperature scale.

We assume that a *prior* temperature estimate $\tilde{T}(\boldsymbol{x})$ is available within a subdomain $\Omega_s \subseteq \Omega$ before initiating the random walk process:

$$\tilde{T}(\boldsymbol{x}) = T(\boldsymbol{x}) + \epsilon(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega_s, \tag{10}$$

where $\epsilon(\boldsymbol{x})$ denotes the prior estimation error. The procedure for obtaining such prior estimates will be described in a later section.

With this prior estimation in place, the random walk path in FasterRW terminates under one of the following two conditions:

1) The path reaches a Dirichlet boundary, or
2) The Feynman–Kac functional $\hat{e}_c(t)$ decreases below a predefined threshold as the local time $L(t)$ increases, *and* the current location lies in the subdomain Ω_s where the prior temperature field is available.

When termination occurs under the second condition at a stopping time τ, FasterRW does not simply return the accumulated term $\mathbb{P}_0^{\tau}$ as in previous approaches. Instead, it

augments this value using the prior estimation, yielding the *prior-corrected single-path estimator*

$$\begin{aligned}\hat{T}_i(\boldsymbol{x}) &\triangleq \mathbb{P}_0^\tau + \hat{e}_c(\tau)\,\tilde{T}(X_\tau) \\ &= \mathbb{P}_0^\tau + \mathbb{E}[\mathbb{P}_\tau^\infty \mid \mathcal{F}_\tau] + \hat{e}_c(\tau)\,\epsilon(X_\tau),\end{aligned} \tag{11}$$

where $\hat{T}_i(\boldsymbol{x})$ is the i-th single-path estimate of the temperature at the query location $\boldsymbol{x}$, $\mathbb{P}_0^\tau$ is the Feynman–Kac sum accumulated up to the truncation time τ, $\tilde{T}(X_\tau)$ is the prior estimate at the truncation location X_τ, and $\epsilon(X_\tau)$ is the prior-estimation error there (cf. Equation (10)). Compared with vanilla truncation, the expected truncation error is reduced from $\hat{e}_c(\tau)\,T(X_\tau)$ to $\hat{e}_c(\tau)\,\epsilon(X_\tau)$. Since in most cases $|\epsilon(X_\tau)| \ll T(X_\tau)$, **the allowable threshold Λ in FasterRW can be chosen much larger than in conventional methods, significantly reducing the number of random walk steps required to achieve the same truncation error**, as illustrated in Fig. 4. The following result makes this bias reduction explicit.

This estimator is the operational rule used by each accelerated path. Instead of discarding the unsimulated tail, it returns the simulated prefix plus a prior-based continuation estimate, which is why a path can stop much earlier than in vanilla Feynman–Kac random walk.

For comparison, define the ideal but infeasible truncation-corrected estimator that replaces the prior with the exact continuation value:

$$\hat{T}_i^\star(\boldsymbol{x}) \triangleq \mathbb{P}_0^\tau + \hat{e}_c(\tau)\,T(X_\tau), \tag{12}$$

which is the oracle counterpart of $\hat{T}_i(\boldsymbol{x})$ obtained by substituting the true temperature $T(X_\tau)$ for the prior $\tilde{T}(X_\tau)$. By Theorem 1, $\hat{T}_i^\star(\boldsymbol{x})$ is unbiased for $T(\boldsymbol{x})$.

Theorem 2 (Bias induced by prior correction)**.** *Let $\hat{T}_i(\boldsymbol{x})$ and $\hat{T}_i^\star(\boldsymbol{x})$ be defined in Equations (11) and (12). Then*

$$\hat{T}_i(\boldsymbol{x}) - \hat{T}_i^\star(\boldsymbol{x}) = \hat{e}_c(\tau)\,\epsilon(X_\tau), \tag{13}$$

and the bias satisfies

$$\mathbb{E}\left[\hat{T}_i(\boldsymbol{x})\right] - T(\boldsymbol{x}) = \mathbb{E}[\hat{e}_c(\tau)\,\epsilon(X_\tau)]\,. \tag{14}$$

Corollary 2 (Uniform bound under threshold truncation)**.** *Suppose the truncation rule enforces $\hat{e}_c(\tau) \le \Lambda$ almost surely and the prior error satisfies $\sup_{\boldsymbol{x}\in\Omega_s} |\epsilon(\boldsymbol{x})| \le \epsilon_{\max}$. Then*

$$\left|\mathbb{E}\left[\hat{T}_i(\boldsymbol{x})\right] - T(\boldsymbol{x})\right| \le \Lambda\,\epsilon_{\max}. \tag{15}$$

Equivalently, to ensure an absolute bias tolerance b_{tol}, it suffices to choose

$$\Lambda \le \frac{b_{\mathrm{tol}}}{\epsilon_{\max}}. \tag{16}$$

Corollary 2 gives a direct threshold-selection rule: once a prior error bound is available, Λ can be chosen from the target truncation-induced bias rather than by empirical tuning. A detailed bias–variance decomposition of this plug-in correction is given in Section B.

Once all paths have terminated, the estimated temperature at the starting point, $\bar{T}(\boldsymbol{x})$, is obtained by averaging the individual path estimators $\{\hat{T}_i(\boldsymbol{x})\}_{i=1}^N$ from Equation (11):

$$\bar{T}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} \hat{T}_i(\boldsymbol{x}), \tag{17}$$

where $\bar{T}(\boldsymbol{x})$ is the empirical mean used as the final temperature estimate at $\boldsymbol{x}$, N is the total number of paths initiated from $\boldsymbol{x}$, and $\hat{T}_i(\boldsymbol{x})$ is the prior-corrected single-path estimator of the i-th path. Since the prior estimation approach in Equation (11) applies a larger truncation threshold Λ and shortens the path length, the resulting variance of individual paths is also smaller than in conventional methods.

By the law of large numbers, N should be chosen so that the variance of the mean remains within an acceptable range:

$$\mathrm{Var}\left[\bar{T}(\boldsymbol{x})\right] = \frac{\sigma^2(\boldsymbol{x})}{N}, \tag{18}$$

where $\sigma^2(\boldsymbol{x}) \triangleq \mathrm{Var}[\hat{T}_i(\boldsymbol{x})]$ is the per-path variance of the prior-corrected single-path estimator at $\boldsymbol{x}$. In practice, N is selected to balance computational cost and accuracy, as increasing N reduces variance but requires more computation.

Operationally, the single-point stage leaves the WOG/WOS transition rule unchanged and only modifies the stopping rule and returned path contribution: a path exits at a Dirichlet boundary as usual, or exits once $\hat{e}_c(t) \le \Lambda$ at a location where the prior is available, in which case it returns $\mathbb{P}_0^\tau + \hat{e}_c(\tau)\,\tilde{T}(X_\tau)$ instead of the uncorrected partial sum.

Obtaining Prior Estimation. The prior field $\tilde{T}$ only needs to be inexpensive and reasonably accurate over a subdomain Ω_s; it is not required to match the accuracy of the final random-walk estimate. Its role is to reduce the correction scale in Equation (11) from the absolute temperature to the prior error, so even a coarse prior can be useful. We examine the dependence between prior accuracy and the allowable truncation threshold in TABLE III.

In FasterRW, we use a coarse FEM solve as the prior. Such a solve is inexpensive because the element count can be kept low, while still providing enough accuracy to reduce the truncation residual. In our chip thermal experiments, the coarse FEM prior typically has a maximum error around 5 K, compared with an on-chip average temperature around 400 K; this allows a truncation threshold up to about $80\times$ larger than in conventional random walk methods.

As described in Section II, we apply WOG in the source region and WOS in the source-free region, using prior predictions only in the source region. When a path terminates at a source-region voxel, the prior temperature at that voxel is returned as specified in Equation (11).

C. Tail-Reuse Observations for Multi-Point Queries

This subsection extends the continuation-value view from path endpoints to intermediate pass-through events. As illustrated in Fig. 5, in a multi-query run, when a path launched from one query voxel enters another query voxel, its remaining segment behaves like a fresh continuation path from the visited location. The already simulated tail can therefore be reused as an additional observation for the visited query, reducing the number of direct paths needed per query.

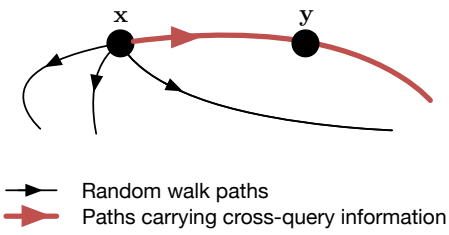


Fig. 5 Random walk paths in multi-point simulation. Some paths contain cross-point information and can correct the prediction.

When the query set contains multiple locations $\{\boldsymbol{x}_q\}_{q=1}^{M}$, conventional approaches estimate each temperature $T(\boldsymbol{x}_q)$ using only paths that originate from $\boldsymbol{x}_q$ (cf. Equation (17)). The truncation residual identity of Theorem 1 reveals that paths starting elsewhere can also carry information about $T(\boldsymbol{x}_q)$.

Consider the n-th path that starts at query point $\boldsymbol{x}_i$, and let $\hat{T}_{i,n}$ denote the prior-corrected single-path estimator from Equation (11) evaluated along this specific path (the realization of $\hat{T}_i(\boldsymbol{x}_i)$ on the n-th path from origin i). In continuous space, hitting an exact query point has probability zero, so we treat "visiting $\boldsymbol{x}_j$" as the event that the walk enters the discretization voxel associated with $\boldsymbol{x}_j$. Suppose the path enters such a voxel at time t_k, where the subscript k indexes successive pass-through events along the path. By the strong Markov property invoked in the proof of Theorem 1, the remainder of the path is, in distribution, a fresh reflected process started from the SRBM position X_{t_k} with its Feynman–Kac weight scaled by $\hat{e}_c(t_k)$. The complete prior-corrected single-path estimator therefore admits the pathwise decomposition

$$\hat{T}_{i,n} \;=\; \mathbb{P}_0^{t_k} + \mathbb{P}_{t_k}^{\infty,\,n}, \tag{19}$$

where $\mathbb{P}_{t_k}^{\infty,\,n}$ denotes the post-t_k contribution along this single path (including the prior-assisted truncation correction in Equation (11)). Solving Equation (19) for $\mathbb{P}_{t_k}^{\infty,\,n}$ and dividing by $\hat{e}_c(t_k)$ produces a pseudo-sample of $T(X_{t_k})$, the temperature at the visited voxel.

Corollary 3 (Tail-reuse pseudo-sample)**.** *For each pass-through event k along the n-th path starting at $\boldsymbol{x}_i$, define the tail-reuse pseudo-sample*

$$S_{j_k,\,k} \;\triangleq\; \frac{\mathbb{P}_{t_k}^{\infty,\,n}}{\hat{e}_c(t_k)} = \frac{\hat{T}_{i,n} - \mathbb{P}_0^{t_k}}{\hat{e}_c(t_k)}, \tag{20}$$

where j_k is the index of the query voxel entered at t_k, $\mathbb{P}_{t_k}^{\infty,\,n}$ is the post-t_k tail contribution from Equation (19), $\hat{T}_{i,n}$ is the complete prior-corrected single-path estimator, $\mathbb{P}_0^{t_k}$ is the pre-t_k Feynman–Kac sum, and $\hat{e}_c(t_k)$ is the Feynman–Kac weight at t_k.

This definition is the conversion rule from a pass-through event to a reusable observation. It removes the contribution accumulated before the path reached the visited query and rescales the remaining tail back to the temperature scale of that query.

Then $S_{j_k,k}$, viewed as an estimate of $T(\boldsymbol{x}_{j_k})$, satisfies

$$\mathbb{E}[S_{j_k,\,k} \mid X_{t_k} = \boldsymbol{y}] \;=\; T(\boldsymbol{y}) \;+\; \mathbb{E}\Big[\tfrac{\hat{e}_c(\tau)}{\hat{e}_c(t_k)}\,\epsilon(X_\tau)\Big], \tag{21}$$

and

$$\mathrm{Var}[S_{j_k,\,k} \mid X_{t_k} = \boldsymbol{y}] \;=\; \sigma^2(\boldsymbol{y}), \tag{22}$$

where τ is the post-t_k truncation time of the path and $\sigma^2(\boldsymbol{y})$ is the per-path variance of the prior-corrected single-path estimator at $\boldsymbol{y}$ from Equation (18).

Corollary 3 follows from Theorems 1 and 2 applied to the post-t_k segment of the path. The pseudo-sample $S_{j_k,k}$ is therefore approximately unbiased for $T(\boldsymbol{x}_{j_k})$, with a residual plug-in bias inherited from Corollary 2 and amplified by the factor $1/\hat{e}_c(t_k)$. To control this amplification, we restrict to events with $\hat{e}_c(t_k) \geq e_{\min}$ (default $e_{\min} = 0.3$). When the same path enters the same target voxel more than once, only the record with the largest $\hat{e}_c(t_k)$ is retained, so that pseudo-samples within a target are uncorrelated.

Per-target Bayesian fusion. Let $\mathcal{S}_q = \{S_{q,\,k} : j_k = q,\ \hat{e}_c(t_k) \geq e_{\min}\}$ denote the set of tail-reuse pseudo-samples collected at the query point $\boldsymbol{x}_q$, with cardinality $n_q \triangleq |\mathcal{S}_q|$. By Corollary 3, each pseudo-sample $S_{q,k}$ has the same conditional variance as a single direct path estimator $\hat{T}_{q,n}$ from $\boldsymbol{x}_q$. Modelling both as conditionally Gaussian and placing an improper flat prior on $T(\boldsymbol{x}_q)$,

$$\begin{aligned} \hat{T}_{q,n} \mid T(\boldsymbol{x}_q) &\sim \mathcal{N}\big(T(\boldsymbol{x}_q),\, \sigma^2(\boldsymbol{x}_q)\big), \\ S_{q,k} \mid T(\boldsymbol{x}_q) &\sim \mathcal{N}\big(T(\boldsymbol{x}_q),\, \sigma^2(\boldsymbol{x}_q)\big), \end{aligned} \tag{23}$$

where $\sigma^2(\boldsymbol{x}_q)$ is the per-path variance at $\boldsymbol{x}_q$ (cf. Equation (18)).

The posterior mean of $T(\boldsymbol{x}_q)$ is the inverse-variance weighted combination of the direct mean $\bar{T}(\boldsymbol{x}_q) \triangleq N^{-1}\sum_n \hat{T}_{q,n}$ and the tail-reuse mean $\bar{S}_q \triangleq n_q^{-1}\sum_k S_{q,k}$:

$$\hat{T}_q \;=\; \frac{N\,\bar{T}(\boldsymbol{x}_q) + n_q\,\bar{S}_q}{N + n_q}, \qquad \mathrm{Var}[\hat{T}_q] \;=\; \frac{\sigma^2(\boldsymbol{x}_q)}{N + n_q}, \tag{24}$$

where $\hat{T}_q$ is the per-target fused estimate of $T(\boldsymbol{x}_q)$. **Thus, paths launched for one query are no longer useful only for their own starting point: each qualified pass-through tail can reduce the direct path budget of another queried point.**

In practice, $\sigma^2(\boldsymbol{x}_q)$ is replaced by the empirical sample variance $\hat{\sigma}^2(\boldsymbol{x}_q)$ of $\{\hat{T}_{q,n}\}_{n=1}^{N}$, and the corresponding empirical sample variance $\widehat{\mathrm{Var}}(\mathcal{S}_q)$ is used for the tail-reuse term, yielding

$$\hat{T}_q = \frac{N\,\bar{T}(\boldsymbol{x}_q) + \frac{\hat{\sigma}^2(\boldsymbol{x}_q)}{\widehat{\mathrm{Var}}(\mathcal{S}_q)}\,n_q\,\bar{S}_q}{N + \frac{\hat{\sigma}^2(\boldsymbol{x}_q)}{\widehat{\mathrm{Var}}(\mathcal{S}_q)}\,n_q}, \quad \mathrm{Var}[\hat{T}_q] = \frac{\hat{\sigma}^2(\boldsymbol{x}_q)}{N + \frac{\hat{\sigma}^2(\boldsymbol{x}_q)}{\widehat{\mathrm{Var}}(\mathcal{S}_q)}\,n_q}. \tag{25}$$

This is the fusion formula used in implementation. It keeps the same idea as the idealized rule, but estimates the relative reliability of direct paths and reused tails from the sampled data rather than assuming identical variances.

Thus the multi-point procedure consists of three statistical steps: record qualified cross-query pass-through events during the usual prior-assisted walks, convert their remaining path

contributions into pseudo-samples through Equation (20), and combine direct and tail-reuse observations through Equation (25). The estimator in Equation (24) provides a direct temperature-space fusion of single-point Monte Carlo estimates and tail-reuse pseudo-samples. The key information source exposed by Corollary 3 is that every step of a multi-point walk, and not only its termination, contributes statistical information about the queried temperatures; this reduces the number of paths required to attain the desired precision in FasterRW. We next reuse the fused estimates and variances as noisy observations of the prior-error field, enabling spatial refinement across the query set.

D. Prior-Error Kriging Refinement

This subsection refines the multi-point estimates by changing the statistical target from temperature to prior error. The temperature field may contain sharp hotspot structure, while the error of a coarse FEM prior is typically smoother over nearby query locations. We use this smoother error field to regularize the Monte Carlo estimates across a query batch through universal kriging.

The tail-reuse fusion above estimates the queried temperatures directly. Because FasterRW already maintains a coarse prior field $\tilde{T}$ for the single-point plug-in correction (Equation (11)), the same per-target observations can equivalently be expressed using the prior-error convention in Equation (10). We model the stacked prior-error vector

$$\boldsymbol{\epsilon} \triangleq [\epsilon(\boldsymbol{x}_1), \ldots, \epsilon(\boldsymbol{x}_M)]^\top$$

where $\epsilon(\boldsymbol{x}_i)$ denotes the prior error at the i-th query point. **Estimating the smoother prior-error field allows FasterRW to borrow correction information across nearby queries, further reducing the path count beyond tail reuse alone.**

Combined per-target observation. Let $\hat{T}_q$ be the per-target fused estimate of $T(\boldsymbol{x}_q)$ from Equation (25). Define the per-target prior-error observation

$$\begin{aligned} d_q &\triangleq \tilde{T}(\boldsymbol{x}_q) - \hat{T}_q, \\ d_q \mid \epsilon(\boldsymbol{x}_q) &\sim \mathcal{N}\Big(\epsilon(\boldsymbol{x}_q),\, \mathrm{Var}\big[\hat{T}_q\big]\Big), \end{aligned} \tag{26}$$

where d_q is the observed discrepancy between the prior $\tilde{T}(\boldsymbol{x}_q)$ and the fused estimate $\hat{T}_q$, and $\epsilon(\boldsymbol{x}_q)$ is the (unknown) prior-estimation error at $\boldsymbol{x}_q$. Stacked over the query set, this gives the vector observation $\boldsymbol{d} \triangleq [d_1, \ldots, d_M]^\top$ with conditional distribution $\boldsymbol{d} \mid \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}, \boldsymbol{\Sigma}_{\mathrm{MC}})$, where $\boldsymbol{\Sigma}_{\mathrm{MC}} \triangleq \mathrm{diag}(\mathrm{Var}[\hat{T}_1], \ldots, \mathrm{Var}[\hat{T}_M])$ is the diagonal Monte Carlo noise covariance.

Universal kriging prior. We model the prior-error vector $\boldsymbol{\epsilon}$ as the sum of an unknown constant trend μ and a zero-mean spatial Gaussian process:

$$\boldsymbol{\epsilon} \;=\; \mu\,\mathbf{1} + \mathbf{g}, \qquad \mathbf{g} \sim \mathcal{N}(\mathbf{0},\, \boldsymbol{K}_\ell), \qquad p(\mu) \propto 1, \tag{27}$$

where μ is the global drift of the prior error, $\mathbf{1} \in \mathbb{R}^M$ is the all-ones vector, $\mathbf{g}$ is the zero-mean spatial residual with covariance matrix $\boldsymbol{K}_\ell$, and $p(\mu) \propto 1$ is an improper flat prior over μ.

The covariance $\boldsymbol{K}_\ell$ uses an RBF kernel,

$$[\boldsymbol{K}_\ell]_{ij} \;=\; \sigma_\epsilon^2 \exp\!\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\ell^2}\right), \tag{28}$$

with marginal variance σ_ϵ^2 and length scale ℓ. The trend μ absorbs any near-uniform bias in the coarse prior, e.g., a global temperature shift between a low-DoF FEM solution and the truth, while $\mathbf{g}$ captures the residual spatial structure of the error. Assigning μ an improper flat prior lets it be inferred from data without anchoring the posterior toward an arbitrary value.

REML hyperparameter selection. Let $\boldsymbol{C} \triangleq \boldsymbol{K}_\ell + \boldsymbol{\Sigma}_{\mathrm{MC}}$ denote the total covariance of $\boldsymbol{d}$ given μ. Integrating μ out against its flat prior yields the restricted maximum-likelihood (REML) marginal log-likelihood, used to select the kernel hyperparameters $(\ell, \sigma_\epsilon^2)$:

$$\begin{aligned} \log p(\boldsymbol{d} \mid \ell, \sigma_\epsilon^2) = &-\tfrac{M-1}{2}\log 2\pi - \tfrac{1}{2}\log\det \boldsymbol{C} - \tfrac{1}{2}\log A \\ &- \tfrac{1}{2}\,(\boldsymbol{d} - \hat{\mu}\mathbf{1})^\top \boldsymbol{C}^{-1}(\boldsymbol{d} - \hat{\mu}\mathbf{1}). \end{aligned} \tag{29}$$

Here, the auxiliary quantities are

$$A \;\triangleq\; \mathbf{1}^\top \boldsymbol{C}^{-1}\mathbf{1}, \qquad \hat{\mu} \;\triangleq\; \frac{\mathbf{1}^\top \boldsymbol{C}^{-1}\boldsymbol{d}}{A}, \tag{30}$$

with A the precision of $\hat{\mu}$ under known $\boldsymbol{C}$ and $\hat{\mu}$ the generalized-least-squares estimate of the global drift. We maximize Equation (29) over a grid in $(\ell, \sigma_\epsilon^2)$, with ℓ scaled to the spatial discretization and σ_ϵ^2 scaled to the empirical variance of $\boldsymbol{d}$.

Posterior estimate. For the selected $(\ell^\star, \sigma_\epsilon^{2,\star})$, the posterior mean of the stacked prior-error vector $\boldsymbol{\epsilon}$ is

$$\hat{\boldsymbol{\epsilon}} \;=\; \hat{\mu}\,\mathbf{1} + \boldsymbol{K}_\ell\,\boldsymbol{C}^{-1}(\boldsymbol{d} - \hat{\mu}\,\mathbf{1}), \tag{31}$$

where $\hat{\boldsymbol{\epsilon}} \in \mathbb{R}^M$ is the universal-kriging posterior mean of $\boldsymbol{\epsilon}$ given $\boldsymbol{d}$.

The marginal posterior variance at $\boldsymbol{x}_i$ is

$$\begin{aligned} \mathrm{Var}[\epsilon(\boldsymbol{x}_i) \mid \boldsymbol{d}] \;=\; &[\boldsymbol{K}_\ell]_{ii} - (\boldsymbol{K}_\ell \boldsymbol{C}^{-1}\boldsymbol{K}_\ell)_{ii} \\ &+ \frac{\big(1 - (\mathbf{1}^\top \boldsymbol{C}^{-1}\boldsymbol{K}_\ell)_i\big)^2}{A}, \end{aligned} \tag{32}$$

where the third term reflects residual uncertainty about μ propagated to $\epsilon(\boldsymbol{x}_i)$.

The final FasterRW temperature estimate is then

$$\hat{\boldsymbol{T}}_\epsilon \;\triangleq\; \tilde{\boldsymbol{T}} - \hat{\boldsymbol{\epsilon}}, \qquad \tilde{\boldsymbol{T}} \;\triangleq\; [\tilde{T}(\boldsymbol{x}_1), \ldots, \tilde{T}(\boldsymbol{x}_M)]^\top, \tag{33}$$

where $\hat{\boldsymbol{T}}_\epsilon \in \mathbb{R}^M$ is the prior-error-corrected temperature vector at the query points and $\tilde{\boldsymbol{T}}$ is the prior temperature vector. Derivations of Equations (29) to (32) are provided in Section C.

The estimator in Equation (31) can be viewed as a statistically regularized correction of the coarse prior field. The observation vector $\boldsymbol{d}$ measures the discrepancy between the prior prediction and the random-walk estimate, the drift μ absorbs the average bias of the coarse FEM solution, and the RBF covariance $\boldsymbol{K}_\ell$ encodes the residual spatial coherence in the prior error. After the tail-reuse fusion (Equation (25)) has produced $\{\hat{T}_q, \mathrm{Var}[\hat{T}_q]\}_{q=1}^M$, the prior-error refinement only solves an M-dimensional Gaussian update involving the $M \times$

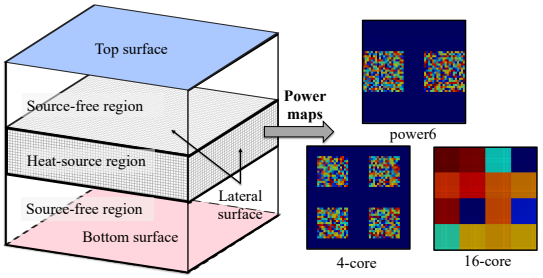


Fig. 6 An illustration of test cases and power maps.

M covariance matrix $\boldsymbol{C}$, whose Cholesky cost is negligible for M in the tens. Therefore, the quantity reported by FasterRW is the prior-error-corrected estimate $\hat{\boldsymbol{T}}_\epsilon$ in Equation (33).

IV. Experiments

In this section, we evaluate FasterRW on widely used 3D steady-state thermal simulation benchmarks [26] and compare against two random-walk baselines.

A. Experimental Settings

For a fair comparison with existing state-of-the-art methods, we closely follow the setups adopted in prior works [26]–[29]. Due to differences in implementation and hardware, some absolute runtimes may vary slightly, but all methods are evaluated under the same conditions.

Test Cases. Following previous works [26]–[29], we consider a 3D chip thermal model illustrated in Fig. 6. The model consists of three layers: the middle layer represents the heat-source region (transistors and metal interconnections), while the top and bottom layers are source-free regions modeling the heat transfer medium and substrate, respectively. The chip measures $2\,\mathrm{cm} \times 2\,\mathrm{cm}$ in the horizontal plane. We study three thickness configurations (top, middle, bottom): Case 1 $(500, 100, 500)\,\mu$m, Case 2 $(1000, 100, 1000)\,\mu$m, and Case 3 $(500, 100, 1000)\,\mu$m.

The heat-source region is discretized into voxels for WOG, with a resolution of $200\,\mu\mathrm{m} \times 200\,\mu\mathrm{m} \times 20\,\mu\mathrm{m}$, consistent with prior work. The thermal conductivities are $395\,\mathrm{W}/(\mathrm{K}\cdot\mathrm{m})$ for source-free regions and $125\,\mathrm{W}/(\mathrm{K}\cdot\mathrm{m})$ for the heat-source region, with an ambient temperature of $293.15\,\mathrm{K}$.

For power maps, we adopt the settings from prior studies, shown in Fig. 6. We consider three types, `4-core`, `16-core`, and `power6`, with total powers $356\,\mathrm{W}$, $352\,\mathrm{W}$, and $174\,\mathrm{W}$. While the precise power-map values differ slightly from those reported in prior work, leading to minor variations in absolute temperatures, these differences do not affect the relative comparison between algorithms.

Since FasterRW focuses on acceleration in cases without Dirichlet boundaries, we follow prior works [26], [27] and impose Robin boundary conditions on the top and bottom surfaces with convection coefficient $h(\boldsymbol{x}) = 8700\,\mathrm{W}/(\mathrm{K}\cdot\mathrm{m}^2)$ for Cases 1 and 2, and $h(\boldsymbol{x}) = 4900\,\mathrm{W}/(\mathrm{K}\cdot\mathrm{m}^2)$ for Case 3. Lateral surfaces are modeled as adiabatic Neumann boundaries with zero normal heat flux, $q_n(\boldsymbol{x}) = 0$.

Implementation Details. Unless otherwise noted, FasterRW uses a truncation threshold $\Lambda = 0.03$. This value follows the threshold-selection rule in Corollary 2: the worst-case maximum point-wise prior error across the three cases is $5.04\,\mathrm{K}$ (Case 2), so $\Lambda = 0.03$ bounds the threshold-induced correction bias by $0.15\,\mathrm{K}$, well below the iso-accuracy targets $\varepsilon \in \{0.4, 0.5\}\,\mathrm{K}$ used in our experiments. We therefore apply a uniform $\Lambda = 0.03$ across all cases.

For multi-point acceleration, FasterRW records at most one tail-reuse pseudo-sample for $T(\boldsymbol{x}_j)$ every 1000 random-walk steps, and only admits pass-through events whose Feynman–Kac weight satisfies $\hat{e}_c(t_k) \geq e_{\min} = 0.3$. In the tail-reuse fusion stage, only cross-target pass-throughs ($j_k \neq i$) are kept, since head and tail samples drawn from the same trajectory would violate the independence assumption of inverse-variance fusion. In the kriging refinement stage, we use observations from all queried locations unless specified otherwise.

The local-time update at Robin and Neumann boundaries follows the boundary-strip allocation used in prior PIRW implementations [26]–[28].

The random walk is implemented in C++ with a Metal compute kernel and runs on the GPU of an Apple M5 Pro workstation. FEM prior estimation, Bayesian tail-reuse fusion, and universal-kriging refinement are CPU stages executed on the same workstation. All baselines use the same C++/Metal codebase and the same hardware, so reported speedups are measured under a single, consistent setup.

To estimate priors for the temperature field, we employ a commercial FEM tool with coarse element discretization. The FEM priors used for the main comparison have approximately 1,288, 2,779, and 2,841 degrees of freedom for Cases 1, 2 and 3. These discretizations are chosen for two reasons: first, they are lightweight enough that the FEM solve time is strictly below $1\,\mathrm{s}$ on each case; second, the resulting prior error is acceptable, with a worst-case point-wise error of $5.04\,\mathrm{K}$ on Case 2. The golden reference is obtained from the finest FEM discretization, whose mesh exceeds $2.5\,\mathrm{M}$ degrees of freedom on every case.

Baseline Methods. We compare FasterRW against two random-walk baselines.

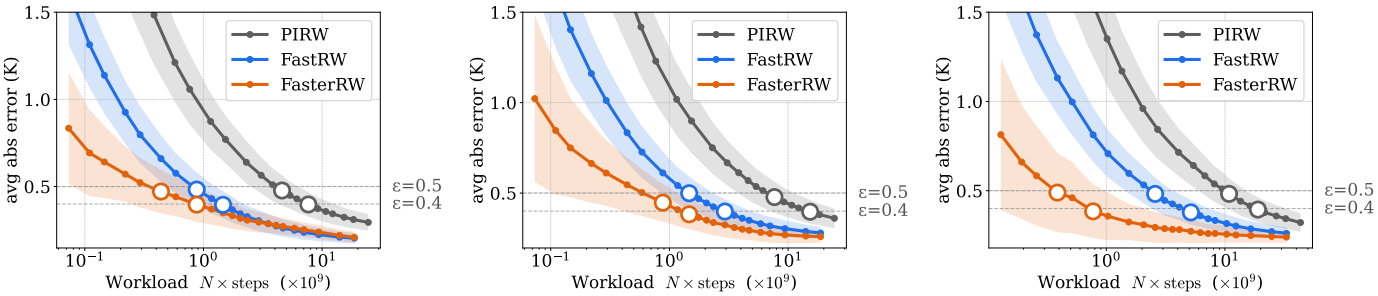
PIRW [26], [27] is the canonical Feynman–Kac solver described in Section II; following [27], we set its truncation threshold to $\Lambda = 10^{-4}$. PIRW has no prior correction, so its effective tail scale is bounded by the absolute temperature magnitude ($\sim 350\,\mathrm{K}$ in our benchmarks), which explains why it must use a much smaller threshold than FasterRW to keep truncation bias small.

FastRW [29] is a recently proposed Feynman–Kac solver that introduces prior-corrected truncation together with Bayesian fusion of cross-point observations. We adopt the same $\Lambda = 0.03$ as in FasterRW and the same multi-point hyperparameters, so any remaining gap reflects the kriging refinement contributed by FasterRW rather than threshold or fusion choices.

A third method, the hybrid random-walk of [25], does not reach the sub-kelvin accuracy range targeted in this paper; we discuss this comparison separately in Section IV-C.

TABLE I Iso-accuracy comparison of FasterRW against the PIRW and FastRW baselines. For each (case, ε, method), we report the path count N, the mean steps per path (in 10^6), the mean$\pm$std of the average absolute error over $M = 16$ query points and $B = 500$ bootstrap trials, the total $N \times$ steps workload (in 10^9), and the speedup over PIRW. Speedup is the ratio of $N \times$ steps workloads; per-stage wallclock costs are reported in TABLE V.

ε	Case	Method	N	Steps (10^6)	Err$\pm$Std (K)	Workload (10^9)	Speedup
0.4	1	PIRW [26]	1280	5.99	0.40 ± 0.07	7.66	1.0×
		FastRW [29]	640	2.28	0.40 ± 0.08	1.46	5.2×
		FasterRW	384	2.28	0.40 ± 0.10	0.88	**8.7**×
	2	PIRW [26]	2560	6.00	0.40 ± 0.07	15.35	1.0×
		FastRW [29]	1280	2.29	0.40 ± 0.07	2.93	5.2×
		FasterRW	640	2.29	0.38 ± 0.08	1.47	**10.5**×
	3	PIRW [26]	1792	10.58	0.39 ± 0.07	18.96	1.0×
		FastRW [29]	1280	4.03	0.38 ± 0.07	5.16	3.7×
		FasterRW	192	4.03	0.38 ± 0.15	0.77	**24.5**×
0.5	1	PIRW [26]	768	5.99	0.48 ± 0.09	4.60	1.0×
		FastRW [29]	384	2.28	0.48 ± 0.10	0.88	5.2×
		FasterRW	192	2.28	0.47 ± 0.13	0.44	**10.5**×
	2	PIRW [26]	1280	6.00	0.48 ± 0.09	7.68	1.0×
		FastRW [29]	640	2.29	0.50 ± 0.09	1.47	5.2×
		FasterRW	384	2.29	0.45 ± 0.10	0.88	**8.7**×
	3	PIRW [26]	1024	10.58	0.48 ± 0.09	10.84	1.0×
		FastRW [29]	640	4.03	0.48 ± 0.09	2.58	4.2×
		FasterRW	96	4.03	0.49 ± 0.22	0.39	**28.0**×



(a) Case 1 (b) Case 2 (c) Case 3

Fig. 7 Bootstrap mean of the average absolute error vs. workload. Error bands show ± 1 std over 500 bootstrap trials. Dashed lines mark the iso-accuracy targets $\varepsilon \in \{0.4, 0.5\}$ K; markers indicate the operating points reported in TABLE I.

B. Main Results

We compare FasterRW against the two baselines on all three cases under two iso-accuracy targets, $\varepsilon \in \{0.4, 0.5\}$ K. For each (case, ε, method) cell, the smallest path count N that meets the target is selected, with per-cell statistics summarized over $M = 16$ uniformly sampled query points and $B = 500$ bootstrap subsamples. The subsamples are drawn with replacement from a single long Monte Carlo run at $N_{\max}$ paths per query, allowing a dense N-sweep at modest compute. Results are reported in TABLE I and visualized as bootstrap-convergence curves in Fig. 7.

FasterRW outperforms both baselines on every (case, ε) cell. Against PIRW, FasterRW achieves **8.7**–**28.0**× compute-unit speedup, with two complementary sources: shorter paths ($\sim$ 2.6× fewer steps from prior-corrected truncation) and fewer paths (from tail-reuse fusion and kriging refinement together). Against the stronger baseline FastRW [29], FasterRW still cuts the path count by 1.7–6.7×: the kriging step extrapolates the prior-error residual across the query set, an information channel that neither PIRW nor FastRW exploits. The largest gap appears on Case 3 at $\varepsilon = 0.5$ K (28.0× over PIRW), where the FEM prior is most accurate and the kriging residual is correspondingly smoothest. The bootstrap-convergence curves in Fig. 7 confirm this ordering: the FasterRW curve crosses both iso-accuracy thresholds at strictly smaller N than the FastRW baseline, which in turn crosses at strictly smaller N than PIRW, with the horizontal gap widening at the looser target.

These ratios reflect algorithmic acceleration in random-walk compute units. The corresponding end-to-end wallclock, which additionally charges the FEM-prior and posterior stages, is examined later in this section (see TABLE V).

C. Further Discussion

The headline numbers in TABLE I aggregate several distinct sources of acceleration. We now isolate them in turn, first verifying the truncation identity that underpins prior-corrected stopping, then quantifying how the kriging refinement scales with co-batched queries, examining the trade-off between prior accuracy and threshold, stress-testing robustness under deliberately weak priors, reporting the end-to-end wallclock

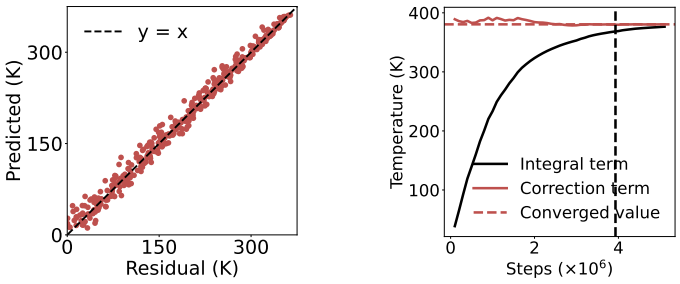


Fig. 8 (Left) Validation of Corollary 1: measured residual vs. predicted $\hat{e}_c(t)\,T(X_t)$ on sampled paths. (Right) Convergence of $\mathbb{P}_0^t$ and the corrected estimate; dashed line marks the truncation step adopted by FasterRW.

TABLE II Per-query equivalent path-count speedup $\hat{N}/N$ as a function of group size on Case 1 ($N = 1000$ paths per query, $B = 500$ bootstrap trials). Both methods grow monotonically with G; FasterRW amplifies the cross-query gain through its kriging refinement.

Group G	FastRW [29]		FasterRW	
	Err. (K)	$\hat{N}/N$	Err. (K)	$\hat{N}/N$
1	0.407	1.00×	0.407	1.00×
4	0.379	$1.20 \times \pm 0.11$	0.362	$1.68 \times \pm 0.47$
8	0.357	$1.39 \times \pm 0.15$	0.323	$2.26 \times \pm 0.71$
16	0.324	$1.85 \times \pm 0.25$	0.307	$\mathbf{3.79} \times \pm \mathbf{0.90}$

breakdown, and finally positioning FasterRW against alternative path-shortening directions.

Empirical Validation of the Truncation Identity. Our acceleration rests on Corollary 1: the expected truncation residual at time t equals $\hat{e}_c(t)\,T(X_t)$. We first verify this empirically. From the query location in Case 3, we sample 400 random-walk trajectories and terminate each at a random time so that $\hat{e}_c(t) \in [0.03, 0.95]$. For each trajectory we record both the realized residual and the predicted scaling $\hat{e}_c(t)\,T(X_t)$. The left panel of Fig. 8 shows the prediction tracking the realized residuals across the full range of $\hat{e}_c(t)$. The right panel shows the consequence for convergence: the corrected estimate $\mathbb{P}_0^t + \hat{e}_c(t)\,T(X_t)$ stabilizes far earlier than $\mathbb{P}_0^t$ alone. This is the mechanism that lets FasterRW terminate paths early without losing accuracy.

Effect of Group Size. The kriging refinement in FasterRW extrapolates the prior-error residual across co-batched queries; its benefit therefore depends on how many queries are batched together. The FastRW baseline only fuses tail-reuse pseudo-samples, without this extrapolation step. We quantify the resulting gap by holding the path count fixed at $N = 1000$ per query on Case 1 and partitioning the $M = 16$ queries into disjoint subgroups of size $G \in \{1, 4, 8, 16\}$. Each pseudo-sample is admitted only if both its source and target lie in the same subgroup, matching the user-facing scenario of co-querying G points. We then report the per-query equivalent path count $\hat{N}/N = \mathrm{Var}_{\mathrm{single}}/\mathrm{Var}_{\mathrm{fused}}$ averaged over queries; this ratio follows directly from the law-of-large-numbers scaling $\mathrm{Var}[\bar{T}] = \sigma^2/N$ in Equation (18), so a $k\times$ variance reduction is equivalent to running $k\times$ more independent paths. By construction $\hat{N}/N = 1$ at $G = 1$. The results are summarized in TABLE II.

TABLE III More accurate priors permit larger Λ and shorter random walks on Case 1. DoF denotes the degrees of freedom in the FEM prior; $\epsilon_{\max}$ is the global maximum point-wise prior error against the golden reference.

Prior		FasterRW		
DoF	$\epsilon_{\max}$ (K)	Λ	Steps (10^6)	RW Time (s)
699	8.71	0.013	2.83	2.14
1,288	3.76	0.030	2.28	1.73
10,254	1.26	0.090	**1.57**	**1.19**

The FastRW baseline grows from 1.00× to 1.85× as G goes from 1 to 16: the number of cross-target tail-reuse pass-throughs scales roughly with $G - 1$, but each pass-through carries limited per-path information under $e_{\min} = 0.3$. FasterRW grows from 1.00× to **3.79**×, roughly doubling the FastRW gain at $G = 16$. The reason is structural: the kriging step exploits the spatial smoothness of the prior-error residual, a signal that scales with the number of training points rather than with per-path pass-through statistics. The mean absolute error decreases in parallel for both methods, consistent with the variance reduction.

Prior Accuracy versus Truncation Threshold. By Equation (11), a smaller prior error $\epsilon_{\max}$ permits a larger truncation threshold Λ, which directly shortens paths. The threshold-selection rule Corollary 2 ties this trade-off to a worst-case bias bound. We test it on Case 1 by varying the FEM prior DoF. We pick three DoFs whose worst-case point-wise prior error $\epsilon_{\max}$ drops roughly monotonically (in general, FEM prior error depends on mesh layout and is not strictly monotone in DoF). For each row, Λ is chosen so that $\Lambda \times \epsilon_{\max}$ stays near 0.11 K, well below both iso-accuracy targets. TABLE III summarizes the result.

Improving prior accuracy nearly halves both the mean steps per path (from 2.83×10^6 to 1.57×10^6) and the per-query random-walk wallclock (from 2.14 s to 1.19 s). Moreover, the FEM prior is computed once per case and can be reused across repeated queries, so in any workflow that probes the same chip many times, paying for a finer prior is essentially a free lunch.

Robustness to Weak Priors. The previous paragraph shows that better priors help. Does FasterRW also *need* them? We stress-test this on Case 1 with a deliberately crude rule-of-thumb prior: a uniform temperature equal to ambient across all voxels. This prior has maximum error 43.7 K, since the hottest voxel sits $\sim$ 44 K above ambient. Under Corollary 2, $\Lambda = 0.001$ bounds the threshold-induced correction bias by $0.001 \times 43.7 \approx 0.044$ K, well below the $\varepsilon = 0.4$ K target, while remaining 10× larger than PIRW's threshold. TABLE IV reports FasterRW and the FastRW baseline under this rule-of-thumb prior, alongside the FEM-prior reference.

Even under this crude 43.7 K-error prior, FasterRW reaches a 3.32× wallclock speedup over PIRW—and the FastRW baseline still gets 1.66×. This confirms the key implication of Equation (11): the prior need not be accurate enough to solve the problem on its own. It only needs to shrink the residual scale from the absolute temperature magnitude ($\sim$ 350 K) to the prior-error magnitude; the random-walk correction recov-

TABLE IV Robustness of FasterRW under a rule-of-thumb (uniform-at-ambient) prior on Case 1 at $\varepsilon = 0.4\,\mathrm{K}$. Time/pt is per-query random-walk wallclock on Apple M5 Pro. Speedup is per-query wallclock vs. PIRW.

Method	Prior	$\epsilon_{\max}$ (K)	Λ	Paths	Steps (10^6)	Time/pt (s)	Speedup
PIRW [26]	none	N/A	10^{-4}	1280	5.99	5.78	1.00×
FastRW [29]	rule-of-thumb	43.7	10^{-3}	1024	4.49	3.48	1.66×
FasterRW	rule-of-thumb	43.7	10^{-3}	512	4.49	1.74	3.32×
FastRW [29]	FEM	3.76	0.030	640	2.28	1.11	5.21×
FasterRW	FEM	3.76	0.030	384	2.28	0.66	**8.76**×

TABLE V Per-query wallclock breakdown on Case 1 at $\varepsilon = 0.4\,\mathrm{K}$, measured on Apple M5 Pro. The FEM prior is amortized over $M = 16$ queries and reported as the steady-state MUMPS linear-solve cost† (excluding one-time mesh-build overhead).

Stage (s)	PIRW	FastRW	FasterRW
FEM prior	—	$< 0.06^{\dagger}$	$< 0.06^{\dagger}$
Random walk	5.78	1.11	0.66
Tail-reuse fusion	—	0.02	0.01
Kriging refinement	—	—	0.02
Total per query	**5.78**	< 1.19	$<$ **0.76**
Wallclock speedup	1.00×	> 4.88×	$>$ **7.61**×

† Commercial tool logs solve time as 0 s (integer-second rounding) for the DoF= 1288 prior; the total is therefore < 1 s, i.e., < 0.06 s amortized over $M = 16$ queries.

ers the remaining accuracy. Switching to the FEM prior shrinks the residual scale by another order of magnitude and unlocks roughly 3× more for both methods: **8.76**× for FasterRW and 5.21× for FastRW. FasterRW's ~ 2× advantage over FastRW persists across the prior-quality spectrum, because the kriging step exploits residual smoothness whether the residual is small (FEM prior) or large (the rule-of-thumb prior leaves the entire physics field as residual, still spatially smooth, just full-amplitude).

End-to-end Wallclock Breakdown. FasterRW adds three stages over PIRW: a FEM-prior stage, a tail-reuse fusion stage, and a kriging refinement stage. The FastRW baseline shares the first two and lacks only the kriging refinement. We report per-query wallclock on Case 1 at $\varepsilon = 0.4\,\mathrm{K}$, broken down by stage, in TABLE V.

End-to-end, FasterRW reaches $>$ **7.61**× wallclock speedup over PIRW on this configuration. This is slightly smaller than the 8.7× compute-unit speedup reported in TABLE I; the residual gap is the FEM-prior cost, charged here as the steady-state MUMPS linear solve (< 0.06 s per query, amortized over $M = 16$ queries). The tail-reuse fusion and kriging refinement together contribute 0.03 s per query, confirming that the speedup gains arise from path- and step-count reductions, not from a cheaper backend. For context, a ground-truth FEM solve on Case 1 takes more than 450 s, two orders of magnitude longer than the entire FasterRW per-query budget, which confirms that the coarse FEM serves as an inexpensive prior rather than a replacement for the random-walk solver.

Comparison with Alternative Methods. We briefly position FasterRW against two alternative path-shortening directions that we considered but excluded from the main tables.

The hybrid random-walk method of [25] replaces parts of the Feynman–Kac formulation with a coarser heuristic for Robin boundaries. We re-implemented it in our setting; the maximum error reaches 7.39 K on Case 1, well outside the sub-kelvin range targeted in this paper, and the other cases show similar behavior. We therefore retain PIRW as the strongest direct baseline in the same reflected Feynman–Kac regime.

A second direction is the empirical curve fit used by the GPU PIRW variant [27]:

$$\mathbb{P}_0^t = T(\boldsymbol{x}) - a\,\exp(b\,t + c), \tag{34a}$$

$$\hat{T}(\boldsymbol{x}) = \frac{\mathbb{P}_0^{t-l}\,\mathbb{P}_0^{t-l} - \mathbb{P}_0^{t}\,\mathbb{P}_0^{t-2l}}{2\,\mathbb{P}_0^{t-l} - \mathbb{P}_0^{t} - \mathbb{P}_0^{t-2l}}, \tag{34b}$$

where l is a fixed sampling gap and a, b, c are unknown constants. Comparing Equation (34a) with Equations (7) and (8) shows that Equation (34a) is an approximation that assumes a strict map between the functional $\hat{e}_c(t)$ and the step count t. In practice, random fluctuations along walks make Equation (34b) unstable, and we did not obtain reasonable results in our implementations. Equations (34a) and (34b) also offer no analogue of FasterRW's multi-point fusion or threshold-selection rule, and so we exclude this approach from the main tables.

V. Conclusion

This paper introduced FasterRW, a random-walk solver that accelerates steady-state thermal analysis by quantitatively analyzing truncation error and incorporating both prior estimation and posterior correction. The truncation residual identity explains why prior-assisted truncation can use substantially larger thresholds while keeping the correction error controlled by the prior error rather than the absolute temperature scale, leading to an explicit threshold-selection rule. The multi-point formulation further reuses the tail segment of every random-walk path as an approximately unbiased pseudo-sample for the other query points and, in prior-error coordinates, refines the result through a universal-kriging Gaussian-process prior on the prior-error field. By leveraging coarse FEM-based priors together with tail-reuse fusion and kriging refinement, FasterRW achieves 8.7–28× compute-unit speedup and over 7.6× end-to-end wallclock speedup over PIRW across the studied cases while maintaining sub-kelvin accuracy, with negligible posterior-correction overhead. Expanded experiments validate the residual behavior, path convergence, group-size effect, prior-accuracy trade-off, robustness under weak priors, and end-to-end runtime breakdown. These results highlight FasterRW as a practical tool for efficient hotspot estimation, and future extensions may explore more accurate priors, in-

cluding learning-based priors, transient conduction, and hardware acceleration.

ACKNOWLEDGMENT

The authors used ChatGPT and Codex only for language polishing and editorial assistance, including minor wording suggestions. All technical content and results were created and approved by the authors.

REFERENCES

[1] Semiconductor Research Corporation, "Microelectronics and advanced packaging technologies roadmap," Semiconductor Research Corporation, Tech. Rep., 2025. [Online]. Available: https://srcmapt.org
[2] K. Puttaswamy and G. H. Loh, "Thermal analysis of a 3d die-stacked high-performance microprocessor," in *Proc. GLSVLSI*, 2006, pp. 19–24.
[3] H. Sultan and S. R. Sarangi, "Varsim: A fast process variation-aware thermal modeling methodology using green's functions," *Microelectronics Journal*, vol. 142, p. 105995, 2023.
[4] Y. Ma, L. Delshadtehrani, C. Demirkiran, J. L. Abellan, and A. Joshi, "Tap-2.5 d: A thermally-aware chiplet placement methodology for 2.5 d systems," in *Proc. DATE*. IEEE, 2021, pp. 1246–1251.
[5] D.-W. Wang, B.-W. Zhang, L.-T. Wang, P. Zhang, and W.-S. Zhao, "A proposal of fast thermal simulation method for 2.5-d advanced packaging to enable efficient thermal-aware placement optimization," *IEEE TCAD*, pp. 1–1, 2025.
[6] Y. Zhan and S. S. Sapatnekar, "High-efficiency green function-based thermal simulation algorithms," *IEEE TCAD*, vol. 26, no. 9, pp. 1661–1675, 2007.
[7] B. Wang and P. Mazumder, "Accelerated chip-level thermal analysis using multilayer green's function," *IEEE TCAD*, vol. 26, no. 2, pp. 325–344, 2007.
[8] S. Ladenheim, Y.-C. Chen, M. Mihajlović, and V. F. Pavlidis, "The mta: An advanced and versatile thermal simulator for integrated systems," *IEEE TCAD*, vol. 37, no. 12, pp. 3123–3136, 2018.
[9] S. Ladenheim, Y.-C. Chen, M. Mihajlović, and V. Pavlidis, "Ic thermal analyzer for versatile 3-d structures using multigrid preconditioned krylov methods," in *Proc. ICCAD*, 2016, pp. 1–8.
[10] T.-Y. Wang and C. Chen, "Thermal-adi - a linear-time chip-level dynamic thermal-simulation algorithm based on alternating-direction-implicit (adi) method," *IEEE TVLSI*, vol. 11, no. 4, pp. 691–700, 2003.
[11] T.-Y. Wang and C. C.-P. Chen, "3-d thermal-adi: a linear-time chip level transient thermal simulator," *IEEE TCAD*, vol. 21, no. 12, pp. 1434–1445, 2002.
[12] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. Stan, "Hotspot: a compact thermal modeling methodology for early-stage vlsi design," *IEEE TVLSI*, vol. 14, no. 5, pp. 501–513, 2006.
[13] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza, "3d-ice: Fast compact transient thermal modeling for 3d ics with inter-tier liquid cooling," in *Proc. ICCAD*, 2010, pp. 463–470.
[14] 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 TCAD*, vol. 41, no. 4, pp. 1048–1061, 2022.
[15] 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 *Proc. ASPDAC*, 2021, pp. 690–696.
[16] L. Chen, W. Jin, and S. X.-D. Tan, "Fast thermal analysis for chiplet design based on graph convolution networks," in *Proc. ASPDAC*, 2022, pp. 485–492.
[17] Z. Liu, Y. Li, J. Hu, X. Yu, S. Shiau, X. Ai, Z. Zeng, and Z. Zhang, "Deepoheat: operator learning-based ultra-fast thermal simulation in 3d-ic design," in *Proc. DAC*, 2023, pp. 1–6.
[18] D. Zhang, D. Niu, Z. Jin, Y. Dong, J. Tan, and C. Sun, "A novel frequency-spatial domain aware network for fast thermal prediction in 2.5d ics," in *Proc. DATE*, 2025, pp. 1–6.
[19] S. Hwang, M. J. Smith, V. C. Do Nascimento, Q. Qiu, C.-K. Koh, G. Subbarayan, and D. Jiao, "Real-time 3-d thermal simulation of advanced packages via generative adversarial networks," *IEEE TCAD*, vol. 44, no. 7, pp. 2439–2450, 2025.
[20] M. Wang, Y. Cheng, W. Zeng, Z. Lu, V. F. Pavlidis, and W. Xing, "Aro: Autoregressive operator learning for transferable and multi-fidelity 3d-ic thermal analysis with active learning," in *Proc. ICCAD*, 2025, pp. 1–9.
[21] Y. Zhan, B. Goplen, and S. S. Sapatnekar, "Electrothermal analysis and optimization techniques for nanoscale integrated circuits," in *Proc. ASPDAC*, 2006, pp. 219–222.
[22] E. Wong and S. K. Lim, "3d floorplanning with thermal vias," in *Proc. DATE*, vol. 1, 2006, pp. 1–6.
[23] H. Qian, S. Nassif, and S. Sapatnekar, "Power grid analysis using random walks," *IEEE TCAD*, vol. 24, no. 8, pp. 1204–1224, 2005.
[24] M. Visvardis, P. Liaskovitis, and E. Efstathiou, "Deep-learning-driven random walk method for capacitance extraction," *IEEE TCAD*, vol. 42, no. 8, pp. 2643–2656, 2023.
[25] Y. Liang, W. Yu, and H. Qian, "A hybrid random walk algorithm for 3-d thermal analysis of integrated circuits," in *Proc. ASPDAC*, 2014, pp. 849–854.
[26] L. Yang, C. Ding, C. Yan, D. Zhou, and X. Zeng, "A high-precision stochastic solver for steady-state thermal analysis with fourier heat transfer robin boundary conditions," in *Proc. ICCAD*, 2022, pp. 1–9.
[27] Y. Luo, Y. Wang, Z. Dong, C. Yan, Z. Bi, K. Zhu, S.-G. Wang, and X. Zeng, "Gpu acceleration of a high-precision stochastic solver for steady-state thermal analysis with robin boundary conditions," in *Proc. ISEDA*, 2025, pp. 287–292.
[28] Z. Dong, L. Yang, C. Ding, C. Yan, Z. Bi, S.-G. Wang, D. Zhou, and X. Zeng, "ppirw: An efficient and accurate precalculation path integral random walk solver for steady-state thermal simulation with robin boundary conditions," *IEEE TCAD*, vol. 44, no. 4, pp. 1489–1502, 2025.
[29] Z. Wang, T. Hou, C. Wang, Z. Zhuang, T.-Y. Ho, F. Farnia, and B. Yu, "FastRW: An efficient random walk method for steady-state thermal analysis," in *Proc. DATE*, Verona, Italy, Apr. 2026.
[30] R. Sawhney, B. Miller, I. Gkioulekas, and K. Crane, "Walk on stars: A grid-free monte carlo method for pdes with neumann boundary conditions," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–20, 2023.
[31] B. Oksendal, *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.
[32] Y. Zhou, W. Cai, and E. Hsu, "Computation of local time of reflecting brownian motion and probabilistic representation of the neumann problem," *arXiv preprint*, 2015, arXiv:1502.01319.

APPENDIX A
PROOF OF THE TRUNCATION RESIDUAL IDENTITY

This appendix proves Theorem 1. The same argument applies to the discrete-time Markov chain estimators used in implementation: conditioning at a stopping time and restarting from the truncation state yields the residual identity, with the continuous reflected diffusion serving as a limiting interpretation.

A. Proof

Let $\{\mathcal{F}_t\}_{t\geq 0}$ be the natural filtration generated by the reflected process $\{X_t\}_{t\geq 0}$ and its local time $\{L(t)\}_{t\geq 0}$. Consider a stopping time τ such that $\tau < t_D$ on the event of interest, and define the post-τ shifted process $\tilde{X}_s \triangleq X_{\tau+s}$ for $s \geq 0$. Let $\tilde{L}(s)$ denote the boundary local time accumulated by $\tilde{X}$ after time τ, so that $L(\tau + s) = L(\tau) + \tilde{L}(s)$. Define the shifted weight

$$\tilde{e}_c(s) \triangleq \exp\left(- \int_0^s c(\tilde{X}_u)\, \mathrm{d}\tilde{L}(u) \right). \tag{35}$$

By construction,

$$\hat{e}_c(\tau + s) = \hat{e}_c(\tau)\, \tilde{e}_c(s). \tag{36}$$

Define $\tilde{\mathbb{P}}_0^{\infty}$ as the full Feynman–Kac functional evaluated along the shifted path $\tilde{X}$ starting from $\tilde{X}_0 = X_\tau$, with the shifted weight $\tilde{e}_c(\cdot)$. By Equation (36), the truncation tail along the original path factors as

$$\mathbb{P}_\tau^{\infty} = \hat{e}_c(\tau)\, \tilde{\mathbb{P}}_0^{\infty}. \tag{37}$$

Condition on $\mathcal{F}_\tau$. By the strong Markov property of the reflected process at the stopping time τ, the conditional law of the shifted process $(\tilde{X}_s, \tilde{L}(s))_{s\geq 0}$ given $\mathcal{F}_\tau$ is the same as that of a fresh reflected process started from X_τ, and is independent of the past beyond the dependence through X_τ. Therefore, by the definition of $T(\cdot)$ in Equation (2),

$$\mathbb{E}\Big[\tilde{\mathbb{P}}_0^\infty \mid \mathcal{F}_\tau\Big] = T(X_\tau). \tag{38}$$

Combining Equations (37) and (38) yields

$$\mathbb{E}[\mathbb{P}_\tau^\infty \mid \mathcal{F}_\tau] = \hat{e}_c(\tau)\, T(X_\tau), \tag{39}$$

which completes the proof.

Appendix B
Bias–Variance Analysis of Prior-Based Truncation Correction

This appendix extends the prior-correction bias result in Theorem 2 and Corollary 2 with a mean-squared-error (MSE) decomposition, an explicit bias–variance split after averaging, and a discussion of how the truncation threshold Λ trades the two. The conclusion is that, in the operating regime targeted by FasterRW, the per-path variance $\sigma^2(\boldsymbol{x})$ has only a weak dependence on Λ, so Λ can be chosen from the bias bound and N from the variance budget without coupling the two knobs.

A. MSE Decomposition

The prior-corrected estimator in Equation (11) can be viewed as the ideal unbiased estimator in Equation (12) plus a weighted prior-error term. Let $\Delta(\boldsymbol{x}) \triangleq \hat{T}_i^\star(\boldsymbol{x}) - T(\boldsymbol{x})$ denote the intrinsic Monte Carlo error of the ideal estimator. From Equation (13),

$$\hat{T}_i(\boldsymbol{x}) - T(\boldsymbol{x}) = \Delta(\boldsymbol{x}) + \hat{e}_c(\tau)\,\epsilon(X_\tau). \tag{40}$$

Therefore, the per-path MSE admits the decomposition

$$\begin{aligned}\mathbb{E}\Big[(\hat{T}_i(\boldsymbol{x}) - T(\boldsymbol{x}))^2\Big] &= \underbrace{\mathbb{E}\big[\Delta(\boldsymbol{x})^2\big]}_{\text{ideal MC variance}} + 2\,\underbrace{\mathbb{E}[\Delta(\boldsymbol{x})\,\hat{e}_c(\tau)\,\epsilon(X_\tau)]}_{\text{cross term}} \\ &\quad + \underbrace{\mathbb{E}\big[\hat{e}_c(\tau)^2\,\epsilon(X_\tau)^2\big]}_{\text{plug-in contribution}}.\end{aligned} \tag{41}$$

Using Cauchy–Schwarz, the cross term is bounded by

$$\begin{aligned}&|\mathbb{E}[\Delta(\boldsymbol{x})\,\hat{e}_c(\tau)\,\epsilon(X_\tau)]| \\ &\quad \leq \sqrt{\mathbb{E}[\Delta(\boldsymbol{x})^2]}\,\sqrt{\mathbb{E}[\hat{e}_c(\tau)^2\,\epsilon(X_\tau)^2]},\end{aligned} \tag{42}$$

which yields the convenient upper bound

$$\begin{aligned}&\mathbb{E}\Big[(\hat{T}_i(\boldsymbol{x}) - T(\boldsymbol{x}))^2\Big] \\ &\quad \leq \Big(\sqrt{\mathbb{E}[\Delta(\boldsymbol{x})^2]} + \sqrt{\mathbb{E}[\hat{e}_c(\tau)^2\,\epsilon(X_\tau)^2]}\Big)^2.\end{aligned} \tag{43}$$

This bound is tight only when $\Delta(\boldsymbol{x})$ and $\hat{e}_c(\tau)\,\epsilon(X_\tau)$ are perfectly correlated. This is not expected in our setting: $\Delta(\boldsymbol{x})$ aggregates Monte Carlo noise over the entire reflected Brownian path including the source/boundary excursions, while $\hat{e}_c(\tau)\,\epsilon(X_\tau)$ depends only on the prior-error field evaluated at the truncation location. Treating the two contributions as approximately uncorrelated gives the working approximation

$$\mathbb{E}\Big[(\hat{T}_i(\boldsymbol{x}) - T(\boldsymbol{x}))^2\Big] \approx \mathbb{E}\big[\Delta(\boldsymbol{x})^2\big] + \mathbb{E}\big[\hat{e}_c(\tau)^2\,\epsilon(X_\tau)^2\big], \tag{44}$$

which separates the contribution of the underlying Monte Carlo estimator from the contribution of the prior-error plug-in.

B. Explicit Bias–Variance Split After Averaging

FasterRW averages N independent paths as in Equation (17). Let $\bar{T}(\boldsymbol{x}) \triangleq \frac{1}{N}\sum_{i=1}^N \hat{T}_i(\boldsymbol{x})$. The standard bias–variance decomposition gives

$$\mathbb{E}\big[(\bar{T}(\boldsymbol{x}) - T(\boldsymbol{x}))^2\big] = \underbrace{\big(\mathbb{E}[\hat{e}_c(\tau)\,\epsilon(X_\tau)]\big)^2}_{\text{squared bias}} + \underbrace{\frac{\sigma^2(\boldsymbol{x})}{N}}_{\text{variance}}, \tag{45}$$

where the bias term is taken from Equation (14) and the variance from Equation (18). Under the threshold rule $\hat{e}_c(\tau) \leq \Lambda$, the squared-bias factor is uniformly bounded by $\Lambda^2\,\epsilon_{\max}^2$ as in Corollary 2. The variance factor decays as $1/N$ and is governed by the per-path variance $\sigma^2(\boldsymbol{x})$, which is itself a function of Λ analyzed below.

C. Behavior of $\sigma^2(\boldsymbol{x})$ Under Λ

The per-path variance $\sigma^2(\boldsymbol{x}) = \mathrm{Var}[\hat{T}_i(\boldsymbol{x})]$ depends on Λ through the stopping location and the returned continuation value:

- *Shorter pre-truncation paths.* Increasing Λ triggers the condition $\hat{e}_c(\tau) \leq \Lambda$ earlier, so the partial Feynman–Kac sum $\mathbb{P}_0^\tau$ accumulates fewer source and boundary contributions.
- *Larger continuation value.* Earlier stopping also increases the magnitude of the returned continuation term $\hat{e}_c(\tau)\,\tilde{T}(X_\tau)$, whose realization depends on the random terminal location X_τ.

The continuation term itself is on the temperature scale; it is not bounded by $\Lambda\epsilon_{\max}$. What the prior-error analysis controls is the extra discrepancy between the practical estimator and the ideal continuation estimator, namely $\hat{e}_c(\tau)\epsilon(X_\tau)$ in Equation (13). Under $\hat{e}_c(\tau) \leq \Lambda$ this additional error is bounded by $\Lambda\epsilon_{\max}$, while the remaining variance of the ideal continuation estimator is governed by the reflected process and the geometry. Empirically, $\sigma^2(\boldsymbol{x})$ exhibits only a weak dependence on Λ in the tested range; this is what allows the path-count knob N to be chosen essentially independently of Λ.

D. Practical Implications

The decoupling above justifies the two-step tuning protocol used in FasterRW:

1) Given a target bias tolerance b_{tol} and a prior error bound $\epsilon_{\max}$, choose $\Lambda \leq b_{\mathrm{tol}}/\epsilon_{\max}$ via Corollary 2.
2) Given a target variance tolerance, choose N so that $\sigma^2(\boldsymbol{x})/N$ is within budget via Equation (18).

We do not attempt to solve for the MSE-optimal Λ jointly with N in closed form: the per-path variance $\sigma^2(\boldsymbol{x})$ is not available analytically for the reflected Brownian process under mixed boundary conditions, and the dependence of both path length and terminal location on Λ is geometry-specific. Instead, the bias bound provides a conservative, deployment-ready rule, and the variance is handled separately through path averaging. The weak-prior experiment in TABLE IV validates this protocol even when $\epsilon_{\max}$ is intentionally large.

APPENDIX C
REML DERIVATION FOR UNIVERSAL KRIGING IN PRIOR-ERROR COORDINATES

This appendix derives the closed-form expressions for the REML marginal likelihood and the universal-kriging posterior used in Section III-D. We adopt the model in Equations (26) and (27):

$$\begin{aligned} \boldsymbol{d} \mid \boldsymbol{\epsilon} &\sim \mathcal{N}(\boldsymbol{\epsilon},\, \boldsymbol{\Sigma}_{\mathrm{MC}}), \\ \boldsymbol{\epsilon} &= \mu\,\mathbf{1} + \mathbf{g}, \\ \mathbf{g} &\sim \mathcal{N}(\mathbf{0},\, \boldsymbol{K}_\ell), \qquad p(\mu) \propto 1, \end{aligned} \tag{46}$$

where $\boldsymbol{d} \in \mathbb{R}^M$ is the observation vector, $\boldsymbol{\epsilon}$ the stacked prior-error vector, $\boldsymbol{\Sigma}_{\mathrm{MC}}$ the diagonal Monte Carlo noise covariance, μ the unknown global drift, and $\mathbf{g}$ the zero-mean spatial residual with RBF covariance $\boldsymbol{K}_\ell$. Define the total covariance $\boldsymbol{C} \triangleq \boldsymbol{K}_\ell + \boldsymbol{\Sigma}_{\mathrm{MC}}$.

A. REML Marginal Likelihood

Marginalising over $\boldsymbol{\epsilon}$ gives the conditional distribution $\boldsymbol{d} \mid \mu \sim \mathcal{N}(\mu\,\mathbf{1},\, \boldsymbol{C})$. The joint density of $(\boldsymbol{d}, \mu)$ is, up to a constant,

$$p(\boldsymbol{d}, \mu) \;\propto\; (\det \boldsymbol{C})^{-1/2} \exp\!\big(-\tfrac{1}{2}(\boldsymbol{d} - \mu\mathbf{1})^\top \boldsymbol{C}^{-1}(\boldsymbol{d} - \mu\mathbf{1})\big)\,. \tag{47}$$

Expanding the quadratic in μ and defining $A \triangleq \mathbf{1}^\top \boldsymbol{C}^{-1}\mathbf{1}$ and $B \triangleq \mathbf{1}^\top \boldsymbol{C}^{-1}\boldsymbol{d}$ yields

$$(\boldsymbol{d}-\mu\mathbf{1})^\top \boldsymbol{C}^{-1}(\boldsymbol{d}-\mu\mathbf{1}) = A\big(\mu - \tfrac{B}{A}\big)^2 + \boldsymbol{d}^\top \boldsymbol{C}^{-1}\boldsymbol{d} - \tfrac{B^2}{A}. \tag{48}$$

Integrating μ over $\mathbb{R}$ against the flat prior,

$$\int_{\mathbb{R}} \exp\!\big(-\tfrac{1}{2}A(\mu - B/A)^2\big)\,\mathrm{d}\mu = \sqrt{2\pi/A}, \tag{49}$$

gives the marginal density of $\boldsymbol{d}$:

$$\begin{aligned} p(\boldsymbol{d}) &\propto (\det \boldsymbol{C})^{-1/2}\, A^{-1/2} \exp\!\Big(-\tfrac{1}{2}\,\boldsymbol{d}^\top \boldsymbol{C}^{-1}\boldsymbol{d} + \tfrac{1}{2}\tfrac{B^2}{A}\Big) \\ &= (\det \boldsymbol{C})^{-1/2}\, A^{-1/2} \exp\!\big(-\tfrac{1}{2}\mathbf{r}^\top \boldsymbol{C}^{-1}\mathbf{r}\big)\,, \end{aligned} \tag{50}$$

where $\mathbf{r} \triangleq \boldsymbol{d} - \hat{\mu}\mathbf{1}$ and $\hat{\mu} \triangleq B/A$ is the generalized-least-squares estimate of the drift. Taking the logarithm reproduces Equations (29) and (30). The estimator $\hat{\mu}$ is the best linear unbiased estimator (BLUE) of μ under the known $\boldsymbol{C}$.

B. Posterior Mean and Variance of $\boldsymbol{\epsilon}$

By Bayes' rule, conditional on μ, the posterior of $\boldsymbol{\epsilon}$ given $\boldsymbol{d}$ is jointly Gaussian. From the linear-Gaussian observation model

$$\boldsymbol{d} \mid \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon},\, \boldsymbol{\Sigma}_{\mathrm{MC}}), \qquad \boldsymbol{\epsilon} \mid \mu \sim \mathcal{N}(\mu\,\mathbf{1},\, \boldsymbol{K}_\ell), \tag{51}$$

standard conjugacy gives

$$\mathbb{E}[\boldsymbol{\epsilon} \mid \boldsymbol{d}, \mu] = \mu\,\mathbf{1} + \boldsymbol{K}_\ell\, \boldsymbol{C}^{-1}(\boldsymbol{d} - \mu\,\mathbf{1}), \tag{52}$$

$$\mathrm{Cov}[\boldsymbol{\epsilon} \mid \boldsymbol{d}, \mu] = \boldsymbol{K}_\ell - \boldsymbol{K}_\ell\, \boldsymbol{C}^{-1}\, \boldsymbol{K}_\ell. \tag{53}$$

The posterior of $\mu \mid \boldsymbol{d}$ is Gaussian with mean $\hat{\mu}$ and variance $1/A$. Marginalising over μ, the posterior mean of the stacked prior-error vector is

$$\mathbb{E}[\boldsymbol{\epsilon} \mid \boldsymbol{d}] = \hat{\mu}\,\mathbf{1} + \boldsymbol{K}_\ell\, \boldsymbol{C}^{-1}(\boldsymbol{d} - \hat{\mu}\,\mathbf{1}), \tag{54}$$

which is Equation (31).

For the posterior variance, write $\epsilon_i \triangleq \epsilon(\boldsymbol{x}_i)$ for the i-th component of $\boldsymbol{\epsilon}$. The law of total variance gives, at each index i,

$$\begin{aligned} \mathrm{Var}[\epsilon_i \mid \boldsymbol{d}] &= \mathbb{E}_\mu[\mathrm{Var}[\epsilon_i \mid \boldsymbol{d}, \mu]] + \mathrm{Var}_\mu[\mathbb{E}[\epsilon_i \mid \boldsymbol{d}, \mu]] \\ &= \big(\boldsymbol{K}_\ell - \boldsymbol{K}_\ell \boldsymbol{C}^{-1} \boldsymbol{K}_\ell\big)_{ii} \\ &\quad + \big(1 - (\mathbf{1}^\top \boldsymbol{C}^{-1}\boldsymbol{K}_\ell)_i\big)^2\, \mathrm{Var}[\mu \mid \boldsymbol{d}]. \end{aligned} \tag{55}$$

Substituting $\mathrm{Var}[\mu \mid \boldsymbol{d}] = 1/A$ gives Equation (32). The second term captures the residual uncertainty about the global drift μ propagated to ϵ_i; it vanishes precisely when $(\mathbf{1}^\top \boldsymbol{C}^{-1}\boldsymbol{K}_\ell)_i = 1$, i.e., when the row of $\boldsymbol{C}^{-1}\boldsymbol{K}_\ell$ sums to one.

C. Algorithmic Use

In practice the matrix inversions in Equations (29), (31) and (32) are computed once per hyperparameter candidate via a single Cholesky factorisation of $\boldsymbol{C}$. The selection over a grid in $(\ell, \sigma_\epsilon^2)$ then costs $O(|grid| \cdot M^3)$, which is negligible for M in the tens.