Pushing a Single GPU to Its Limits and Scaling to Tens of Thousands: RL-Guided, Physically Consistent KMC for Nuclear Materials Simulation

Haozhi Han[1‡*] , Qi Li[2‡], Ruge Zhang[3,4‡*], Haipeng Jia[3,4], Yunquan Zhang[3,4], Yifeng Chen[1], Ting Cao[5], Yunxin Liu[5], and Kun Li[5†]

[1]School of Computer Science, Peking University, Beijing, China
[2]University of Science and Technology of China, Hefei, China
[3]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[4]University of Chinese Academy of Sciences, Beijing, China
[5]Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China

***Abstract*—Kinetic Monte Carlo is a cornerstone for rare-event dynamics in materials, but its strict sequentiality and super-basin trapping create a *scalability deadlock*: massive computation advances physical time only marginally, throttling both spatial and temporal scalability.**

We present EscapeKMC, the first reinforcement learning–guided KMC framework that resolves the scalability deadlock under strict physical alignment. EscapeKMC introduces three techniques: (1) *Poisson Clock Alignment* anchors event timing to the Poisson law and enables $O(1)$ escape-time estimation; (2) *Adaptive Swarm Reasoning* distributes decision-making across lightweight atomic agents coordinated by a centralized critic, yielding system-size-invariant policies transferable across scales; and (3) *Sparse Dynamics Regularization* restructures irregular dynamics into batched, accelerator-friendly execution efficiently.

On reactor-pressure-vessel benchmarks spanning concentrations from 10^{-5} to 10^{-1} and temperatures from 663–773 K, EscapeKMC achieves $17.5\times$ over OpenKMC while maintaining physical fidelity. At extreme scale, it simulates 5.8×10^{10} atoms on a single NVIDIA A100 GPU with a $500\times$ improvement in memory efficiency. Beyond single-GPU execution, EscapeKMC scales efficiently to large GPU clusters. It sustains 91% parallel efficiency on up to 16,384 GPUs and enables KMC simulations of up to 1.7×10^{14} atoms, pushing atomistic irradiation modeling to previously unattainable scales.

I. Introduction

Monte Carlo (MC) is one of the canonical algorithmic "dwarfs" in high-performance computing [1], with broad applications spanning statistical physics [2]–[4], uncertainty quantification [5]–[7], and computational finance [8], [9]. By generating statistically independent random samples, MC exhibits negligible data dependencies, making it embarrassingly parallel and ideally suited to modern GPU-accelerated architectures.

However, not all Monte Carlo methods share this property. Kinetic Monte Carlo, a widely used variant for rare-event dynamics such as atomic diffusion [10], [11], precipitation [12]–[14], and defect evolution [15], [16], stands in sharp contrast. As a continuous-time Markov chain, KMC executes one event at a time under a state-dependent dwell. This strict sequentiality fundamentally cripples its *spatial scalability*—precisely where rare-event simulations demand extreme scale.

Attempts at parallelization expose an additional barrier: intrinsic load imbalance. Local event dependencies induce ghost-region conflicts across spatial partitions, forcing frequent global synchronizations that leave large portions of hardware idle. As shown in Fig. 1, these locality-driven conflicts translate directly into barrier idle time, further undermining spatial scalability.

Beyond these spatial limits, KMC also collapses in *temporal scalability*. Systems often cycle endlessly among shallow barriers, while the rare high-barrier escapes that drive substantive evolution occur only after long dwell times. As a result, event streams are dominated by trivial transitions that consume vast computation yet advance physical time only marginally, stalling long-horizon evolution.

Fig. 2 quantifies the imbalance. At 663 K, 87.5% of all KMC steps occur inside the super-basin yet account for only 6.1% of the total simulated time; at 773 K, 84.3% of steps contribute just 25.3%. Coupled with KMC's sequentiality, these spatial and temporal bottlenecks together create a **scalability deadlock**: parallel resources remain idle, time advances slowly, and scalability collapses despite massive step counts.

Prior work tackles only one side of this deadlock. Learning-based surrogates [17], [18] claim both spatial and temporal scalability but function as black boxes detached from the Poisson process that grounds KMC, incurring severe distortion over long horizons. Physics-based accelerations, such as bias potentials [19]–[21] and basin compression [22], [23], suppress low-barrier flickers to advance temporal evolution,

‡Equal contribution.
*Work done during an internship at Institute for AI Industry Research (AIR), Tsinghua University.
†Corresponding author (likungw@gmail.com)

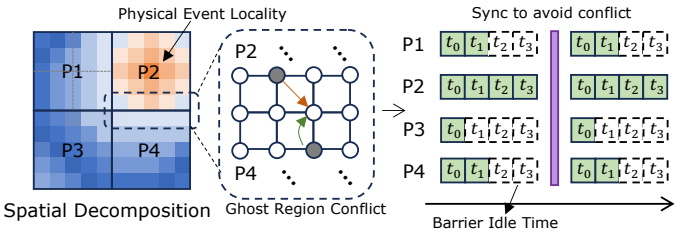

Fig. 1. Load imbalance in spatial scaling of KMC.

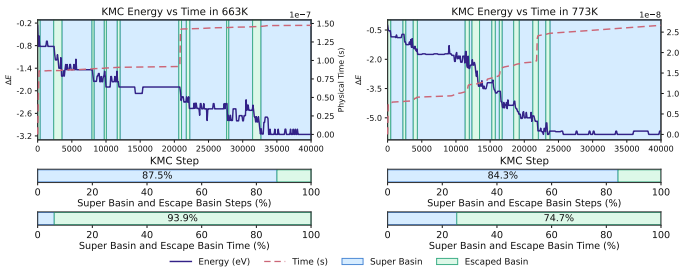

Fig. 2. The super-basin traps in temporal scaling of KMC.

but require explicit basin identification and costly recalibration, confining them to toy-scale demonstrations. HPC parallelization systems, exemplified by OpenKMC [24] and TensorKMC [25], scale via spatial decomposition and hardware-tuned kernels, but depend on massive over-provisioning and remain throttled by the super-basin trap, leaving temporal scalability unresolved. To date, no method unifies spatial and temporal scalability under strict physical alignment—the scalability deadlock persists.

In this paper, we present EscapeKMC, the first reinforcement learning–guided KMC system that breaks this scalability deadlock under strict physical alignment, delivering fidelity-preserving simulations at extreme scale on a single GPU and achieving near-linear strong scaling at ultra-large GPU scale.

The name EscapeKMC is inspired by an escape-room analogy: atoms trapped in low-barrier basins resemble players stuck in puzzles, circling endlessly without making meaningful progress in time. This reframes super-basin trapping as a decision problem rather than a sampling task, motivating reinforcement learning (RL) to discover physically valid escape strategies. Building on this reformulation, EscapeKMC points toward an emerging paradigm that we refer to as **embodied science**, in which microscopic agents interact with physical laws as an environment, and macroscopic scientific time and long-horizon evolution emerge from these embodied interactions rather than from passive sampling alone.

EscapeKMC system introduces three technical components:

First, **Poisson Clock Alignment (PCA)** aligns event timing strictly with physical laws, reconciling fidelity with temporal scalability. PCA defines event time as the drop of a *time potential* anchored to KMC's Poisson law, enforcing physical fidelity while enabling $O(1)$ escape-time estimation from fixed-size patches. Instantiated as the *Poisson Potential Network (PPN)*, PCA eliminates costly flicker cycles and restores temporal scalability across million-atom systems.

Second, **Adaptive Swarm Reasoning (ASR)** reconciles fidelity with spatial scalability by framing atomic evolution as a distributed reasoning process. Each atom acts as a lightweight *local agent* observing a finite neighborhood and making decisions with an $O(1)$ policy network—sustaining massive parallelism. A *centralized critic*, trained with Poisson-aligned rewards, enforces physical fidelity, while a *global arbitration* aggregates local proposals into consistent escape paths. ASR also supports *transferable training*, enabling models trained on small systems to scale to larger simulations without sacrificing accuracy, thereby maintaining swarm-level spatial scalability across systems of varying sizes.

Third, **Sparse Dynamics Regularization (SDR)** unlocks system-level scalability by regularizing sparse, irregular dynamics into accelerator-friendly structured execution. SDR integrates three mechanisms: an *event-driven cache* that updates only perturbed neighborhoods, *batch compaction* that converts patches of scattered active atoms into dense tensors for high-throughput inference, and *localized normalization* that restricts computation from all atoms to only the active set. Together, these techniques regularize memory access, saturate accelerator throughput, and preserve exact KMC semantics, enabling fidelity-preserving simulations on accelerator architectures.

We evaluate EscapeKMC on nuclear irradiation benchmarks that model solute cluster formation in complex alloys, a key mechanism underlying the embrittlement of reactor pressure vessel (RPV) steels. To evaluate **correctness**, EscapeKMC faithfully replicates dynamics validated by experimental data [26], [27], ensuring high fidelity across all test cases. In terms of **efficiency**, EscapeKMC, running on a single GPU, achieves a $17.5\times$ speedup over OpenKMC [24]. Beyond raw speedups, EscapeKMC simulates 5.8×10^{10} atoms on a single NVIDIA A100 GPU, achieving $500\times$ higher memory efficiency than OpenKMC. For **scalability**, EscapeKMC scales efficiently with 91% parallel efficiency on the ORISE supercomputer up to 16,384 GPUs and enabling simulations of up to 1.7×10^{14} atoms.

II. Background and Challenges

A. Kinetic Monte Carlo

Kinetic Monte Carlo is a *Continuous-Time Markov Chain* (CTMC) framework for simulating rare-event dynamics over long time scales. It proceeds as a sequence of KMC steps, each consisting of two strictly coupled operations: (1) **Event Selection**: selecting an event with probability proportional to its physical rate, and (2) **Clock Advance**: advancing the simulation time according to a Poisson process.

Event Selection determines *what happens next* in the KMC trajectory. In a given system state, the physical rate of each possible event is determined by the Arrhenius form:

$$\Gamma_X = \Gamma_0 \exp\left(-\frac{E_a^X}{kT}\right), \tag{1}$$

where Γ_0 is the attempt frequency, E_a^X is the activation energy barrier associated with event X, k is the Boltzmann constant, and T is the absolute temperature. The event is then chosen with probability

$$p_X = \frac{\Gamma_X}{\Gamma_{\text{tot}}}, \quad \Gamma_{\text{tot}} = \sum_{a \in \mathcal{A}(s)} \Gamma_a \tag{2}$$

where $\mathcal{A}(s)$ is the set of enabled events in state s.

Because the rate in Eq. (1) scales as $\exp(-E_a^X/kT)$, a seemingly minor reduction in activation energy can increase Γ_X by an order of magnitude, making low-barrier events overwhelmingly more likely to be chosen and trapping trajectories in cycles of rapid, low-impact transitions.

Clock Advance determines *how far the simulation clock moves* after the chosen event. In KMC, the probability of *leaving* s in a short interval Δt is approximately $\Gamma_{\text{tot}}(s)\,\Delta t$. In the continuous limit, the survival probability $P_{\text{surv}}(t)$, the probability that the system remains in s up to time t, satisfies:

$$P_{\text{surv}}(t) = \exp\big(-\Gamma_{\text{tot}}(s)\,t\big), \tag{3}$$

which is exactly the survival function of an *exponential distribution* with rate parameter $\Gamma_{\text{tot}}(s)$. This distribution has mean waiting time $\mathbb{E}[\Delta t] = \frac{1}{\Gamma_{\text{tot}}(s)}$. Therefore, In practice, KMC samples Δt by inverse transform:

$$\Delta t \;=\; -\frac{\ln u}{\Gamma_{\text{tot}}(s)}, \qquad u \sim \mathcal{U}(0,1). \tag{4}$$

B. Reinforcement Learning

Reinforcement Learning models sequential decision-making problems as a *Markov Decision Process* (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $P(s' \mid s, a)$ the transition probability from s to s' after taking action a, $r(s,a)$ the immediate reward, and $\gamma \in [0,1)$ the discount factor controlling the weight of future rewards.

In practice,*Proximal Policy Optimization* (PPO) [28] is among the most widely adopted for its stability and empirical performance. PPO maintains an explicit, parameterized policy $\pi_\theta(a \mid s)$ and updates its parameters θ by maximizing the *clipped surrogate objective*:

$$L^{\text{CLIP}}(\theta) \;=\; \mathbb{E}_t\left[\min\Big(r_t(\theta)\,\hat{A}_t,\; \text{clip}\big(r_t(\theta), 1-\epsilon, 1+\epsilon\big)\,\hat{A}_t\Big)\right], \tag{5}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is an estimator of the advantage function, and ϵ is the clipping parameter.

C. Breaking the Scalability Deadlock: RL for Escaping the Super-Basin Trap

Reinforcement learning excels at goal-directed escape in complex environments, from solving escape rooms to navigating mazes, by actively searching for rare yet decisive outcomes.

Classical Kinetic Monte Carlo, in contrast, is locked in a **scalability deadlock**: temporally, its passive, rate-driven sampling overwhelmingly favors low-barrier transitions, trapping trajectories in local basins; spatially, its inherently sequential updates hinder scaling to large systems. Together, these factors prevent efficient exploration of critical high-barrier events and severely constrain scalability as system size increases.

By modeling each atom as an "escape-room agent" equipped with RL-driven intelligence, we can enable atoms to actively pursue strategies that transcend passive sampling. Moreover, RL's parallelizable inference offers a promising route to scale this approach to millions or even billions of atoms, providing an opportunity to break the scalability deadlock that has long limited KMC.

D. Challenges

However, translating this opportunity into reality involves overcoming significant challenges:

(1) *How can we inject strategy-driven bias into a system whose evolution is strictly determined by physical laws?* In KMC, each step selects an event with probability proportional to its physical rate and advances time according to a Poisson process. This rigid coupling between "what happens" and "when it happens" leaves no room for arbitrary intervention. In contrast, RL optimizes behavior by introducing purposeful bias to maximize returns, unconstrained by any physical measure, so that any deviation from the native probabilities risks breaking the physical fidelity of the simulation.

(2) *How can we design atom-agents that rely only on local information for massive parallelism, yet still identify escape routes from super-basins that typically require global strategic knowledge?* At first glance, treating moving atoms as independent agents seems natural. But in realistic KMC systems, their number can reach millions or even billions, making strictly local reasoning the only way to sustain massive parallelism. However, escaping a deep super-basin is a fundamentally global problem: the most promising routes emerge only from the collective evolution of the entire system, information that no single atom-agent can observe. Designing agents that operate in local informational yet behave as if they perceive the global landscape forces a direct collision between scalability and strategic awareness.

(3) *How can we turn KMC's sparse, locally coupled event updates into the batched computation patterns that modern accelerators excel at through RL integration?* Modern accelerators deliver peak efficiency on large, dense batches with regular memory access and minimal dependencies. However, KMC executes tiny, scattered updates confined to local neighborhoods, with each step's valid events determined by the previous one. This combination of fine-grained dependencies and dynamically changing active regions makes it difficult to form large, conflict-free batches without expensive reordering or synchronization.

III. EscapeKMC

In this section, we propose EscapeKMC, a full-stack rearchitecture of Kinetic Monte Carlo that resolves its long-standing **scalability deadlock** through algorithm–system co-design. Unlike prior approaches that address only isolated

symptoms of the deadlock, EscapeKMC leverages reinforcement learning to fundamentally restructure the evolution process and introduces distributed optimizations that enable scalable execution across heterogeneous nodes.

A. Physical Poisson Alignment

Physical Poisson Alignment provides the foundation for modeling temporal evolution in RL–based dynamics. A key distinction between RL-based and classical KMC dynamics lies in how actions are chosen: in KMC, both the event and the time it consumes are determined by physical rates; in RL, however, actions are selected actively by the policy, independent of those rates. This decoupling motivates the core problem of Physical Poisson Alignment: given an agent-selected action, what is the physically consistent amount of time that should be assigned to it? This is answered through *Local Poisson Law* and *Scalable Poisson Network*.

Local Poisson Approximation derives an event–time expression from the generator of KMC dynamics, establishing a physically grounded definition of time for any agent-selected action. Crucially, we show that this event-time function can be accurately approximated using local features that are independent of system size, with an exponentially decaying approximation error. We formalize the derivation as follows:

Consider the mean first-passage time (MFPT) $\tau(s)$ to an absorbing set A. By Dynkin's formula, τ satisfies the *Poisson equation*

$$\sum_{a\in\mathcal{A}(s)} \Gamma_a(s)\left[\tau(\Phi(s,a)) - \tau(s)\right] + 1 = 0, \tag{6}$$

which characterizes the expected physical time to absorption directly in terms of the generator. The induced event–time increment is

$$\delta\tau(s,a) := \tau(s) - \tau(\Phi(s,a)). \tag{7}$$

However, $\delta\tau(s,a)$ is still a global quantity: it depends on the full state s, whose representation grows explosively with system size, making direct computation intractable. This motivates the central problem of *Local Poisson Representation*:

a) Problem (Local Poisson Representation).: can $\delta\tau(s,a)$ be expressed by a function of only local information in a finite-radius patch around (s,a), independent of the overall system size?

To solve the problem, we first introduce some formal definitions.

Definition 1 (Dimensionless potential)**.** *For state s, Define dimensionless potential*

$$u(s) := \Gamma_{\mathrm{tot}}(s)\,\tau(s).$$

Definition 2 (Twisted kernel)**.** *Let $p_a(s) =$ be the transition probability of event a from state s to $s' = \Phi(s,a)$. The* twisted kernel *acts on a function $u : \mathcal{S} \to \mathbb{R}$ as*

$$(\tilde{P}u)(s) \;:=\; \sum_{a\in\mathcal{A}(s)} p_a(s)\,\frac{\Gamma_{\mathrm{tot}}(s)}{\Gamma_{\mathrm{tot}}(s')}\,u(s').$$

With these definitions, the KMC setting we consider naturally satisfies the following locality:

Assumption 1 (Finite-range update)**.** *For the radius-r neighborhood $B_r(s,a)$ around event (s,a), there exists $R_0 > 0$ such that each one-step update depends on and modifies only $B_{R_0}(s,a)$.*

The assumption underpins the subsequent lemmas:

Lemma 1 (Geometric drift)**.** *Let $V_\alpha(x) := e^{\alpha d(x)}$, then exist $\alpha > 0$ and $\rho \in (0,1)$ such that*

$$\sup_{x\notin A} \sum_{a\in\mathcal{A}(x)} p_a(x)\,\frac{V_\alpha(\Phi(x,a))}{V_\alpha(x)} \;\le\; \rho.$$

Proof. *Let $\Delta_a := d(\Phi(x,a)) - d(x)$. By locality, $|\Delta_a| \le \Delta_{\max}$; by inward bias outside a finite set C, $\sum_a p_a(x)\Delta_a \le -\kappa$ for some $\kappa > 0$. Using $e^t \le 1 + t + \frac{1}{2}t^2$,*

$$\sum_a p_a(x)\frac{V_\alpha(\Phi)}{V_\alpha(x)} = \sum_a p_a(x)e^{\alpha\Delta_a} \le 1 - \alpha\kappa + \tfrac{1}{2}\alpha^2\Delta_{\max}^2.$$

Choosing $0 < \alpha \le \kappa/\Delta_{\max}^2$ gives $\sum_a p_a(x)\frac{V_\alpha(\Phi)}{V_\alpha(x)} \le 1 - \frac{1}{2}\alpha\kappa < 1$ for all $x \notin A\cup C$. On the finite set C, continuity in α yields the same bound after shrinking α. Taking the supremum over $x \notin A$ completes the proof. □

Lemma 2 (Contractivity in the V_α–weighted norm)**.** *Assume Lemma 1 and a uniform one–step rate ratio bound: $\sup_{x\to x'} \frac{\Gamma_{\mathrm{tot}}(x)}{\Gamma_{\mathrm{tot}}(x')} \le C_\Gamma < \infty$. With $\|f\|_{\alpha,*} := \sup_{x\notin A} \frac{|f(x)|}{e^{\alpha d(x)}}$. , there exists $\rho \in (0,1)$ such that for all bounded f,*

$$\|\tilde{P}f\|_{\alpha,*} \le \rho\,\|f\|_{\alpha,*}.$$

Proof. *Let $V_\alpha = e^{\alpha d}$ and $x' = \Phi(x,a)$. By Lemma 1, for sufficiently small $\alpha > 0$,*

$$\rho_0(\alpha) := \sup_{x\notin A} \sum_a p_a(x)\,\frac{V_\alpha(x')}{V_\alpha(x)} < 1.$$

For any $x \notin A$,

$$\frac{|\tilde{P}f(x)|}{V_\alpha(x)} \le \sum_a p_a(x)\frac{\Gamma_{\mathrm{tot}}(x)}{\Gamma_{\mathrm{tot}}(x')}\frac{V_\alpha(x')}{V_\alpha(x)}\,\|f\|_{\alpha,*} \le C_\Gamma\,\rho_0(\alpha)\,\|f\|_{\alpha,*}.$$

Choose α such that $C_\Gamma\rho_0(\alpha) < 1$ and set $\rho := C_\Gamma\rho_0(\alpha) \in (0,1)$.

Lemma 3 (Boundary-layer residual)**.** *Let $u^{(R)}$ solve $u^{(R)} = \tilde{P}^{(R)}u^{(R)} + 1$, where $\tilde{P}^{(R)} = \tilde{P}$ on $B_R(s_0)$. Define $r := (\tilde{P} - \tilde{P}^{(R)})u^{(R)}$. Then*

$$\operatorname{supp} r \;\subset\; \{x :\, d(x,s_0) \ge R - R_0\}, \qquad \|r\|_{\alpha,*} \;\le\; C_r\,e^{-\alpha(R-R_0)},$$

for a constant C_r independent of R.

Proof. *Finite-range updates give the support statement. Moreover, $u^{(R)} = \sum_{k\ge0}(\tilde{P}^{(R)})^k 1$ and Lemma 2 yield $\|u^{(R)}\|_{\alpha,*} \le (1-\rho)^{-1}$; on the boundary layer $V_\alpha(x) \ge e^{\alpha(R-R_0)}$, hence dividing by V_α gives the stated decay (absorbing local rate factors into C_r).* □

Theorem 4 (Exponential locality of u at s_0)**.** *Let* $\Delta := u - u^{(R)}$. *Then*

$$|u(s_0) - u^{(R)}(s_0)| \;\le\; C\, e^{-\alpha R}, \qquad C = \frac{C_r\, e^{\alpha R_0}}{1-\rho}.$$

Proof. *From* $u = \tilde{P}u + 1$ *and* $u^{(R)} = \tilde{P}^{(R)}u^{(R)} + 1$ *we get* $\Delta = \tilde{P}\Delta + r$. *Lemma 2 gives* $\|\Delta\|_{\alpha,*} \le \rho\|\Delta\|_{\alpha,*} + \|r\|_{\alpha,*}$, *hence* $\|\Delta\|_{\alpha,*} \le (1-\rho)^{-1}\|r\|_{\alpha,*}$. *Apply Lemma 3 and note* $V_\alpha(s_0) = 1$. □

According to Theorem 4, the potential $u(s)$ admits a patch-based approximation with exponentially decaying error, which provides the theoretical foundation for our *Scalable Poisson Network*.

Scalable Poisson Network (SPN) is a local neural operator designed to approximate the potential $u(s)$ from bounded patches. SPN is motivated by the exponential locality of $u(s)$ and is tailored for scalable, physically consistent time modeling. Unlike direct regression of event-time increments, which risks inconsistency across trajectories, SPN predicts potentials and derives time increments as potential differences. This ensures global path consistency while retaining strictly local inputs. Furthermore, by restricting inputs to bounded patches, SPN achieves an inference cost that is independent of system size, requiring only $O(1)$ forward evaluations per event. We next describe the architecture and training objective of SPN.

Architecture. As shown in Fig. 3, SPN consists of an Encoder, an Aggregator, and a Head. The Encoder maps bounded patches to node and edge features, capturing local interactions. The Aggregator condenses these into a single patch-level representation h_0 via attention, ensuring scalability and permutation invariance. The Head then maps h_0 to a scalar time potential μ, and the potential μ and μ' define the physical time increment.

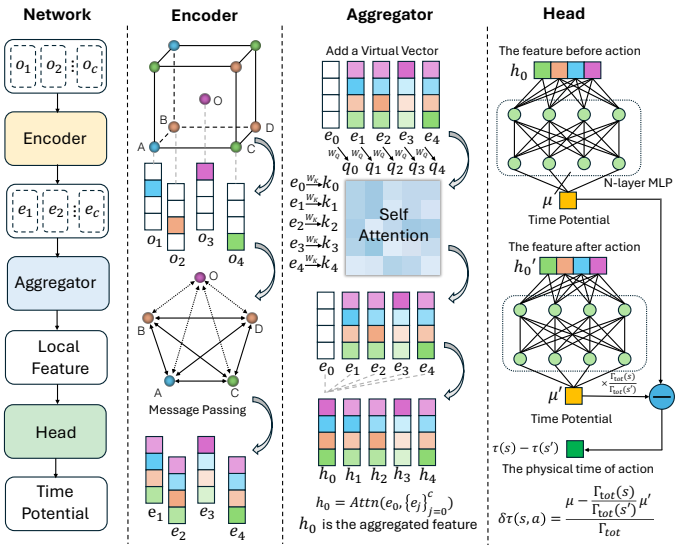

Fig. 3. Overall architecture of Scalable Poisson Network.

Training Objective. Our goal is to learn the dimensionless potential $u(s) := \Gamma_{\mathrm{tot}}(s)\,\tau(s)$, which satisfies the twisted Bellman fixed point

$$u(s) \;=\; \sum_a p_a(s)\,\frac{\Gamma_{\mathrm{tot}}(s)}{\Gamma_{\mathrm{tot}}(s')}\,u(s') + 1, \quad s' = \Phi(s,a),\;\; u|_A = 0, \tag{8}$$

We therefore train f_θ purely by minimizing the Bellman residual. Let $\mu = f_\theta(\mathcal{P}_R(s))$ and $\mu'_a = f_\theta(\mathcal{P}_R(s'))$, define the residual

$$\mathcal{R}(s;\theta) \;:=\; \mu - \sum_a p_a(s)\,\frac{\Gamma_{\mathrm{tot}}(s)}{\Gamma_{\mathrm{tot}}(s')}\,\mu'_a \;-\; 1, \tag{9}$$

and minimize

$$\mathcal{L}(\theta) \;=\; \mathbb{E}_s\big[\mathcal{R}(s;\theta)^2\big]. \tag{10}$$

At optimality ($\mathcal{R} \equiv 0$), f_θ realizes u and the event-time increment is recovered locally via

$$\widehat{\delta\tau}(s,a) = \frac{\mu - \frac{\Gamma_{\mathrm{tot}}(s)}{\Gamma_{\mathrm{tot}}(s')}\mu'_a}{\Gamma_{\mathrm{tot}}(s)} \;=\; \tau(s) - \tau(s'). \tag{11}$$

B. Adaptive Swarm Reasoning

Adaptive Swarm Reasoning is designed with two fundamental objectives: (1) to enable large-scale scalability and efficient parallelism by grounding each atom's decision-making on purely local information. (2) To ensure that, despite local autonomy, the collective dynamics converge toward globally optimal evolution paths through coordinated training. As illustrated in Fig. 4, these objectives are achieved through **Local Agent** and **Centralized Critic**.

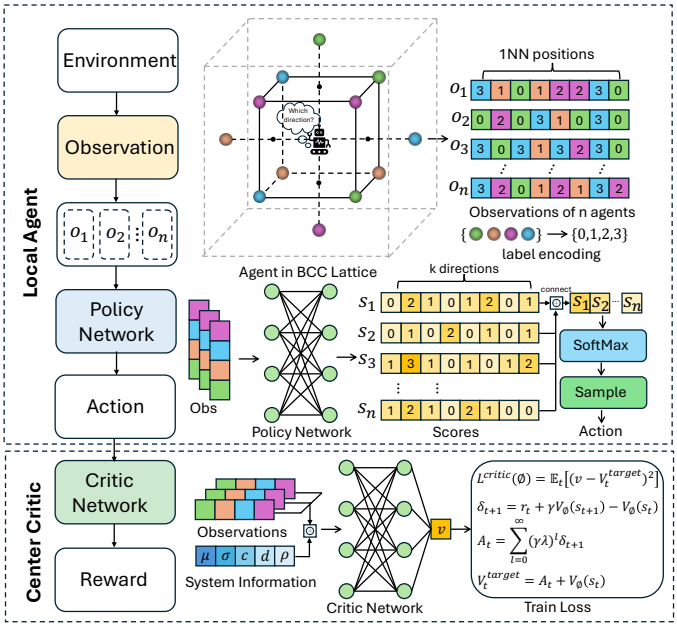

Fig. 4. Local Agent and Centralized Critic of the Adaptive Swarm Reasoning.

Local Agent consists of three components: (1) a local observation, (2) a lightweight policy network, and (3) a global arbitration mechanism. Together they enable scalable, swarm-level coordination while preserving physical fidelity.

(1) Local Observation. As illustrated in the upper panel of Fig. 4, each atom i acts as an agent that observes only a finite neighborhood in the lattice. Formally, we define

$$\mathcal{N}_i = \{\, j : \|\mathbf{r}_j - \mathbf{r}_i\| \le R_c,\ j \neq i \,\}, \tag{12}$$

where R_c is the cutoff radius. The local configuration is then encoded as a discrete vector

$$o_i = [\,\sigma_{ij}\,]_{j\in\mathcal{N}_i}, \quad o_i \in \mathbb{Z}^n, \tag{13}$$

where σ_{ij} denotes the species label of neighbor j. In practice, labels (colors in Fig. 4) are mapped to integers for input into the network. This discrete, topology-preserving encoding avoids the noise of continuous coordinates, respects lattice symmetries, and, most importantly, is invariant to system size and defect density, ensuring scalability.

(2) Policy Network. Given the local observation o_i in (13), each agent applies a shared MLP to produce directional logits. Each entry of $o_i \in \mathbb{Z}^n$ is a discrete species label σ_{ij} for neighbor $j \in \mathcal{N}_i$, where σ_{ij} is an integer index into the species alphabet Σ. Since integer indices have no intrinsic metric meaning, they are first converted into categorical vectors. Specifically, σ_{ij} is represented as a one-hot vector $u_{ij} \in 0, 1^{|\Sigma|}$, which is then mapped through a trainable embedding matrix $E \in \mathbb{R}^{|\Sigma|\times d}$ to obtain a dense representation:

$$e_{ij} = E^\top u_{ij} \in \mathbb{R}^d, \qquad j \in \mathcal{N}_i, \tag{14}$$

where d is the embedding dimension. The local observation is then formed by concatenating all neighbor embeddings,

$$\tilde{o}_i = \mathrm{concat}(e_{ij} : j \in \mathcal{N}_i) \in \mathbb{R}^{nd}. \tag{15}$$

This embedding-based representation removes spurious numerical relationships between integer labels, while allowing the network to learn latent similarities between species types.

The hidden features are computed through L nonlinear layers:

$$h_i^{(\ell)}; =; \sigma\big(W^{(\ell)} h_i^{(\ell-1)} + b^{(\ell)}\big), \qquad \ell = 1, \dots, L, \tag{16}$$

with $\sigma(\cdot)$ a pointwise nonlinearity, chosen as the GELU [29], which provides a smooth activation widely used in modern neural architectures. The parameter set

$$\theta = \{\, E,\, W^{(1:L)},\, b^{(1:L)},\, W^{\mathrm{out}},\, b^{\mathrm{out}} \,\} \tag{17}$$

is shared across all agents, ensuring the policy scales independently of system size.

A linear head then produces logits for K candidate hop directions:

$$z_i = W^{\mathrm{out}} h_i^{(L)} + b^{\mathrm{out}}, \qquad z_i \in \mathbb{R}^K. \tag{18}$$

To enforce physical feasibility, logits are masked and optionally scaled by a temperature $\tau > 0$:

$$\hat{z}_{i,k} = \begin{cases} z_{i,k}/\tau, & m_{i,k} = 1, \\ -\infty, & m_{i,k} = 0, \end{cases} \qquad k = 1, \dots, K, \tag{19}$$

where $m_{i,k} \in \{0, 1\}$ encodes whether direction k is physically admissible for atom i. Since both the neighborhood size in (12) and the action dimension K are constant, the per-agent forward cost remains $O(1)$, enabling batched inference for millions of agents in parallel and thereby underpinning the scalability of our framework.

TABLE I
MESOSCOPIC SYSTEM DESCRIPTORS DEFINED FOR EACH ATOMIC SPECIES $i = 1, \dots, M$.

Symbol	Dimension	Description
$\mu_t^{(i)}$	$\mathbb{R}^3$	Spatial mean of atomic positions
$\sigma_t^{(i)}$	$\mathbb{R}^3$	Spatial variance of atomic positions
$c_t^{(i)}$	$\mathbb{R}$	Variance of local atomic density
$d_t^{(i)}$	$\mathbb{R}$	Mean vacancy–atom distance
$\rho_t^{(i)}$	$\mathbb{R}$	Clustering degree around vacancies

(3) Global Arbitration. Although logits are generated locally by (18)–(19), event selection must reflect system-wide competition. All masked logits are concatenated,

$$Z = \mathrm{concat}\,(\hat{z}_1, \hat{z}_2, \dots, \hat{z}_N) \in \mathbb{R}^{NK}, \tag{20}$$

where $\hat{z}_i \in \mathbb{R}^K$ denotes the feasibility-masked logits of agent i. The system-wide action distribution is then obtained via a global softmax:

$$p_\theta(a \mid o_{1:N}) = \frac{\exp(Z_a)}{\sum_{a=1}^{NK} \exp(Z_a)}, \qquad a \in 1, \dots, NK. \tag{21}$$

This differentiable arbitration layer in (21) preserves the inherently global nature of KMC event competition while enabling end-to-end training. Importantly, it induces swarm intelligence: local proposals are forced to compete on equal footing at the system level, allowing kinetically significant escape events to dominate the collective dynamics.

Centralized Critic rests on three parts: (1) Multi-scale Input Representation, (2) Physics-aligned Reward, (3) Transferable Training. These ensure faithful and scalable coordination across the swarm.

(1) Multi-scale Input Representation. The critic operates on a two-level input representation. First, microscopic observations from all agents are concatenated to preserve local detail:

$$o_{1:N,t} = \big(o_{1,t}, o_{2,t}, \dots, o_{N,t}\big), \tag{22}$$

which encodes the discrete neighborhood states of each atom. Second, mesoscopic system descriptors are appended to capture the coarse-grained collective behavior of all atomic species. For a system with M species, we define

$$(\mu_t^{(i)}, \sigma_t^{(i)}, c_t^{(i)}, d_t^{(i)}, \rho_t^{(i)})_{i=1}^M, \tag{23}$$

where the definitions are summarized in Table I. The complete critic input is therefore

$$s_t = \mathrm{concat}\Big(o_{1:N,t},\ \{\,\mu_t^{(i)}, \sigma_t^{(i)}, c_t^{(i)}, d_t^{(i)}, \rho_t^{(i)}\,\}_{i=1}^M\Big). \tag{24}$$

(2) Physics-aligned Reward. To align learning with physical dynamics, we define rewards using SPN (in Sec. III-A), which

approximates the Poisson time potential locally. For each transition $(s, a \to s')$, the reward is

$$r_t = \frac{\hat{u}^{(R)}(s)}{\Gamma_{\text{tot}}(s)} - \frac{\hat{u}^{(R)}(s')}{\Gamma_{\text{tot}}(s')}, \tag{25}$$

providing a faithful measure of elapsed time. Here, SPN acts as a fixed reward model: its outputs supply GAE targets for critic training, while gradients are stopped to avoid feedback instability. This design yields two benefits: (i) Since SPN evaluates only fixed-radius patches, reward computation is $O(1)$ per step, ensuring scalability. (ii) Unlike heuristic shaping, this design does not bias toward greedy descent but supplies the correct temporal scale, enabling RL to discover globally shortest-time escape trajectories.

(3) Transferable Training. A distinctive feature of our framework is that a policy trained on small systems generalizes *unchanged* to arbitrarily large ones. This invariance arises from the factorized structure of the global selection probability.

Let $\mathcal{U}$ be the set of local contexts. With $n_u(s_t)$ the count of context u in state s_t and $\nu_{s_t}(u) := n_u(s_t)/N$ its frequency (N atoms total), summing the global softmax over identical contexts yields

$$\Pr_\theta(u, k \mid s_t) = \frac{\nu_{s_t}(u)\, \exp(z_\theta(u)_k)}{\sum_{v\in\mathcal{U}} \nu_{s_t}(v) \sum_\ell \exp(z_\theta(v)_\ell)}. \tag{26}$$

Dividing numerator and denominator by N shows explicit cancellation; the logits $z_\theta(u)_k$ depend only on the local context u and are thus independent of N. The expected one-step reward becomes

$$\mathbb{E}[r_t \mid s_t] = \sum_{u\in\mathcal{U}} \sum_k \Pr_\theta(u, k \mid s_t)\, \Delta R(u, k), \tag{27}$$

which depends only on the local mapping $\{z_\theta(u)_k\}$ and the context distribution ν_{s_t}, which are both size-invariant. Hence once the local ranking $u \mapsto z_\theta(u)_k$ is learned on small systems, the policy transfers directly to larger ones, yielding true zero-shot scalability.

C. Sparse Dynamics Regularization

KMC updates are inherently sparse and irregular, making them ill-suited to modern accelerators. As illustrated in Fig. 5, Sparse Dynamics Regularization resolves this mismatch through *Event-Driven Cache*, *Batch Compaction*, and *Localized Normalization*, which together convert irregular event dynamics into accelerator-friendly workloads while preserving exact physical consistency.

Event-Driven Cache. As illustrated in Fig. 5, KMC updates perturb only a vanishing fraction of atoms per step, resulting the overwhelming majority of neighborhoods remain unchanged and can be safely reused across steps. We exploit this asymmetry with an event-driven cache: *dirty* patches trigger recomputation, whereas all others reuse cached logits. This selective reuse removes redundant work, converts irregular sparsity into regular memory streams, and yields dense batched inference that both preserves physical fidelity and saturates accelerator throughput.

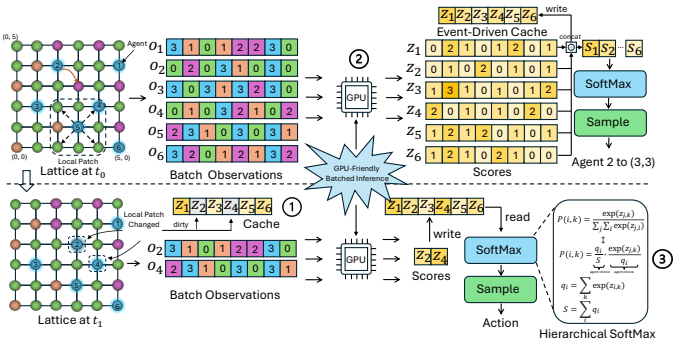

Fig. 5. Overview of Sparse Dynamics Regularization. ① *Event-Driven Cache*, reusing unchanged neighborhoods. ② *Batch Compaction*, packing scattered frontiers into dense batches. ③ *Localized Normalization*, hierarchical softmax with incremental updates.

Batch Compaction. In KMC steps, active atoms after each event are scattered irregularly across the lattice. Invoking the MLP per atom would fragment GPU kernels and yield poor occupancy. Instead, we compact these scattered frontiers into contiguous, fixed-size batches, converting a sparse worklist into a dense tensor. This yields fully vectorized inference, regularized memory streams, and accelerator-saturating throughput, while preserving exact KMC semantics.

Localized Normalization. A key bottleneck in KMC inference is the global softmax, which naïvely rescans all NK logits at every step. We eliminate this overhead with a two-level scheme: each atom maintains a local weight $q_i = \sum_k e^{z_{i,k}}$, while a global accumulator $S = \sum_i q_i$ tracks their total. Only frontier atoms update their q_i, with S corrected incrementally. Sampling then factorizes hierarchically: first selecting an atom with probability q_i/S, then an event within that atom with probability $e^{z_{i,k}}/q_i$. This semantics-preserving decomposition reduces complexity from $O(NK)$ global reductions to $O(|D_t|K)$ localized updates, removing the system-wide bottleneck.

IV. Evaluation

A. Setup

Hardware. (1) Single-node system contains an AMD EPYC 7543 CPU and an NVIDIA A100 80 GB. (2) ORISE is a large-scale heterogeneous supercomputer comprising over 4,100 compute nodes. Each node is equipped with one CPU and four GPGPU-like accelerators, interconnected via a high-speed network with a peak bandwidth of 25 GB/s. The CPU adopts a 4-way, 8-core x86 architecture. The GPGPU-like accelerators, also referred to as HIP-based GPUs, employ HIP as the backend programming model and deliver performance comparable to AMD MI60 GPUs.

State-of-the-arts. To evaluate EscapeKMC, we compare against both simulation baselines and experimental references that represent the state of the art in kinetic Monte Carlo studies of reactor-pressure-vessel (RPV) steels.

On the simulation side, we compare against OpenKMC [24], a state-of-the-art parallel KMC design that scales to hundred-billion-atom simulations on leadership-class supercomputers.

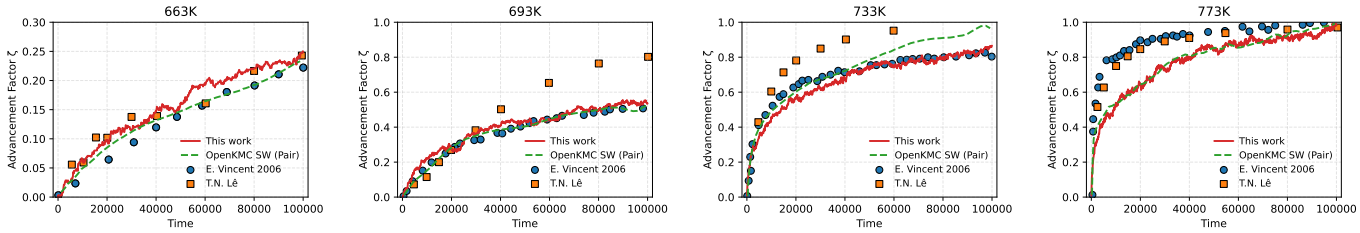

Fig. 6. Correctness Validation of EscapeKMC against OpenKMC and literature.

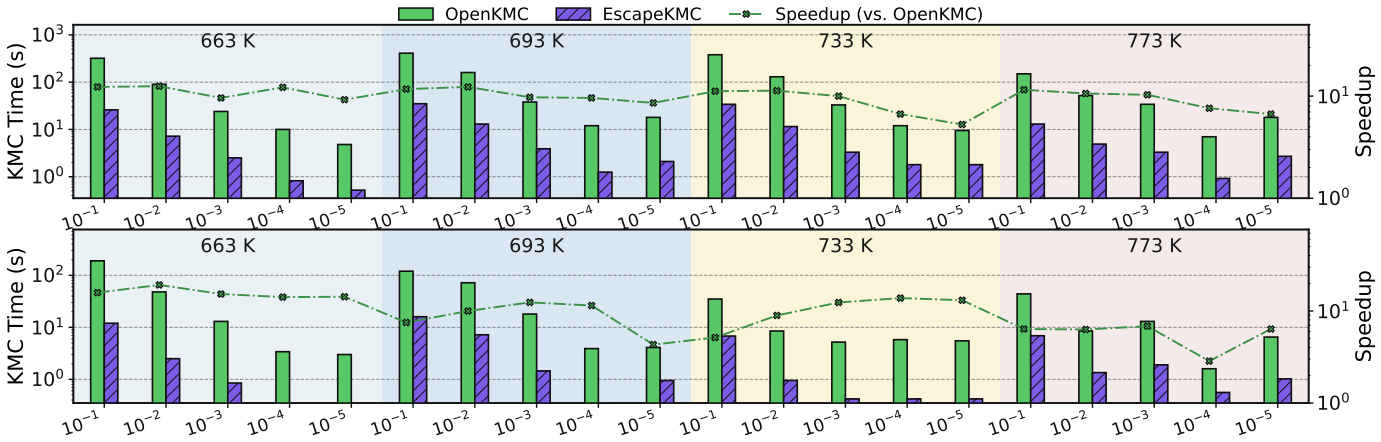

Fig. 7. State-of-the-art performance comparison of EscapeKMC (Top: PAIR potential; Bottom: EMA potential).

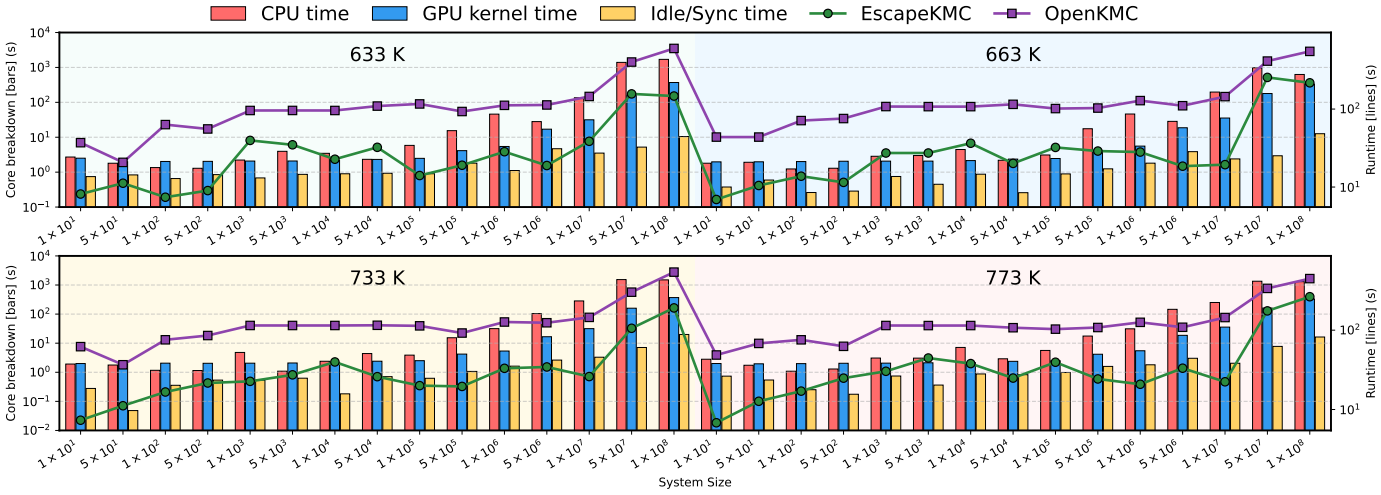

Fig. 8. Supercomputing-scale runtime breakdown and scalability analysis.

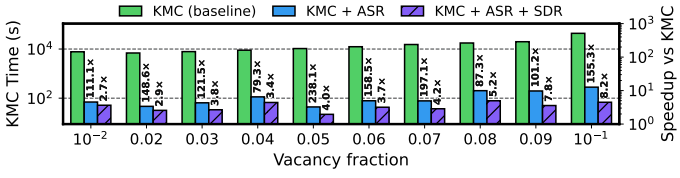

Fig. 9. Ablation study of EscapeKMC with ASR and SDR.

Experimentally, we validate against canonical benchmarks for irradiated Fe–Cu alloys, including the seminal studies of the datasets of Lê *et al.* [26] and Vincent *et al.* [27]

Benchmarks. We evaluate on a classical KMC application, Cu-rich precipitate formation in RPV steels under neutron irradiation, a key driver of material degradation. Our benchmarks start from the $40 \times 40 \times 40$ RPV cell. Furtherly, we explore a wide range of system sizes, from 10^1 atoms to 10^{10}, and vacancy concentrations spanning 10^{-5} to 10^{-1}.

Metrics. We introduce a time-to-target metric, which captures the wall-clock time required for the system energy to

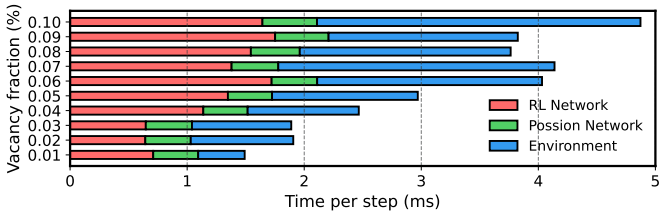

Fig. 10. Per-step runtime components.

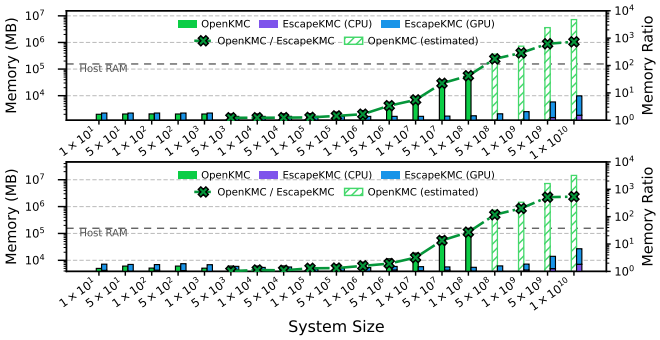

Fig. 11. Supercomputing-scale memory analysis (Top: 1.5×10^{-4} at.% of Cu; Bottom: 3.0×10^{-4} at.% of Cu).

relax to a given fraction of its initial residual:

$$t_\alpha \;=\; \inf\left\{\, t \;\middle|\; \frac{E(t) - E_\infty}{E(0) - E_\infty} \;\le\; \alpha \right\}, \tag{28}$$

where $E(t)$ denotes the system energy at time t, and E_∞ is the asymptotic equilibrium energy. In our evaluation, we set $\alpha = 0.05$ to balance accuracy with feasible testing time.

B. Correctness Verification

We verify correctness by examining the advancement factor $\zeta(t)$. As shown in Fig. 6, EscapeKMC closely matches OpenKMC and experimental data in $\zeta(t)$, capturing both the slow evolution at low temperatures and the rapid saturation at high temperatures. These results show that EscapeKMC preserves both configurational and energetic evolution, validating PCA as an accurate correction of the physical time scale.

C. State-of-the-art Comparison

Fig. 7 compares EscapeKMC against the baseline OpenKMC across vacancy fractions and temperatures. EscapeKMC achieves a $17.52\times$ speedup over OpenKMC, with consistently strong performance across all evaluated regimes. These results demonstrate the high computational efficiency and robustness of EscapeKMC.

D. Ablation Study

Fig. 9 quantifies the contribution of ASR and SDR. Introducing ASR yields an average $139.8\times$ speedup over baseline KMC across vacancy fractions 10^{-2}–10^{-1}, indicating that RL-driven restructuring resolves super-basin trapping efficiently. Adding SDR provides an additional $2.7\times$–$8.2\times$ gain over ASR, with a clear upward trend as the vacancy fraction

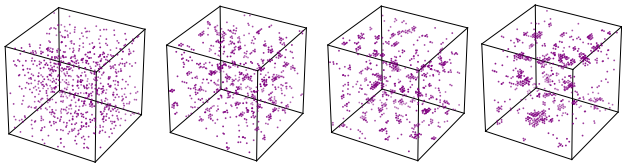

Fig. 12. Clustering dynamics in Fe–0.67 at.% Cu.

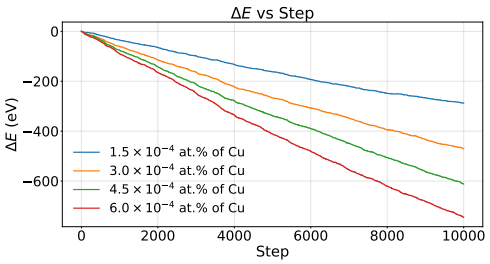

Fig. 13. Supercomputing-scale energy relaxation on a single GPU.

increases, indicating growing benefit of SDR under higher activity.

The per-step runtime analysis in Figure 10 shows that EscapeKMC eliminates the classical hotspot of event sampling. In traditional KMC, sampling dominates the cost, whereas in EscapeKMC, the inference stage (RL in red, PCA in green) remains small and stable—RL rises only slowly with vacancy fraction due to SDR batching and caching, while PCA stays nearly constant. Furthermore, the environment update (blue) is an algorithm-independent cost, stemming solely from applying events to the physical KMC lattice. Its dominance underscores that all algorithmic hotspots have been removed, leaving only the intrinsic cost of simulating the environment.

E. Supercomputing-Scale on a Single A100

We evaluate scalability and practicality on a single NVIDIA A100 GPU, focusing on compute performance, memory scalability, and a 5.8×10^{10}-atom simulation previously requiring leadership-class supercomputers.

Compute performance. Fig. 8 shows that across 60 benchmarks, EscapeKMC outperforms OpenKMC with a $4.04\times$ geometric-mean speedup, sustaining $4.1\times$ even at 10^8 atoms. Runtime increases slowly with system size and remains consistently below OpenKMC. The runtime breakdown indicates that computation dominates execution, while Idle/Sync overhead is only 7.6% on average, indicating high hardware utilization.

Memory efficiency. As shown in Fig. 11, EscapeKMC scales to 10^{10} atoms within a single A100 GPU, whereas OpenKMC exceeds host memory beyond 10^8 atoms. EscapeKMC uses on average $5\times$ less memory than OpenKMC, with reductions reaching $27\times$, $200\times$, and $500\times$ at 10^8, 10^9, and 10^{10} atoms, respectively. This efficiency is insensitive to Cu concentration and remains GPU-dominated.

Supercomputing-scale energy relaxation on a single GPU. Fig. 13 presents the energy relaxation involving 5.8×10^{10} atoms performed on a single 80 GB A100 GPU. Across four Cu concentrations, the trajectories exhibit smooth descent

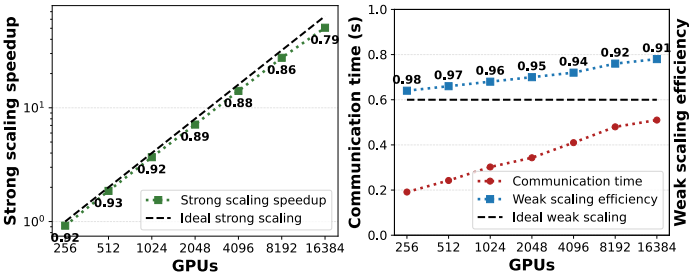

Fig. 14. Strong and weak scaling behavior of large-scale KMC simulations in Fe–0.67 at.% Cu with **140 trillion** atoms.

and closely match the trend of a 40^3 reference system, confirming physical validity and consistent cross-scale behavior. To the best of our knowledge, this constitutes the first demonstration of RPV precipitate dynamics at supercomputing scale achieved on a single GPU.

F. Scalability

Fig. 14 demonstrates that the proposed method achieves structural scalability with consistent performance gains. In strong scaling, speedup closely follows the ideal trend from 256 to 16,384 GPUs, maintaining parallel efficiency above 92% up to 1,024 GPUs and above 88% at 4,096 GPUs, despite the highly irregular and event-driven nature of KMC execution.

Under weak scaling, efficiency degrades only mildly as the system scales, decreasing from 98% at 256 GPUs to 91% at 16,384 GPUs, while communication time increases smoothly from 0.19 s to 0.51 s. This monotonic growth in communication overhead does not induce efficiency collapse, as synchronization is restricted to localized, neighbor-only interactions without horizon-wide barriers, enabling stable weak scaling at trillion-atom scales.

G. Physical Visualization.

Fig. 12 shows long-term evolution of Cu in Fe–0.67 at.% Cu, from dispersed solutes to nucleation, growth, and coarsening of precipitates, which are consistent with experimental observations in RPV steels. This demonstrates that EscapeKMC faithfully reproduces the underlying physical processes.

V. Related Work

Kinetic Monte Carlo is a key tool for modeling thermally activated processes, by sampling transitions from physically derived rates [30]–[37]. To access longer timescales, acceleration methods like hyperdynamics [20], [38], [39] and parallel-replica dynamics [40]–[42] boost effective event rates, while adaptive and on-the-fly KMC reduce transition search via automated catalogs and NEB exploration [43]–[45]. At the system scale, frameworks such as OpenKMC [24], SPPARKS [46], and TensorKMC [25] exploit parallelism to reach billions of atoms. More recently, machine learning has entered KMC: surrogate models predict rates [47]–[50], graph networks encode structures [51]–[53], and reinforcement learning guides event selection [17], [18]. These approaches show promise, but applications remain mostly limited to small or reduced models, and scaling them to large, physically faithful systems is still an open challenge.

VI. Conclusion

This paper introduces EscapeKMC, the first RL-guided KMC that resolves the scalability deadlock of rare-event simulations while preserving physical fidelity. By introducing Scalable Swarm Reasoning, Sparse Dynamics Regularization, and Decentralized Domain Scheduling, It demonstrates order-of-magnitude efficiency gains and near-linear scalability at extreme GPU scale, unlocking new opportunities for large-scale science.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful and constructive feedback. This work was supported by the Carbon Neutrality and Energy System Transformation (CNEST) Program and XiongAn AI Institute. Part of the numerical calculations in this work was carried out on the ORISE Supercomputer and computing resources provided by Tecorigin, and the authors gratefully acknowledge their support.

REFERENCES

[1] K. Asanovic, R. Bodik, B. Catanzaro, J. Gebis, P. Husbands, K. Keutzer, D. Patterson, W. Plishker, J. Shalf, and S. W. Williams, "The landscape of parallel computing research: A view from berkeley," 2006.

[2] J. A. Anderson, E. Jankowski, T. L. Grubb, M. Engel, and S. C. Glotzer, "Massively parallel monte carlo for many-particle simulations on gpus," *Journal of Computational Physics*, vol. 254, pp. 27–38, 2013.

[3] M. Weigel and T. Yavorskii, "Gpu accelerated monte carlo simulations of lattice spin models," *Physics Procedia*, vol. 15, pp. 92–96, 2011.

[4] M. Weigel, "Monte carlo methods for massively parallel computers," *Order, Disorder and Criticality*, vol. 5, pp. 271–340, 2018.

[5] J. Zhang, "Modern monte carlo methods for efficient uncertainty quantification and propagation: A survey," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 13, no. 5, p. e1539, 2021.

[6] T. J. Dodwell, C. Ketelsen, R. Scheichl, and A. L. Teckentrup, "A hierarchical multilevel markov chain monte carlo algorithm with applications to uncertainty quantification in subsurface flow," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 3, no. 1, pp. 1075–1108, 2015.

[7] Z. Yin, S. Strebelle, and J. Caers, "Automated monte carlo-based quantification and updating of geological uncertainty with borehole data (autobel v1. 0)," *Geoscientific Model Development*, vol. 13, no. 2, pp. 651–672, 2020.

[8] M. F. Dixon, T. Bradley, J. Chong, and K. Keutzer, "Monte carlo–based financial market value-at-risk estimation on gpus," in *GPU Computing Gems Jade Edition*. Elsevier, 2012, pp. 337–353.

[9] D. Liyanage, G. Fernando, D. Arachchi, R. Karunathilaka, and A. S. Perera, "Utilizing intel advanced vector extensions for monte carlo simulation based value at risk computation," *Procedia Computer Science*, vol. 108, pp. 626–634, 2017.

[10] M. Andersen, C. Panosetti, and K. Reuter, "A practical guide to surface kinetic monte carlo simulations," *Frontiers in chemistry*, vol. 7, p. 202, 2019.

[11] X. He, F. Cheng, and Z.-X. Chen, "The lattice kinetic monte carlo simulation of atomic diffusion and structural transition for gold," *Scientific Reports*, vol. 6, no. 1, p. 33128, 2016.

[12] K. Y. Khromov, V. Vaks, and I. Zhuravlev, "Studies of concentration and temperature dependences of precipitation kinetics in iron-copper alloys using kinetic monte carlo and stochastic statistical simulations," *Journal of Experimental and Theoretical Physics*, vol. 116, no. 2, pp. 236–251, 2013.

[13] E. Clouet, M. Nastar, A. Barbu, C. Sigli, and G. Martin, "Precipitation in al-zr-sc alloys: A comparison between kinetic monte carlo, cluster dynamics and classical nucleation theory," *arXiv preprint cond-mat/0507259*, 2005.

[14] L. Messina, N. Castin, C. Domain, and P. Olsson, "Introducing ab initio based neural networks for transition-rate prediction in kinetic monte carlo simulations," *Physical Review B*, vol. 95, no. 6, p. 064112, 2017.

[15] I. Martin-Bragado, A. Rivera, G. Valles, J. L. Gomez-Selles, and M. J. Caturla, "Mmonca: An object kinetic monte carlo simulator for damage irradiation evolution and defect diffusion," *Computer Physics Communications*, vol. 184, no. 12, pp. 2703–2710, 2013.

[16] H. Xu, Y. N. Osetsky, and R. E. Stoller, "Simulating complex atomistic processes: On-the-fly kinetic monte carlo scheme with selective active volumes," *Physical Review B—Condensed Matter and Materials Physics*, vol. 84, no. 13, p. 132103, 2011.

[17] H. Tang, B. Li, Y. Song, M. Liu, H. Xu, G. Wang, H. Chung, and J. Li, "Reinforcement learning-guided long-timescale simulation of hydrogen transport in metals," *Advanced Science*, vol. 11, no. 5, p. 2304122, 2024.

[18] T. A. Bojesen, "Policy-guided monte carlo: Reinforcement-learning markov chain dynamics," *Physical Review E*, vol. 98, no. 6, Dec. 2018. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.98.063303

[19] K. Ferasat, Y. N. Osetsky, A. V. Barashev, Y. Zhang, Z. Yao, and L. K. Béland, "Accelerated kinetic monte carlo: A case study; vacancy and dumbbell interstitial diffusion traps in concentrated solid solution alloys," *The Journal of Chemical Physics*, vol. 153, no. 7, 2020.

[20] A. Chatterjee and A. F. Voter, "Accurate acceleration of kinetic monte carlo simulations through the modification of rate constants," *The Journal of Chemical Physics*, vol. 132, no. 19, p. 194101, 2010.

[21] K. M. Bal and E. C. Neyts, "Merging metadynamics into hyperdynamics: accelerated molecular simulations reaching time scales from microseconds to seconds," *Journal of chemical theory and computation*, vol. 11, no. 10, pp. 4545–4554, 2015.

[22] K. A. Fichthorn and Y. Lin, "A local superbasin kinetic monte carlo method," *The Journal of chemical physics*, vol. 138, no. 16, 2013.

[23] M. Novotny, "Monte carlo algorithms with absorbing markov chains: Fast local algorithms for slow dynamics," *Physical review letters*, vol. 74, no. 1, p. 1, 1995.

[24] K. Li, H. Shang, Y. Zhang, S. Li, B. Wu, D. Wang, L. Zhang, F. Li, D. Chen, and Z. Wei, "Openkmc: a kmc design for hundred-billion-atom simulation using millions of cores on sunway taihulight," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3295500.3356165

[25] H. Shang *et al.*, "Tensorkmc: Kinetic monte carlo simulation of 50 trillion atoms with tensorflow," in *SC '21: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2021, pp. 1–14.

[26] T. N. Lê, A. Barbu, D. Liu, and F. Maury, "Precipitation kinetics of dilute fecu and fecumn alloys subjected to electron irradiation," *Scripta Metallurgica et Materialia*, vol. 26, no. 5, pp. 771–776, 1992.

[27] E. Vincent, C. S. Becquart, and C. Domain, "Solute interaction with point defects in α-fe during thermal ageing: A combined *ab initio* and atomic kinetic monte carlo approach," *Journal of Nuclear Materials*, vol. 351, no. 1–3, pp. 88–99, 2006.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[29] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[30] T. S. Rahman, C. Gosh, O. Trushin, A. Kara, and A. Karim, "Atomistic studies of thin film growth," in *Nanomodeling*, vol. 5509. SPIE, 2004, pp. 1–14.

[31] P. Kratzer, "Monte carlo and kinetic monte carlo methods," *arXiv preprint arXiv:0904.2556*, 2009.

[32] J. Yang, M. I. Monine, J. R. Faeder, and W. S. Hlavacek, "Kinetic monte carlo method for rule-based modeling of biochemical networks," *Physical Review E*, vol. 78, no. 3, p. 031910, 2008. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.78.031910

[33] D. T. Gillespie, "Stochastic simulation of chemical kinetics," *Annual Review of Physical Chemistry*, vol. 58, pp. 35–55, 2007.

[34] Z. Xi, L. Hector, A. Misra, and L. Qi, "Kinetic monte carlo simulations of solute clustering during quenching and aging of al-mg-zn alloys," *Acta Materialia*, vol. 269, p. 119795, 03 2024.

[35] B. D. Hehr, "Anaylsys of defect clustering in semiconductors using kinetic monte carlo methods." Sandia National Lab. (SNL-NM), Albuquerque, NM (United States), 07 2013. [Online]. Available: https://www.osti.gov/biblio/1666149

[36] C. S. Becquart and B. D. Wirth, "Kinetic monte carlo simulations of irradiation effects," in *Comprehensive Nuclear Materials*. Elsevier, 2012, pp. 393–410.

[37] C. S. Becquart and F. Soisson, "Monte carlo simulations of precipitation under irradiation," in *Handbook of Mechanics of Materials*. Springer, 2019, pp. 703–731. [Online]. Available: https://link.springer.com/10.1007/978-981-10-6884-3_24

[38] A. F. Voter, "Hyperdynamics: Accelerated molecular dynamics of infrequent events," *Physical Review Letters*, vol. 78, no. 20, p. 3908, 1997.

[39] R. A. Miron and K. A. Fichthorn, "Accelerated molecular dynamics with the parallel replica method," *The Journal of Chemical Physics*, vol. 119, no. 12, pp. 6210–6216, 2003.

[40] A. F. Voter, "Parallel replica method for dynamics of infrequent events," *Physical Review B*, vol. 57, no. 22, p. R13985, 1998.

[41] D. Perez, B. P. Uberuaga, Y. Shim, J. G. Amar, and A. F. Voter, "Accelerated molecular dynamics methods: Introduction and recent developments," *Annual Reports in Computational Chemistry*, vol. 5, pp. 79–98, 2009.

[42] G. Simpson and M. Luskin, "Numerical analysis of parallel replica dynamics," *Multiscale Modeling & Simulation*, vol. 9, no. 2, pp. 593–606, 2011.

[43] G. Henkelman, B. P. Uberuaga, and H. Jónsson, "A climbing image nudged elastic band method for finding saddle points and minimum energy paths," *The Journal of Chemical Physics*, vol. 113, no. 22, pp. 9901–9904, 2000.

[44] L. Xu and G. Henkelman, "Adaptive kinetic monte carlo for first-principles accelerated dynamics," *The Journal of Chemical Physics*, vol. 129, no. 11, p. 114104, 2008.

[45] A. Chatterjee and D. G. Vlachos, "An overview of spatial microscopic and accelerated kinetic monte carlo methods," *Journal of Computer-Aided Materials Design*, vol. 14, no. 2, pp. 253–308, 2007.

[46] J. A. Mitchell, F. Abdeljawad, C. C. Battaile *et al.*, "Parallel simulation via spparks of on-lattice kinetic and metropolis monte carlo models for materials processing," *Modelling and Simulation in Materials Science and Engineering*, vol. 31, no. 5, p. 055001, 2023.

[47] J. Kimari, V. Jansson, S. Vigonski, E. Baibuz, R. Domingos, V. Zadin, and F. Djurabekova, "Application of artificial neural networks for rigid lattice kinetic monte carlo studies of cu surface diffusion," *Computational Materials Science*, vol. 183, p. 109789, 2020.

[48] J. R. Gonzalez and D. G. Vlachos, "Machine learning-based kinetic monte carlo simulations," *AIChE Journal*, vol. 65, no. 3, pp. 1006–1016, 2019.

[49] N. Castin, L. Messina, C. Domain, R. C. Pasianot, and P. Olsson, "Improved atomistic monte carlo models based on ab-initio-trained neural networks: Application to fecu and fecr alloys," *Phys. Rev. B*, vol. 95, p. 214117, Jun 2017. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.95.214117

[50] Q. Lin, C. Zhang, X. Meng, and Z. Guo, "Monte carlo physics-informed neural networks for multiscale heat conduction via phonon boltzmann transport equation," *arXiv preprint arXiv:2408.10965*, 2024.

[51] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoesel, H. Schopmans, T. Sommer *et al.*, "Graph neural networks for materials science and chemistry," *Communications Materials*, vol. 3, no. 1, p. 93, 2022.

[52] K. Choudhary and B. DeCost, "Atomistic line graph neural network for improved materials property predictions," *npj Computational Materials*, vol. 7, no. 1, p. 185, 2021. [Online]. Available: https://doi.org/10.1038/s41524-021-00650-1

[53] V. Fung, J. Zhang, E. Juarez, and B. G. Sumpter, "Benchmarking graph neural networks for materials chemistry," *npj Computational Materials*, vol. 7, no. 1, p. 84, 2021. [Online]. Available: https://doi.org/10.1038/s41524-021-00554-0