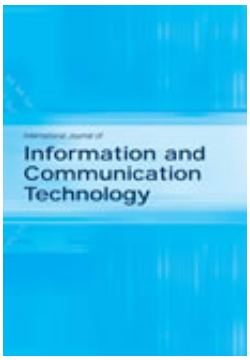
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Power control in semiconductor chips: integration of reinforcement learning and dynamic modelling

Zhanhan Hu

DOI: 10.1504/IJICT.2025.10074944

Article History:

Received:	30 August 2025
Last revised:	26 September 2025
Accepted:	26 September 2025
Published online:	17 December 2025

Power control in semiconductor chips: integration of reinforcement learning and dynamic modelling

Zhanhan Hu

School of Computer Science,
North China Institute of Aerospace Engineering,
Langfang, 065000, China
Email: xiaohan9192@126.com

Abstract: As semiconductor technology enters the nanoscale era, power control has become a major challenge impeding chip performance. Traditional heuristic methods fail to handle complex dynamic workloads, while pure reinforcement learning lacks stability and safety. This paper proposes a hybrid intelligent control framework integrating reinforcement learning with dynamic physical modelling. Offline-trained power and performance models offer prior guidance and safety constraints for online decisions. Experiments using Google production cluster data show the framework achieves an average power of 101.3 W – 21.3% lower than on-demand strategies – and a tail latency of 34.1 ms with only 1.2% violation rate. The energy efficiency reaches 15.6 instructions per joule, outperforming existing methods. This study provides an effective solution for energy-aware chip-level power management and introduces new ideas for intelligent cyber-physical systems.

Keywords: power consumption control; reinforcement learning; dynamic modelling; energy efficiency optimisation; chip management.

Reference to this paper should be made as follows: Hu, Z. (2025) 'Power control in semiconductor chips: integration of reinforcement learning and dynamic modelling', *Int. J. Information and Communication Technology*, Vol. 26, No. 46, pp.76–94.

Biographical notes: Zhanhan Hu received her Bachelor's degree from the North China Institute of Aerospace Engineering in 2013, and received her Master's degree from the North China Institute of Aerospace Engineering in 2015. Currently, she is employed at the North China Institute of Aerospace Engineering. Her research interests are semiconductor devices, semiconductor and communications.

1 Introduction

The rapid development of conductor technology has always followed the guidance of Moore's law, successfully integrating billions of transistors on chips between inches, which have greatly promoted the exponential growth of computing performance. However, this increasing level of integration has also brought unprecedented challenges, the most serious of which is the 'power wall' problem. As the process node enters the deep submicron and even nanometre era, the static leakage current power consumption increases sharply, and the dynamic power density continues to rise, making thermal

management and energy efficiency become the core bottleneck restricting the further improvement of computing power (Horowitz, 2014). In the fields of high-performance computing, data centre and mobile computing, the peak power consumption and total energy consumption of the chip are not only directly related to operating costs and carbon emissions, but also pose a limit challenge to the reliability, stability and physical heat dissipation design of the system (Borkar, 2007). Therefore, the development of advanced, intelligent and adaptive chip power control technology is no longer a simple academic pursuit, but an urgent engineering demand for the sustainable development of the entire information technology (IT) industry.

In response to this challenge, industry and academia have long been committed to studying various chip-level power management technologies. Dynamic voltage and frequency scaling (DVFS) and power gating are the most classical and widely used techniques. These technologies are usually based on pre-set heuristic rules or look-up tables, and periodically adjust the operating voltage and frequency according to limited indicators such as current processor utilisation, temperature, etc. (Min-Allah et al., 2007). Such methods are simple to implement, but their static and passive control strategies are inadequate when dealing with complex, variable and highly non-stationary workloads generated by modern applications. They are often unable to make a fast and refined optimal trade-off between performance and power consumption, either too conservative to lead to low energy efficiency, or too aggressive to cause performance jitter or quality of service (QoS) violations (Basmadjian and De Meer, 2012). On the other hand, optimisation methods based on control theory [such as model predictive control (MPC)] attempt to achieve better control by establishing accurate mathematical models (Zanini et al., 2010). However, the power consumption dynamics of the chip is a complex nonlinear process involving electrical, thermal, and workload characteristics. It is extremely difficult to model it with high fidelity, and the model parameters often drift with process, voltage, and temperature (PVT) changes, resulting in limited generalisation ability and practicality of such methods.

In recent years, artificial agent technology represented by reinforcement learning (RL) has provided a new paradigm for solving the above problems. Through continuous interaction with the environment, the RL agent can autonomously learn the best action strategy that should be taken in a specific state to maximise the long-term cumulative reward. This feature makes it very suitable for sequence decision problems such as power control. Some groundbreaking studies have demonstrated the great potential of RL in managing power consumption on single-core, multi-core processors and even data centre levels (Ranjbar et al., 2023). These pure data-driven methods do not need to master the precise physical model of the system in advance. In theory, they can adaptively capture complex workload patterns through training and make decisions that go beyond static rules (Mnih et al., 2015). However, the direct application of RL to chip control scenarios with high security and stability requirements still faces inherent severe challenges. First of all, the training process requires a lot of exploration, and random exploration behaviour can lead to catastrophic performance degradation or hardware damage on real hardware. Secondly, the sample efficiency of the RL algorithm is usually low, the convergence speed is slow, and the trained strategy may lack robustness and be unstable in the face of unknown loads outside the training data distribution (Al-Saadi et al., 2023). These limitations make the deployment of pure end-to-end RL solutions in production environments extremely risky.

In view of this, the frontier of current research is gradually shifting from the isolated application of data-driven methods or physical models to exploring how to deeply integrate the advantages of the two. An emerging idea is to construct a hybrid framework that combines the domain knowledge contained in the first-principles-based physical model with the powerful learning ability of data-driven RL (Chen et al., 2022). The physical model can provide critical system dynamic prior information, guide RL agents to explore more safely and efficiently, and accelerate the training process as a virtual environment. In turn, RL can learn to optimise and adjust model parameters, or deal with complex nonlinear relationships that are difficult to describe with simple mathematical models. This fusion indicates the birth of a new paradigm: it not only has the interpretability and security of the model method, but also has the adaptability and optimisation ability of the learning method. However, how to design an effective architecture to achieve the synergy between model and learning, especially in the specific field of semiconductor chip power control, there are still a lot of open problems to be explored in terms of model construction, state space definition, reward function design and learning algorithm selection. The research of this paper is carried out in this context and is committed to addressing the key challenges in this cross-cutting field.

2 Relevant work

2.1 Traditional chip power management technology

The early and existing chip power management technologies mostly rely on predefined heuristic rules or static strategies. DVFS and power gating are the most representative technologies, which are widely used in industry. DVFS balances performance and power consumption by dynamically adjusting the operating voltage and frequency of the processor core. Hanumaiah and Vrudhula (2012) proposed a unified energy efficiency optimisation scheme for multi-core processors that integrates dynamic frequency and voltage adjustment, thread migration and active cooling for the energy efficiency problem of cores in various computing devices. The formula contains an accurate power consumption thermal model and an extended performance per watt (PPW) index. The simulation experiment of quad-core processors shows that the energy efficiency of this strategy is 3.2 times higher than that of the optimal performance scheme. It can also quickly explore design space (such as finding the optimal number of cores to maximise PPW and has been implemented on quad-core Intel Sandy Bridge processors. Power gating reduces static power consumption by turning off the power of the idle module. These methods are usually based on look-up tables or simple control loops, and make decisions based on current central processing unit (CPU) utilisation, temperature, and other indicators. However, these methods are inadequate in the face of the complexity and variability of modern computing load. They lack adaptive ability and cannot achieve fine power control under the premise of guaranteeing QoS, and often perform poorly in the trade-off between performance and energy efficiency (Hathwar et al., 2024). In addition, these strategies are usually designed for specific hardware or workloads, and have poor portability and are difficult to adapt to chips with different architectures or dynamic environments (Basmadjian and De Meer, 2012).

2.2 Model-based optimal control method

In order to overcome the limitations of traditional methods, researchers have explored optimisation methods based on control theory and high-fidelity mathematical models. MPC is one of the typical representatives. It solves the optimisation problem in each control cycle to determine the optimal action (such as voltage and frequency setting) by constructing a dynamic model of chip power consumption, temperature and performance. Zanini et al. (2009) applied MPC to the thermal management of multi-core processors, demonstrating its potential to optimise power consumption under constraints. Aiming at the problem that the performance of the current thermal management method drops sharply due to the drastic change of the forced operation point, the thermal management is expressed as a discrete-time optimal control problem and solved by MPC to achieve smooth thermal control action and minimise the performance tracking error variance. The optimisation process takes into account the thermal characteristics of the system, time evolution and time-varying load requirements. Experiments show that the thermal balance effect is significantly improved compared with the previous methods. The effectiveness of such methods is highly dependent on the accuracy of the model. The power consumption dynamics of the chip involve complex interactions of electrical, thermal, and load characteristics. It is a nonlinear, time-varying system, so the modelling process often requires in-depth domain knowledge and complex parameter identification (Basmadjian and De Meer, 2012). Basmadjian and De Meer (2012) analysed and pointed out that the existing multi-core processor power consumption model (assuming that the multi-core power consumption in parallel computing is equal to the sum of the power consumption of each active core) is not accurate enough when applied to modern processors such as quad-core processors. Then, a multi-core processor power consumption estimation method considering resource sharing and energy saving mechanism is proposed, and the maximum error of this method is within 5%, which ensures the estimation accuracy.

However, accurate physical models are difficult to construct, and their computational complexity is high, which may be difficult to deploy in resource-constrained real-time control scenarios. In addition, Zhuo et al. (2023) proposed a dynamic power management framework based on RL to solve the problem that multi-core chips are limited by heat and power when supplying energy to mobile intelligent terminals. The framework first relies on Gem5 Simulator (GEM5) to build a multi-core chip dynamic voltage and frequency adjustment simulation system, and then uses a chip power model considering complementary metal-oxide-semiconductor (CMOS) physical characteristics to achieve real-time monitoring. Finally, a gradient reward method is designed and combined with deep Q network (DQN) learning management strategy. Simulation results show that its computational performance is 2.12% and 4.03% higher than that of the traditional Ondemand and MaxBIPS schemes, respectively. The model parameters will change due to PVT drift, resulting in model mismatch and control performance degradation (Zhuo et al., 2023).

2.3 Research on power control driven by machine learning

With the development of artificial agent technology, machine learning, especially RL, provides a new data-driven paradigm for chip power control. The RL agent can autonomously learn the optimal control strategy through interaction with the

environment, without needing to master the accurate mathematical model of the system in advance. Mnih et al. (2015) proposed the DQN algorithm, which shows the powerful ability of RL in complex decision-making tasks. It can learn end-to-end learning strategies from high-dimensional sensory input. In the Atari 2600 game test, only pixels and scores are used as input, and the unified algorithm and parameters are used to surpass the previous algorithms in 49 games, reaching the level of professional testers, bridging the gap between high-dimensional sensory input and action, and creating the first artificial agent that can learn and perform well in a variety of complex tasks.

In the field of chip power control, Yang et al. (2017) proposed a runtime framework PowerChief, an RL-based intelligent power allocation framework for multi-core systems, to solve the problem that multi-stage user applications have variable latency and existing methods do not consider their multi-stage characteristics and are difficult to improve responsiveness under power constraints. It identifies bottleneck services through joint design of services and queries, adaptively selects acceleration techniques, and dynamically redistributes power budgets, which has been evaluated by real multi-stage applications. It improves the average latency of Sirius and natural language processing applications by 20.3 times and 32.4 times, respectively (13.3 times and 19.4 times for 99% tail latency). It also reduces the power consumption of Sirius and web search applications by 23% and 33%, respectively, which is better than the previous work.

However, the pure data-driven RL method faces significant challenges in practice. First of all, its training process requires a lot of exploration, and on real hardware platforms, random exploration behaviour may lead to performance jitter, system instability, and even hardware damage (Zhang et al., 2019). Secondly, the sample efficiency of RL algorithm is usually low, the convergence speed is slow, and the trained strategy may lack robustness and perform poorly in the face of loads outside the distribution of training data (Basmadjian and De Meer, 2012). These security problems severely limit the direct deployment of pure RL schemes in the production environment.

2.4 Mixed methods research

Recognising the limitations of purely model-based or purely learning-based methods, the current research frontier is turning to a hybrid framework that combines the two. These studies aim to use the prior knowledge of the physical model to improve the safety, sample efficiency and interpretability of the learning process, and use the powerful learning ability of RL to optimise and adaptively adjust the model parameters. For example, Li et al. (2024) proposed a collaborative design optimisation method combining offline PDS design optimisation and online power management, and introduced an online collaborative management control scheme based on centralised DQN to solve the problem that Chiplet multi-core system brings challenges to power delivery system (PDS) design while improving performance and PDS efficiency is affected by workload changes. This method realises workload-aware adaptive control by designing state space and reward function. In the evaluation of 64-core system, the energy delay product (EDP) is reduced by 67% on average when the 90% performance target (PT) is reached, which is 4% and 16% higher than the advanced modular Q-learning (MQL) and heuristic method, respectively, and the action selection is better, the control is more stable, and the implementation overhead is lower.

At the macro data centre level, Zhao et al. (2024) proposed an energy optimisation framework based on RL to solve the problem of low energy efficiency in sustainable

cloud data centres due to the intermittency of renewable energy, the complexity of equipment status and action space, and the difficulty of coordinating IT and cooling resources in existing methods. The framework first improves the accuracy of RL state space information through multi-task learning long short-term memory (MTL-LSTM) joint prediction method, and then designs Bayesian double deep Q-network (BayesDDQN) method to synchronously adjust virtual machine migration and cooling parameters, and integrates pre-cooling technology. Experiments show that it reduces 2.83% overall energy consumption, 4.74% brown energy consumption and 13.48% cooling energy consumption on average, with the lowest hot spot frequency, and MTL-LSTM reduces the root mean square error (RMSE) of energy consumption and inlet temperature prediction by nearly half compared with LSTM and extreme gradient boosting (XGBoost). These works show that the integration of domain knowledge (such as power consumption, thermodynamic model) and data-driven learning is a promising direction to solve complex system control problems (Chen et al., 2022). However, how to design an efficient fusion architecture, especially to achieve deep collaboration between model and learning in chip-level control (for example, using models for reward shaping, security constraints, or simulators), is still a key issue that current research needs to explore in depth.

3 Problem formalisation: a Markov decision process (MDP) framework

3.1 Sorting target feature extraction

We model the chip power control problem as a MDP, which is the standard framework for dealing with sequential decision problems (Thrun and Littman, 2000). An MDP consists of the tuple (S, A, P, R, γ), where S stands for the state space, A for the action space, P is the state transfer probability, R is the reward function, and $\gamma \in (0, 1)$ is a discount factor to weigh the immediate versus the long-term payoff. In our application scenario, the intelligent body (controller) observes the state $s_t \in S$ acquired from the environment (system-on-chip) at each discrete time step t, and subsequently selects an action $a_t \in A$ and executes it based on its policy π. After execution, the environment transfers to a new state s_{t+1}, and gives the agent scalar reward feedback r_t. The ultimate goal of the controller is to learn an optimal policy π^* that maximises the expectation of future cumulative discounted rewards.

$$\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right] \tag{1}$$

Specifically, we define the state space based on the Google ClusterData dataset. The state vector s_t is a multi-dimensional real-valued vector containing key performance metrics (KPMs) obtained from the system monitoring unit at time step t:

$$s_t = \left[u_{cpu}(t), u_{mem}(t), f_{io}(t), T_j(t), P_{meas}(t-1)\right]^T \in \mathbb{R}^5 \tag{2}$$

where $u_{cpu} \in [0, 1]$ denotes the CPU utilisation (%), $u_{men} \in [0, 1]$ denotes the memory utilisation (%), f_{io} denotes the disk I/O frequency (in MB/s), T_j denotes the chip junction temperature (in °C), and $P_{meas}(t - 1)$ denotes the total power consumption (in W)

measured at the previous time step. Together, these state variables portray the load, thermal state, and energy consumption of the system at time \$t\$.

The action space \$A\$ is defined as the discrete or continuous control commands that can be executed by the controller. We use continuous action space to support finer control:

$$a_t = \left[\alpha_f(t), \alpha_v(t)\right]^T \in [0.7, 1.0]^2 \subset \mathbb{R}^2 \tag{3}$$

where $\alpha_f(t)$ and $\alpha_v(t)$ are scaling factors applied to the chip's core clock frequency F_{base} (Gigahertz, GHz) and operating voltage V_{base} (V), respectively. The actual frequency and voltage after the action is executed are $F(t) = \alpha_f(t) \cdot F_{base}$ and $V(t) = \alpha_v(t) \cdot V_{base}$. The reason for limiting the action to between 70% and 100% of the rated value. This relatively conservative action range setting is based primarily on considerations of system reliability and correct function. Dropping the voltage and frequency below 70% of the rated value may cause the chip to approach or enter its uncertainty region, where transistor switching delays are significantly increased and the probability of timing errors rises nonlinearly, resulting in a high risk of failure of the computational task or system lock-up. Therefore, the 70% lower limit is an engineering balance between the pursuit of energy efficiency and the safeguarding of absolute system functionality, which allows aggressive energy saving strategies while fundamentally avoiding catastrophic hardware or software failures caused by excessive voltage and frequency reduction.

The design of the reward function $R(s_t, a_t)$ is key to fusing modelling and learning, which must encode both the sometimes-conflicting goals of reducing power consumption and maintaining performance (QoS). We design the reward function as follows:

$$R(s_t, a_t) = -\left[P_{total}(s_t, a_t) + \lambda \cdot \max\left(0, L_{pred}(s_t, a_t) - L_{\max}\right)^2\right] \tag{4}$$

where $P_{total}(s_t, a_t)$ is the total power consumption (W) measured or predicted after executing the action a_t in state s_t. $L_{pred}(s_t, a_t)$ is the possible latency of the current task request as predicted by a performance model (ms). $L_{\max}$ is the maximum latency threshold that the application can tolerate (ms). $\lambda > 0$ is a hyperparameter used to fine-tune the trade-off between power consumption targets and performance penalties. $L_{pred}(s_t, a_t)$ is the likely latency of the current task request as predicted by a performance model (e.g., an increase in instruction completion time). $L_{\max}$ is the maximum latency threshold tolerated by the application, with a quadratic penalty term once the predicted latency exceeds that threshold. The $\lambda > 0$ is a hyperparameter used to fine-tune the trade-off between power consumption goals and performance penalties. This reward function is designed to encourage the agent to look for actions that reduce power consumption as much as possible while satisfying performance constraints.

3.2 Chip dynamic power modelling

In order to provide physical prior knowledge to the RL agents and to assist in reward computation, we construct a high-fidelity model of the chip's dynamic power consumption. The total power consumption P_{total} of a modern CMOS chip can usually be decomposed into three main components: dynamic power consumption $P_{dynamic}$, static power consumption P_{static}, and short-circuit power consumption P_{short} (Weste and Harris,

2015). Since the share of short-circuit power is usually small, our model focuses mainly on the first two.

The dynamic power consumption is mainly generated by the circuit switching activity, which is classically calculated as:

$$P_{dynamic} = A \cdot C \cdot F \cdot V^2 \tag{5}$$

where A is the activity factor (dimensionless, depending on the frequency of transistor switching), C is the load capacitance (F), F is the clock frequency (Hz), and V is the operating voltage (V). In practical modelling, the activity factor is highly correlated with CPU utilisation. Therefore, we use a simplified practical model:

$$P_{dynamic} = k_d \cdot F(t) \cdot V(t)^2 \cdot u_{cpu}(t) \tag{6}$$

where k_d is a process correlation coefficient (in F^{-1}, or equivalently the reciprocal of C) that needs to be fitted through the data.

Static power consumption (or leakage power consumption) is mainly caused by subthreshold leakage currents and occurs even when the transistor is off. It is strongly voltage and temperature dependent:

$$P_{static} = I_{leak} \cdot V(t) \tag{7}$$

Whereas the leakage current I_{leak} is itself a complex function of the temperature T_j and the threshold voltage s, a commonly used empirical model is (Weste and Harris, 2015):

$$I_{leak} \propto T_j^{3/2} \cdot \exp\left(-q \cdot V_{th} \middle/ \left(k \cdot T_j\right)\right) \tag{8}$$

where q is the electron charge (~1.602×10^{-19} C), k is the Boltzmann constant (~1.381×10^{-23} J/K), and is the threshold voltage (V) of the transistor. For simplicity, we fit this to a term related to voltage and temperature:

$$P_{static} = k_s \cdot V(t) \cdot T_j(t)^{3/2} \cdot \exp\left(-\frac{\beta}{T_j(t)}\right) \tag{9}$$

where k_s and β are the parameters to be fitted.

In summary, our integrated power consumption model P_{model} can be expressed as:

$$P_{model}(t) = k_d \cdot F(t) \cdot V(t)^2 \cdot u_{cpu}(t) + k_s \cdot V(t) \cdot T_j(t)^{3/2} \cdot \exp\left(-\frac{\beta}{T_j(t)}\right) + \epsilon \tag{10}$$

where ϵ is the modelling error. We use u_{cpu}, F, V, T_j, P_{meas} historical data extracted from Google ClusterData to fit the parameters k_d, k_s, β via nonlinear regression, (e.g., using the Levenberg-Marquardt algorithm) or neural networks to obtain a power consumption model $\hat{P}_{model}$ that can be used for prediction.

Performance models are used to predict the effect of control actions on task execution latency. We build a simple linear model to predict latency growth:

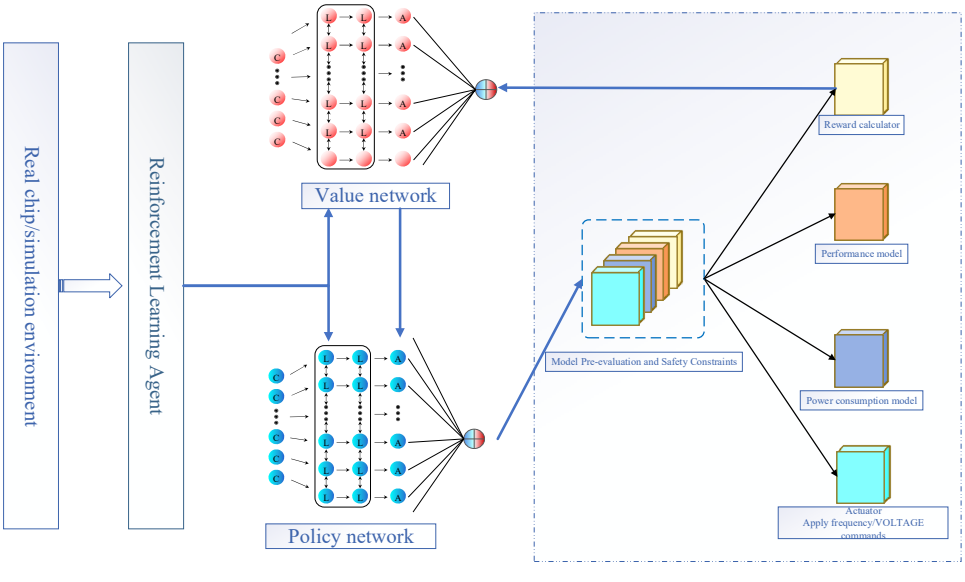
$$L_{pred}(t) = L_0 \cdot \left(1 + \beta_1 \cdot \frac{u_{cpu}(t)}{F(t)} + \beta_2 \cdot \left(1 - \alpha_f(t)\right)\right) \tag{11}$$

where L_0 is the baseline latency (ms) and β_1 (ms-GHz) and β_2 (dimensionless) are the fitting parameters. The model captures two main factors: the computational resource constraints represented by the utilisation-to-frequency ratio, and the increase in single-cycle instruction execution time due to the frequency reduction itself.

3.3 *A hybrid control framework incorporating modelling and learning*

The core idea of our proposed hybrid control framework is to use dynamic models trained offline to guide, constrain and accelerate the online RL process, the architecture of which is shown in Figure 1.

Figure 1 A hybrid control framework for chip power consumption incorporating RL and dynamic modelling (see online version for colours)



The framework consists of two core modules: offline dynamic models and online RL agents. The offline dynamic models (power model $\hat{P}model$ and performance model $Lpred$) are trained before deployment and encapsulate the physical characteristics of the chip and the general laws of the workload. The online RL agents are based on an actor-critic architecture, [e.g., soft actor-critic (SAC) or TD3] and are responsible for learning optimal control policies.

The workflow of the framework is a closed-loop control process: at each time step t, the agent observes the current state s_t from the environment. The policy network (actor) $\pi_\phi(a_t|\ s_t)$ outputs a candidate action a_t (frequency and voltage scaling factor) based on the state s_t. Before executing this action into the real system, it is first fed into an offline dynamic model for 'pre-evaluation'. The dynamic model quickly derives predicted power consumption $\hat{P}model(t)$ and performance latency $\hat{L}pred(t)$ based on the current state s_t and the candidate action a_t. These predictions are fed into a reward calculator that computes a predicted reward $\hat{r}_t = R(s_t, a_t)$. This predicted reward is compared to a safety threshold, and if the predicted reward is too low, (e.g., the predicted performance

violation is too severe), the action can be rejected or adjusted to ensure the safety of the real system.

Subsequently, the action a_t is applied to the real environment (or simulated environment). The environment is transferred to a new state s_{t+1}, and a real reward r_t is generated (based on actual measured power consumption $P_{meas}(t)$ and performance metrics). The transition experience d is stored in the experience playback buffer for training. It is worth noting that the reward calculator can flexibly fuse predicted and real rewards. In the early stages of training, the agent knows little about the environment and can rely more on the model's predictions to shape the rewards and guide the learning direction. As more real data is collected, the weight of real rewards can be gradually increased so that the strategy eventually converges to a direction that can realistically optimise the system's performance.

Critic networks s learn the action value function by minimising the mean square Bellman error:

$$\mathcal{L}(\theta) = \mathbb{E}(s, a, r, s') \sim \mathcal{D}\left[\left(Q\theta(s, a) - \left(r + \gamma, Q_{\bar{\theta}}\left(s', \pi_j\left(s'\right)\right)\right)\right)^2\right] \tag{12}$$

where s is the empirical playback buffer, s is a parameter of the target critic network (usually obtained from θ via a soft update), and $\gamma \in (0, 1)$ is a discount factor defined in Section 4.1. The actor network then maximises the expected value by gradient ascent:

$$\nabla_{\phi} J(\phi) = \mathbb{E}s \sim \mathcal{D}\left[\nabla a Q_{\theta}(s, a)\middle| a = \pi\phi(s)\nabla_{\phi}\pi_{\phi}(s)\right] \tag{13}$$

In this way, the dynamic model acts as a 'built-in tutor' that provides additional supervisory signals and security, greatly enhancing the sample efficiency and security of RL learning (Nagabandi et al., 2018).

3.4 Algorithmic implementation

We chose the SAC algorithm as the basis for the implementation of our RL agents (Haarnoja et al., 2018). SAC is a maximum entropy RL algorithm that maximises not only the standard cumulative rewards but also the entropy of the strategy. Its goal is to learn a strategy that is as random as possible, yet capable of completing the task. This property gives it significant advantages over other deterministic algorithms, (e.g., DDPG): stronger exploration, better sample efficiency, and higher robustness to hyperparameters, properties that are well suited for applications in complex chip-control environments.

The core of SAC is the introduction of an entropy term in its value function:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}\left(s_t, a_t\right) \sim \rho\pi\left[\gamma^t\left(r\left(s_t, a_t\right) + \alpha\mathcal{H}\left(\pi\left(\cdot\middle|s_t\right)\right)\right)\right] \tag{14}$$

where $\mathcal{H}$ is the entropy of the strategy π and $\alpha > 0$ is the temperature parameter (temperature parameter), which is used to trade-off between reward maximisation and entropy maximisation, i.e., the balance between exploration and exploitation.

SAC typically maintains four neural networks: an actor network (the policy network π_{ϕ}), two critic networks ($Q_{\theta_1}, Q_{\theta_2}$, to mitigate value overestimation) and a value network (V_{ψ}). Its critic loss function and strategy loss function are as follows:

$$\mathcal{L}Q(\theta_i) = \mathbb{E}(s, a, r, s') \sim \mathcal{D}\left[\left(Q_{\theta_i}(s, a) - \left(r + \gamma(V_{\bar{\psi}}(s'))\right)\right)^2\right], \quad \text{for } i = 1, 2 \tag{15}$$

$$\mathcal{L}V(\psi) = \mathbb{E}s \sim \mathcal{D}\left[\left(V_{\psi}(s) - \left(\min_{j=1,2} Q_{\theta_j}(s, \tilde{a}') - \alpha \log \pi_j(\tilde{a}' \mid s)\right)\right)^2\right] \tag{16}$$

$$\text{where, } \tilde{a}' \sim \pi_{\phi}(\cdot \mid s) \tag{17}$$

$$\mathcal{L}\pi(\phi) = \mathbb{E}s \sim \mathcal{D}\left[\alpha \log \pi_{\phi}\left(f_{\phi}(\epsilon; s) \middle| s\right) - \min_{j=1,2} Q_{\theta_j}(s, f_{\phi}(\epsilon; s))\right] \tag{18}$$

where $f_{\phi}(\epsilon; s)$ is the action sampled from the strategy using the reparameterisation technique, and ϵ is the input noise vector, usually sampled from a standard normal distribution. $\bar{\psi}$ is the parameter of the target value network.

We integrate the hybrid framework into the SAC algorithm. Specifically, experiences sampled from the playback buffer are used to update the network during the training loop. Meanwhile, the predicted rewards are computed by querying the dynamic model before the action is executed and fused with the real rewards returned from the environment in a certain ratio to form the final reward signal $r_{hybrid} = \eta\hat{r} + (1-\eta)r_{true}$, where $\eta \in [0, 1]$ is a mixing coefficient that decays from 1 to 0 over time.

4 Experimental validation

4.1 Experimental setup and baseline methodology

To validate the effectiveness of the hybrid RL framework proposed in this paper, we conduct a comprehensive experimental evaluation based on the ClusterData 2019 dataset released by Google. The dataset contains detailed workload trace records of over 12,000 machines in a large-scale production cluster for eight consecutive days in May 2019, covering CPU usage, memory usage, disk I/O, machine specifications, and task scheduling information, etc. with a sampling interval of 5 minutes (Sliwko, 2024). We randomly sampled data from 1,000 machines from which key metrics characterising computationally intensive loads: CPU utilisation (u_cpu), memory utilisation (u_mem), disk I/O rate (f_io), and power consumption values estimated based on machine specifications and loads (P_est) were selected as our main experimental data. The data from the first five days (about 1.44 million data points) is used as a training set and the data from the last two days (about 0.576 million data points) is used as a test set to evaluate the generalisation ability of the model.

We chose four representative baseline algorithms for comparison. The first one is the ondemand regulator, widely used in the Linux kernel, which rapidly ramps up to the maximum frequency when the CPU utilisation exceeds a certain threshold (typically 95%), and gradually reduces the frequency in the opposite direction, representing the heuristic practice in the industry. The second is the Powersave regulator, which always fixes the CPU at the lowest frequency, providing a lower bound reference for power consumption. The third is the deep deterministic policy gradient (DDPG) algorithm, as a classical deep RL algorithm, which was proposed and widely used for continuous control tasks by Mnih et al. (2015), and represents here the model-free RL approach. The fourth is the MPC-based method, which is adapted based on the model predictive controller for

thermal management proposed by Zanini et al. (2010), which uses the power consumption model we constructed for rolling optimisation in a finite time domain, representing the model-based optimal control method.

For the evaluation metrics, we focus on the following four aspects: average power (W): the average power consumption during the test period, which directly reflects the energy efficiency; tail latency (ms): we simulate the task processing latency and report its 99th percentile latency for evaluating the QoS; QoS violation rate (QoS violation rate, %): the percentage of time points where the latency exceeds a preset threshold (100 ms in this paper); this threshold is set with reference to widely accepted user experience standards for interactive services (e.g., web search, online recommendation systems). Numerous studies have shown that for such services, a response time of 100 ms is considered 'instantaneous' and critical to the user experience, while above this threshold the service may be considered slow. Therefore, setting it to 100 ms aims to simulate a realistic and stringent service level agreement (SLA) target, so that the optimisation strategy must seek to optimise the power consumption under this constraint, which is more in line with the optimisation requirements of real production environments. Energy efficiency (instructions/Joule): the amount of work that can be done per unit of energy consumption, which is calculated by dividing the total energy consumption by the estimated total number of instructions (which correlates with the CPU utilisation).

The hybrid framework proposed in this paper is implemented based on the SAC algorithm (Haarnoja et al., 2018). Both the policy network and the value function network use a multilayer perceptron (MLP) with two hidden layers (256 neurons per layer), and the activation function is ReLU. The learning rate is set to 3×10^{-4}, and the discount factor γ is 0.99, the empirical replay buffer size is 10^6, and the batch size is 256. The trade-off coefficient in the reward function, λ is determined by a grid search was determined to be 0.1.

4.2 Overall performance results and analysis

The overall performance comparison is shown in Figure 2. Among all the compared algorithms, the Linux ondemand regulator exhibits a low tail delay (25.3 ms), but it has the highest average power consumption (128.7 W) due to its tendency to aggressively boost the frequency at high loads to safeguard performance. On the contrary, the power save regulator achieves the lowest power consumption (89.2 W), but its tail delay is as high as 215.8 ms and its QoS violation rate is more than 30%, which is completely unable to meet the performance requirements. The pure RL approach (DDPG) shows some optimisation ability, with its average power consumption (115.5 W) lower than ondemand and tail delay (41.7 ms) staying within acceptable limits. This demonstrates the potential of RL in terms of power-performance trade-offs. However, the training process of DDPG is unstable and the final performance fails to outperform well-designed modelling approaches. The MPC-based approach (Bartolini et al., 2012) utilises an offline model for optimisation and achieves good results (average power consumption of 105.8 W and tail latency of 38.9 ms), but its performance is heavily dependent on the accuracy of the model and the computational overhead is large.

The SAC hybrid framework proposed in this paper achieves the best or near-optimal balance in all key metrics. It achieves the lowest average power consumption (101.3 W), which is 4.3% lower than the suboptimal MPC approach and 21.3% lower than the ondemand regulator. At the same time, it manages to keep tail latency at 34.1 ms,

significantly lower than powersave and DDPG, and very close to ondemand's performance level. Its QoS violation rate is only 1.2%, well below the practical threshold of 5%. This makes our approach the highest in terms of energy efficiency metrics (15.6 instructions/Joule), fully demonstrating the superiority of the convergence framework in maximising energy efficiency while safeguarding user experience.

Figure 2 Comparison of average power consumption and 99th tail delay under different control strategies (see online version for colours)

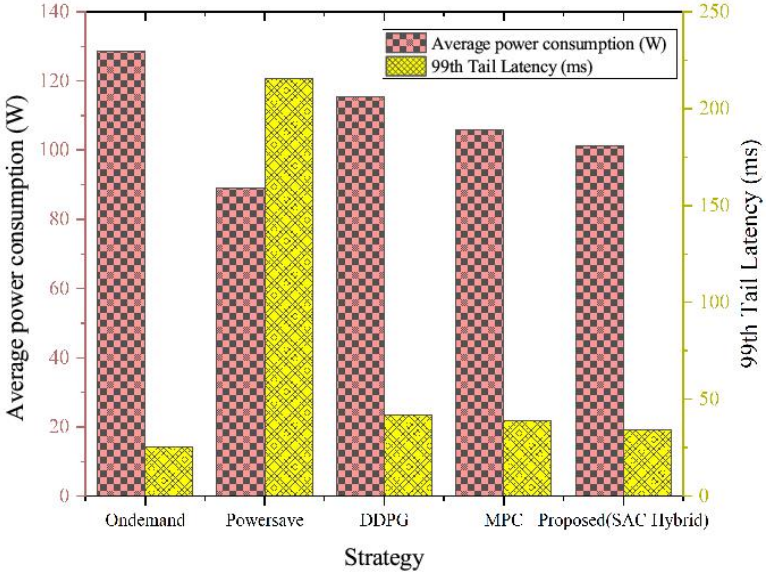
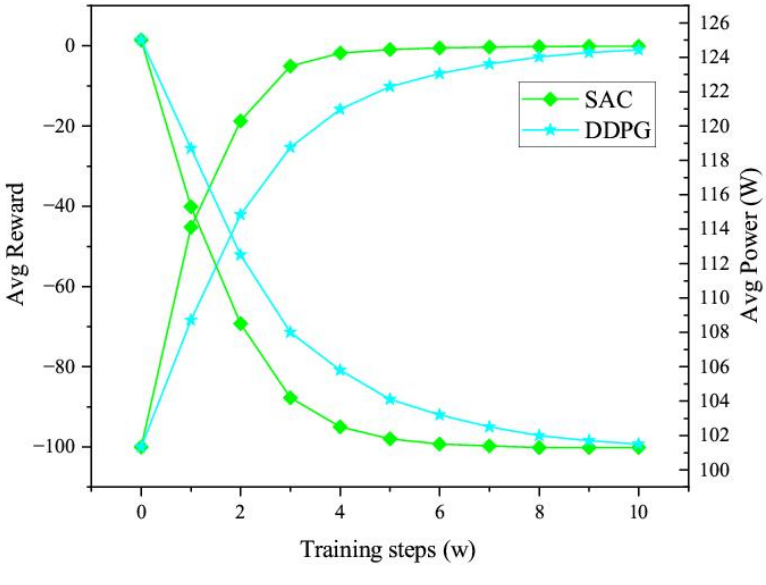


Figure 3 Average reward and power convergence curves during training of hybrid SAC and DDPG algorithms (see online version for colours)



4.3 Training efficiency and stability analysis

In order to deeply explore the efficiency and stability of the learning process, we plotted the change curves of average reward and average power consumption during the training period, as shown in Figure 3. It can be clearly observed that the training curve of the pure DDPG algorithm fluctuates drastically, and even shows a sudden degradation of performance in the middle of training, which reflects the instability and high risk of model-free RL in the exploration process. In contrast, the SAC hybrid framework in this paper benefits from the a priori reward signals and security constraints provided by the dynamic model, and its training process is very smooth and stable, exhibits low power consumption from the initial stage, and quickly converges to the vicinity of the optimal solution. This suggests that the offline model effectively steers the exploration direction of the agents, dramatically improves the sample efficiency and reduces the training risk, which is crucial for deploying learning algorithms on real physical systems.

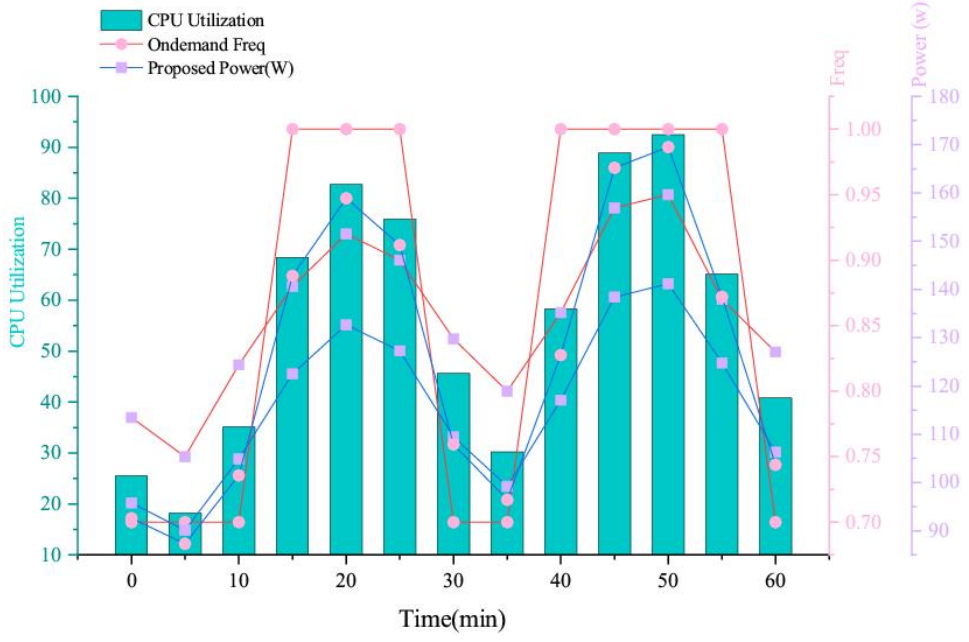
4.4 Experimental ablation studies

In order to validate the necessity of each component in the framework, we conducted an ablation study (AFS) and the results are shown in Table 1. We designed two variants: SAC (w/o model reward): remove the model prediction part of the reward function and use only the real reward returned by the environment; SAC (w/o safety check): remove the model pre-evaluation and safety checking mechanism before action execution. The experimental results show that after removing the model rewards, the agents cannot be effectively guided at the early stage of training, and their final performance is comparable to that of pure SAC but with a slower convergence rate. And once the security checking mechanism is removed, although the final average power consumption is slightly lower (100.1 W), its tail latency and QoS violation rate increase dramatically (52.4 ms and 8.5%, respectively), as the intelligent body performs some overly aggressive actions that may trigger performance risks in pursuit of extreme energy efficiency. This result clearly reveals the essential behavioural pattern of a RL intelligent in an unconstrained environment: it will spare no effort to maximise its cumulative reward. In this study, the reward function contains both goals of reducing power consumption and avoiding delay violations. After removing the safety mechanisms, the intelligence finds that it can achieve more significant energy savings, (i.e., higher positive rewards) by performing more aggressive and frequent voltage and frequency reductions, (e.g., keeping the frequency close to the 70% lower bound for a longer period of time), even if this occasionally triggers higher latency. The intelligence will ‘weigh’ the fact that occasional severe latency penalties (negative rewards) can be ‘offset’ by the positive rewards of large power savings over time, leading to a strategy that favours ‘desperate measures’. This illustrates the need for external safety constraints – they act as a no-compromise ‘guardrail’, forcing the intelligence to optimise within a feasible domain that absolutely satisfies the performance SLAs, thus avoiding this dangerous behaviour of sacrificing user experience for extreme energy efficiency to avoid the dangerous behaviour of sacrificing user experience for extreme performance. This fully demonstrates that both core designs of our framework – model-assisted reward shaping and model-based security constraints – are indispensable for achieving safe, efficient, and high-performance learning.

Table 1 Results of ablation experiments show the need for model-assisted rewards and safety constraints

Methodologies	*Average power consumption (W)*	*99th tail delay (ms)*	*QoS case rate (%)*
Full framework	101.3	34.1	1.2
SAC (w/o model reward)	103.8	36.5	2.1
SAC (w/o safety check)	100.1	52.4	8.5

Figure 4 Comparison of dynamic response of each control strategy during load fluctuation (see online version for colours)



4.5 Case studies

The control problem under steady state load is relatively simple, and the performance difference of each strategy is often not obvious, making it difficult to fully expose its advantages and disadvantages. However, the period of severe load fluctuation is the 'stress test' and 'litmus test' for the response speed, prediction ability, stability and robustness of the control strategies. During this period, the strategy needs to quickly provide sufficient computing resources to prevent performance deterioration when the load increases suddenly, and quickly converge to save energy when the load decreases suddenly, while avoiding unnecessary performance jitter introduced by too frequent or violent control actions. Therefore, this extreme scenario best highlights the essential differences between different control strategies: the lags and overshoots of heuristics, (e.g., ondemand), the unstable oscillations of pure RL methods, (e.g., DDPG), the smooth but sluggishness of modelling methods, (e.g., MPC), and the forward-looking, fast, and smoothness superiority demonstrated by our hybrid framework. Finally, we visualise the

dynamic behaviour of the different strategies through a case study. Figure 4 shows the frequency tuning and resulting power consumption and latency of each control strategy during a period of severe load fluctuations on the test set (lasting about 1 hour). The Linux ondemand regulator pulls the frequency full quickly during a sudden increase in load, resulting in a spike in power consumption, followed by a lag in frequency tuning when the load drops. The action of the DDPG appears to be a bit oscillatory and unsmooth. The MPC approach has smooth action but slightly sluggish response. The hybrid framework in this paper exhibits the most desirable control characteristics: it is able to predict the rising trend of the load in advance (thanks to the state sequence information), and smoothly and accurately boosts the frequency to cope with the load, avoiding unnecessary performance jitter; and then quickly reduces the frequency when the load is falling, thus realising the smoothest power consumption curve and stable low-latency performance. This demonstrates that the fusion framework combines the adaptive capability of the learner and the forward optimisation capability of the model.

4.6 Experimental results and analysis

The results of this study strongly confirm the great potential and unique advantages of blending physically-based dynamic modelling with data-driven RL in solving the complex sequential decision-making problem of chip power control. Compared to traditional heuristics, (e.g., Linux ondemand) and pure MPC, our hybrid framework achieves significantly better power performance tradeoffs. The key to its success lies in effectively circumventing the inherent shortcomings of each of the pure data-driven and pure modelling approaches. As pointed out by Al-Saadi et al. (2023) in their review, the application of pure RL in real systems is limited by the high risk and low sample efficiency of its exploration process. Our framework provides a safe 'simulation environment' and a priori knowledge base for agents by introducing dynamic models trained offline, which greatly constrains the exploration space and avoids catastrophic failure actions, which is in line with Nagabandi et al. (2018) proposal for a 'model-based RL' approach to robotics. 'Model-based RL with model-free fine-tuning' idea. The model-assisted reward shaping mechanism guides the agents to learn the high-performance strategies faster, while the model-based safety check acts as a guardian to ensure the safety of the learning process, effectively addressing the safety of RL deployment as emphasised by Zhuo et al. (2023).

In terms of theoretical contributions, this study goes beyond the simple application of existing algorithms to propose a novel and generalisable architectural paradigm. It deeply integrates a priori models from control theory with a posteriori learning capability from RL. Instead of simply being used as an emulator, the model is embedded as an intrinsic reasoning module in the decision loop of RL, participating in value judgement and safety verification. This architecture provides a blueprint that can be used to solve more complex system control problems, (e.g., autonomous driving, precision robot manipulation, industrial process control) that require high safety and high sample efficiency. It responds to Al-Ani and Das (2022) call for 'how to effectively incorporate domain knowledge into deep RL' by demonstrating a viable path for embedding knowledge through embodied models, rather than just adjusting network structure or reward functions.

However, there are several limitations to this study, which at the same time point to future research directions. First, our dynamic power consumption model, although

effective, is still a relatively simplified model. It does not take into account more microscopic effects within the chip, such as power consumption variations of different functional units [arithmetic and logic unit (ALU), cache, Ctrl unit], PVT variation, and characteristic drift due to aging. Future work can explore online adaptive models, where model parameters can slowly self-update as the chip ages, or integrate more refined simulation-based models (e.g., using the Gem5+McPAT simulator). Second, the current framework still requires an offline model training and RL training phase before deployment. Although the sample efficiency has been significantly improved, it is still an open challenge to realise true online lifelong learning so that the intelligences can continuously adapt to the slow variation of workload characteristics. In the future, online fine-tuning (OFT) mechanisms can be explored to mitigate the catastrophic forgetting problem through regular small updates of the policy network parameters and the introduction of continual learning (CL) techniques such as elastic weight consolidation (EWC) or progressive networking. Finally, although we used real production load data, the validation was performed in a simulation environment. The next step of research should be to move towards hardware-in-the-loop (HIL) simulation and even deployment on real chip platforms to fully evaluate its latency, overhead and robustness in real systems.

Although the hybrid framework proposed in this study shows significant advantages in chip power control, there is still room for further expansion and deepening. Based on the limitations of the current study, we propose the following three directions for future focused exploration. The first is to develop an online adaptive optimisation mechanism based on lifelong learning. The current framework relies on models and strategies trained offline, but in the future, we can introduce CL and Meta-learning to design lightweight OFT algorithms, so that the agents can continuously adjust their strategies and model parameters based on real-time system feedback, without forgetting their existing knowledge. In practice, this mechanism can enable the chip control system to adapt to the dynamic changes in workloads and the characteristic drift caused by hardware aging in the long term, significantly extend the effective life cycle of the system, and reduce the cost of re-training and manual intervention due to the failure of strategies caused by environmental changes. Continuously adjust the strategy and model parameters without forgetting the existing knowledge. The second is to build a multi-scale and cross-layer collaborative power management architecture. This study focuses on chip-level control, and in the future, we can explore the cross-level integration of chip-level control with the task scheduler, operating system, and virtual machine monitor, etc. to build a collaborative decision-making framework under the global optimisation goal. For example, a joint optimisation strategy is introduced to simultaneously adjust the chip voltage frequency and task allocation strategy. Finally, it is to promote open environment-oriented explainable and secure RL methods Current methods work well under security constraints, but in the face of highly uncertain open environments, there are still problems such as non-explainable policies and insufficient response to anomalous states. In the future, Bayesian inference, uncertainty quantification and explainable AI (XAI) techniques can be integrated to enhance the perception and decision transparency of agents to out-of-distribution (OOD) states. The above directions not only have high academic value, but also have a wide range of application prospects, which can promote smart chip power management from theoretical prototypes to large-scale engineering applications.

5 Conclusions

In this paper, a hybrid intelligent control framework integrating physical modelling and RL (SAC algorithm) is proposed for chip power control challenges. The framework provides key security constraints and a priori guidance for core decision-making by embedding offline-trained dynamic power consumption models into the online learning process, effectively balancing the efficiency and risk of exploration. Experimental validation based on Google production cluster data shows that the proposed method outperforms traditional heuristics, (e.g., Linux ondemand), pure model-free RL (DDPG), and MPC methods in key metrics such as average power consumption, tail latency, and quality-of-service violation rate, and achieves a better power-performance trade-off. This not only confirms the effectiveness of hybrid frameworks in dealing with complex sequential decision problems, but also provides practical new solutions for developing energy-efficient computing systems. What's more, this study provides a promising paradigm for achieving safe, efficient, and adaptive optimal control of complex information-physical systems, i.e., to address decision-making challenges in uncertain environments by deeply integrating domain knowledge and data-driven learning. Follow-up work will focus on adaptive refinement of the model and final HIL validation to drive the technology towards practical deployment.

Acknowledgements

This work is supported by the Youth Fund Project of Hebei Provincial Department of Education (No. QN2025375).

Declarations

All authors declare that they have no conflicts of interest.

References

Al-Ani, O. and Das, S. (2022) 'Reinforcement learning: theory and applications in hems', *Energies*, Vol. 15, No. 17, p.6392.

Al-Saadi, M., Al-Greer, M. and Short, M. (2023) 'Reinforcement learning-based intelligent control strategies for optimal power management in advanced power distribution systems: a survey', *Energies*, Vol. 16, No. 4, p.1608.

Bartolini, A., Cacciari, M., Tilli, A. and Benini, L. (2012) 'Thermal and energy management of high-performance multicores: distributed and self-calibrating model-predictive controller', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, No. 1, pp.170–183.

Basmadjian, R. and De Meer, H. (2012) 'Evaluating and modeling power consumption of multi-core processors', *Where Energy, Computing and Communication Meet*, Vol. 5, pp.1–10.

Borkar, S. (2007) 'Thousand core chips: a technology perspective', *The Design Automation*, Vol. 5, pp.746–749.

Chen, T., Chen, X., Chen, W., Heaton, H., Liu, J., Wang, Z. and Yin, W. (2022) 'Learning to optimize: a primer and a benchmark', *Journal of Machine Learning Research*, Vol. 23, No. 189, pp.1–59.

Haarnoja, T., Zhou, A., Abbeel, P. and Levine, S. (2018) 'Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor', *Machine Learning*, Vol. 13, No. 1, pp.1861–1870.

Hanumaiah, V. and Vrudhula, S. (2012) 'Energy-efficient operation of multicore processors by DVFS, task migration, and active cooling', *IEEE Transactions on Computers*, Vol. 63, No. 2, pp.349–360.

Hathwar, D.K., Bharadwaj, S.R. and Basha, S.M. (2024) 'Power-aware virtualization: dynamic voltage frequency scaling insights and communication-aware request stacking', *Computational Agent for Green Cloud Computing and Digital Waste Management*, pp.84–108 , IGI Global Scientific Publishing, New York.

Horowitz, M. (2014) '1.1 computing's energy problem (and what we can do about it)', *IEEE International Solid-State Circuits Digest of Technical Papers*, Vol. 2, pp.10–14.

Li, X., Chen, L., Chen, S., Jiang, F., Li, C., Zhang, W. and Xu, J. (2024) 'Deep reinforcement learning-based power management for chiplet-based multicore systems', *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 32, No. 9, pp.1726–1739.

Min-Allah, N., Wang, Y., Xing, J., Nisar, W. and Kazmi, A. (2007) 'Towards dynamic voltage scaling in real-time systems-a survey', *International Journal of Computer Sciences and Engineering Systems*, Vol. 1, No. 2, pp.93–103.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K. and Ostrovski, G. (2015) 'Human-level control through deep reinforcement learning', *Nature*, Vol. 518, No. 7540, pp.529–533.

Nagabandi, A., Kahn, G., Fearing, R.S. and Levine, S. (2018) 'Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning', *Robotics and Automation*, Vol. 9, No. 3, pp.7559–7566.

Ranjbar, B., Singh, A.K., Sahoo, S.S., Dziurzanski, P. and Kumar, A. (2023) 'Power management of multicore systems', *Handbook of Computer Architecture*, pp.1–33, Springer, Singapore.

Sliwko, L. (2024) 'Cluster workload allocation: a predictive approach leveraging machine learning efficiency', *IEEE Access*, Vol. 12, No. 1, p.4150.

Thrun, S. and Littman, M.L. (2000) 'Reinforcement learning: an introduction', *AI Magazine*, Vol. 21, No. 1, pp.103–103.

Weste, N.H. and Harris, D. (2015) *CMOS VLSI Design: A Circuits and Systems Perspective*, Vol. 1, p.1, Pearson Education, India.

Yang, H., Chen, Q., Riaz, M., Luan, Z., Tang, L. and Mars, J. (2017) 'Powerchief: intelligent power allocation for multi-stage applications to improve responsiveness on power constrained CMP', *Annual International Symposium on Computer Architecture*, Vol. 21, pp.133–146.

Zanini, F., Atienza, D., Benini, L. and De Micheli, G. (2009) 'Multicore thermal management with model predictive control', *Circuit Theory and Design*, Vol. 12, pp.711–714.

Zanini, F., Jones, C.N., Atienza, D. and De Micheli, G. (2010) 'Multicore thermal management using approximate explicit model predictive control', *IEEE International Symposium on Circuits and Systems (ISCAS)*, Vol. 6, pp.3321–3324.

Zhang, Z., Zhang, D. and Qiu, R.C. (2019) 'Deep reinforcement learning for power system applications: an overview', *CSEE Journal of Power and Energy Systems*, Vol. 6, No. 1, pp.213–225.

Zhao, D., Zhou, J., Zhai, J. and Li, K. (2024) 'A reinforcement learning based framework for holistic energy optimization of sustainable cloud data centers', *IEEE Transactions on Services Computing*, Vol. 15, No. 1, p.10041.

Zhuo, C., Zeng, X., Chen, Y., Sun, S., Luo, G., He, Q. and Yin, X. (2023) 'Multi-core chip dynamic power management framework based on reinforcement learning', *Journal of Electronics & Information Technology*, Vol. 45, No. 1, pp.24–32.