

RHODES: Robust Optimization for Uncertainty-Aware Design of CO₂-Efficient Computing Systems

Mariam Elgamal¹, Abdulrahman Mahmoud², Gu-Yeon Wei¹, David Brooks¹, Gage Hills¹

¹Harvard University, ²MBZUAI

Abstract—The *inherent uncertainty in quantifying carbon footprint* is a major challenge for designing environmentally sustainable computing systems. While existing efforts propose a variety of approaches to address this uncertainty, optimizing carbon footprint and energy efficiency together, while also accounting for *uncertainty in carbon footprint*, remains challenging. We identify two types of uncertainty in CO₂-aware hardware design: *data value* ($\mathcal{D}_V$) uncertainty, where probability distributions of CO₂-related parameters exhibit high standard deviation (e.g., fluctuations in CO₂ emissions of the power grid), and *data availability* ($\mathcal{D}_A$) uncertainty, in which probability distributions of CO₂-related parameters cannot even be reliably estimated due to a lack of data (e.g., carbon emissions during integrated circuit fabrication, with very few data points available). $\mathcal{D}_A$ uncertainty is especially challenging. To address this challenge, we leverage mathematical Robust Optimization (RO) techniques that enable uncertainty-aware decision-making *without* requiring explicit probability distributions. We present RHODES, a robust optimization framework for designing CO₂-efficient computing systems under carbon footprint data uncertainty. RHODES jointly models $\mathcal{D}_V$ and $\mathcal{D}_A$ uncertainties at multiple layers of the computing stack to optimize total carbon (tC) under uncertainty, and can target 3 different primary objectives (we include example results for specific systems, details in the full paper): (1) *minimizing total carbon given a performance constraint*: designs that do not account for uncertainty incur $1.7\times$ worse tC for the same performance constraint; (2) *minimizing execution time given a tC constraint*: RHODES produces robust designs with $1.36\times$ lower tC and only 0.2% execution time degradation, for complex heterogeneous system-on-chip (SoC), considering domain specific accelerators and workload level parallelism; and (3) *minimizing total carbon delay product (tCDP, a metric of CO₂ efficiency)*: improves tCDP by $1.3\text{--}3.17\times$ vs. state-of-the-art CO₂-aware optimization frameworks.

I. INTRODUCTION

Computing is becoming increasingly resource-intensive. For instance, datacenter power demand in the United States alone is projected to grow by 5–15% annually, primarily driven by Artificial Intelligence (AI) workloads and the rapid adoption of large-scale Large Language Models [22]. This growth entails an increase in carbon footprint emissions, which reflects in 2025 corporate sustainability reports, including 23.4% increase of Microsoft’s emissions since 2020 [47] and 51% increase of Google’s emissions since 2019 [30].

Notably, recent investigations show that carbon footprint emissions of in-house datacenters at major technology companies are $7.62\times$ higher than officially reported between 2020 and 2022 [52]. A recent study projects US datacenter AI emissions to incur between 24 to 44 million metric tons of CO₂ equivalents (CO₂e) annually by 2030 [74]. The nearly twofold spread between projections is equivalent to adding 5.6 to 10.3 million gasoline-powered cars on the road annually¹, highlighting the

scale and uncertainty in carbon emissions. The rapid growth and variation in emissions reporting reveal deep uncertainties in carbon footprint quantification. This variability and uncertainty in quantifying carbon footprint are not merely a reporting problem, but a fundamental design challenge that needs to be explicitly accounted for when optimizing CO₂-aware designs.

Quantifying the total carbon footprint of a computing system over its lifetime (tC, quantified in mass of CO₂e) includes both *operational carbon* due to hardware use, and *embodied carbon* due to hardware manufacturing [33]. Existing works have developed models and tools to quantify the carbon footprint of a wide range of computing systems [13], [15], [26], [27], [31], [32], [39], [41], [60]. Despite these efforts, carbon quantification data sources are rampant with variability and limited data availability resulting in major uncertainties in the underlying parameters for quantifying carbon footprint.

Uncertainty in carbon footprint results from multiple sources at multiple levels of the computing stack, described in detail in this paper. These include embodied carbon parameters, such as: Materials Per Wafer (MPW) from materials sourcing, Energy Per Wafer (EPW) from energy consumption in integrated circuit (IC) fabrication facilities, and Gases Per Wafer (GPW) from direct gas emissions during fabrication. These parameters suffer from either limited data, e.g. few data points with low statistical confidence, or the use of outdated data, e.g. data from previous generations of technology nodes applied inadequately to more advanced nodes. For instance, embodied carbon parameters from semiconductor supply chains are often inconsistent across vendors and reporting methodologies, and may consist of as few as one data point per technology node (detailed examples in §III-B). Operational carbon parameters also have significant variations and fluctuations beyond the designer’s control, these include Carbon Intensity of use (CI_{use}) dependent on the availability of renewables/non-renewables, and operational time ($t_{\text{operational}}$), which quantifies the duration of hardware use throughout the system’s lifetime. In contrast, carbon intensity data from electrical grids are abundant, regularly updated and in cases live-trackable [1], [53]. This dichotomy creates a fundamental challenge for CO₂-aware hardware design optimization, where to optimize the tC of computing systems, methods must remain effective in *both*.

Toward formalizing a framework to quantify carbon footprint in the presence of such wide variability and uncertainty, we classify two types of uncertainty in CO₂-aware hardware design:

- **Data Value ($\mathcal{D}_V$)**: This uncertainty arises when data are sufficiently available but exhibit variability, such as high standard deviation. Such uncertainty can be addressed using established empirical and learning-based methods [42], [72] that leverage the statistical distributions of the data. A representative example is CI_{use} .

¹Quantified using the US Environmental Protection Agency (EPA) greenhouse gas equivalencies calculator [69].

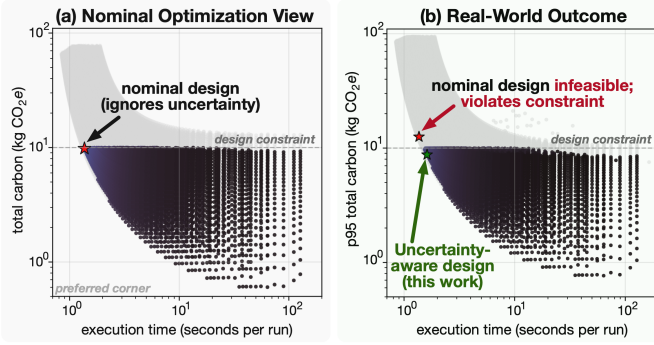


Fig. 1. Design space exploration of heterogeneous SoC across CPU cores and GPU SMs. (a) The nominal design, optimized without accounting for carbon footprint uncertainty, appears feasible but violates the total carbon constraint once realistic variability is considered. (b) RHODES optimizes with uncertainties in multiple carbon footprint parameters, producing an uncertainty-aware design that satisfies design constraints and achieves better carbon efficiency.

- **Data Availability ($\mathcal{D}_A$):** This uncertainty emerges when data are limited (e.g. few data points with wide spread). In this case, the underlying probability distribution functions (e.g. mean, standard deviation, higher order moments) cannot be reliably estimated, due to a lack of statistical significance from a small sample size (as demonstrated in II-A). This includes embodied carbon parameters MPW, EPW, and GPW.

Fig. 1 illustrates the impact of uncertainty on designing carbon-efficient computing systems. In this example, we optimize the configuration of a system-on-chip (SoC), including CPU cores and GPU streaming multiprocessors (SMs), to minimize execution time subject to a tC constraint. To quantify the benefits of our approach, we refer to both the *nominal design* and the *robust design*. The *nominal design* is the design optimized without accounting for uncertainty in tC parameters (i.e. assuming that all parameters have their nominal values). This serves as our baseline. The *robust design* is the design optimized while explicitly accounting for uncertainty in tC parameters (this work). We consider the impact of variations on the total carbon footprint of *both* designs. In Fig. 1a, the nominal design appears optimal when evaluated under fixed parameter assumptions. However, under realistic variations the nominal design becomes suboptimal and violates the design constraint (Fig. 1b). This highlights the need for tractable design optimization methodologies that can jointly optimize over multiple uncertain tC parameters, especially when data and distribution knowledge are limited. Even then, optimizing the tC of a computing system under uncertainty along with conventional power-performance-area optimizations becomes a different undertaking.

To that end, how can we, as computer designers and researchers, systematically and tractably reason about multiple uncertainties, some with unknown distributions, in multiple design parameters while optimizing for different workloads on different architectures? To address this question, we leverage Robust Optimization (RO) [7], [9], a mathematical optimization paradigm for uncertainty-aware decision-making with computational tractability in high dimension problem spaces. RO enables systematic design under uncertainty without requiring explicit probability distributions. This makes RO particularly well-suited for CO₂-aware design, where some parameters may have enough data to rely on distribution properties ($\mathcal{D}_V$) and other

parameters may have limited data ($\mathcal{D}_A$) as few as one or four data points! Fig. 1b demonstrates how the robust design (green star) remains feasible and carbon-efficient under these uncertainties.

We develop RHODES², a Robust Optimization framework to design carbon-efficient computing systems under carbon footprint data uncertainty. RHODES, implemented in Julia, takes input parameters from multiple layers of the computing stack, including fabrication (e.g. process technology node, die area, embodied carbon data), architectural specifications (e.g. CPU cores, GPUs, domain-specific accelerators, memory, power and bandwidth constraints), and workload characterization data based on HILP’s [55] software profiling. Given designer-specified constraints (e.g. power, performance, area, and tC), RHODES jointly models $\mathcal{D}_A$ and $\mathcal{D}_V$ uncertainties and creates the vectors and linearized constraints required by our RO formulation. RHODES then invokes an optimization solver (e.g. Gurobi [34]) to identify the carbon-efficient system configuration under uncertainty.

RHODES targets *three primary optimization objectives*, enumerated below. For each objective, we perform Monte Carlo simulations to evaluate the impact of uncertainty on both *nominal designs* and *RHODES designs*, where we show for a variety of case studies:

- 1 **minimize total carbon given a performance constraint (§V-A):** For SoC configurations with CPU cores and GPU SMs, we show nominal designs incur $1.7\times$ worse tC for the same performance constraint when accounting for uncertainty.
- 2 **minimize execution time given a total carbon constraint (§V-B):** For SoC configurations with CPU cores and GPU SMs, RHODES produces designs with $1.58\times$ lower tC with only 0.2% execution time degradation and only 1.1% tC constraint violation versus 54.4% tC violation for nominal designs. We show accounting for memory tC in SoC optimization is important, where ignoring it can underestimate tC by $2.28\times$ and lead to over-provisioned compute configurations that violate tC constraints. We further demonstrate the versatility of RHODES by modeling complex heterogeneous design spaces, spanning CPUs, GPUs, domain specific accelerators (DSAs), main memory, and workload-level parallelism (WLP): RHODES consistently produces robust designs within $0.951\text{--}1.147\times$ of tC simulated using Monte Carlo, versus $1.17\text{--}1.55\times$ underestimation by nominal optimization.
- 3 **minimize total carbon delay product (tCDP) (§V-C):** tCDP [23] is a carbon efficiency that balances energy efficiency and embodied carbon. Compared to CORDOBA [23], a state-of-the-art optimization framework for CO₂-aware design, which optimizes tCDP across uncertain operational time, RHODES achieves $1.3\text{--}3.17\times$ better tCDP and further prunes 80–90% of CORDOBA’s design space to uncover a more carbon-efficient design not captured by CORDOBA. Beyond operational time, we show that fabrication uncertainties (e.g. variable abatement rates) can shift the optimal robust architecture, including when operational carbon dominates improving tCDP by $1.02\times$. We show incorporating architectural redundancy improves tCDP by $1.01\text{--}2.98\times$ in GPU dominant SoCs.

²<https://github.com/mariamalgamal/RHODES>

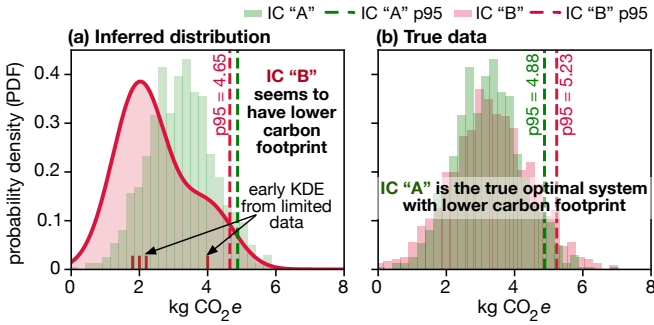


Fig. 2. While statistical methods are useful in characterizing uncertainties, inferring distributions, e.g. using kernel density estimation (KDE), with limited data can lead to incorrect conclusions and optimizations.

II. RELATED WORKS AND BACKGROUND

We begin by discussing prior statistical methods used to tackle uncertainty in quantifying carbon footprint (§II-A), and related works (§II-B). §II-C provides a brief primer for RO.

A. Understanding the Impact of $\mathcal{D}_A$ Uncertainty

Here we focus on the impact of $\mathcal{D}_A$ uncertainty, when there is very little data. Toward understanding uncertainty in CO₂-aware computing, recent works have proposed statistical frameworks, e.g. using kernel density estimations (KDE), to capture uncertainties in carbon footprint parameters through estimating their probability density functions (PDFs) [4], [18], mapping $\mathcal{D}_A$ uncertainty into $\mathcal{D}_V$ uncertainty. In Fig. 2, we demonstrate how statistical inference with limited data can lead to misleading conclusions.

Consider, as an illustrative example, two integrated circuits (ICs) “A” and “B”. IC “A” is manufactured at a lagging technology node with abundant data to quantify the embodied carbon footprint estimate. IC “B” is fabricated at a more advanced technology node, which has only four available data points. By applying KDE using a gaussian kernel to IC “B”’s sparse data and computing the 95th-percentile (p95) carbon footprint, the designer may conclude that IC “B” has lower carbon footprint than IC “A” (Fig. 2a). However, as more data become available for IC “B”’s process technology node, the estimated p95 reverses, revealing that IC “A” has the lower carbon footprint (Fig. 2b). This example highlights how statistical methods alone are insufficient when data are sparse, limiting their effectiveness for large-scale CO₂-aware design optimization across many uncertain parameters.

Unlike power or performance which can be empirically measured and calibrated [49], carbon quantification depends on incomplete and a multitude of data sources from supply chain life cycle assessments [5], [13], [15], and other carbon footprint models [27], [31], [32], [35], [39], [41]. Capturing these diverse sources of uncertainty and encoding them appropriately within optimization frameworks is essential for robust CO₂-aware design.

B. Related Works

There are a range of works on CO₂-aware design in the presence of uncertainty, and a wide range of statistical methods to deal with variability. Such methods, including stochastic optimization [72], Bayesian optimization [42], and statistical machine learning methods [42], along with many of the works described

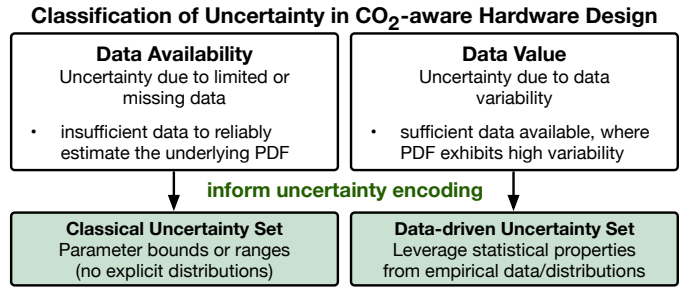


Fig. 3. RHODES classification of uncertain data in CO₂-aware hardware design.

below are well-suited when uncertain data are available ($\mathcal{D}_V$ uncertainty). However, many of these approaches are not specifically designed to deal with $\mathcal{D}_A$ uncertainty, and thus their effectiveness degrades when data are extremely sparse (e.g. the KDE example in §II-A). The benefit of *Robust Optimization* (RO) is it does not rely on the underlying probability distribution function, and instead relies on the formulation of deterministic uncertainty sets (§III-A). RHODES is specifically designed to deal with $\mathcal{D}_A$ uncertainty and also simultaneously deals with $\mathcal{D}_V$ uncertainty, making it especially suitable for CO₂-aware design optimization.

Prior works have quantified and modeled the rising environmental impact and carbon emissions of computing systems [13], [16], [24], [27], [31], [33], [36]–[39], [43], [54], [57], [71]. To enable designers to trade-off environmental impacts of competing systems, GreenChip [41] quantifies embodied and operational carbon of a wide range of computing systems, and integrates with architecture simulators to evaluate system trade-offs using indifference point analysis. To quantify carbon footprint of chips manufactured at sub-28 nm technology nodes, ACT leverages fabrication characterization and industry data, and proposes carbon-centric design metrics [32]. ECO-CHIP [60], 3D-Carbon [75], and EPiCarbon [26] model the carbon footprint of emerging integration techniques and technologies, such as three-dimensional ICs, chiplets and electro-photonics accelerators.

To address the inherent uncertainties in carbon footprint data, FOCAL proposes a parameterized first-order model using first principles to evaluate processor sustainability [21]. Prior efforts such as GreenChip utilize indifference point analysis to identify break-even points between two designs [31], [41]. CarbonClarity leverages statistical methods, such as KDE, to model temporal and spatial uncertainties in embodied carbon footprint [18]. To optimize computing systems for carbon efficiency, CORDOBA utilizes the Lagrange multiplier when the carbon intensity of use is unknown at design time, as well as heuristic methods to optimize over uncertain hardware operational time [23]. Neoscope proposes an Integer Linear Programming (ILP) tool to investigate SoC architectures resilient to workload churn and leverages FOCAL to quantify SoC carbon efficiencies [56].

Apart from sustainable computing, prior works have explored a plethora of statistical methods to better characterize and quantify the impact of uncertainty, such as runtime variability, on design decisions and simulations [3], [14], [20], [28], [29], [44], [46], [59]. To quantify architecture risk, prior work utilizes Monte Carlo-based uncertainty injection to model process, design, and projection variations using synthetic distributions [20].

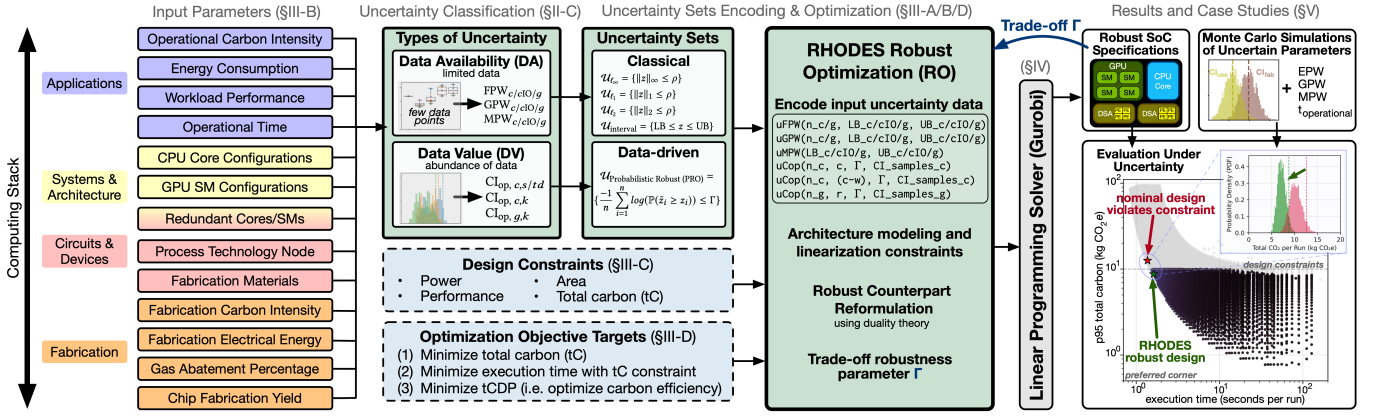


Fig. 4. RHODES, our robust optimization framework for uncertainty-aware design of carbon-efficient computing systems.

C. Robust Optimization Primer

Optimization methods offer tractable and practical solutions for high dimensional problems. We leverage *Robust Optimization (RO)*, a subfield of optimization that explicitly encodes the uncertainty in the optimization problem, by systematically separating uncertain numerical data from the deterministic structure of the optimization problem, which is known in advance and common across all realizations of the uncertainty [7], [9]. RO transforms the underlying uncertainties in the problem into tractable deterministic formulations by defining *uncertainty sets* (described in §III-A), which bound the possible realizations of the uncertain parameters [9]. RO does not rely or require assumed or known data distributions. This is a key advantage of RO, where uncertainty sets can be constructed using the properties of existing distributions or empirical data, without needing the full data distribution. This enables RO to maintain the tractability of deterministic optimization models, while enabling robustness to parameter variations.

Implementing an RO problem begins similarly as a standard deterministic optimization formulation: defining the objective function, design constraints, and the uncertain parameters. However, the key difference is the robust solution must remain feasible for all realizations of the uncertain parameters within the uncertainty set, which introduces an infinite number of constraints that cannot be solved directly. To preserve tractability, we reformulate the problem into an equivalent deterministic optimization problem, the *robust counterpart*, introducing additional variables and constraints to model the uncertainty. This reformulation method collapses all possible uncertainty realizations, as defined by the uncertainty set, into a deterministic problem that can be efficiently solved using conventional optimization solvers. Deriving the robust counterpart typically requires advanced knowledge of duality theory (we refer interested readers to literature [7], [9], [11]).

RHODES leverages RO and bridges the gap between RO theory and practical CO₂-aware design. Fig. 3 illustrates how we classify uncertain parameters in CO₂-aware hardware design, which informs the mathematical formulations of our uncertainty set encoding detailed in §III-A.

III. RHODES FRAMEWORK

In this section, we dive into the implementation details of our RHODES framework. First, §III-A details the mathematical

TABLE I

PARAMETERS IN RHODES. PARAMETERS WITH UNCERTAINTY ARE **BOLDED**.

Type	Symbol	Description	Units
Input Parameters	EPW	Electrical energy consumption in IC fabrication	kWh cm ⁻²
	MPW	Emissions due to materials procured for fabrication	g CO ₂ e cm ⁻²
	GPW	Direct gases emitted during IC fabrication	g CO ₂ e cm ⁻²
	D ₀	Defect density for yield estimation	defects cm ⁻²
	t_{operational}	Operational time quantified by number of cycles	workload runs
	CI_{fab}	Carbon intensity of fabrication energy source	g CO ₂ e kWh ⁻¹
	CI_{use}	Use-phase carbon intensity	g CO ₂ e kWh ⁻¹
	T _{k,i}	Execution time of workload <i>k</i> on die <i>i</i>	s
	P _{k,i}	Power per workload <i>k</i> on die <i>i</i>	W
	A _i	Area of die <i>i</i>	mm ²
Decision Variables	c_n	Number of CPU cores <i>n</i>	count
	g_m	Number of GPU SMs <i>m</i> and frequency <i>f</i> in MHz	count
Output Parameters	A _{total}	Total area of SoC components (c _n , g _m)	mm ²
	T _k	Delay of workload <i>k</i> on SoC	s
	E _k	Operational energy per workload <i>k</i> on SoC	kWh
	K _k	Total cost per workload <i>k</i> on SoC	\$ USD
	tC _k	Total carbon footprint of SoC	g CO ₂ e

formulations of RHODES uncertainty sets implementation. Second, §III-B details the parameters we consider for quantifying total carbon and their uncertainties, which are then encoded into the *uncertainty sets*. Third, we detail how RHODES performs architectural modeling based on heterogeneous System-on-Chip (SoCs) to model a rich set of design requirements, constraints, and choices for uncertainty-aware design space optimization (§III-C). While this paper focuses on heterogeneous SoC design space, RHODES is a modularized uncertainty-aware framework for carbon-efficient design that can be easily extended to other interesting architectural design spaces. Lastly, §III-D illustrates the variety of optimization targets and constraints supported by RHODES. Fig. 4 shows an overview of our RHODES framework. Table I describes the input parameters, decision variables, and output parameters of RHODES.

A. Encoding the Uncertain Data into Uncertainty Sets

We formalize the mathematical formulations of the uncertainty sets used to encode the uncertainty parameters described in §III-B, following the classification of $\mathcal{D}_A$ versus $\mathcal{D}_V$ in Fig. 3, where each uncertainty set is denoted by $\mathcal{U}$. Uncertainty sets capture the range of possible realizations for each uncertain parameter (extensive details in [9]) and are the foundation of the RO formulation in RHODES.

$\mathcal{D}_A$ Uncertainty Sets. $\mathcal{D}_A$ uncertainties are modeled using fundamental classical uncertainty sets (eq. 1), that do not require

complete knowledge of the underlying data distribution. Common choices include the box uncertainty $\mathcal{U}_{\infty}$, which bounds uncertain parameters within the interval $[-\rho, \rho]$, as well as the interval set $\mathcal{U}_{\text{interval}}$, where users can define explicit upper (UB) and lower (LB) bounds for their uncertain parameters (others in [9]). Both uncertainty sets represent cases where no additional information is available on the statistical relationships among uncertain variables, which currently holds for many tC parameters (e.g. EPW, MPW, GPW). With more knowledge of the parameter relationships (even without more data), less conservative uncertainty sets, such as $\mathcal{U}_{\ell_1}$ based on the L1 norm and $\mathcal{U}_{\ell_2}$ uncertainty set based on the euclidean (i.e. L2) norm³ can be employed.

$$\begin{aligned}\mathcal{U}_{\infty} &= \{z \in \mathbb{R}^n : \|z\|_{\infty} \leq \rho\}, \quad \mathcal{U}_{\ell_1} = \{z \in \mathbb{R}^n : \|z\|_1 \leq \rho\}, \\ \mathcal{U}_{\ell_2} &= \{z \in \mathbb{R}^n : \|z\|_2 \leq \rho\}, \quad \mathcal{U}_{\text{interval}} = \{z \in \mathbb{R}^n : \text{LB} \leq z \leq \text{UB}\} \quad (1) \\ \mathcal{U}_{\text{budget}} &= \{z \in \mathbb{R}^n : \|z\|_1 \leq \rho, \|z\|_{\infty} \leq \Gamma\}.\end{aligned}$$

We modularize these diverse uncertainty sets in RHODES for users to freely determine their uncertainty conservativeness and the relationships of their tC parameters, especially as more data and parameters become available. In this paper, we use $\mathcal{U}_{\text{interval}}$ and $\mathcal{U}_{\infty}$ with tuned robustness parameters (see Fig. 12), to explore the trade-off between robustness and performance optimality in uncertainty-aware carbon-efficient design. Another uncertainty set $\mathcal{U}_{\text{budget}}$ (§III-A) avoids being overly conservative by leveraging Central Limit Theorem conclusions (we refer interested readers to [9]), where it is statistically unlikely for all uncertain parameters to reach their extreme values at the same time. We leverage $\mathcal{U}_{\text{budget}}$ in our WLP case study (§V-B) to prevent optimizing for worst-case values simultaneously across EPW, GPW, and MPW across the different SoC components. **$\mathcal{D}_V$ Uncertainty Sets.** As more data becomes available, it is important to leverage available data to construct more accurate and less conservative uncertainty sets. We particularly focus on the probabilistic robust (PRO) uncertainty set $\mathcal{U}_{\text{PRO}}$ [6], due to its ability to model heavy-tailed data samples and its effectiveness in reducing the average constraint violations compared to fundamental $\mathcal{U}$.

$$\mathcal{U}_{\text{PRO}} = \{z \in \mathbb{R}^n : \frac{-1}{n} \sum_{i=1}^n \log(\mathbb{P}(\tilde{z}_i \geq z_i)) \leq \Gamma\}, \quad (2)$$

where $\Gamma > 0$ is the probabilistic parameter threshold (set to 1 in our experiments, more details in [6]), and n is the number of independent uncertain parameters. This uncertainty set is derived from the reliability function (or log-survival), that is the probability of survival (complementary to the cumulative distribution function, i.e. CDF, which represents the probability of failure). Therefore, $\mathcal{U}_{\text{PRO}}$ fully leverages the available information about the uncertainty, where the reliability function captures the entire distribution, including tail behavior, and can distinguish between different distributions even when they share the same mean and variance. Fig. 7 illustrates the reliability functions (i.e. log-survival curves) of the yearly CAISO $\mathcal{D}_V$ data used to construct $\mathcal{U}_{\text{PRO}}$ for CI_{use} and CI_{fab} . Furthermore, $\mathcal{U}_{\text{PRO}}$ can be formulated from sampled empirical data (e.g. using Monte Carlo simulations). Using ordered samples $y_{i,1} \leq \dots \leq y_{i,N_{\text{data}}}$,

³Lp norm is defined as $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$, where $p > 0$.

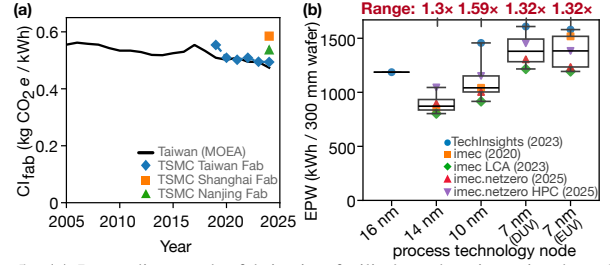


Fig. 5. (a) Depending on the fabrication facility's carbon intensity data, CI_{fab} may exhibit either a data availability ($\mathcal{D}_A$) or a data value ($\mathcal{D}_V$) uncertainty. Here, CI_{fab} is treated as a $\mathcal{D}_A$ parameter based on the limited data reported in TSMC's public sustainability reports [61]–[63]. If grid level data such as CAISO (Fig. 7) are applicable, CI_{fab} should be modeled as a $\mathcal{D}_V$ parameter to capture temporal variations in carbon intensity. (b) EPW data [13], [27], [35], [39] illustrate a $\mathcal{D}_A$ uncertainty (up to 5 data points per technology node).

we can estimate $\mathbb{P}(\tilde{z}_i \geq y_{i,j}) \approx \frac{N_{\text{data}} + 1 - j}{N_{\text{data}}}$ [6]. Hence, PRO can generalize to enable uncertainty sets constructed from empirical data to accurately capture real-world variability.

Given the uncertain parameter $z_i = y_{i,j}$, and using linear strong duality, we implement the robust counterpart [6], which adds one constraint per data sample N_i and is equivalent to

$$\begin{cases} n\Gamma\alpha + \sum_{i=1}^n \beta_i \leq b, \\ -\log\left(\frac{N_i + 1 - j}{N_i}\right)\alpha + \beta_i \geq x_i y_{i,j}, \quad \forall i \in [n], \forall j \in [N_i], \\ \alpha \geq 0 \end{cases} \quad (3)$$

Eq. 3 is the robust counterpart of $\mathcal{U}_{\text{PRO}}$ in eq. 2 when solved with sample distributions, where α and β are dual variables (extensive details in [6]). This enables us to sample any dataset to model and optimize over PRO uncertainty.

B. Parameters Contributing to Uncertainty in tC

The total carbon footprint (tC) of an IC is quantified as follows [23], [32]:

$$\begin{aligned}\text{tC} &= \text{C}_{\text{embodied}} + \text{C}_{\text{operational}} \\ &= (\text{CI}_{\text{fab}} \cdot \text{EPW} + \text{MPW} + \text{GPW}) \cdot \frac{A}{Y} + \int_0^{t_{\text{operational}}} \text{CI}_{\text{use}}(t) P(t) dt \quad (4)\end{aligned}$$

where $\text{C}_{\text{embodied}}$ is the embodied carbon footprint due to fabrication, and $\text{C}_{\text{operational}}$ is the operational carbon footprint due to hardware use. $\text{C}_{\text{embodied}}$ is quantified by summing the contributions from the total CO_2e impact of the fabrication electrical energy consumption EPW, the fabrication materials MPW, and the direct gases emitted during the fabrication process GPW. A and Y are the chip area and yield. CI_{fab} is the carbon intensity of the energy source used in fabrication. $\text{C}_{\text{operational}}$ is computed by quantifying the total CO_2e impact of system use over its operational lifetime $t_{\text{operational}}$, where CI_{use} is the carbon intensity of the electricity consumed to power P the system. Extensive details for quantifying tC are in [23], [31], [32], and we describe the relevant information that relates to uncertainty in the remainder of this sub-section.

Building on the classification in Fig. 3, we construct the corresponding uncertainty sets in RHODES RO models. Each parameter is classified into $\mathcal{D}_V$ or $\mathcal{D}_A$ uncertainty. Table II shows the quantitative parameters for $\mathcal{D}_A$ uncertainty set formation. Exact equations for deriving values used for $\mathcal{D}_V$ and $\mathcal{D}_A$ uncertainty sets are in our github repository.

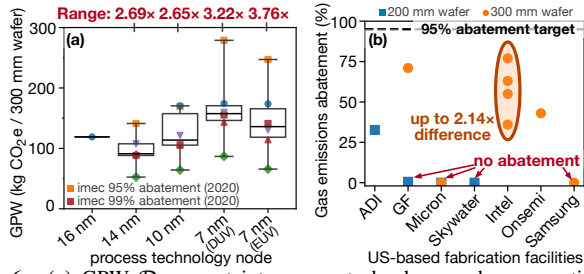


Fig. 6. (a) GPW $\mathcal{D}_A$ uncertainty across technology nodes accounting for 95–99% abatement [13], [27], [35], [39] (legend follows Fig 5b). (b) Reported gas abatement percentages in US-based fabrication facilities [25] often do not meet the 95% abatement target, including many with 0% abatement.

CI_{fab}. While grid level carbon intensity data may be available through public platforms, such as CAISO [53] and Electricity Maps [1], fabrication facilities often source power from multiple sources, including on-site renewable generation and uninterruptible power supply systems [68]. For example, Fig. 5a shows the carbon intensity of Taiwan’s grid (between 0.474–0.562 kg CO₂e per kWh) versus TSMC fabrication facilities (between 0.494–0.5849 kg CO₂e per kWh) [61]–[63], [68]. The duration of IC fabrication is typically on the order of several months [58], which also results in temporal fluctuations in carbon intensity that need to be accounted for. Therefore, we classify CI_{fab} as a data value $\mathcal{D}_V$ uncertainty parameter, to capture temporal and regional variations of the energy sources ($\mathcal{D}_V$ example shown in Fig. 7). In cases of data scarcity, CI_{fab} can alternatively be modeled as a $\mathcal{D}_A$ uncertainty parameter (such as Fig. 5a). Both uncertainty formulations are supported in RHODES (further details in Table II).

EPW. The electrical energy consumption of IC fabrication tools is dependent on wafer throughput, process technology node, and fabrication facility-specific process flows, tools and utilization. Fig. 5b highlights the wide range of EPW data uncertainty per process technology node for a 300 mm Si wafer (up to $1.59\times$ variation per technology node) [13], [27], [35], [39]. Reported datasets also differ in scope and system boundaries. For instance, the fabrication tools model in [27] imec (2020) [27] assumes 100% tool utilization, whereas imec LCA (2023) [13] assumes 70% utilization with 50% idle energy consumption (◆). Such variations are largely beyond the designer’s control, and large-scale production often spans multiple fabrication facilities with distinct infrastructures and energy profiles. These variations underscore the importance of incorporating EPW as a $\mathcal{D}_A$ uncertainty to model variations across fabrication environments.

GPW. Direct gas emissions and abatement practices vary widely across fabrication facilities, where abatement refers to on-site treatment systems that reduce or neutralize greenhouse gases produced during IC fabrication. Fig. 6a shows GPW data with up to $3.76\times$ variation per technology node at $\geq 95\%$ abatement, exhibiting $\mathcal{D}_A$ uncertainty. Fig. 6b illustrates the gap between reported abatement values in US-based fabrication facilities from [25] (between 0%–77%) versus the 95–99% abatement levels commonly assumed in carbon quantification models [13], [27]. Even within the same vendor, abatement percentages can vary significantly across facilities. For example, in 2023 Intel US-based 300 mm wafer fabrication facilities reported abatement percentages ranging between 36%–77% ($2.14\times$ difference). As a case

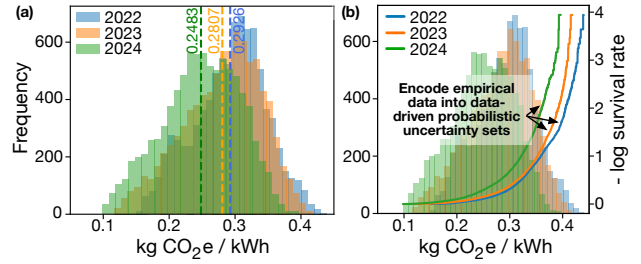


Fig. 7. (a) CI_{use} $\mathcal{D}_V$ uncertainty illustrated through variations in the carbon intensity of the California grid (CAISO [53]) across 12 months between years 2022–2024, due to fluctuations in renewable versus non-renewable energy generation. The dataset provides hourly measurements equivalent to 720 points per month. (b) We encode the CAISO data into a probabilistic robust uncertainty set $\mathcal{U}_{\text{PRO}}$, to capture temporal and energy source fluctuations, modeling CI_{use} as a $\mathcal{D}_V$ uncertainty independent of the data distribution.

study, we evaluate how uncertainty in GPW, due to abatement percentages, impacts robust CO₂-aware SoC design in §V-C.

MPW. MPW is one of most difficult and least quantified parameters in carbon quantification efforts. Many prior works either omit or assume a constant value across all process technology nodes due to outdated or scarce data [32]. Recent works have begun to quantify MPW emissions more thoroughly through detailed analysis of fabrication process flows [31]. Accordingly, we model MPW using a conservative $\pm 5\%$ $\mathcal{D}_A$ uncertainty for nominal MPW (500 g CO₂e per cm²) across all technology nodes. As improved data become available, RHODES uncertainty sets can be readily updated, reflecting the generality of our methodology rather than dependence on any specific dataset.

Yield. Fabrication yield is vendor and process dependent, varying with defect density (D_0) and node maturity, making it a $\mathcal{D}_A$ uncertainty. Since the SoCs in our design space are fabricated at mature nodes, 7 nm (EUV) and 12 nm (in mass-production since 2018 and 2013, respectively [67]), we model yield deterministically using a Murphy yield model [48] with 0.1–0.15 defects per cm² (D_0). Yield is therefore not explicitly treated as an uncertainty in RHODES, although its area dependence is inherently captured in the optimization. In §V-C, we evaluate the impact of redundant cores/SMs on the carbon efficiency of robust designs.

CI_{use}. Similar to CI_{fab}, CI_{use} depends on the energy mix and fluctuations of the grid during system operation. The use of renewable versus non-renewables significantly impacts CI_{use} and the overall operational carbon of the system (e.g. the difference between a coal versus a solar-powered grid is $19.5\times$ [32]). We leverage the CAISO dataset with hourly granularity [53], which enables a $\mathcal{D}_V$ uncertainty parameter. Fig. 7 shows the empirical CI_{use} data we encode into a $\mathcal{D}_V$ probabilistic robust uncertainty set ($\mathcal{U}_{\text{PRO}}$ [6]), that is independent of the underlying data distribution.

t_{operational}. Operational time is the total execution time of the system, including background processes and idle time (i.e. when the system is consuming energy). Since t_{operational} is dependent on hardware use and user behavior, it is difficult to determine apriori at design time. Operational time also influences whether a system is embodied or operational carbon dominant, determining how embodied carbon amortizes over the system’s lifetime [21], [23], [41]. While major hardware vendors may have internal profiling data, we account for the common case where hardware use data are unavailable and model operational time as a $\mathcal{D}_A$ uncertainty.

TABLE II

UNCERTAINTY SETS IN RHODES. FPW AND GPW ARE DISTINCT FOR CPU CORES (c), CPU I/O DIE (cIO), AND GPU SMS (g), DEPENDING ON THE TECHNOLOGY NODE. C_{OP} DEPENDS ON THE SETUP/TEARDOWN (s/td) ON CPU CORES, AND COMPUTE PHASE SELECTION (k) ON CPU OR GPU. n_c/g IS NUMBER OF CPU CORES/GPU SMS. WE SHOW EXAMPLE PARAMETER VALUES USED TO ENCODE UNCERTAINTY SET PARAMETERS IN RHODES. $CI_{SAMPLES}$ ARE VECTORS OF SAMPLED CI_{USE} (FIG. 7).

Parameter	Type	Uncertainty Set Function ($\mathcal{U}$)				
FPW _{c/cIO/g}	$\mathcal{D}_A$	$\mathcal{U}_{interval}: uFPW(n_c/g, LB_c/cIO/g, UB_c/cIO/g)$				
GPW _{c/cIO/g}	$\mathcal{D}_A$	$\mathcal{U}_{interval}: uGPW(n_c/g, LB_c/cIO/g, UB_c/cIO/g)$				
MPW _{c/cIO/g}	$\mathcal{D}_A$	$\mathcal{U}_{interval}: uMPW(LB_c/cIO/g, UB_c/cIO/g)$				
C _{op, c, s/td}	$\mathcal{D}_V$	$\mathcal{U}_{PRO}: uCop(n_c, c, \Gamma, CI_samples_c)$				
C _{op, c, k}	$\mathcal{D}_V$	$\mathcal{U}_{PRO}: uCop(n_c, (c-w), \Gamma, CI_samples_c)$				
C _{op, g, k}	$\mathcal{D}_V$	$\mathcal{U}_{PRO}: uCop(n_g, r, \Gamma, CI_samples_g)$				
Example Parameter Values (kg CO ₂ e / 300 mm wafer)						
	LB_c	UB_c	LB_cIO	UB_cIO	LB_g	UB_g
FPW	565.5	888.0	380.6	587.3	565.5	904.3
GPW	65.7	247.1	52.5	141.2	65.7	247.1
MPW	318.1	388.8	318.1	388.8	318.1	388.8

Modeling Choices. To maintain the linearity of our optimization, we ensure that the tC formulation remains linear in the uncertain parameters (eq. 4). For instance, directly accounting for uncertainties in CI_{fab} and EPW, or CI_{use} and $t_{operational}$ would result in nonlinearity in the uncertain parameters. To preserve linearity and improve tractability at large design spaces, we recognize without loss of generality that unknown multiplied by unknown is the same as unknown multiplied by a constant. Therefore, we treat the products of the uncertain parameters as new linearized variables; CI_{fab} and EPW are combined into FPW and CI_{use} and $t_{operational}$ are combined into the new linear parameter C_{op} . This transformation maintains the model's linearity without explicitly introducing nonlinear uncertainty terms. We leave explicit definition of the nonlinear uncertainties, which can be done by leveraging state-of-the-art RO techniques such as the Reformulation Perspectification Technique (RPT) [8], for future work. Table II summarizes example uncertainty sets implemented in RHODES for each uncertain parameter. For brevity, the robust counterpart formulations, such as eq. 3, are omitted due to space limitations. The mathematical derivations and code implementation are available in our github repository.

C. Modeling Heterogeneous Architectures

Each heterogeneous SoC comprises one or more CPU cores, and an optional GPU (i.e. zero or one) with a configurable number of SMs. The candidate CPU core configurations form the set $C = \{1, \dots, 32\}$. $c \in \mathbb{Z}_2^{(|C|)}$ is the binary decision vector of length $|C|$ indexed by $i \in C$, where exactly one element equals 1 to indicate the selected CPU core configuration and the remaining elements are zero. The candidate GPU configurations form the set $G = \{(m, f) \mid m \in \{1, \dots, 128\}, f \in \{f_{210}, \dots, f_{765}\}\}$, where m indicates SM count and f indicates one of 11 candidate clock frequencies (in MHz). $g \in \mathbb{Z}_2^{(|G|)}$ is the binary decision vector of length $|G|$ indexed by $j \in G$, where at most one element equals 1 to indicate the optional GPU SM configuration (SM count and clock frequency) and the remaining elements are zero.

Workload Execution Time. To model workloads on heterogeneous SoCs, each workload execution is composed of

three phases: setup (s), compute (k), and teardown (td). The setup and teardown phases are sequential and therefore executed on CPU cores; the compute phase is parallelizable and can run on either CPU or GPU. Notice that each workload has one setup, one compute, and one teardown phase as shown in [55]. For a single workload, the compute phase is dependent on the setup phase executing first, therefore both phases for the same workload cannot execute in parallel. Similarly, the teardown phase cannot execute before the compute phase is executed.

To model GPU versus CPU selection for the compute phase, we introduce a binary variable $x \in \mathbb{Z}_2$, where 1 indicates GPU selection and 0 indicates CPU selection. We introduce auxiliary variable $w_i \in [0, 1]$ as a linearization of the product $w_i = c_i x$, to represent the active CPU and GPU configurations in the optimization. The term $(c_i - w_i)$ activates the CPU compute phase when $x=0$, and is zero when $x=1$. The GPU compute phase is activated by g_j directly, where the selection constraint $\sum_{j \in G} g_j = x$ enforces $g_j=0$ when $x=0$ and selects exactly one GPU configuration when $x=1$. These are expressed as follows:

$$(c_i - w_i) = \begin{cases} c_i, & x=0 \quad (\text{CPU compute phase}) \\ 0, & x=1 \quad (\text{GPU compute phase}) \end{cases} \quad \forall i \in C \quad (5)$$

$$\sum_{j \in G} g_j = \begin{cases} 0, & x=0 \quad (\text{CPU compute phase}) \\ 1, & x=1 \quad (\text{GPU compute phase}) \end{cases}$$

The total execution time of a workload on the selected SoC configuration is as follows:

$$(T_{c,s} + T_{c,td})^\top c + T_{c,k}^\top (c-w) + T_{g,k}^\top g \leq T_{\max}, \quad (6)$$

where $T_{c,s} \in \mathbb{R}_+^{(|C|)}$ and $T_{c,td} \in \mathbb{R}_+^{(|C|)}$ are vectors indexed by $i \in C$ denoting the setup and teardown times of the workload executed sequentially on a CPU core configuration c_i . $T_{c,k} \in \mathbb{R}_+^{(|C|)}$ is the vector of the CPU compute-phase execution time indexed by $i \in C$, and multiplied by $(c-w)$ to activate only when the compute phase runs on CPU ($x=0$). $T_{g,k} \in \mathbb{R}_+^{(|G|)}$ is the vector of the GPU compute-phase execution time indexed by $j \in G$, and multiplied by g to capture the GPU configuration only when the GPU is active ($x=1$). $T_{\max}$ is the user-defined execution time constraint.

Power Consumption. The total SoC power consumption accounts for active and idle power of the selected CPU and GPU configurations. $P_c \in \mathbb{R}_+^{(|C|)}$ and $P_g \in \mathbb{R}_+^{(|G|)}$ are vectors of active power consumption, while $P_{c,idle} \in \mathbb{R}_+^{(|C|)}$ and $P_{g,idle} \in \mathbb{R}_+^{(|G|)}$ are vectors of idle power consumption, indexed over C and G respectively. Note that power consumptions are estimated from thermal design power (TDP) as in [55]. The first constraint applies to cases where the CPU is active and the GPU is idle, which includes the setup and teardown phases always executed on CPU, and the compute phase when $x=0$. The second constraint applies to the compute phase when the GPU is active and the CPU is idle ($x=1$). $P_{\max}$ is the user-defined SoC total power constraint.

$$P_c^\top c + P_{g,idle}^\top g \leq P_{\max} \quad (7)$$

$$P_{c,idle}^\top c + P_g^\top g \leq P_{\max}$$

Area. The total SoC area includes the CPU cores area $A_c \in \mathbb{R}_+^{(|C|)}$, I/O die area $A_{cIO} \in \mathbb{R}_+^{(|C|)}$, which scales linearly with the number of CPU cores, and the GPU SMs area

❶ Optimize Carbon Footprint	❷ Carbon-Constrained Optimization	❸ Optimize Carbon Efficiency
emin tC	min T	min tCDP
s.t. (6)–(9)	s.t. (7)–(10)	s.t. (7)–(9)

Fig. 8. RHODES optimization targets subject to (s.t.) design constraints.

$A_g \in \mathbb{R}_+^{(|G|)}$ depending on the selected configuration. $A_{\max}$ is the user-defined SoC total area constraint.

$$(A_c + A_{cIO})^\top c + A_g^\top g \leq A_{\max} \quad (8)$$

Cost. The total monetary cost accounts for both fabrication costs $\mathcal{K}_{\text{fabrication}}$ and operational costs $\mathcal{K}_{\text{operational}}$. $\mathcal{K}_{\text{fabrication}}$ is quantified by multiplying die area with the user-configurable parameter fabrication cost per area $\mathcal{K}^f$, dependent on technology node (in \$ per mm²). $\mathcal{K}_{\text{operational}}$ is quantified by multiplying the total energy consumption of the SoC E_{total} with the user-configurable operational energy cost per kWh $\mathcal{K}^o$ (in \$ per kWh). Refer to §IV for our cost estimation methodology. $\mathcal{K}_{\max}$ is the user-defined SoC total cost constraint.

$$\begin{aligned} \mathcal{K}_{\text{fabrication}} &= (A_c \mathcal{K}_c^f + A_{cIO} \mathcal{K}_{cIO}^f)^\top c + (A_g \mathcal{K}_g^f)^\top g \\ \mathcal{K}_{\text{operational}} &= E_{\text{total}} \mathcal{K}^o \\ \mathcal{K}_{\text{fabrication}} + \mathcal{K}_{\text{operational}} &\leq \mathcal{K}_{\max} \end{aligned} \quad (9)$$

D. Optimization Formulation

RHODES incorporates 3 optimization targets (Fig. 8), to model trade-offs in power-performance-area-total carbon (PPATC):

❶ **Optimizing Carbon Footprint.** PPA-constrained formulation that minimizes total carbon footprint (tC) of the system.

❷ **Carbon-Constrained Optimization.** In this formulation, the designer can optimize for a variety of objectives, including performance (e.g. execution time, T), energy efficiency, throughput, or cost (in \$), while constraining total carbon emissions:

$$(tC_c + tC_{cIO})^\top c + tC_g^\top g \leq tC_{\max} \quad (10)$$

❸ **Optimizing Carbon Efficiency.** Here, the optimization target is the carbon efficiency metric, total carbon-delay product (tCDP) [23], which quantifies the trade-off between embodied carbon, operational carbon, and energy efficiency.

Below we detail an example RO formulation for carbon-constrained optimization:

$$\begin{aligned} &\underset{c, g, x, w}{\text{minimize}} \quad (T_{c,s} + T_{c,td})^\top c + T_{c,k}^\top (c - w) + T_{g,k}^\top g \\ &\text{subject to} \quad [(FPW_c + GPW_c + MPW_c) \odot A_c]^\top c \\ &\quad + [(FPW_{cIO} + GPW_{cIO} + MPW_{cIO}) \odot A_{cIO}]^\top c \\ &\quad + [(FPW_g + GPW_g + MPW_g) \odot A_g]^\top g \\ &\quad + C_{\text{op}, c, s}^\top c + C_{\text{op}, c, td}^\top c + C_{\text{op}, c, k}^\top (c - w) + C_{\text{op}, g}^\top g \leq tC_{\max}, \\ &\quad \forall FPW_c \in \mathcal{U}_{FPW, c}^{\text{DA}}, GPW_c \in \mathcal{U}_{GPW, c}^{\text{DA}}, MPW_c \in \mathcal{U}_{MPW, c}^{\text{DA}}, \\ &\quad FPW_{cIO} \in \mathcal{U}_{FPW, cIO}^{\text{DA}}, GPW_{cIO} \in \mathcal{U}_{GPW, cIO}^{\text{DA}}, \\ &\quad MPW_{cIO} \in \mathcal{U}_{MPW, cIO}^{\text{DA}}, FPW_g \in \mathcal{U}_{FPW, g}^{\text{DA}}, GPW_g \in \mathcal{U}_{GPW, g}^{\text{DA}}, \\ &\quad MPW_g \in \mathcal{U}_{MPW, g}^{\text{DA}}, C_{\text{op}, c, s} \in \mathcal{U}_{\text{CI}, c, s}^{\text{DV}}, C_{\text{op}, c, td} \in \mathcal{U}_{\text{CI}, c, td}^{\text{DV}}, \\ &\quad C_{\text{op}, c, k} \in \mathcal{U}_{\text{CI}, c, k}^{\text{DV}}, C_{\text{op}, g} \in \mathcal{U}_{\text{CI}, g}^{\text{DV}}, \\ &\quad P_c^\top c + P_{g, \text{idle}}^\top g \leq P_{\max} \\ &\quad P_{c, \text{idle}}^\top c + P_g^\top g \leq P_{\max} \\ &\quad (A_c + A_{cIO})^\top c + A_g^\top g \leq A_{\max} \end{aligned}$$

Here we minimize total workload execution time subject to tC, power, and area constraints. The tC constraint ensures the selected SoC configuration does not exceed $tC_{\max}$ for any realization of the uncertain embodied (FPW, GPW, MPW) and operational (C_{op}) carbon parameters within their respective $\mathcal{D}_A$

and $\mathcal{D}_V$ uncertainty sets, annotated as $\mathcal{U}^{\text{DA}}$ and $\mathcal{U}^{\text{DV}}$, respectively (see uncertainty set functions and parameters in Table II).

The embodied carbon of the CPU core die, CPU IO die, and GPU die are computed as the element-wise product ($\odot$) of the uncertain embodied carbon parameters (FPW, GPW, MPW) with their respective areas (A_c, A_{cIO}, A_g), selecting the active configuration using decision vectors c and g . Each of the embodied carbon parameters are bounded by interval uncertainty sets $\mathcal{U}_{\text{interval}}$, with lower and upper bounds detailed in Table II. The operational carbon terms $C_{\text{op}, c, s}$, $C_{\text{op}, c, td}$, $C_{\text{op}, c, k}$, and $C_{\text{op}, g}$ represent the uncertain operational carbon parameters of each workload phase on the selected configuration. C_{op} parameters are implemented using PRO uncertainty sets $\mathcal{U}_{\text{PRO}}$ due to $\mathcal{D}_V$ uncertainty. Note that RHODES C_{op} implementation accounts for uncertain CI_{use} , uncertain $t_{\text{operational}}$, and the power and execution time of the corresponding phase to compute operational carbon under all realizations of the corresponding uncertainty set. Each of the operational carbon parameters are multiplied by c , $(c - w)$, or g depending on whether each phase executes on CPU or GPU.

RHODES can be flexibly extended to incorporate other CO₂-aware design metrics, due to our modular uncertainty set implementation. Each RO problem is reformulated and transformed into a standard Mixed Integer Linear Programming (MILP) problem, solvable by a wide array of linear solvers (§IV).

IV. EVALUATION METHODOLOGY

Workloads. We leverage profiled Rodinia benchmark [17] data provided publicly by HILP [55]. HILP breaks down each workload to three dependent phases of setup, compute, and teardown, and reports phase level execution time data. HILP focuses on the performance modeling of workload level parallelism in early-stage design space exploration. Therefore, our parameters, decision variables, modeling and design objectives in §III are fundamentally distinct.

SoC Configurations. Our SoC configurations leverage HILP's profiled workloads, and incorporates a design space of ranging from 1 to 32 CPU cores, and 0 to 128 GPU SMs with a range of GPU clock frequencies from 210 to 765 MHz. For our configurations, c_n indicates a CPU with n number of cores, and g_m^f indicates a GPU with m number of SMs at f MHz clock frequency. The default clock frequency for designs reported in this paper is 765 MHz unless indicated otherwise.

The area of the CPU cores is quantified based on the AMD EPYC 7453 32-core processor [65], where each CPU core is estimated as 10.125 mm² implemented at a 7 nm (EUV) technology node, and a corresponding 6.5 mm² I/O die per core implemented at a 12 nm technology node. To estimate the GPU area per SM based on an NVIDIA A100 GPU [66], we divide the total GPU area of 826 mm² implemented at a 7 nm (EUV) technology node by the number of active SMs (108 SMs), resulting in 7.65 mm² per SM.

For full system analysis (§V-B), we model a high bandwidth memory (HBM3) comprising an HBM stack of 8 DRAM die totaling 16 GB, with an area footprint of 107.09 mm² and average access energy of 7 pJ/bit [19], [55], [73]. Additionally, we model domain specific accelerators (DSAs) at 4× the GPU

TABLE III

WE EVALUATE THE NOMINAL DESIGN (NO UNCERTAINTY) VERSUS RHODES ROBUST DESIGN, USING MONTE CARLO ASSUMING FULL KNOWLEDGE OF PARAMETER DISTRIBUTIONS. BOTH DESIGNS ARE EVALUATED USING THE 95TH-PERCENTILE (P95) TOTAL CARBON FROM THE MONTE CARLO.

Case	Data Uncertainty Handling	Method
Nominal	No uncertainty considered	Nominal optimization
RHODES	Uncertainty sets	Robust optimization
p95	Full knowledge of parameter distributions	Monte Carlo evaluation

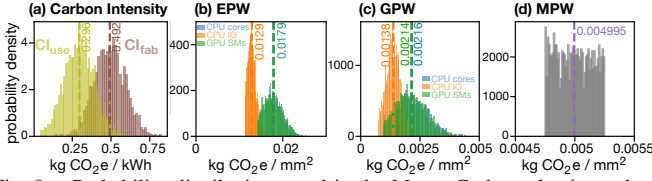


Fig. 9. Probability distributions used in the Monte Carlo evaluation, where $\mathcal{N}(\mu, \sigma)$ denotes a normal distribution with mean μ and standard deviation σ , and $\mathcal{U}(a, b)$ denotes a uniform distribution with lower bound a and upper bound b . (a) CI_{fab} : $\mathcal{N}(0.492, 0.098)$, CI_{use} : $\mathcal{N}(0.296, 0.098)$. (b) EPW: $\mathcal{N}(0.0179, 0.00215)$ for CPU cores, $\mathcal{N}(0.0179, 0.00206)$ for GPU SMs, $\mathcal{N}(0.0129, 0.000898)$ for CPU IO die. (c) GPW (>90% abatement): $\mathcal{N}(0.00216, 0.000653)$ for CPU cores, $\mathcal{N}(0.00214, 0.000635)$ for GPU SMs, $\mathcal{N}(0.00138, 0.000298)$ for CPU IO die. (d) MPW: $\mathcal{U}(0.004745, 0.005245)$ for CPU cores, CPU IO die, and GPU SMs. Units are $\text{kg CO}_2\text{e/kWh}$ for CI and $\text{kg CO}_2\text{e/mm}^2$ for EPW, GPW, MPW.

efficiency in line with prior works [55], for our workload level parallelism (WLP) analysis (§V-B). d_m^n indicates m number of DSAs, each with n number of processing elements (PEs).

Furthermore, we model monetary cost (as described in §III-C), including fabrication and operational costs, in line with prior works [50], [51], [56]. $\mathcal{K}^f$ is a user-defined production cost parameter accounting for fabrication, design, assembly and testing. For example, we set $\mathcal{K}^f$ to \$0.10 per mm^2 for 12 nm technology node ICs and \$0.24 per mm^2 for 7 nm technology node ICs as described in [40]. $\mathcal{K}^o$ is a user-defined energy cost parameter, which amounts to \$0.17 per kWh (residential sector, e.g., consumer devices) and \$0.13 per kWh (commercial sector, e.g., datacenters) based on the average US energy cost in 2025 [70]. Note that, while §III-C describes how to formulate cost as part of our RHODES framework, the primary optimization objectives we target in this paper focus on uncertainty in total carbon footprint (rather than cost), since RO techniques for cost optimization have been extensively studied in operations research [2], [7], [9]–[11].

Configuring the Solver. We implement RHODES optimization models in Julia [12]. Our robust model implementations are solver-agnostic, therefore they can be utilized with a wide variety of solvers. In this work, we use the Gurobi solver [34] for our implementation and results (example scripts provided in our github).

Solver Computational Runtime. The computational runtime of our RHODES framework for a single RO run, incorporating all the uncertainty parameters for all configurations, is under 5 minutes (for 45,088 unique configurations). Our experiments are conducted with a 0 optimality gap on an ARM-based 8-core CPU using up to 8 threads. Optimizing WLP is more compute intensive (§V-B), therefore we run our experiments on Intel(R) Xeon(R) CPU E5-2680 at 2.40 GHz using up to 28 threads.

Evaluation & Validation. To quantify the tC and carbon efficiency benefits of the robust designs compared to the nominal design (i.e. not accounting for tC uncertainty), we perform Monte

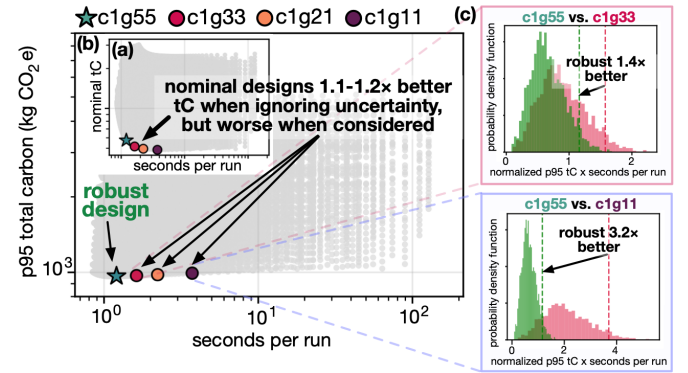


Fig. 10. (a) Without accounting for uncertainty, the nominal designs (colored circles) appear to have better 1.1–1.2 $\times$ carbon footprint. (b) Once uncertainties in total carbon including operational time are considered, the nominal designs perform worse than the robust design. (c) RHODES robust design (green star) is 1.4–3.2 $\times$ more carbon-efficient than the nominal designs under uncertainty.

Carlo (MC) evaluations using synthetic distributions generated for each parameter to simulate a range of manufacturing variations and hardware use conditions (see Table III and Fig. 9). The MC evaluations draw 2,000 samples per uncertain parameter. CI_{fab} and CI_{use} are sampled from bounded normal distributions with a minimum value of 0.041 $\text{kg CO}_2\text{e/kWh}$, corresponding to the carbon intensity of solar energy [32]. EPW and GPW parameters are sampled from bounded normal distributions, with minimum thresholds enforced by rejection resampling to ensure non-negative values. MPW is sampled from a uniform distribution over $\pm 5\%$ of the mean. EPW, GPW, MPW are normalized and divided by the wafer area $A_w = 70,685.83 \text{ mm}^2$ prior to sampling. The mean and standard deviation values reported in Fig. 9 caption are the empirical statistics derived from the generated samples, where the standard deviation is derived from the spread value divided by 2 (i.e. $\sigma = \text{spread}/2$).

Part of the challenge in evaluating CO_2 -aware designs in sustainable computing in general is the lack of data, which is also why optimizing for carbon efficiency is inherently difficult [23]. We validate our RHODES designs on multiple distributional assumptions, following common practices in RO [6], and our robust designs consistently outperform nominal designs as shown in Table IV. We note, however, as with any modeling-based evaluation, *garbage in, garbage out* still applies—if the simulated distributions are irrelevant to the uncertainties modeled, both nominal and robust designs lose practical relevance. We validate RHODES designs across a wide range of tC uncertain data distributions, including normal, uniform, exponential, bernoulli, and cauchy distributions (Table IV). For each uncertain parameter, we enforce a minimum threshold during Monte Carlo sampling to prevent physically unrealistic values (e.g. negative energy or operational time).

V. RESULTS

Here, we present our results demonstrating the effectiveness of RHODES to improve performance and tC, despite realistic $\mathcal{D}_A$ and $\mathcal{D}_V$ uncertainties, for the three optimization targets (Fig. 8).

A. Minimizing tC Given a Performance Constraint

Fig. 10 shows the total carbon (y-axis) versus execution time in seconds (x-axis) for both nominal and RHODES robust designs

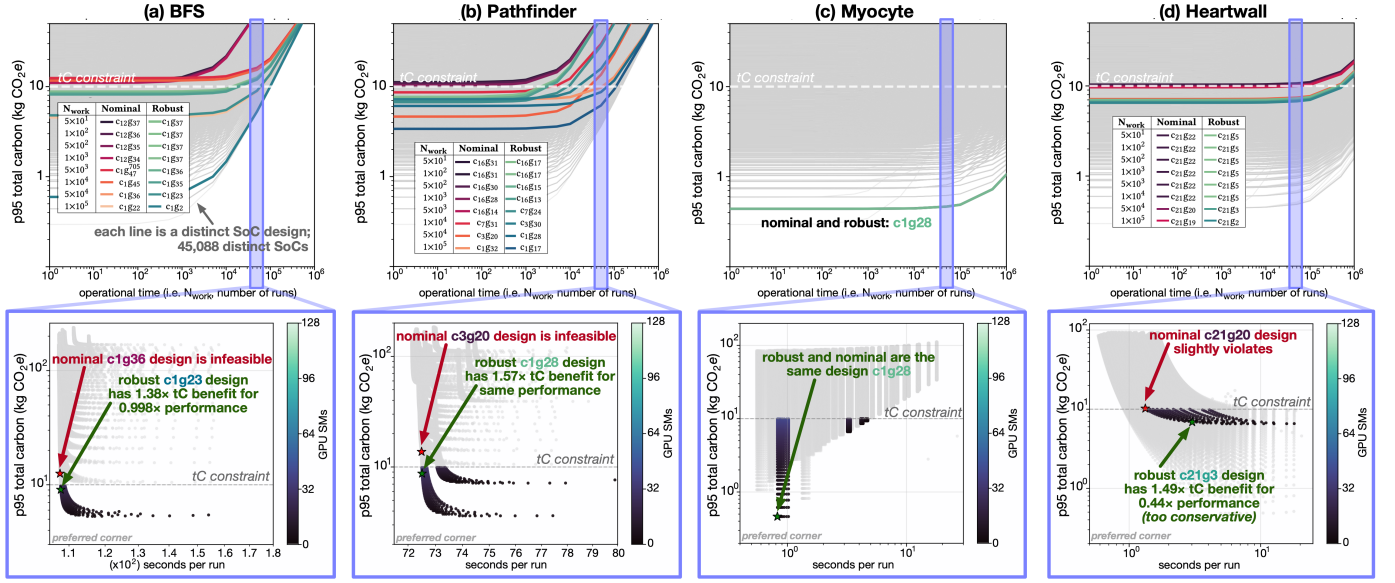


Fig. 11. (Top) each line represents a distinct SoC configuration, showing p95 total carbon of each design (y-axis) versus operational time (quantified by number of runs) for a variety of Rodinia workloads. RHODES robust designs consistently achieve better carbon footprint under uncertainty compared to nominal designs.

optimized for Rodinia “lavaMD” workload. The nominal designs (colored circles) in Fig. 10a appear to have 1.1–1.2× lower total carbon than the robust design c_{1g55} when tC uncertainty is not accounted for. However, this presents an incomplete picture of the total carbon of the design space. Using MC simulations with synthetic distributions for each uncertain parameter (Fig. 9), we compute the p95 total carbon to reflect real-world variations. Under these conditions, the nominal designs exhibit up to 1.7× worse tC for the c_{1g33} SoC, primarily due to unaccounted uncertainty in operational time. Designs optimized only under nominal carbon footprint conditions can therefore mislead designers and researchers into overestimating tC and carbon efficiency benefits.

In contrast, RHODES yields designs that are more realistic and resilient to uncertainties, through explicit incorporation of uncertainty at the optimization phase. Fig. 10c demonstrates that RHODES robust design c_{1g55} achieve 1.4–3.2× better carbon efficiency than the nominal designs c_{1g33} , c_{1g21} , c_{1g11} , due to its better energy efficiency which results in lower operational carbon over longer operational times that the nominal optimization does not account for. While it is unlikely for a designer to compare both c_{1g11} and c_{1g55} for the same workload given the large difference in compute resources, the comparison illustrates the consequence of ignoring uncertainty. *Note that the robust design c_{1g55} is not the design with the lowest tC in the full design space (grey designs). This is because both robust and nominal designs are evaluated on Monte Carlo simulations that neither of the optimizations were exposed to, to ensure fair and unbiased validation.*

B. Minimizing Execution Time Given a tC Constraint

Optimizing SoCs with CPU cores and GPU SMs. Here, we first consider SoC case studies excluding the carbon footprint of main memory, as this information may not be available at early design stages. Fig. 11 (a)–(d) present the p95 total carbon footprint (y-axis) of the hardware design space exploration for a range of Rodinia workloads [17]; Breadth-First Search

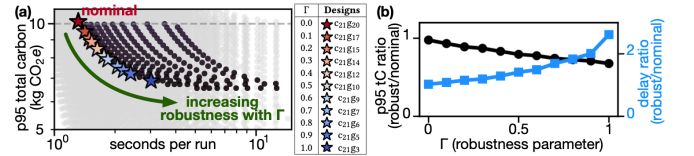


Fig. 12. Designers can tune the level of robustness in RHODES by sweeping the robustness parameter, Γ . (a) Higher Γ increases robustness. (b) Highlights the design trade-offs between robustness (i.e. lower tC) and objective target performance (i.e. execution time) for Heartwall Fig. 11e.

(BFS), Pathfinder, Myocyte, and Heartwall. The x-axis represents operational time quantified by number of workload runs. Each curve corresponds to a distinct SoC design, where red-hued lines indicate nominal designs and blue-hued lines indicate RHODES robust designs. The bottom panels of Fig. 11 capture a specific snapshot of the design space at 5×10^4 runs, which serves two purposes. It allows us to evaluate the performance of our robust designs excluding the impact of operational time highlighted in Fig. 10, and to explore designs at the cusp of embodied to operational carbon dominance, where embodied carbon dominates at shorter operational times, and operational carbon dominates at longer operational times. We enforce a $tC \leq 10$ kg CO₂e constraint (dashed line), alongside power and area constraints, where grey points/lines indicate designs that do not meet the design constraints.

In addition to tC and performance metrics, we quantify violation (%) as the fraction of MC samples in which a design exceeds the tC constraint. In Fig. 11 (a)–(b), nominal designs consistently exceed 10 kg CO₂e, violating the tC constraint in 49.85%–54.4% of the samples. RHODES designs, on the other hand, have 1.38–1.57× lower tC than their nominal counterparts, with a maximum violation rate of 1.1% tC and negligible performance degradation. This demonstrates two key points. First, *even when operational time is known exactly, uncertainties in other carbon footprint parameters (§III-B) impact designs measurably.* Second, *designers do not necessarily have to sacrifice performance to meet carbon constraints*

TABLE IV

EVALUATION FOR A RANGE OF MONTE CARLO DISTRIBUTIONS SAMPLING ACROSS WORKLOADS. TC CONSTRAINT: $t_C \leq 10$ kg CO₂e. **N** INDICATES NOMINAL DESIGN AND **R** INDICATES RHODES DESIGN. RHODES ROBUST DESIGNS HAVE 1–1.5 $\times$ LOWER TC (I.E. BETTER) VERSUS NOMINAL DESIGNS.

Distribution	Function	bfs (N:c1g36, R:c1g23)			pathfinder (N:c3g20, R:c1g28)			myocyte (N:c1g28, R:c1g28)			nn (N:c1g48, R:c1g37)			heartwall (N:c21g20, R:c21g3)		
		N _{p95}	R _{p95}	Impr.	N _{p95}	R _{p95}	Impr.	N _{p95}	R _{p95}	Impr.	N _{p95}	R _{p95}	Impr.	N _{p95}	R _{p95}	Impr.
Normal	$\mathcal{N}(\mu, \sigma)$	13.39	9.56	1.40 $\times$	14.00	9.31	1.50 $\times$	6.95	6.95	1.00 $\times$	14.60	10.03	1.46 $\times$	11.09	7.50	1.48 $\times$
Exponential	$\mu - \sigma + \mathcal{E}(1/\sigma)$	13.98	9.87	1.42 $\times$	14.75	9.88	1.49 $\times$	7.36	7.36	1.00 $\times$	14.89	10.21	1.46 $\times$	11.37	7.62	1.49 $\times$
Uniform	$\mathcal{U}(\mu \pm \sqrt{3}\sigma)$	13.53	9.55	1.42 $\times$	13.92	9.48	1.47 $\times$	7.02	7.02	1.00 $\times$	14.77	10.13	1.46 $\times$	11.21	7.59	1.48 $\times$
Bernoulli	$\mu - \sigma + 2\sigma \mathcal{B}(1/2)$	14.03	9.73	1.44 $\times$	13.46	9.81	1.37 $\times$	7.14	7.14	1.00 $\times$	15.01	10.29	1.46 $\times$	11.23	7.63	1.47 $\times$
Cauchy	$\mathcal{C}(\mu, \sigma)$	13.10	9.30	1.41 $\times$	13.42	9.20	1.46 $\times$	6.75	6.75	1.00 $\times$	14.15	9.71	1.46 $\times$	10.82	7.46	1.45 $\times$

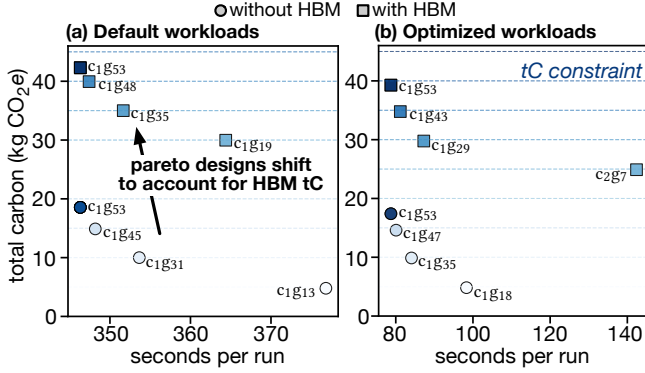


Fig. 13. We show RHODES robust designs when excluding versus including HBM carbon footprint across tC constraints for 9 Rodinia workloads.

under uncertainty. We show RHODES identifies designs that satisfy both objectives simultaneously. We also validate Fig. 11 RHODES designs under a wide range of uncertain parameter distributions simulated using MC in Table IV (see §IV).

Fig. 11c highlights the case where both nominal and RHODES RO independently select the same design, c_{1g28} . This outcome is expected when the feasible design space provides sufficient margin from the tC constraint. Even in such cases, optimizing with RHODES is valuable for designers to verify that the selected design is inherently robust. This shows that *when sufficient total carbon design margin exists, the convergence of nominal and RHODES optimizations, validates the designer's choice at no additional cost to performance or carbon efficiency.*

In contrast, Fig. 11d shows a robust design c_{21g3} , while it has 1.49 $\times$ lower total carbon than the nominal design c_{21g20} , it incurs significant performance degradation, indicating that the robust design is overly conservative. To address such scenarios, designers can tune the robustness parameter Γ in the uncertainty sets (§III-A), by sweeping Γ from zero (no uncertainty) to 1 (full uncertainty set), as demonstrated in Fig. 12a (zoomed in). Fig. 12b shows that as Γ increases, the robustness of the designs to total carbon variations improves (black line), while performance degrades (blue line). Therefore, in cases when optimizing under tC uncertainty incurs significant performance degradation, *designers can use robustness parameter Γ in RHODES to continuously trade-off the level of tC robustness versus performance degradation according to their design requirements.*

Optimizing SoCs considering main memory. We also explore the impact of accounting for the tC, both operational and embodied, of main memory (e.g. HBM) into RHODES robust designs. This is particularly important as design stages progress, given that memory has been shown to contribute significantly to the tC of server-grade systems [32]. Fig. 13 shows the shift in RHODES robust designs when excluding (circles) versus including (squares)

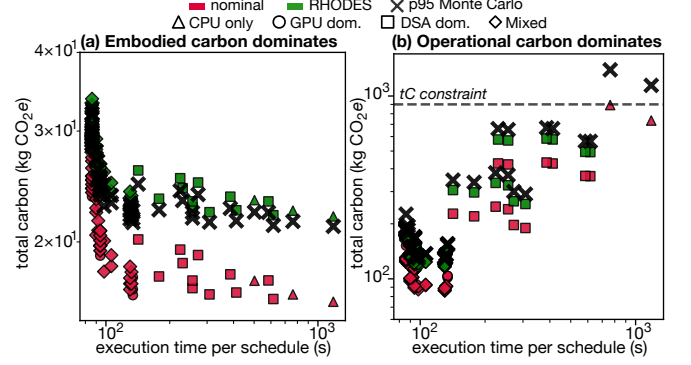


Fig. 14. We extend RHODES to optimize for overall execution time by modeling workload level parallelism (WLP) scheduling for heterogeneous SoCs spanning CPU cores, GPU SMs, DSAs and HBM.

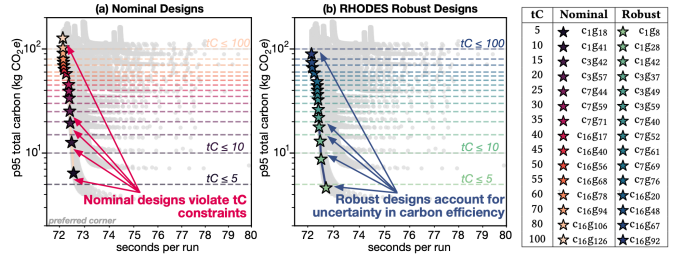


Fig. 15. (a) Nominal designs violate tC constraints when evaluated under uncertainty (suboptimal), (b) RHODES robust designs satisfy the same design constraints while accounting for uncertainty. For example, the pathfinder nominal design c_{16g126} exceeds the $tC \leq 100$ kg CO₂e constraint 58.3% of the time (for 5×10^4 runs), while the corresponding robust design c_{16g92} violates it only 0.4%, with only 0.1% lower performance (execution time in seconds). RHODES robust design yields 1.43 $\times$ carbon efficiency (tCDP) improvement.

HBM carbon footprint across a range of tC constraints (y-axis) and execution time (x-axis) for nine Rodinia workloads. Several Rodinia workloads are setup and teardown-heavy [55], which are CPU-bound. Therefore, we show results for both (a) default setup and teardown times, and (b) optimized setup and teardown times ($5 \times [55]$) for efficiency-sensitive deployments.

Fig. 13 shows that accounting for HBM tC results in two primary changes in RHODES robust designs. First, tC increases to account for the additional embodied carbon of HBM and the operational carbon of memory transfers, requiring designers to accommodate for higher tC constraints. For example, c_{1g53} with HBM incurs 2.28 $\times$ higher tC for the same operational time and performance, shifting the satisfied tC constraint from $tC \leq 20$ kg CO₂e to $tC \leq 45$ kg CO₂e (Fig. 13a). Second, for the same tC constraint (e.g. $tC \leq 30$ kg CO₂e), robust designs with HBM have less compute resources to remain within the tC constraint, e.g., c_{1g53} to c_{1g19} in (a), and c_{1g53} to c_{1g29} in (b). Note that designs in Fig. 13b can sustain higher compute resources (e.g. more GPU SMs) for the same tC constraint compared to (a), since optimized workloads reduce operational carbon.

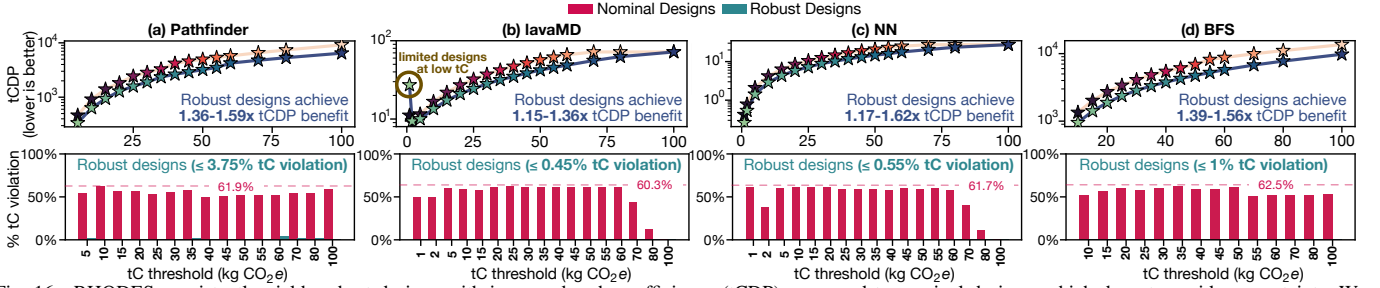


Fig. 16. RHODES consistently yields robust designs with improved carbon efficiency (tCDP) compared to nominal designs, which do not consider uncertainty. We calculate the percentage violation rate for the tC threshold based on the number of Monte Carlo samples for each design, where RHODES designs consistently outperform nominal designs. Each distinct tCDP value represents a distinct SoC design (shown in Fig. 15 legend for Pathfinder).

As design stages progress and memory configurations become known, incorporating memory carbon footprint into RHODES is important to avoid underestimating tC or over-provisioning compute resources that violate tC constraints in practice.

Modeling WLP considering DSAs and HBM. We further showcase the versatility of RHODES by extending it to model workload level parallelism (WLP) [55], i.e. independent application phases concurrently executing on the SoC, by leveraging HILP’s ILP Job-Shop Scheduling Problem (JSSP) architecture formulation [55]. Notice that formulating WLP as a JSSP results in an NP-complete problem for large design spaces, making it a time-intensive scheduling optimization problem (we refer interested readers to the details here [55]). Our modeling in §III-B and §III-A applies consistently across problem formulations.

We explore the design space of SoCs with 1, 2 or 4 CPUs, an optional GPU with 4, 16, or 32 SMs, and optional 2 or 4 DSAs with 4 or 16 PEs, scheduling 7 Rodinia workloads. In Fig. 14, each point represents an SoC configuration spanning CPU only, GPU dominant (no DSAs), DSA dominant (no GPU), and mixed (both GPU and DSAs) designs. We show the tC footprint modeled by the nominal optimization, RHODES optimization, and the p95 Monte Carlo simulations to reflect realistic uncertainty.

As in previous case studies, robust design trends observed in Fig. 11 generalize to these more complex heterogeneous SoCs, since the added design complexity does not fundamentally change how RHODES models and accounts for uncertainty. In Fig. 14a, when embodied carbon dominates (e.g. 1 year of hardware use), nominal optimization underestimates total carbon by $1.166\times$ – $1.318\times$ when evaluated against p95 tC, while RHODES tC modeling remains within $0.951\times$ – $0.970\times$, reducing the average impact of tC uncertainty by $1.17\times$. In Fig. 14b, when operational carbon dominates (e.g. 5–7 years of hardware use), nominal optimization underestimates total carbon by $1.453\times$ – $1.552\times$, while RHODES tC modeling is within $1.093\times$ – $1.147\times$ of p95 tC. Furthermore, at a $tC \leq 26 \text{ kg CO}_2\text{e}$ constraint (Fig. 14a), the robust design $c_1g_4d_4^{16}$ achieves $1.16\times$ lower tC than the nominal design $c_1g_{32}d_0^0$ with $0.91\times$ the performance. Fig. 14b also shows that CPU only designs fail to satisfy the $tC \leq 900 \text{ kg CO}_2\text{e}$ constraint under uncertainty when operational carbon dominates, due to their limited performance and high energy consumption. *These results demonstrate that RHODES scales to complex design spaces, spanning CPUs, GPUs, DSAs, main memory, and workload-level parallelism modeling, while consistently producing more robust designs than nominal optimization under uncertainty.*

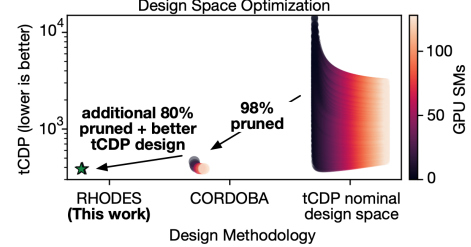


Fig. 17. CORDOBA [23] eliminates 98% of the design space as suboptimal by considering uncertainty only in operational time. RHODES expands carbon-efficient optimization by incorporating multiple uncertainties across tC parameters. RHODES further prunes 80–90% of the remaining designs and uncovers a more carbon-efficient optimum not present in CORDOBA’s search space.

C. Minimizing tCDP

Directly optimizing tCDP, by multiplying total carbon and execution time, in RHODES introduces non-linearity in the decision variables (modeling choices discussed in §III-A). Instead, RHODES traces the Pareto front of total carbon versus execution time by solving a series of linear subproblems: for each tC threshold in an increasing sequence, we minimize execution time subject to the tC threshold constraint. By iterating across tC thresholds, we find the set of Pareto-optimal points for which we can compute tCDP and find the tCDP optimal design without introducing non-linearity into the optimization.

Optimizing SoCs with CPU cores and GPU SMs. Fig. 15 shows that nominal designs, when evaluated against p95 tC, violate tC thresholds by 61.9% for “pathfinder” at a tC threshold of $10 \text{ kg CO}_2\text{e}$. Fig. 16 (a)–(d) quantify tCDP trends across a range of tC thresholds (1–100 $\text{kg CO}_2\text{e}$) and workloads. Across workloads, RHODES consistently yields more carbon-efficient (lower tCDP) and more robust (lower tC violation) designs than nominal counterparts. The exception is Fig. 16b “lavaMD” at a tC threshold of $1 \text{ kg CO}_2\text{e}$, where the severely constrained design space forces a conservative design c_1g_1 with low performance. In such cases, designers can explore tuning robustness parameter Γ (Fig. 12, §III-A) directly from RHODES optimization output, without requiring Monte Carlo simulation.

Evaluating RHODES versus CORDOBA. Beyond nominal optimization, we evaluate RHODES versus CORDOBA’s optimization methodology [23]. CORDOBA employs a discretized approach to optimize tCDP across operational time. Consequently, CORDOBA’s output designs depend on the discrete operational times selected by the designer. For example, to optimize SoC configurations for operational time between 3–6 years (10^6 – 8×10^7 runs) for “heartwall”, CORDOBA identifies five tCDP

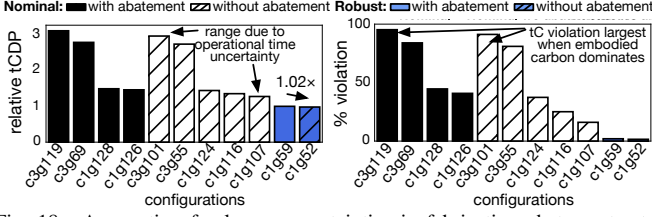


Fig. 18. Accounting for larger uncertainties in fabrication abatement rates (<95%) yields a different robust design with little carbon efficiency (tCDP) improvement (1.02 $\times$). The range of CORDOBA designs (ordered in increasing operational time), consistently perform worse versus RHODES designs.

optimal designs c_{1g2} , c_{1g4} , c_{1g10} , c_{1g14} , and c_{1g19} . Depending on the range of operational times and the size of the design space, a designer may obtain a non-trivial number of candidate designs. The designer then manually selects among these candidate designs, based on available data and practical factors, such as cost and tC-performance trade-offs. RHODES eliminates this manual process by encoding uncertainties across multiple parameters directly into the optimization model, automatically selecting c_{1g16} as the robust design, achieving better carbon efficiency than all CORDOBA candidates. Fig. 17 highlights that while CORDOBA eliminates 98% of the original design space as suboptimal, RHODES further prunes 80–90% of the remaining designs, and uncovers a more carbon-efficient design not captured in CORDOBA’s candidate design space.

RHODES designs also significantly reduce the simulation-to-real gap. For example, under uniform and normal distributions of operational time, the nominal-to-real gap (including CORDOBA candidate designs) is 1.23 $\times$ –3 $\times$ and 1.4–3.49 $\times$, respectively. RHODES robust designs reduce this gap to 1.12 $\times$ and 1.01 $\times$, respectively. These results highlight a critical takeaway: while CORDOBA effectively narrows the design space by sweeping tCDP across operational time, it relies on nominal tC parameters. Therefore, CORDOBA is useful when quantifying the relative carbon efficiency of designs, however, its absolute tC estimate can deviate substantially. In cases where a designer must ensure that tC does not exceed a specific threshold, manually selecting an inappropriate CORDOBA design (e.g. due to limited knowledge of uncertain parameters) can result in 95.6% tC threshold violation with a tCDP 3.17 $\times$ worse than the RHODES robust design (discussed next in Fig. 18). *RHODES automates robust carbon-efficient optimization, which is especially useful when optimizing manually over large design spaces with multiple uncertain parameters becomes practically infeasible.*

Quantifying the impact of abatement uncertainty. During chip fabrication, abatement systems are used to reduce harmful gas emissions, and abatement rates quantify their effectiveness in reducing GPW’s associated emissions. While many carbon quantification models assume high abatement rates (95–99%), foundry reported data are much lower and highly variable (Fig. 6b). To evaluate the impact of this uncertainty, we perform two RHODES optimizations: one with 95% abatement data in the uncertainty set and one with near 0% abatement. Fig. 18 shows that the resulting robust designs differ only slightly, where c_{1g52} (near 0% abatement) is only 1.02 $\times$ more carbon-efficient than c_{1g59} (95% abatement) when evaluated under no abatement

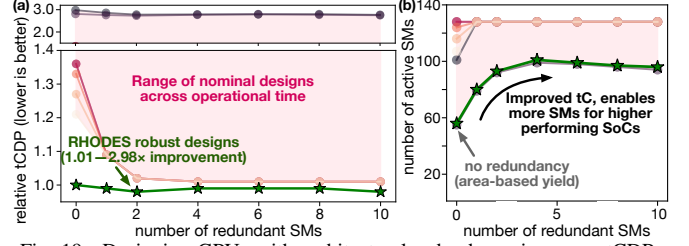


Fig. 19. Designing GPUs with architectural redundancy improves tCDP.

MC simulations. *This demonstrates that even when operational carbon dominates, uncertainties in embodied carbon parameters can still shift the robust design. RHODES enables architects to quantify how fabrication uncertainties impact SoC designs and optimizes robust architectures under fabrication variability.*

Quantifying the impact of architectural redundancy on carbon efficiency. In architectural redundancy [45], [64], additional SMs (cores) are manufactured on a GPU (CPU) die to tolerate more manufacturing defects than designs with no redundancy and area-based yield models [48]. This redundancy introduces a trade-off, where more redundant cores increase die area and accordingly embodied carbon, but simultaneously improve die yield. We quantify how optimizing with tC uncertainty can impact architectural redundancy benefits for robust hardware design. To model redundancy, we calculate the probability that at least the required number of SMs are functional as follows: $\sum_{k=n_{SMsNeed}}^{n_{SMsTotal}} \binom{n_{SMsTotal}}{k} (e^{-A_{SM}D_0})^k (1 - e^{-A_{SM}D_0})^{n_{SMsTotal}-k}$. Fig. 19a shows that RHODES robust design c_{1g56} achieves up to 2.98 $\times$ tCDP improvement with 2 redundant SMs compared to nominal designs (shaded region). Fig. 19b shows that as redundancy improves yield, tC decreases which enables higher performing SoCs with more active SMs. *This highlights that a small number of redundant SMs (2-4 SMs) can significantly improve carbon efficiency, by improving defect-tolerance and yield which amortizes embodied carbon. However, over-provisioning redundancy shows diminishing carbon efficiency returns as the embodied carbon overhead outweighs the yield benefits.*

VI. CONCLUSION

RHODES addresses the key challenge of $\mathcal{D}_A$ uncertainty in optimizing CO₂ efficiency of computing systems, where the severe lack of data prevents designers from being able to precisely quantify underlying probability distributions of key parameters that impact carbon footprint. Methods that rely on probability distributions for uncertainty-aware design are thus insufficient for optimizing CO₂ efficiency of computing systems. Instead, RO techniques in RHODES leverage *uncertainty sets* to jointly optimize carbon footprint and performance, in the presence of $\mathcal{D}_A$ and $\mathcal{D}_V$ uncertainties. RHODES can also be extended to account for additional objectives, e.g., related to cost and demand, where RO techniques have been employed extensively.

ACKNOWLEDGMENTS

We thank Carole-Jean Wu, Benjamin Boucher, Theodoros Koukouvinos, Periklis Petridis, Dick den Hertog, Dimitris Bertsimas & Flavio Calmon for their valuable discussions. We acknowledge valuable support from the Harvard Salata Institute, NSF Expeditions in Computing: 2326605 & NSF CAREER: 2443303.

REFERENCES

- [1] "Electricity Maps," 2025. [Online]. Available: <https://app.electricitymaps.com/map/>
- [2] E. Adida and G. Perakis, "A robust optimization approach to dynamic pricing and inventory control with no backorders," *Mathematical Programming*, vol. 107, no. 1, pp. 97–129, 2006. [Online]. Available: <https://doi.org/10.1007/s10107-005-0681-5>
- [3] A. Alameldeen and D. Wood, "Variability in architectural simulations of multi-threaded workloads," in *The Ninth International Symposium on High-Performance Computer Architecture*, 2003. *HPCA-9 2003. Proceedings.*, 2003, pp. 7–18.
- [4] Anonymous, "Uncertainty-aware Carbon Accounting for Large-scale AI models with Market-based Attribution," in *Submitted to The Fourteenth International Conference on Learning Representations*, 2025, under review. [Online]. Available: <https://openreview.net/forum?id=T4PMT8ZCZo>
- [5] Apple, "iPhone 17 Product Environmental Report," 2025. [Online]. Available: https://www.apple.com/la/environment/pdf/products/iphone/iPhone_17_PER_Sept2025.pdf
- [6] D. Bertsimas and B. Boucher, "Probabilistic robust optimization," 2023. [Online]. Available: https://www.dropbox.com/scl/fi/ju1h8nbwdzqfdugslneq/Probabilistic_Robust_Optimization.pdf?rlkey=2tzei01om7a217m6hrqg0mjb1&e=1&dl=0
- [7] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011. [Online]. Available: <https://doi.org/10.1137/080734510>
- [8] D. Bertsimas, D. de Moor, D. den Hertog, T. Koukouvinos, and J. Zhen, "An exact method for a class of robust nonlinear optimization problems," *25th International Symposium on Mathematical Programming (ISMP 2024)*, 2024. [Online]. Available: <https://optimization-online.org/2024/07/an-exact-method-for-a-class-of-robust-nonlinear-optimization-problems/>
- [9] D. Bertsimas and D. Den Hertog, *Robust and Adaptive Optimization*. Dynamic Ideas LLC, 2022.
- [10] D. Bertsimas and A. G. Koulouras, "Marginal pricing in adaptive robust unit commitment under load and capacity uncertainty," *IEEE Transactions on Power Systems*, vol. 40, no. 1, pp. 341–354, 2025.
- [11] D. Bertsimas and M. Sim, "The price of robustness," *Operations Research*, 2004. [Online]. Available: <https://doi.org/10.1287/opre.1030.0065>
- [12] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017. [Online]. Available: <https://doi.org/10.1137/141000671>
- [13] L. Boakes, M. Garcia Bardon, V. Schellekens, I.-Y. Liu, B. Vanhouche, G. Mirabelli, F. Sebaai, L. Van Winckel, E. Gallagher, C. Rolin, and L.-Ragnarsson, "Cradle-to-gate life cycle assessment of cmos logic technologies," in *2023 International Electron Devices Meeting (IEDM)*, 2023, pp. 1–4.
- [14] J. Bornholt, T. Mytkowicz, and K. S. McKinley, "Uncertainty: a first-order type for uncertain data," *SIGPLAN Not.*, vol. 49, no. 4, p. 51–66, Feb. 2014. [Online]. Available: <https://doi.org/10.1145/2644865.2541958>
- [15] S. B. Boyd, *Life-cycle assessment of semiconductors*. Springer Science & Business Media, 2011.
- [16] E. Brunvand, D. Kline, and A. K. Jones, "Dark silicon considered harmful: A case for truly green computing," in *2018 Ninth International Green and Sustainable Computing Conference (IGSC)*, 2018, pp. 1–8.
- [17] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, S.-H. Lee, and K. Skadron, "Rodinia: A benchmark suite for heterogeneous computing," in *2009 IEEE International Symposium on Workload Characterization (IISWC)*, 2009, pp. 44–54.
- [18] X. Chen, L. Han, A. Bhagavathula, and U. Gupta, "CarbonClarity: Understanding and Addressing Uncertainty in Embodied Carbon for Sustainable Computing," 2025. [Online]. Available: <https://arxiv.org/abs/2507.01145>
- [19] J. Choe, "HBM (High Bandwidth Memory) Technology Trend: 2024 and Beyond," TechInsights, Tech. Rep., 2024.
- [20] W. Cui and T. Sherwood, "Estimating and understanding architectural risk," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-50 '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 651–664. [Online]. Available: <https://doi.org/10.1145/3123939.3124541>
- [21] L. Eckhout, "FOCAL: A First-Order Carbon Model to Assess Processor Sustainability," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ser. ASPLOS '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 401–415. [Online]. Available: <https://doi.org/10.1145/3620665.3640415>
- [22] Electric Power Research Institute (EPRI), "Powering Intelligence: Analyzing Artificial Intelligence and Data Center Energy Consumption," 2024. [Online]. Available: <https://www.epri.com/research/products/000000003002028905>
- [23] M. Elgamal, D. Carmean, E. Ansari, O. Zed, R. Peri, S. Manne, U. Gupta, G.-Y. Wei, D. Brooks, G. Hills, and C.-J. Wu, "CORDOBA: Carbon-Efficient Optimization Framework for Computing Systems," in *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2025, pp. 1289–1303.
- [24] C. Elsworth, K. Huang, D. Patterson, I. Schneider, R. Sedivy, S. Goodman, B. Townsend, P. Ranganathan, J. Dean, A. Vahdat, B. Gomes, and J. Manyika, "Measuring the environmental impact of delivering AI at Google Scale," 2025. [Online]. Available: <https://arxiv.org/abs/2508.15734>
- [25] EPA Facility Level Information on GreenHouse gases Tool (flight), "2023 Greenhouse Gas Emissions from Large Facilities," 2023. [Online]. Available: <https://ghgdata.epa.gov/ghgp/main.do#/listFacility/>
- [26] F. Fayza, C. Demirkiran, S. P. Rao, D. Bunandar, U. Gupta, and A. Joshi, "Epicarbon: A carbon modeling tool for electro-photonics accelerators," in *2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2025, pp. 1–9.
- [27] M. Garcia Bardon, P. Wuytens, L.-Ragnarsson, G. Mirabelli, D. Jang, G. Willems, A. Mallik, A. Spessot, J. Ryckaert, and B. Parvais, "DTCO including Sustainability: Power-Performance-Area-Cost-Environmental score (PPACE) Analysis for Logic Technologies," in *2020 IEEE International Electron Devices Meeting (IEDM)*, 2020, pp. 41.4.1–41.4.4.
- [28] B. Gaudette, C.-J. Wu, and S. Vruthula, "Improving smartphone user experience by balancing performance and energy with probabilistic qos guarantee," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2016, pp. 52–63.
- [29] —, "Optimizing user satisfaction of mobile workloads subject to various sources of uncertainties," *IEEE Transactions on Mobile Computing*, vol. 18, no. 12, pp. 2941–2953, 2019.
- [30] Google, "Environmental Report 2025," 2025. [Online]. Available: <https://www.gstatic.com/gumdrop/sustainability/google-2025-environmental-report.pdf>
- [31] D. Grey-Stewart, D. Kong, M. Elgamal, G. Kyriazidis, J. Morris, and G. Hills, "Quantifying trade-offs in power, performance, area, and total carbon footprint of future three-dimensional integrated computing systems," in *2025 Design, Automation Test in Europe Conference (DATE)*, 2025, pp. 1–7.
- [32] U. Gupta, M. Elgamal, G. Hills, G.-Y. Wei, H.-H. S. Lee, D. Brooks, and C.-J. Wu, "ACT: Designing Sustainable Computer Systems with an Architectural Carbon Modeling Tool," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: <https://doi.org/10.1145/3470496.3527408>
- [33] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H.-H. S. Lee, G.-Y. Wei, D. Brooks, and C.-J. Wu, "Chasing Carbon: The elusive environmental footprint of computing," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. Los Alamitos, CA, USA: IEEE Computer Society, mar 2021, pp. 854–867.
- [34] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2024. [Online]. Available: <https://www.gurobi.com>
- [35] imec, "imec.netzero Public." [Online]. Available: <https://netzero.imec-int.com>
- [36] S. Ji, Z. Yang, X. Chen, S. Cahoon, J. Hu, Y. Shi, A. K. Jones, and P. Zhou, "Scarif: Towards carbon modeling of cloud servers with accelerators," 2024. [Online]. Available: <https://arxiv.org/abs/2401.06270>
- [37] A. K. Jones, Y. Chen, W. O. Collinge, H. Xu, L. A. Schaefer, A. E. Landis, and M. M. Bilec, "Considering fabrication in sustainable computing," in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2013, pp. 206–210.
- [38] A. K. Jones, L. Liao, W. O. Collinge, H. Xu, L. A. Schaefer, A. E. Landis, and M. M. Bilec, "Green computing: A life cycle perspective," in *2013 International Green Computing Conference Proceedings*, 2013, pp. 1–6.
- [39] S. W. Jones, "Modeling 300mm Wafer Fab Carbon Emissions," in *2023 International Electron Devices Meeting (IEDM)*, 2023, pp. 1–4.
- [40] S. M. Khan and A. Mann, "Ai chips: What they are and why they matter," Center for Security and Emerging Technology, Tech. Rep., 2020.
- [41] D. Kline, N. Parshook, X. Ge, E. Brunvand, R. Melhem, P. K. Chrysanthis, and A. K. Jones, "GreenChip: A tool for evaluating holistic sustainability of modern computing systems," *Sustainable Computing: Informatics and Systems*, vol. 22, pp. 322–332, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2210537917300823>

- [42] S. Krishnan, A. Yazdanbakhsh, S. Prakash, J. Jabbour, I. Uchendu, S. Ghosh, B. Boroujerdian, D. Richins, D. Tripathy, A. Faust, and V. Janapa Reddi, "ArchGym: An Open-Source Gymnasium for Machine Learning Assisted Architecture Design," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ser. ISCA '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: <https://doi.org/10.1145/3579371.3589049>
- [43] S. Longofono, D. Kline, R. Melhem, and A. K. Jones, "Toward secure, reliable, and energy efficient phase-change main memory with MACE," in *2019 Tenth International Green and Sustainable Computing Conference (IGSC)*, 2019, pp. 1–8.
- [44] Y. Luo and L. K. John, "Using Statistical Theory to Study Issues in Microprocessor Simulation," in *Technical Report TR-040225-01. University of Texas, Department of ECE.*, 2008. [Online]. Available: <https://lca.ece.utexas.edu/pubs/yue-ibm-cas04.pdf>
- [45] R. E. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," *IBM Journal of Research and Development*, vol. 6, no. 2, pp. 200–209, 1962.
- [46] F. Mazurek, A. Tschand, Y. Wang, M. Pajic, and D. Sorin, "Rigorous evaluation of computer processors with statistical model checking," in *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 1242–1254. [Online]. Available: <https://doi.org/10.1145/3613424.3623785>
- [47] Microsoft, "2025 Environmental Sustainability Report," 2025. [Online]. Available: <https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/msc/documents/presentations/CSR/2025-Microsoft-Environmental-Sustainability-Report.pdf>
- [48] B. Murphy, "Cost-size optima of monolithic integrated circuits," *Proceedings of the IEEE*, vol. 52, no. 12, pp. 1537–1545, 1964.
- [49] A. C. Newkirk, J. Fernandez, J. Koomey, I. Latif, E. Strubell, A. Shehabi, and C. Samaras, "Empirically-Calibrated H100 Node Power Models for Reducing Uncertainty in AI Training Energy Estimation," 2025. [Online]. Available: <https://arxiv.org/abs/2506.14551>
- [50] A. Ning, G. Tziantzioulis, and D. Wentzlaff, "Supply Chain Aware Computer Architecture," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ser. ISCA '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: <https://doi.org/10.1145/3579371.3589052>
- [51] A. Ning and D. Wentzlaff, "Chip Architectures Under Advanced Computing Sanctions," in *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, ser. ISCA '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 1225–1239. [Online]. Available: <https://doi.org/10.1145/3695053.3731012>
- [52] I. O'Brien, "Data center emissions probably 662% higher than big tech claims. Can it keep up the ruse?" 2024. [Online]. Available: <https://www.theguardian.com/technology/2024/sep/15/data-center-gas-emissions-tech>
- [53] C. I. S. Operator, "California ISO," 2023. [Online]. Available: <https://www.caiso.com/TodaysOutlook/Pages/emissions.html>
- [54] T. Pirson, T. P. Delhay, A. G. Pip, G. Le Brun, J.-P. Raskin, and D. Bol, "The environmental footprint of ic production: Review, analysis, and lessons from historical trends," *IEEE Transactions on Semiconductor Manufacturing*, vol. 36, no. 1, pp. 56–67, 2023.
- [55] J. Rogers, L. Eeckhout, and M. Jahre, "HILP: Accounting for Workload-Level Parallelism in System-on-Chip Design Space Exploration," in *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2025, pp. 1275–1288.
- [56] J. Rogers, L. Eeckhout, T. Soliman, and M. Jahre, "Neoscope: How Resilient Is My SoC to Workload Churn?" in *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, ser. ISCA '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 1296–1310. [Online]. Available: <https://doi.org/10.1145/3695053.3731014>
- [57] G. Roussilhe, T. Pirson, M. Xhonneux, and D. Bol, "From silicon shield to carbon lock-in? the environmental footprint of electronic components manufacturing in taiwan (2015–2020)," *Journal of Industrial Ecology*, vol. 28, no. 5, p. 1212–1226, Jul. 2024. [Online]. Available: <http://dx.doi.org/10.1111/jiec.13487>
- [58] Semiconductor Industry Association (SIA), "Chipmakers Are Ramping Up Production to Address Semiconductor Shortage. Here's Why that Takes Time," 2021. [Online]. Available: <https://www.semiconductors.org/chipmakers-are-ramping-up-production-to-address-semiconductor-shortage-heres-why-that-takes-time/#~:text=Manufacturing%20a%20finished%20chip%20for,additional%206%20weeks%20to%20complete.>
- [59] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder, "Automatically characterizing large scale program behavior," *SIGPLAN Not.*, vol. 37, no. 10, p. 45–57, Oct. 2002. [Online]. Available: <https://doi.org/10.1145/605432.605403>
- [60] C. C. Sudarshan, N. Matkar, S. Vrudhula, S. S. Sapatnekar, and V. A. Chhabria, "Eco-chip: Estimation of carbon footprint of chiplet-based architectures for sustainable vlsi," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2024, pp. 671–685.
- [61] Taiwan Ministry of Economic Affairs, Energy Administration, "2021 Electricity Carbon Emission Factor." [Online]. Available: https://www.moeaea.gov.tw/ECW/English/content/Content.aspx?menu_id=20721
- [62] —, "2022 Electricity Carbon Emission Factor." [Online]. Available: https://www.moeaea.gov.tw/ECW/English/content/Content.aspx?menu_id=24200
- [63] —, "Energy Statistics Handbook 2024," 2024. [Online]. Available: <https://www.esist.org.tw/a0303/02/en/publication/handbook/?tab=Summary&subtab=>
- [64] E. Talpes, D. D. Sarma, G. Venkataramanan, P. Bannon, B. McGee, B. Floering, A. Jalote, C. Hsiang, S. Arora, A. Gorti, and G. S. Sachdev, "Compute solution for tesla's full self-driving computer," *IEEE Micro*, vol. 40, no. 2, pp. 25–35, 2020.
- [65] TechPowerUP, "AMD EPYC 7543," 2025. [Online]. Available: <https://www.techpowerup.com/cpu-specs/epyc-7543.c2379>
- [66] —, "NVIDIA A100 PCIe 40 GB," 2025. [Online]. Available: <https://www.techpowerup.com/gpu-specs/a100-pcie-40-gb.c3623>
- [67] TSMC, "Logic technology." [Online]. Available: <https://www.tsmc.com/english/dedicatedFoundry/technology/logic>
- [68] —, "TSMC 2024 Sustainability Report," 2024. [Online]. Available: <https://esg.tsmc.com/file/public/2024-TSMC-Sustainability-Report-e.pdf>
- [69] United States Environmental Protection Agency (EPA), "Greenhouse Gas Equivalencies Calculator," 2024. [Online]. Available: <https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator>
- [70] U.S. Energy Information Administration, Form EIA-861M, "Electric Power Monthly," U.S. Energy Information Administration (EIA), Tech. Rep., 2026. [Online]. Available: https://www.eia.gov/electricity/monthly/current_month/january2026.pdf
- [71] J. Wang, D. S. Berger, F. Kazhamiaka, C. Irvine, C. Zhang, E. Choukse, K. Frost, R. Fonseca, B. Warriar, C. Bansal, J. Stern, R. Bianchini, and A. Sriraman, "Designing cloud servers for lower carbon," in *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, 2024, pp. 452–470.
- [72] M. Wang, W. Lv, F. Yang, C. Yan, W. Cai, D. Zhou, and X. Zeng, "Efficient yield optimization for analog and sram circuits via gaussian process regression and adaptive yield estimation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 10, pp. 1929–1942, 2018.
- [73] Wissolik, Mike and Zacher, Darren and Torza, Anthony and Day, Brandon, "Virtex UltraScale+ HBM FPGA: A Revolutionary Increase in Memory Performance," XILINX, Tech. Rep., 2019. [Online]. Available: <https://docs.amd.com/api/khuf/documents/spQZaozIMEJCwco8GILxPg/content>
- [74] T. Xiao, F. F. Nerini, H. D. Matthews, M. Tavoni, and F. You, "Environmental impact and net-zero pathways for sustainable artificial intelligence servers in the USA," in *Nature Sustainability*, vol. 8, 2025, p. 1541–1553.
- [75] Y. Zhao, Y. K. Zhao, C. Wan, and Y. C. Lin, "3d-carbon: An analytical carbon modeling tool for 3d and 2.5d integrated circuits," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, ser. DAC '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: <https://doi.org/10.1145/3649329.3658482>