Quantifying the Computing Energy Efficiency Paradox

PRANJALI JAIN, JONATHAN BALKIND, and TIMOTHY SHERWOOD, UC Santa Barbara, USA

Computer hardware continues to deliver improvements in energy-efficiency year-over-year at an incredible pace. Performing a billion floating point operations today takes less than 10% of the energy the same computation would take just 10 years ago. Yet, somewhat paradoxically, the total amount of energy spent on computation has never been larger. This "rebound effect", where improvements in efficiency and increased overall resource consumption interrelate, is not unique to computing and was first observed by economists as Jevons Paradox when trying to understand the rising demand for coal in the 1800s. While this connection is familiar to many as an intuition, in this paper we establish the relationship concretely and quantitatively. In addition to providing new ways of understanding and visualizing the role of an "energy elasticity" at play here, we explore this directly through an analysis of two different power-intensive computing workloads over time: GPUs for AI model training and ASICs for cryptocurrency mining. In the end, we find that 10% improvements in energy efficiency are correlated with 76.8% and 21.9% increases in energy demand respectively for AI model training and cryptocurrency mining. If current exponential trends were to somehow continue despite the physical and economic limits of energy supply, fabrication capacity, and capital investment, by 2035 the carbon emissions resulting from training the largest AI model alone could surpass the total carbon emissions of the entire USA in 2024.

CCS Concepts: • **Hardware** → **Power and energy**; • **Computing methodologies** → *Artificial intelligence*; • **Applied computing** → Economics.

Additional Key Words and Phrases: Energy Efficiency, Carbon Footprint

1 Introduction

Reducing the energy consumption of computer systems has long been a fundamental research objective for computer architects and system designers, focusing on innovative hardware designs, more efficient semiconductor materials, advanced cooling techniques, and algorithmic and software optimizations. However, despite continuous improvements in energy efficiency, the overall demand for computation seems to continually outpace those improvements resulting in a net increase in total energy consumption. This phenomenon closely mirrors "Jevons Paradox" (also known as the rebound effect), which posits that technological improvements increasing the efficiency of utilization of a resource often lead to increased, rather than decreased, total resource consumption [1, 10]. While rebound effects have been mathematically explored in other domains (e.g., transport [19], energy sectors [15]), existing discussions of rebound effects in computing [5, 12, 16] are largely qualitative or specifically focus on how hardware or algorithm driven energy efficiency gains reduce environmental impacts of specific technologies. These analyses do not explore how energy efficiency interacts with rising computational demand, how cost of compute and energy are linked, and how these dynamics translate into environmental impacts. As such, assessing the environmental footprint of individual computing systems alone is insufficient. As computer scientists we would be well served by better understanding how efficiency improvements are subject to the market forces that drive wider deployment and result in broader environmental impact [5, 11].

In this paper, we seek to establish a precise and concrete mathematical relationship between rising computational demand and its corresponding energy consumption, and further the carbon emissions resulting from this energy consumption, grounded in well understood economic theory. The challenge in *directly* applying the textbook understanding of the rebound effect is it requires many assumptions about revenue, pricing, utility, and more. In the case of energy efficiency and total carbon, we have a set of variables that are not unrelated to these, but are also not directly equivalent either. We start by briefly reviewing the fundamentals of elasticity of demand and the dynamical relationships that lead to the rebound effect (Section 2). From there we can establish the connections to energy and carbon more concretely (Sections 2.1, 2.2). As long as the relationship between the revenue-per-unit and the energy-per-unit of compute is smooth, our approximations should be sound. When these relationships are also both constant over time and revenue is directly proportional to the energy invested, a clear power law emerges. In addition to describing this phenomenon analytically, we examine this relationship quantitatively through case studies (Section 3) in two different highly energy intensive domains: AI model training and cryptocurrency mining. Of course when dealing with such models there are many assumptions to be made, and we attempt to lay those assumptions out clearly and explore different options when they exist (e.g. the impact of renewable energy sources). Making additional assumptions about how these relationships play out over time (e.g. assuming continued hardware scaling is available), one can project forward as these power laws drive exponential increments in both energy consumption and carbon footprint (Section 4).

2 Elasticity of Computational Demand

The rebound effect, first observed in 1865 by William Jevons in the context of coal consumption in steam engines [1, 10], is closely tied to the economic concept of price elasticity of demand [13], which quantifies how sensitive the quantity demanded of a product is to changes in its price, expressed as the percentage change in quantity demanded relative to the percentage change in price. The quantity of a product sold is dependent on its price. As such, given a quantity $Q(P)$ of a particular product sold at price P per product, the price elasticity of demand, ϵ is expressed as: $\epsilon = \frac{\frac{\partial Q}{Q(P)}}{\frac{\partial P}{P}} = \frac{P}{Q(P)}\frac{\partial Q}{\partial P}$. Note that, selling quantity $Q(P)$ of a product at price P generates revenue $R(P)$, that is, $R(P) = P \cdot Q(P)$. Further, driven by consumer behavior and diminishing marginal utility, the law of demand states that all else being equal, as the price of a product decreases, the quantity demanded increases, and vice versa, that is, $\frac{\partial Q}{\partial P} < 0 \iff \frac{\partial P}{\partial Q} < 0$. Since both price and quantity are always positive numbers, and the change in quantity with respect to changes in price is always negative, the price elasticity of demand ϵ is also always negative.

Authors' Contact Information: Pranjali Jain, pranjali_jain@ucsb.edu; Jonathan Balkind, jbalkind@ucsb.edu; Timothy Sherwood, sherwood@cs.ucsb.edu, UC Santa Barbara, Santa Barbara, CA, USA.

Economists characterize demand responses as inelastic, elastic, or unit elastic. Inelastic demand occurs when the quantity demanded remains relatively stable despite shifts in price, or $|\epsilon| < 1$. In this case, a dollar saved due to a decrease in the price of a product results in less than a dollar being reinvested in buying more of the product, so total expenditure declines even with more product sold. Unit elasticity represents a balanced state where percentage changes in price lead to equivalent changes in quantity demanded, reflected by $|\epsilon| = 1$. A dollar saved is fully reinvested in purchasing more of the same product, keeping the total expenditure constant. Elastic demand occurs when the total expenditure actually grows with price cuts or $|\epsilon| > 1$. This is the rebound effect, where a dollar saved in price leads to more than a dollar of additional demand.

2.1 Computational Energy and the Rebound Effect

The total revenue generated from a datacenter scale computer is practically impossible to model in accurate detail, especially given how little detailed information is available publicly. However, in order to investigate the relationship between revenue and computational demand, we need to assume that there exists some approximately smooth and continuous relationship between revenue and the price per unit of compute. We make the assumption that the price to execute one unit of compute comprises a base price and then some multiple of the total energy expenditure associated with performing the computation.

It is important to note that when we start to talk about the energy required to deliver some computation, it is natural to think primarily about the operational energy associated with executing the computation. However, here we mean energy in a more holistic way that includes not only power consumption, power transmission loss, cooling, etc., but also the capital energy expenditures required to manufacture and integrate the systems running the computation. We wrap all of this into a parameter η, that captures the energy intensity of the computation (which is the energy expended per unit of compute), and is inversely proportional to energy efficiency. Given these assumptions, we can think about the price of delivering a unit of compute as comprising a portion that is proportional to the energy intensity η (which covers any operational or capital expense that grows in direct proportion with the number of watts required) and then some constant factor (capturing the components of price that are not proportional to energy expended).

The price per unit of compute is a function of the energy intensity of compute η weighed by some constant k_1 and offset by some constant base price k_2, that is, $P(\eta) = (k_1 \cdot \eta + k_2)$. Since revenue is simply the total price charged per unit of compute, in this case, revenue is then a smooth and continuous function of price per compute. The computational demand also depends on the price per unit of compute, and as a result on the energy intensity of compute, making it a function of energy intensity, or $Q(P(\eta)) = Q_c(\eta)$. Thus, the total revenue generated by a datacenter scale computer can be approximated as: $R(P(\eta)) = P(\eta) \cdot Q(P(\eta)) = (k_1 \cdot \eta + k_2) \cdot Q_c(\eta)$.

We see that the revenue generated from a datacenter scale computer varies in direct proportion to the total energy invested (or consumed) to service the computational demand. The total energy consumption E is dependent on the computational operations being executed $Q_c(\eta)$ and their energy intensity η, as $E(\eta) = \eta \cdot Q_c(\eta)$.

From a market behavior standpoint, fluctuations in the price of one unit of compute can impact the overall revenue to be generated from servicing the computational demand. To quantify this variability, we take the derivative of revenue with respect to price per compute. We find that this derivative is a function of the price elasticity of demand (ϵ), which represents the percentage change in computational demand ($Q_c(\eta)$) with percentage change in price per compute ($P(\eta) = \eta \cdot k_1 + k_2$) . Since price per compute $P(\eta)$ is a function of energy intensity η, upon simplification, the price elasticity of demand becomes the percentage change in computational demand $Q_c(\eta)$ with respect to percentage change in the energy intensity factor $\left(\eta + \frac{k_2}{k_1}\right)$, where the ratio $\frac{k_2}{k_1}$ has the dimensions of energy per compute. The price elasticity of demand (ϵ) is:

$$\epsilon = \frac{\left(\eta + \frac{k_2}{k_1}\right)}{\partial \eta} \cdot \frac{\partial Q_c(\eta)}{Q_c(\eta)} \tag{1}$$

We can establish the relationship between computational demand and energy intensity of these computations, by integrating the price elasticity of demand (Eq. (1)) while assuming constant elasticity. However, in reality the price elasticity of demand for a product is not necessarily constant and can vary across different price ranges or under significant changes in market conditions. But if the product is within its usual price range, elasticity tends to remain relatively stable since the demand response shifts with price fluctuations in a predictable manner. So, under the assumption that the price per compute does not vary drastically and the price elasticity of demand remains constant over the price range of interest, we can integrate Eq. (1) to determine the relationship between computational demand $Q_c(\eta)$ and the energy intensity factor $\left(\eta + \frac{k_2}{k_1}\right)$. Assuming the constant of integration to be $C = -\log E_0$, where E_0 is also a constant, the integration yields:

$$\begin{aligned} \log\left(\eta + \frac{k_2}{k_1}\right) &= \frac{1}{\epsilon} \cdot \log Q_c(\eta) - \frac{1}{\epsilon} \cdot \log E_0 \\ \iff Q_c(\eta) &= E_0 \cdot \left(\eta + \frac{k_2}{k_1}\right)^{\epsilon} \end{aligned} \tag{2}$$

The integration is a power law relationship between energy intensity η and computational demand $Q_c(\eta)$, as represented graphically in log-log scale in Figure 1. Here, the baseline energy investment E_0, captures the total energy consumption at a reference computational demand. Any future changes in the total energy consumption (resulting from variations in computational demand due to the elasticity of the market) are assessed relative to E_0. In other words, E_0 is a constant reference energy investment and the market is evaluated relative to it. Changes in energy intensity measured with respect to this baseline determine the corresponding changes in computational demand, which in turn impacts the total energy consumption of the computational demand.

The relationship between computational demand and energy intensity (Eq. (2)) presented in Figure 1 is dependent on ϵ, k_1, k_2, E_0. When $k_2 \to 0$ or $k_1 \gg k_2$, the $\frac{k_2}{k_1}$ factor effectively disappears from the energy intensity factor $\left(\eta + \frac{k_2}{k_1}\right)$. This scenario happens when

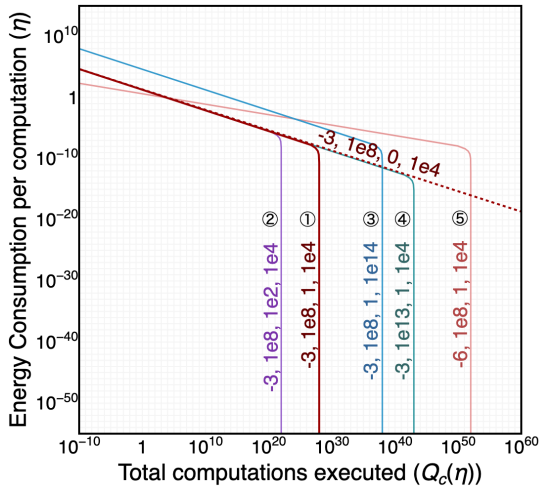


Fig. 1. Relationship between energy intensity and computational demand in log-log scale. Each line on the log-log plot is annotated with a tuple representing (ϵ, k_1, k_2, E_0).

the base price per unit of compute (k_2) is negligible compared to the portion of price due to the energy investment ($\eta \cdot k_1$), leading to the market remaining elastic with variations in computational demand. For a given ϵ and E_0, the Eq. (2) becomes a straight line ($y = m \cdot x + c$) in log-log scale (dotted line in Figure 1):

$$\log(\eta) = \frac{1}{\epsilon} \cdot \log Q_c(\eta) - \frac{1}{\epsilon} \cdot \log E_0 \iff Q_c(\eta) = E_0 \cdot \eta^{\epsilon} \quad (3)$$

When $k_2 > 0$ (or $\frac{k_2}{k_1} > 0$), this is a knee-shaped curve in log-log scale (①, ②, ④) in Figure 1). As long as $\eta > \frac{k_2}{k_1}$, the market is elastic with changes in demand. Once the ratio $\frac{k_2}{k_1}$ starts dominating over energy intensity η, (that is, $\eta < \frac{k_2}{k_1}$), the curve becomes knee-shaped and then quickly approaches a vertical asymptote with any further reduction in energy intensity, indicating that demand has become effectively inelastic. This means further improvements in energy efficiency no longer lead to increasing computational demand. When $\frac{k_2}{k_1}$ and E_0 are constant, the slope of the straight line before the knee varies with ϵ (①, ③), whereas when $\frac{k_2}{k_1}$ and ϵ are constant, E_0 influences the offset of the curve along the y-axis (①, ⑤).

We can now extend the relationship between computational demand and energy intensity to determine the total energy consumption ($E(\eta)$) needed to support the computational demand. Since the total energy consumption $E(\eta)$ is directly proportional to the computational demand $Q_c(\eta)$ (that is, $E(\eta) = \eta \cdot Q_c(\eta)$), when $k_2 \to 0$, the total energy consumption becomes exclusively dependent on energy intensity raised to the power of the price elasticity of demand, $E(\eta) = \eta \cdot Q_c(\eta) = E_0 \cdot \eta^{\epsilon+1}$. This relationship is the rebound effect. For a given price elasticity of demand, as computational energy efficiency increases (energy intensity η decreases), the total energy consumption $E(\eta)$ increases as well.

The value of $|\epsilon|$ relative to 1 governs how total energy consumption shifts in response to changes in energy efficiency. When $|\epsilon| > 1$, computational demand is elastic. Energy efficiency gains result in disproportionately large increases in demand, leading to an overall rise in energy use, and hence, the rebound effect. Conversely, when $|\epsilon| < 1$, computational demand is energy inelastic, so improvements in energy efficiency lead to proportionally smaller increases in computational demand, and thus total energy consumption decreases. If $|\epsilon| = 1$, the system is unit elastic, where the increase in computational demand exactly offsets the energy efficiency gains, keeping total energy consumption constant. Addressing the rebound effect requires maintaining computational demand at inelastic or unit elastic levels with respect to energy efficiency, yet, in reality, computational demand is often highly elastic, limiting the effectiveness of energy efficiency gains. The only way for computational demand to have constant elasticity is if it follows an exponential relationship with energy intensity. When computational demand is unit elastic $|\epsilon| = 1$, the total energy consumption is equal to E_0. This represents a fixed total energy budget, equal to the total energy consumption when computational demand is unit elastic. This is an **iso-energy line** on log-log scale:

$$\log(\eta) = -\log(Q_c(\eta)) + \log(E_0) \iff Q_c(\eta) \cdot \eta = E_0 \quad (4)$$

2.2 Carbon Footprint and the Rebound Effect

The total energy consumed (E) to meet the computational demand also results in carbon emissions, determined by the carbon intensity CI_{source} of the energy source. Carbon intensity quantifies the amount of carbon emissions produced per unit of energy consumption, measured in kilograms of carbon-dioxide equivalent emissions per unit energy consumed (kg CO_2e /J). The carbon intensity of the energy source and the energy intensity of computation determine the carbon footprint per compute f, that is $f = \eta \cdot CI_{\text{source}}$. To quantify how changes in computational demand impact the carbon footprint per compute, we can express the price elasticity of demand (ϵ) as the percentage change in computational operations ($Q_c(\eta)$) with percentage change in carbon footprint per compute (f). Similar to energy consumption, integrating this elasticity equation yields a power law relationship between the factor $\left(f + \frac{k_2}{k_1} \cdot CI_{\text{source}}\right)$ and computational demand $Q_c(\eta)$ (where C_0 is the baseline carbon footprint), as follows:

$$\log\left(f + \frac{k_2}{k_1} \cdot CI_{\text{source}}\right) = \frac{1}{\epsilon} \cdot \log Q_c(\eta) - \frac{1}{\epsilon} \cdot \log C_0 \quad (5)$$

When $k_2 \to 0$, the $\frac{k_2}{k_1}$ factor disappears from Eq. (5), revealing a power law relationship between computational demand $Q_c(\eta)$ and carbon footprint per compute f, as follows:

$$\log(f) = \frac{1}{\epsilon} \cdot \log Q_c(\eta) - \frac{1}{\epsilon} \cdot \log C_0 \quad (6)$$

When computational demand is unit elastic ($\frac{1}{\epsilon} = -1$), the carbon footprint due to total energy consumption remains constant, even as computational demand $Q_c(\eta)$ changes, because the carbon footprint per computation f changes proportionally. This is an **iso-carbon line**, expressed as:

$$\log(f) = -\log(Q_c(f)) + \log(C_0) \implies E_0 = \eta \cdot Q_c(\eta) \cdot CI_{\text{source}} \quad (7)$$

Decarbonizing the energy supply can reduce the total carbon footprint if computational demand is inelastic or unit elastic. Even if computational demand is elastic, reducing the carbon intensity of energy supply has the potential to yield a reduction in total carbon emissions, provided that the reduction occurs at a rate that matches or exceeds the growth in total energy consumption.

3 Case Studies

Using the mathematical framework established earlier, we analyze two prominent computation and energy intensive application domains, namely AI model training and cryptocurrency mining. Note that while these areas of computing are important drivers of system designs at scale today, they may not reflect the elasticity of computing as a whole. Additionally, our predictions about future behavior of these computing domains are not guarantees given how quickly these domains evolve. Leveraging available data, we present log-log plots of the elasticity of computational demand and the corresponding carbon emissions from energy consumption for these application domains. We treat the rated power of GPUs and ASICs as a proxy for the energy proportional portion of price per compute and accordingly use this to estimate the energy intensity.

For carbon footprint calculations, we utilize the average annual carbon intensity of the USA grid [6], and median carbon intensities of renewable energy sources [3], namely 12 g CO_2e/kWh for nuclear energy and 27 g CO_2e/kWh for solar power. We consider 100% solar grid to be one where electricity is generated using only solar power, while in a 100% nuclear grid, it comes solely from nuclear energy. Even 100% solar and nuclear grids have a carbon intensity due to emissions from manufacturing, maintenance and decommissioning of the infrastructure. The total carbon emission by the entirety of the USA in 2024 amounted to 5.912×10^{12} kg CO_2e [4].

3.1 AI Model Training

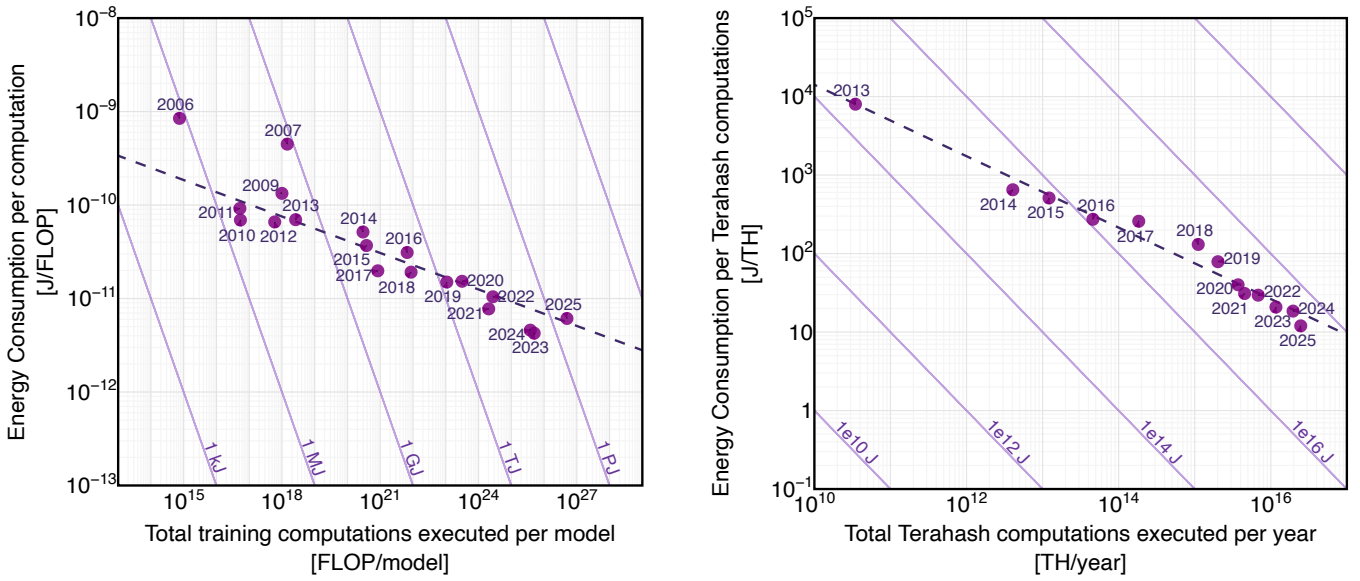
Computational demand per AI model training can be estimated through a combination of the most energy efficient GPU hardware available [9, 18] in relation to AI model complexity [7, 8]. Specifically, we plot the energy consumption per floating-point operation (FLOP) for the highest-throughput GPU released in the market (energy intensity η) annually on the y-axis against the total number of training FLOP required by the largest AI model (largest in terms of number of training FLOP) introduced in the subsequent year (total number of computations Q_c) on the x-axis. The log-log scale plot can be seen in Figure 2a. The markers corresponding to each datapoint represent the release year of the largest AI model. The dashed line shows the power law fitted to these datapoints, as indicated in Eq. (3). The solid angled lines on the graph are *iso-energy lines*, with markers denoting the associated total energy consumption in Joules. As shown in Figure 2a, the slope of the best-fit line corresponding to the power law equation is $\frac{1}{\epsilon_C} = -0.130$, which indicates the computational demand for AI model training is highly elastic ($|\epsilon_C| = 7.68$). A 10% improvement in energy efficiency of GPUs (or 10% reduction in GPU energy intensity) is correlated with a 76.8% increase in computational demand for AI model training.

We also estimate the carbon footprint for those same AI models. The carbon footprint per FLOP for the highest throughput GPU released annually is plotted on the y-axis against the total number of training FLOP for the largest AI model on the x-axis. Again this is shown in log-log scale, now in Figure 3a. The carbon footprint per FLOP is calculated by multiplying the carbon intensity of the energy source with the energy consumption per floating-point operation (FLOP) of the highest throughput GPU. The carbon emissions per FLOP for the plotted points are estimated assuming a carbon intensity of the average USA electricity grid. The markers for the datapoints indicate the release year of the AI model. The green dashed line shows the power law fit to these datapoints, based on Eq. (6). The solid angled lines now are *iso-carbon* lines, with markers representing the estimated total carbon emissions in kg CO_2e. The solid red iso-carbon line represents the total USA carbon emissions in 2024, which amounted to 5.912 trillion kg CO_2e [17]. The brown and orange dashed lines represent the power law fit lines calculated assuming a 100% nuclear-powered grid and a 100% solar-powered grid respectively, considering median carbon intensities for these energy sources and that the computational demand exhibits the same elasticity as the energy consumption curve (Figure 2a). The power law fit line calculated based on the carbon intensity of the average USA electricity grid (green dashed line in Figure 3a) has the slope of $\frac{1}{\epsilon_C} = -0.144$. The elasticity of computational demand in this case ($|\epsilon_C| = 6.92$) is smaller than that derived from the energy consumption curve ($|\epsilon_C| = 7.68$), due to consistent reduction in the carbon intensity of the USA electricity grid by 2.25% year-on-year.

3.2 Cryptocurrency Mining

We examine the computational demand for cryptocurrency mining (specifically Bitcoin mining) by exploring the energy efficiency improvements in cryptocurrency mining ASICs [14], utilizing the metric of energy consumption per terahash computations [2]. Similar to the AI case study, we plot on log-log scale the energy efficiency of the highest hashrate ASIC released in the market (energy intensity η) annually on the y-axis against the total terahash (TH) computations executed in the subsequent year (total computations Q_c) on the x-axis, as shown in Figure 2b. Each datapoint is marked with the year when the corresponding number of terahashes were executed. The solid angled lines are *iso-energy lines* representing total energy consumption in Joules and the dashed line shows the power law fit line based on Eq. (3). The power law fit to the data in Figure 2b had a log-log slope of $\frac{1}{\epsilon_C} = -0.454$, meaning the elasticity of computational demand for cryptocurrency mining is estimated to be $|\epsilon_C| = 2.19$. In other words, for every improvement of 10% in the ASIC mining hardware energy efficiency (or 10% reduction in ASIC energy intensity), is correlated with the computational demand for cryptocurrency mining increasing by 21.9%. While this would be highly elastic, it is less elastic than AI model training.

In Figure 3b, we plot the carbon footprint per terahash computations of the highest hashrate ASIC hardware on the y-axis against the total terahashes executed annually on the x-axis in log-log scale. The carbon footprint per terahash is again calculated based on the average annual carbon intensity of the USA electricity grid. The datapoint are labeled with the year terahash computations were executed. The power law fit to these datapoints based on Eq. (6), is shown by the green dashed line in Figure 3b, the solid angled lines



(a) AI Model Training

(b) Cryptocurrency Mining

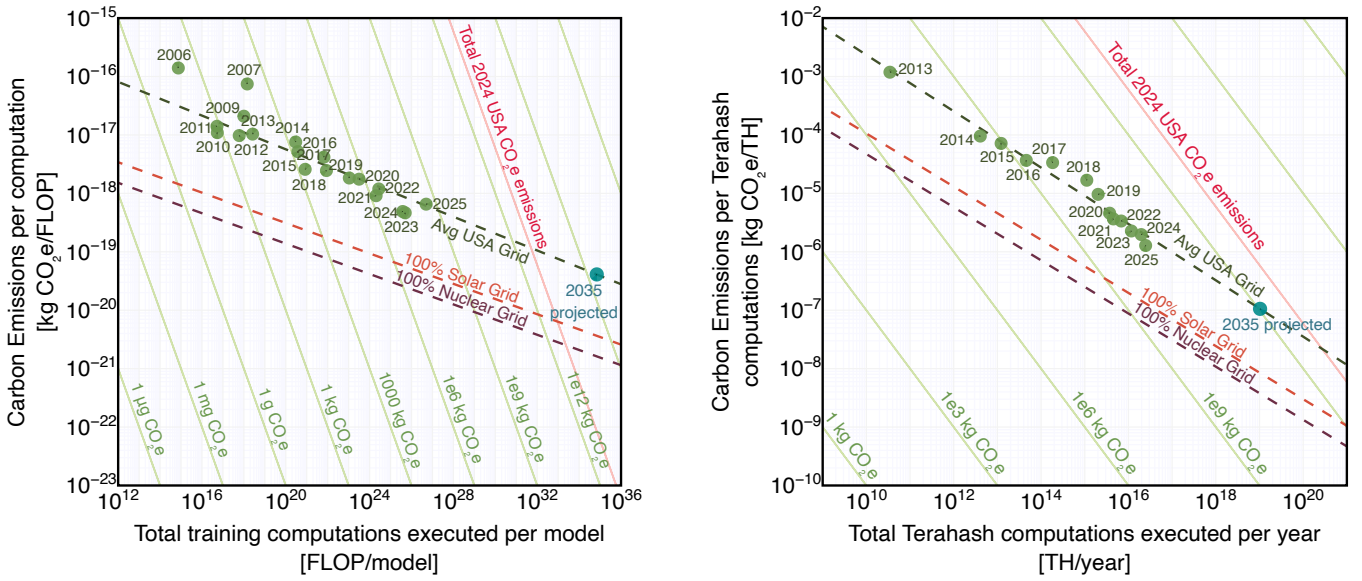
Fig. 2. Estimated total energy consumption per AI model training and annual cryptocurrency mining in log-log scale. The solid angled purple lines are iso-energy lines, and the dashed purple line is the power law of total energy consumption fitted to the datapoints. *For AI model training, each datapoint indicates energy intensity per computation of the highest-throughput GPU Hardware introduced in a given year against the total training FLOP for the largest AI model released in the subsequent year. The markers indicate the release year of the AI model. For cryptocurrency mining, each datapoint indicates energy intensity per computation of the highest-throughput ASIC released in a given year against the total terahashes executed in the subsequent year. The markers indicate the year in which the terahashes were executed.*

again represent the iso-carbon lines annotated with total carbon emissions in kg CO_2e, and the solid red iso-carbon line indicates the estimated total USA carbon emissions in 2024 for reference (5.912 trillion kg CO_2e). The brown and orange dashed lines again represent the power law fit lines for a 100% nuclear-powered grid and a 100% solar-powered grid respectively. The power law fit line indicated by the green dashed line in Figure 3b) once again has a smaller elasticity of computational demand ($|\epsilon_C| = 2.07$) compared to the energy consumption curve ($|\epsilon_C| = 2.19$), owing to the carbon intensity improvements in the USA electricity grid.

4 Results

Starting from the basic economic principle that revenue generated from a datacenter scale computer equals the total computational demand it serves multiplied by the price of one unit of compute, we establish a mathematical basis for the rebound effect by deriving a power law relationship between computational demand and the energy intensity of computation, with the exponent equal to the elasticity of the market. Our analysis is contingent on the assumptions we make about how the price per unit of compute is a combination of some base price plus a component tied to the energy investment to perform the computation, that the base price is negligible compared to the portion of price due to the energy investment, and that the market remains elastic within a reasonable range of price per compute. Under these assumptions, we extend the power law to determine how total energy consumption and corresponding carbon emissions respond to shifts in computational demand and energy intensity, and plot these relationships in log-log for AI model training and cryptocurrency mining. The near constant slope in log-log is consistent with significant elasticity. This is true when measuring cost in terms of either energy or carbon, although the slope for carbon is driven slightly lower by the consistent reduction in the carbon intensity of the USA electricity grid, which has decreased by about 2.25% annually on average. However, while there is a difference, those changes are hard to see given the other exponential constants. The smaller elasticity value is evidence that decarbonizing the electricity grid can slow the rate of increase of carbon emissions relative to total energy consumption.

The dashed lines in Figure 3 show the potential impact of decarbonizing energy generation for computation. We show the linear regression lines for a 100% solar-powered grid and a 100% nuclear powered grid, with an elasticity of computational demand that is the same as the energy consumption curve in Figure 2 since we assume a constant median carbon intensity for these renewable sources. Clearly changes in energy source can lead to lower carbon footprint. However, to contextualize those improvements, it is worth considering projection into the future assuming both elasticity and exponential energy scaling over time continue to hold. Here, any increases in computational demand will quickly outpace any reduction in carbon intensity of the energy supply. To keep up with this computational demand without increasing total carbon emissions, the carbon intensity of energy generation would need to reduce by approximately 79% *annually* on average for AI, and 31% *annually* for cryptocurrency mining.



(a) AI Model Training

(b) Cryptocurrency Mining

Fig. 3. Estimated total carbon footprint per AI model training and annual cryptocurrency mining in log-log scale. The solid angled green lines are iso-carbon lines, and the dashed green line is the power law of total carbon footprint fitted to the datapoints. *For AI model training, each datapoint indicates carbon footprint per computation of the highest-throughput GPU Hardware introduced in a given year against the total training FLOP for the largest AI model released in the subsequent year. The markers indicate the release year of the AI model. For cryptocurrency mining, each datapoint indicates carbon footprint per computation of the highest-throughput ASIC released in a given year against the total terahashes executed in the subsequent year. The markers indicate the year in which the terahashes were executed.*

Extrapolating median year-on-year increase in computational demand for training AI models (~551% in terms of training FLOP), and assuming consistent decline in the carbon intensity of the USA electricity grid, and the improvements in GPU energy efficiency, as illustrated by the regression line, the carbon footprint of training the largest AI model introduced in 2035 could be 8 million times that of the largest AI model released in 2025 (Figure 3(a)). Similarly, considering that the number of terahashes executed annually has been increasing by 82.7% year-on-year, we project that the carbon footprint of executing all terahashes in 2035 would be 34 times compared to 2025.

The near-constant elasticity on the log-log plots also indicates that either the price per compute is heavily influenced by the price associated with the energy investment (that is, $k_1 \gg k_2$), or that energy intensity is still greater than the ratio $\frac{k_2}{k_1}$ (that is, $\eta > \frac{k_2}{k_1}$). We do not see any evidence in the data that suggests an inflection point or *knee* in the curve between computational demand and energy intensity yet. However, unbounded exponential growth cannot continue indefinitely, leaving open the possibility that computational demand could eventually become inelastic with respect to energy intensity, and rebound effect would no longer exist.

5 Discussion

Continued growth in computational demand is ultimately limited by the scaling rate of physical and economic resources from energy capacity, and chip fabrication output to available capital. Additional investment can shift these constraints outward over time, but cannot sustain exponential growth in demand growth indefinitely against these limits. Policy interventions regulating energy usage and capital investments also cap demand independent of price. Despite continued gains in energy efficiency, computational demand growth would plateau in the light of resource and policy constraints.

The elasticity of the market is jointly governed by computational demand and price per unit of compute, which we model as a function of energy intensity. For bitcoin mining, improvements in energy efficiency of the hashing operation directly lowers price per compute, raising profitability and driving up the compute demand for mining. AI training demand, by contrast, is also shaped by improving model capability, research and investment priorities, and expanding applications. This "utility" can drive AI adoption, and its rebound effects, independent of any gains in energy efficiency.

Though Jevons paradox has prompted considerable discussion in the computing community, a rigorous quantitative framework for analyzing its implications has not been previously considered. This paper takes a first step toward establishing such a foundation. While energy efficiency is, and will remain, an important objective for computer systems developers, if one wishes to make statements about reducing the total carbon footprint from design improvements, the elasticity of the market must be considered. That being said, computation also provides immense utility in people's lives, which we do not attempt to quantify in this work. A broader conversation is imminent in the computing community to evaluate how much does the societal and economic utility of computation compensate for its environmental consequences.

References

[1] Blake Alcott. 2005. Jevons' paradox. *Ecological economics* 54, 1 (2005), 9–21.

[2] Blockchain.com. 2025. *Bitcoin Hash Rate Chart.* https://www.blockchain.com/explorer/charts/hash-rate

[3] Thomas Bruckner, Lew Fulton, Edgar Hertwich, Alan McKinnon, Daniel Perczyk, Joyashree Roy, Roberto Schaeffer, Steffen Schlömer, Ralph Sims, Pete Smith, et al. 2014. Technology-specific cost and performance parameters [annex III]. In *Climate change 2014: mitigation of climate change*. Cambridge University Press, 1329–1356.

[4] Monica Crippa, Diego Guizzardi, Federico Pagani, Manjola Banja, Marilena Muntean, et al. 2025. GHG Emissions of All World Countries: 2025 Report. doi:10.2760/9816914

[5] Lieven Eeckhout. 2024. 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*. 401–415.

[6] Ember, Energy Institute, and Our World in Data. 2024. Carbon Intensity of Electricity Generation âĂŞ Ember and Energy Institute. Dataset. https://ourworldindata.org/grapher/carbon-intensity-electricity Major processing by Our World in Data.

[7] Epoch AI. 2024. Data on Notable AI Models. https://epoch.ai/data/notable-ai-models

[8] Epoch AI. 2025. Data on Large-Scale AI Models. https://epoch.ai/data/large-scale-ai-models

[9] Epoch AI. 2025. Data on Machine Learning Hardware. https://epochai.org/data/machine-learning-hardware

[10] W. Stanley Jevons. 1866. The Coal Question. In *The Economics of Population*. Routledge, 193–204.

[11] Benjamin C Lee, David Brooks, Arthur van Benthem, Mariam Elgamal, Udit Gupta, Gage Hills, Vincent Liu, Linh Thi Xuan Phan, Benjamin Pierce, Christopher Stewart, et al. 2025. A view of the sustainable computing landscape. *Patterns* 6, 7 (2025).

[12] Alexandra Sasha Luccioni, Emma Strubell, and Kate Crawford. 2025. From efficiency gains to rebound effects: The problem of jevons' paradox in AI's polarized environmental debate. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*. 76–88.

[13] Alfred Marshall. 2013. *Principles of economics*. Springer.

[14] ASIC Miner. 2025. *ASIC Miner Efficiency Data.* https://www.asicminervalue.com/efficiency

[15] Jonas Nässén and John Holmberg. 2009. Quantifying the rebound effects of energy efficiency improvements and energy conserving behaviour in Sweden. *Energy efficiency* 2, 3 (2009), 221–231.

[16] David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R So, Maud Texier, and Jeff Dean. 2022. The carbon footprint of machine learning training will plateau, then shrink. *Computer* 55, 7 (2022), 18–28.

[17] Hannah Ritchie, Max Roser, and Pablo Rosado. 2020. CO2 and Greenhouse Gas Emissions. *Our World in Data* (2020). https://ourworldindata.org/co2-and-greenhouse-gas-emissions.

[18] Yifan Sun, Nicolas Bohm Agostini, Shi Dong, and David Kaeli. 2019. Summarizing CPU and GPU design trends with product data. *arXiv preprint arXiv:1911.11313* (2019).

[19] Shafqut Ullah, Tahir Mahmood, and Muhammad Zamir Khan. 2022. Energy efficiency and energy rebound, intensity, and output effects in transport sector of Pakistan. *Environmental Science and Pollution Research* 29, 50 (2022), 75402–75416.