FastRW: An Efficient Random Walk Method for Steady-State Thermal Analysis

Zixiao Wang, Tianshu Hou, Chenghan Wang, Zhen Zhuang, Tsung-Yi Ho, Farzan Farnia, Bei Yu
The Chinese University of Hong Kong

Key Results

FastRW accelerates Feynman–Kac-based random walk thermal simulation through exact truncation error analysis and prior/posterior estimation, achieving **over 6.5× speedup** over prior state-of-the-art methods without sacrificing accuracy.

Motivation

Accurate thermal simulation is critical for modern IC design with increasing power density. Random walk methods based on the Feynman–Kac formula [4] enable efficient **localized** temperature estimation without solving the global field [5, 1, 3]. However:

- In practical scenarios **without Dirichlet boundaries**, long random walk paths are needed for convergence.
- Existing methods [2, 3] select truncation thresholds **empirically** with no principled error guarantee.
- **FastRW (ours)**: provides quantitative truncation error analysis and uses prior/posterior estimation to dramatically shorten paths.

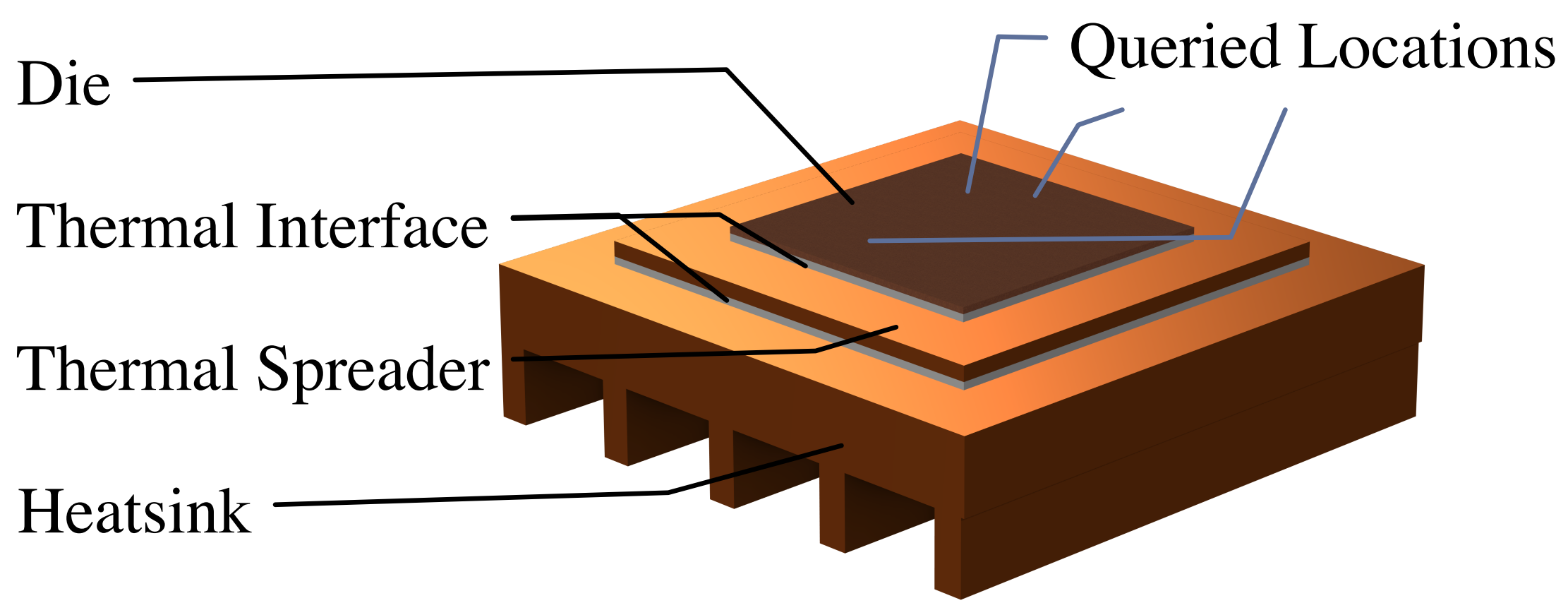

Figure 1. In many thermal design scenarios, only temperatures at a few queried hotspot locations are needed — ideal for random walk methods.

Feynman–Kac Representation

The temperature at $\boldsymbol{x}$ is expressed via the Feynman–Kac formula as an expectation over reflecting Brownian motion (SRBM) paths:

$$\begin{aligned}T(\boldsymbol{x}) = \mathbb{E}\left\{\int_0^\infty \hat{e}_c(t)\, f(X_t)\,\mathrm{d}t\right\} + \mathbb{E}\left\{\hat{e}_c(t_D)\,\phi_D(X_{t_D})\right\} \\ + \mathbb{E}\left\{\int_0^\infty \hat{e}_c(t)\,\phi_N(X_t)\,\mathrm{d}L(t)\right\} + \mathbb{E}\left\{\int_0^\infty \hat{e}_c(t)\,\phi_R(X_t)\,\mathrm{d}L(t)\right\},\end{aligned} \tag{1}$$

where $\hat{e}_c(t) = \exp\left(-\int_0^t c(X_s)\,\mathrm{d}L(s)\right)$ and $L(t)$ is the local time at the boundary.

Truncation Error Analysis

For a threshold $\Lambda \in (0,1)$, we truncate the path at t_Λ when $\hat{e}_c(t) \le \Lambda$. The key result:

$$\mathbb{E}\left[\mathbb{P}_{t_\Lambda}^{\infty}\right] = \hat{e}_c(t_\Lambda)\, T\big(X_{t_\Lambda}\big) = \Lambda\, T\big(X_{t_\Lambda}\big). \tag{2}$$

Corollary: The expected truncation error equals the **weighted temperature** at the truncation location X_t. This provides a principled guideline for choosing Λ.

Single-Point Acceleration via Prior Estimation

We use a coarse FEM prior $\tilde{T}(\boldsymbol{x}) = T(\boldsymbol{x}) + \epsilon$ to compensate the truncation error:

$$T_i(\boldsymbol{x}) = \mathbb{P}_0^t + \hat{e}_c(t)\,\tilde{T}(\boldsymbol{x}) = \mathbb{P}_0^t + \mathbb{E}[\mathbb{P}_t^{\infty}] + \hat{e}_c(t)\,\epsilon. \tag{3}$$

Since $\epsilon \ll T(\boldsymbol{x})$, the threshold Λ can be **350× larger**, cutting path length by more than half.

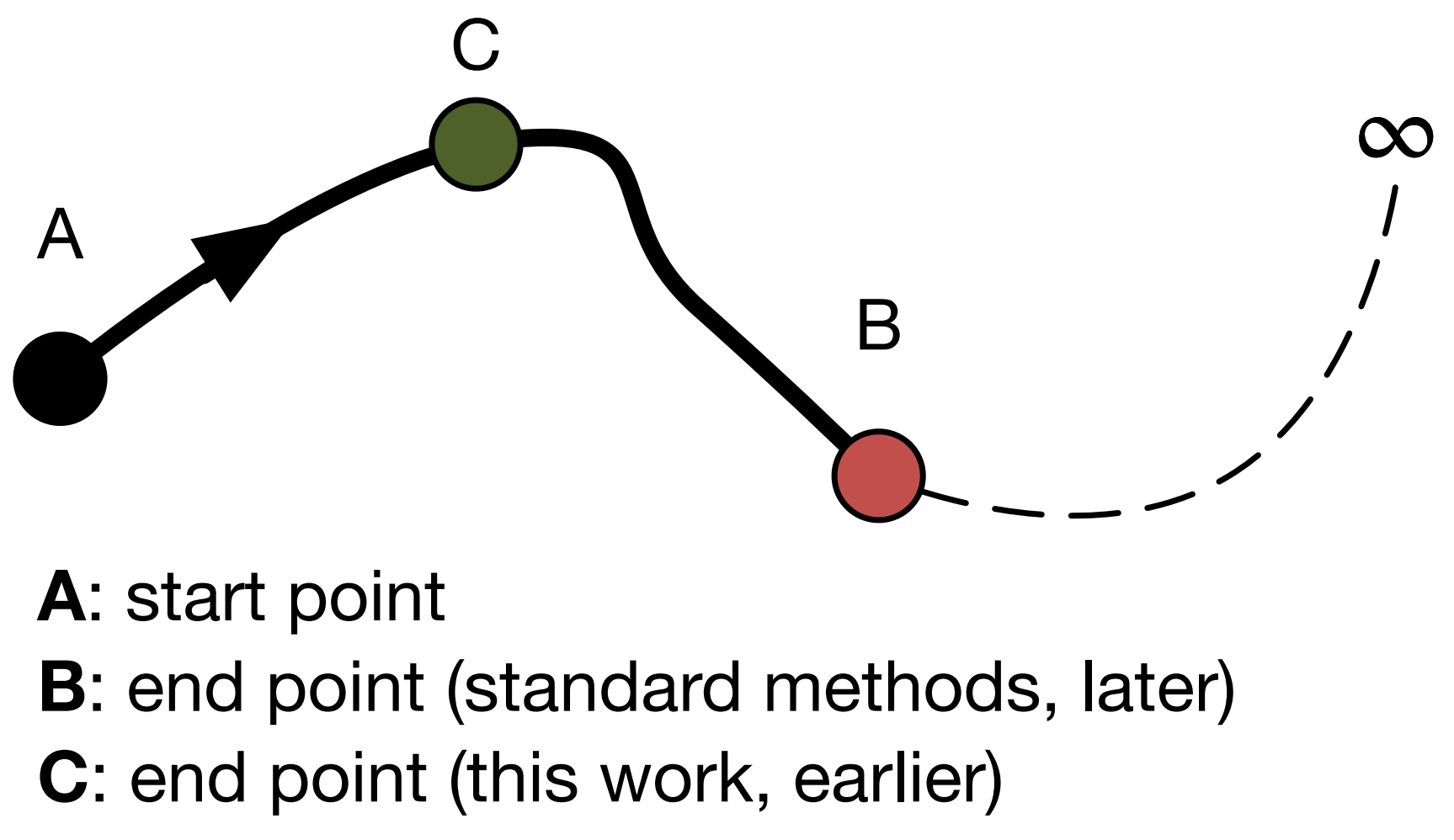

Figure 2. Standard methods compute the full path A→B. **FastRW** truncates at C and estimates the remaining contribution using the prior.

Multi-Point Acceleration via Posterior Observation

When estimating temperatures at M points $\{\boldsymbol{x}_q\}_{q=1}^{M}$, paths starting at $\boldsymbol{x}$ that visit $\boldsymbol{y}$ yield a noisy linear observation:

$$b_k = \mathbf{T}_{i_k} - \alpha_k\,\mathbf{T}_{j_k} + \eta_k, \quad \text{stacked as } \mathbf{B} = \boldsymbol{A}\,\mathbf{T} + \boldsymbol{\eta}. \tag{4}$$

With Bayesian inference (Gaussian prior from single-point estimates):

$$\hat{\mathbf{T}} = \boldsymbol{\Sigma}_{\text{post}}\left(\boldsymbol{A}^\top \boldsymbol{R}^{-1}\mathbf{B} + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right), \tag{5}$$

cross-point information further reduces the number of paths needed.

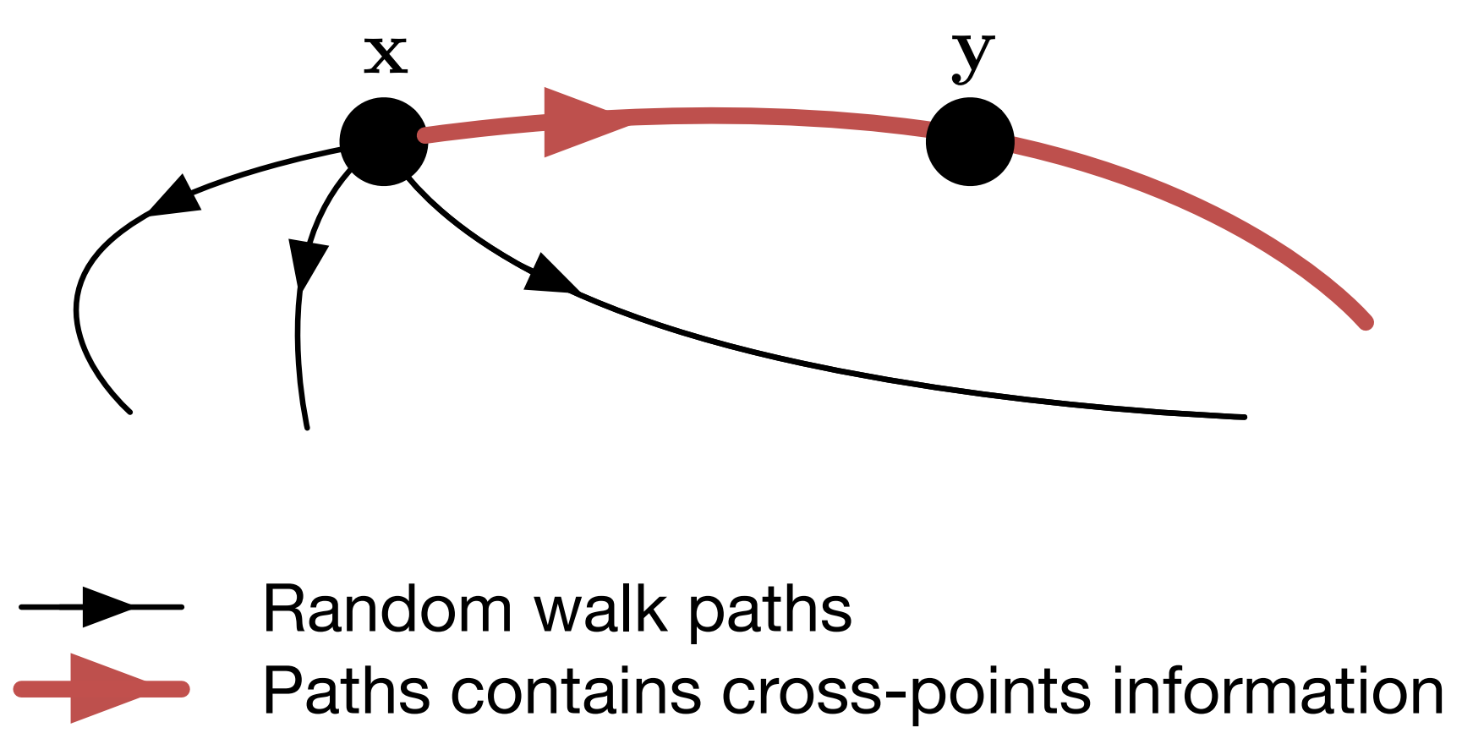

Figure 3. Paths may intersect other query points, providing cross-point linear observations for joint Bayesian refinement.

Experimental Setup

- **Test cases:** 3D chip model [5, 1] (3 layers: source-free / heat-source / source-free), 3 thickness configs, 3 power maps (`power6`, `4-core`, `16-core`).
- **Baseline:** PIRW [5], a Feynman–Kac method with standard truncation ($\Lambda = 0.0001$, following [3]).
- **Hardware:** 2× Intel Xeon Silver 4214 (24 cores), 128 GB RAM.

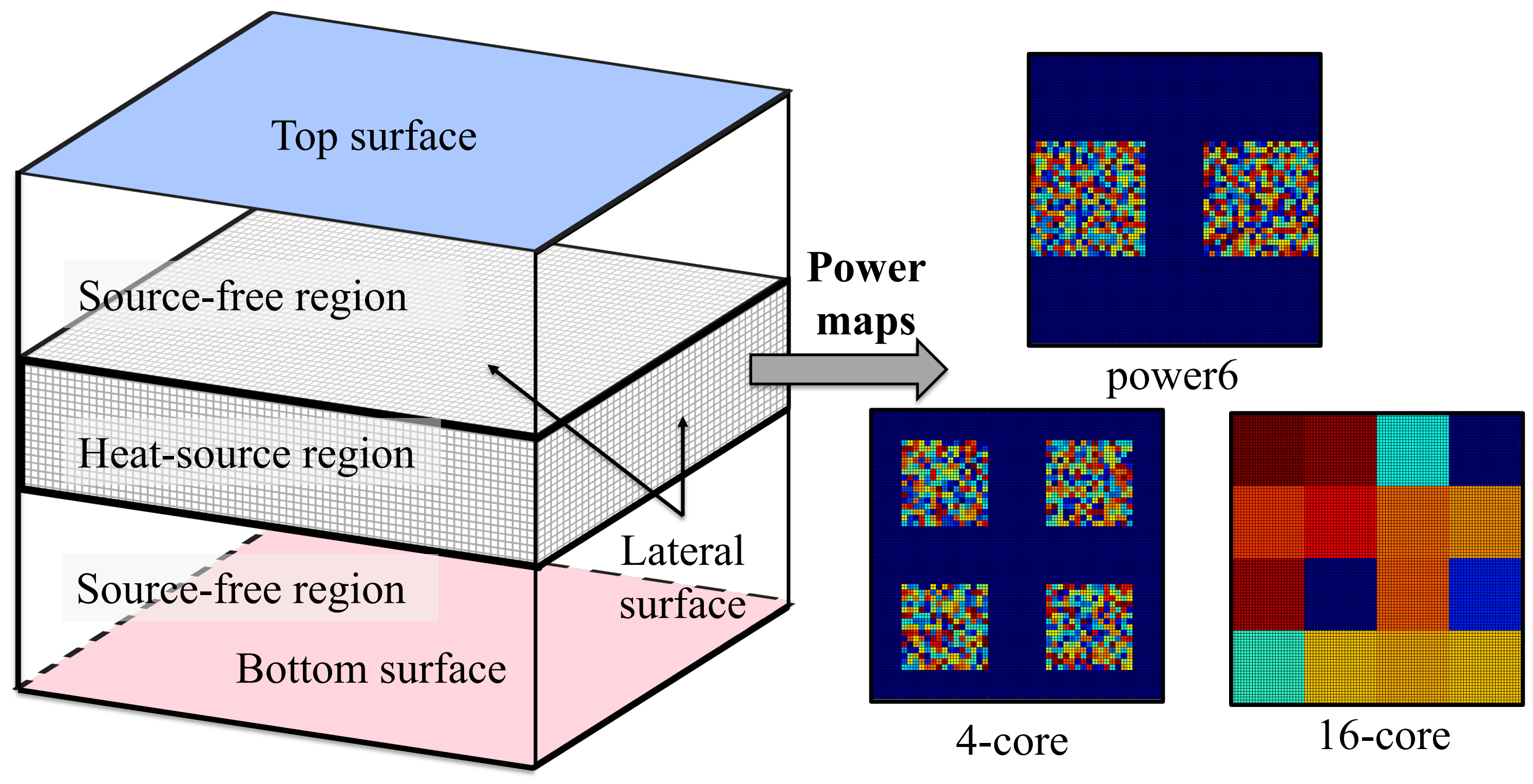

Figure 4. Test case geometry and power map configurations.

Single-Point Results

Table 1. Temperature estimation comparison. FastRW uses $\Lambda = 0.03$ with coarse FEM prior.

Case	Power Map	PIRW [5] Error/K	StdDev/K	Steps	Paths	FastRW Error/K	StdDev/K	Steps	Paths	Speedup
1	`power6`	0.38	0.47	5.79e6	1000	**0.36**	**0.45**	2.20e6	400	**6.58×**
2	`4-core`	0.43	0.55	5.83e6	1000	**0.42**	**0.54**	2.22e6	400	**6.57×**
3	`16-core`	0.38	0.46	1.03e7	1000	**0.32**	**0.41**	3.93e6	400	**6.55×**

- **FastRW** achieves **6.5×+ speedup** with **improved accuracy** across all cases.
- Path length reduced by >50%; number of required paths reduced from 1000 to 400.

Multi-Point Acceleration

Table 2. Multi-point estimation (Case 1, `power6`).

Group Size	Error(K)	StdDev(K)	$\hat{N}$	Speedup
1	0.356	0.446	1000	1.00×
4	0.292	0.379	1385	1.39×
16	0.221	0.281	2519	**2.52×**

Table 3. Efficiency analysis: time per point (Case 1).

Group Size	Prior(s)	RW(s)	Posterior(s)	Extra Cost(%)
1	0.28	15.41	0.00	1.82
4	0.28	15.41	0.02	1.95
16	0.28	15.41	0.61	5.78

- Larger group sizes yield more cross-point observations ⇒ additional **2.52×** acceleration.
- Prior + posterior overhead is only **5.78%** even at group size 16.

Effect of Prior Accuracy

Table 4. Better FEM prior enables larger Λ and shorter paths.

Prior DoF	Max-Error (K)	Random Walk Threshold Λ	Steps	Time (s)
1875	1.93	0.01	2.89e6	20.2
4800	0.88	0.02	2.45e6	17.1
12500	0.59	0.03	2.20e6	15.4

More accurate priors (finer FEM) reduce truncation error, allowing larger Λ and further runtime savings.

Conclusion

FastRW accelerates thermal analysis by:

1. **Exact truncation error analysis** — the expected error equals the weighted temperature at the truncation point.
2. **Prior estimation** — coarse FEM priors shorten paths by >50% and reduce variance, yielding **6.5×** overall single-point speedup.
3. **Posterior correction** — cross-point Bayesian inference provides up to **2.5×** additional acceleration with <6% overhead.

Future extensions: more accurate priors (learning-based), transient conduction, and GPU acceleration.

References

[1] Zhixuan Dong, Longlong Yang, Cuiyang Ding, Changhao Yan, Zhaori Bi, Sheng-Guo Wang, Dian Zhou, and Xuan Zeng. ppirw: An efficient and accurate precalculation path integral random walk solver for steady-state thermal simulation with robin boundary conditions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 44(4):1489–1502, 2025.

[2] Yuan Liang, Wenjian Yu, and Haifeng Qian. A hybrid random walk algorithm for 3-d thermal analysis of integrated circuits. In *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 849–854, 2014.

[3] Yonghan Luo, Yuyan Wang, Zhixuan Dong, Changhao Yan, Zhaori Bi, Keren Zhu, Sheng-Guo Wang, and Xuan Zeng. Gpu acceleration of a high-precision stochastic solver for steady-state thermal analysis with robin boundary conditions. In *2025 International Symposium of Electronics Design Automation (ISEDA)*, pages 287–292. IEEE, 2025.

[4] Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer Science & Business Media, 2013.

[5] Longlong Yang, Cuiyang Ding, Changhao Yan, Dian Zhou, and Xuan Zeng. A high-precision stochastic solver for steady-state thermal analysis with fourier heat transfer robin boundary conditions. In *2022 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9, 2022.

Design, Automation and Test in Europe Conference (DATE) 2026, Verona, Italy