

# CO2-Meter: A Comprehensive Carbon Footprint Estimator for LLMs on Edge Devices

Zhenxiao Fu, Fan Chen, Lei Jiang

Indiana University Bloomington  
{zhfu,fc7,jiang60}@iu.edu

## Abstract

LLMs have transformed NLP, yet deploying them on edge devices poses great carbon challenges. Prior estimators remain incomplete, neglecting peripheral energy use, distinct prefill/decode behaviors, and SoC design complexity. This paper presents *CO2-Meter*, a unified framework for estimating operational and embodied carbon in LLM edge inference. Contributions include: (1) equation-based peripheral energy models and datasets; (2) a GNN-based predictor with phase-specific LLM energy data; (3) a unit-level embodied carbon model for SoC bottleneck analysis; and (4) validation showing superior accuracy over prior methods. Case studies show *CO2-Meter*'s effectiveness in identifying carbon hotspots and guiding sustainable LLM design on edge platforms.

**Code** — <https://github.com/fuzhenxiao/CO2-Meter>

## Introduction

LLMs (Bai et al. 2023) now reach human-level performance on diverse NLP tasks, enabled by large transformers, extensive training, and massive pre-training corpora. While once cloud-only, privacy and QoS concerns (Adekanye 2024) are pushing inference to edge devices, powering applications from autonomous driving (Adekanye 2024) and VR assistants (Min and Jeong 2024) to human-robot interaction (Kim, Lee, and Mutlu 2024) and healthcare robots (Venkataswamy, Janamala, and Cherukuri 2024). This shift could sharply raise emissions: ARM projects 40% annual growth in edge devices through 2035 (Sparks 2017), expanding *operational* footprints from usage and *embodied* footprints from manufacturing (Gupta et al. 2022). LLMs exacerbate both—high inference costs increase operational emissions, while demand for NPUs (Song et al. 2019), GPUs, and memory boosts embodied emissions. Without intervention, edge-device emissions may surpass global data centers by 2028 (Sparks 2017), highlighting the urgency of measuring LLM carbon footprints on edge platforms.

Previous work lacks a comprehensive carbon footprint modeling tool for LLM inference on edge devices, overlooking both operational and embodied carbon emissions:

- *Operational Carbon*: Existing LLM carbon/energy models (Faiz et al. 2024; Fu et al. 2024; Luccioni, Jernite,

and Strubell 2024; Ukarande et al. 2024) often ignore peripheral energy costs—data acquisition (sensors, cameras, mics), transmission (WiFi, Bluetooth), and output (audio, display)—despite their importance for on-device LLMs. Prior work mainly profiles energy for LLM training (Faiz et al. 2024) and inference (Fu et al. 2024; Luccioni, Jernite, and Strubell 2024; Ukarande et al. 2024) in the cloud, or for small CNNs on edge devices (Tu et al. 2024; Kasioulis et al. 2024; Chen et al. 2024). But LLM inference energy on constrained edge platforms is largely unstudied, and CNN-based estimators (Tu et al. 2024) fail to capture the distinct compute and memory demands of LLM prefill and decode phases.

- *Embodied Carbon*: Studies (Chen et al. 2024; Pirson and Bol 2021) show non-computing parts (casings, PCBs, batteries) dominate embodied carbon in low-end IoT (Internet of Things) devices. LLMs, however, demand high-performance NPUs (Ale et al. 2024), GPUs, and large memory, driving emissions higher. Cloud servers estimate embodied carbon by multiplying carbon per unit area by total chip area (CPUs, GPUs, DRAMs) (Gupta et al. 2022; Faiz et al. 2024), but edge devices consolidate units into a single SoC—making chip-level models inadequate for capturing unit-level emissions and pinpointing embodied carbon bottlenecks.

To address the limitations in prior work, this paper presents *CO2-Meter*, a comprehensive model for estimating the end-to-end carbon footprint of deploying LLMs on edge devices. Our contributions can be summarized as:

- *Peripheral Operation Energy Models and Dataset*: We profile operational energy consumption from peripheral operations—data acquisition (e.g., cameras), transmission (e.g., WiFi, Bluetooth), and output (e.g., audio, display)—on edge devices, and construct a dataset. Equation-based models are proposed to estimate the energy of various peripheral operations on edge devices.
- *LLM Inference Energy Prediction and Dataset*: An LLM inference energy dataset is compiled from real-world request traces across multiple devices. A GNN-based predictor is presented to accurately predict operational energy consumption for the prefill and decode phases of an LLM inference under diverse configurations.
- *Unit-Level Embodied Carbon Modeling for SoCs*: We

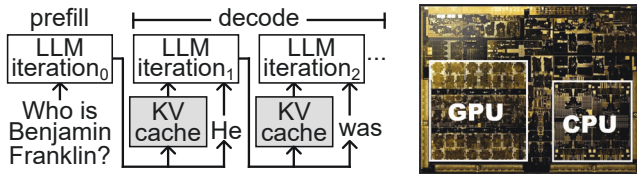


Figure 1: Autoregressive inference. Figure 2: A SoC die.

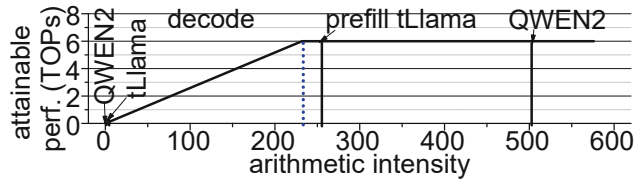
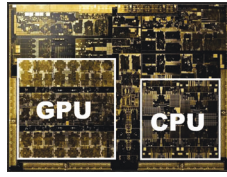


Figure 3: The distinctive hardware characteristics of the prefill and decode phases in an LLM inference on rk3588.

propose a unit-level embodied carbon model to assess the carbon overhead of critical computing units in edge SoCs. This model identifies embodied carbon bottlenecks, supporting efficient design and deployment of edge devices optimized for LLM inferences.

- *Model Validation & Use Case Studies:* Extensive validation demonstrates the accuracy of our models and their superiority over previous approaches. Use case studies highlight CO2-Meter’s ability to pinpoint operational and embodied carbon hotspots, informing sustainable LLM deployment strategies on edge platforms.

## Background and Motivation

### Prior Operational Energy Estimators

**Peripheral Operations on Edge Devices.** Most CNN (Tu et al. 2024; Kasioulis et al. 2024; Chen et al. 2024) and LLM (Fu et al. 2024; Luccioni, Jernite, and Strubell 2024; Ukarande et al. 2024; Faiz et al. 2024) energy estimators overlook peripheral energy use—data collection (sensors, cameras, mics), transmission (WiFi, Bluetooth), and output (audio, displays). Existing WiFi (Sun et al. 2014) and Bluetooth (Negri, Beutel, and Dyer 2006) models target multi-device setups and are overly complex for single-edge devices, while models for key peripherals like cameras, microphones, speakers, and video output remain absent.

**LLM Inference.** LLM training (Faiz et al. 2024) and inference (Fu et al. 2024; Luccioni, Jernite, and Strubell 2024; Ukarande et al. 2024) are well profiled in cloud settings, but edge studies largely focus on CNN latency (Zhang et al. 2021; Liu et al. 2023; Hu et al. 2024; Yi et al. 2023) with limited energy analysis (Tu et al. 2024; Kasioulis et al. 2024; Chen et al. 2024). No prior work models the energy cost of LLM autoregressive inference on edge. CNN-based estimators (Tu et al. 2024; Kasioulis et al. 2024; Chen et al. 2024) treat inference as one phase, missing LLMs’ distinct compute-heavy prefill (parallel token processing) and memory-heavy decode (sequential KV-cache access) phases, as shown in Figures 1 and 3.

## Limitations of Prior Embodied Carbon Models

Unlike in the cloud, where operational emissions dominate (Wu et al. 2022), embodied carbon is often the main contributor on edge devices (Gupta et al. 2022). Studies (Chen et al. 2024; Pirson and Bol 2021) link embodied emissions in low-end IoT devices to non-computing parts (casings, PCBs, batteries), which lack the NPUs/GPUs needed for LLMs (Süzen, Duman, and Şen 2020; Rockchip 2024; NVIDIA 2024). In cloud servers, discrete chips (CPUs, GPUs) are modeled by chip area (Gupta et al. 2022; Faiz et al. 2024), but edge devices merge CPUs, GPUs, and NPUs into a single SoC (Figure 2). Chip-level models miss these SoC designs, limiting embodied carbon analysis for LLM inference on edge.

## Comparison with Prior Work

Most studies address LLM carbon footprints in the cloud (Faiz et al. 2024; Fu et al. 2024; Luccioni, Jernite, and Strubell 2024; Ukarande et al. 2024), while edge research largely targets CNN/vision transformer latency (Zhang et al. 2021; Liu et al. 2023; Hu et al. 2024; Yi et al. 2023) with limited energy focus (Tu et al. 2024; Kasioulis et al. 2024; Chen et al. 2024). No work models LLM inference energy on edge or separates prefill and decode phases. Embodied carbon studies (Pirson and Bol 2021) mostly cover non-computing parts in low-end IoT. CO2-Meter fills these gaps by jointly modeling operational and embodied carbon for autoregressive LLM inference, capturing core and peripheral operations, phase-specific energy, and unit-level SoC emissions.

## CO2-Meter

To estimate the end-to-end carbon footprint of LLM inference on edge devices, CO2-Meter separates the analysis into operational and embodied carbon modeling. Operational carbon is computed by estimating the energy consumption of both peripheral operations and LLM inferences, scaled by the carbon intensity of the edge device location ( $\text{kgCO}_2/\text{kWh}$ ) (Gupta et al. 2022). The energy consumption of various peripheral operations such as data sensing, transmission, and output is modeled by equation-based approaches, while the energy consumption of LLM inferences is estimated using a GNN model. Embodied carbon is quantified by modeling the contributions of individual computing units within integrated edge SoCs.

## Operational Energy Modeling

**Peripheral Operation** This section models the operational energy and power consumption of peripheral operations. For certain operations, only power models are provided, with energy consumption computed as the product of power and execution time.

- *Networking (WiFi/Bluetooth):* Simplifying existing complex power models for WiFi (Sun et al. 2014) and Bluetooth (Negri, Beutel, and Dyer 2006), we propose a straightforward energy model:

$$E_{net} = P_{static} \cdot t + E_{bit} \cdot S_{data}, \quad (1)$$

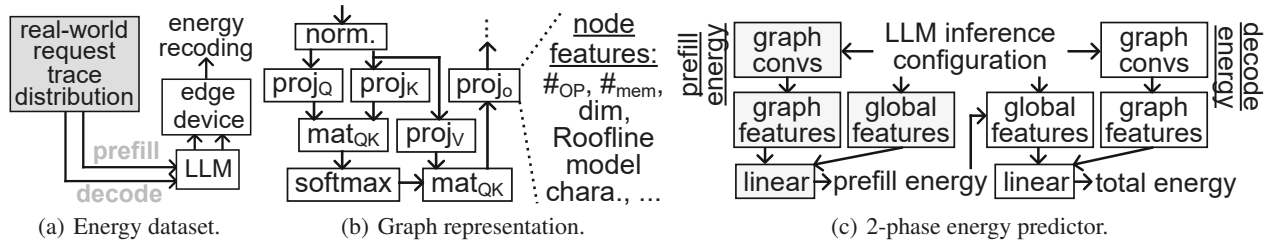


Figure 4: The LLM inference operational energy predictor of CO2-Meter.

where  $E_{net}$  is the network energy,  $P_{static}$  represents the static power of the network interface (including chips and antenna),  $t$  is the transmission time,  $E_{bit}$  denotes the energy per bit of data transfer, and  $S_{data}$  is the transferred data size.

- *Camera*: The energy consumption for camera operations is modeled as:

$$E_{cam} = P_{static} \cdot t + E_{frame} \cdot num_{frame}, \quad (2)$$

where  $E_{cam}$  is the total energy for camera usage,  $P_{static}$  denotes the static power of the camera interface (including image-capturing chips and lens),  $t$  is the duration of camera usage,  $E_{frame}$  indicates the energy per frame, and  $num_{frame}$  is the total number of frames captured.

- *Microphone*: The energy consumption of microphone operations is modeled as:

$$E_{mic} = P_{static} \cdot t + E_{sample} \cdot num_{sample}, \quad (3)$$

where  $E_{mic}$  is the total microphone energy,  $P_{static}$  is the static power (e.g., analog-to-digital converters (Kim 2022) and supporting circuits),  $t$  is the microphone usage duration,  $E_{sample}$  is the energy per sample, and  $num_{sample}$  is the total samples captured.

- *Video Output*: Modern edge SoCs utilize multimedia units to generate HDMI output signals. The power consumption of the multimedia unit during HDMI signal generation is modeled as:

$$P_{video} = P_{static} + P_p \cdot N_p, \quad (4)$$

where  $P_{video}$  represents the video signal generation power,  $P_{static}$  is the static power consumed by the multimedia unit,  $P_p$  is the power required per pixel, and  $N_p$  is the number of pixels.

- *Speaker*: The speaker power consumption is primarily used for membrane vibrations and modeled as:

$$P_{spk} = \frac{1}{1 + \exp(\alpha \cdot V) + \beta}, \quad (5)$$

where  $P_{spk}$  is the total speaker power,  $\alpha$  and  $\beta$  are two fitting parameters, and  $V$  is the sound volume.

- *Display*: We used a TFT Liquid Crystal Display (LCD) as our monitor. We adopted a TFT LCD power model from (Cheng and Pedram 2004):

$$P_{TFT} = a + b \cdot x + c \cdot x^2 \quad (6)$$

where  $P_{TFT}$  is the LCD power consumption,  $x$  is the pixel grey value  $\in [0, 255]$ , and  $a$ - $c$  are fitting parameters.

- *Image/Voice-to-Text Conversions*: In certain LLM applications, such as in-home healthcare systems (Venkataswamy, Janamala, and Cherukuri 2024), inputs and outputs may include images or voice instead of text. Image-to-text conversion is performed using OpenOCR (Du et al. 2024). Voice-to-text conversion is handled by RealtimeSTT (Beigel 2024), while text-to-voice conversion is done via TTS (coqui.ai 2024). The inference energy consumption of OpenOCR, RealtimeSTT, and TTS is estimated using a prior energy predictor (Tu et al. 2024) for small-scale CNNs.

- *System Background*: System background energy is calculated as the product of the SoC idle power and the application’s execution time.

**LLM Autoregressive Inference** To estimate LLM inference energy on edge devices, CO2-Meter employs a GNN model trained on a real-world LLM inference energy dataset. Given an LLM configuration and target edge device, the GNN predicts inference energy for unseen requests. Addressing limitations in prior CNN-based estimators (Tu et al. 2024; Kasioulis et al. 2024; Chen et al. 2024), our approach introduces the following innovations. We choose GNNs because their flexibility ensures that future variants of LLM architectures, each with potentially different graph structures, can all be accommodated by the same model.

- *Energy Dataset from Real-World Traces*: Using LLM inference traces from Azure Cloud (Microsoft 2024) (Figure 4(a)), we build an energy dataset assuming user behavior is consistent across cloud and edge, differing only in execution site. Prefill and decode are treated separately: prefill measures energy for a sampled prompt plus one token, while decode records energy for the same prompt with a sampled output length.

- *Graph Representation*: As shown in Figure 4(b), our GNN models each transformer layer as a graph, with nodes as computational kernels and edges as data dependencies. Node features comprehensively encode kernel-level metrics. Aside from those included in Figure 4(b), arithmetic intensity, weight loads, activation loads/stores, KV cache loads/stores, and per-kernel inference time are also included. Edges capture the data flow between these kernels. The resulting graph is encoded using GNN layers such as GraphSAGE (Hamilton, Ying, and Leskovec 2017), graph attention networks (Velickovic et al. 2018), and graph isomorphism networks (Xu et al. 2019).



Figure 5: The power measurement infrastructure.

SoC	Hardware Configuration
Rockchip rk3588	CPU: 4 Cortex-A76 & 4 A55; NPU: 6-TOPS Ethos NPU; DRAM: 51.2GB/s 8GB 64-bit LPDDR5
Rockchip rk3568	CPU: 4 Cortex-A55; NPU: 1-TOPS Ethos NPU; DRAM: 34.1GB/s 8GB 32-bit LPDDR4
NVIDIA AGX Orin	CPU: 12 Cortex-A78 v8.2; GPU: 275-TOPS CUDA cores; DRAM: 204.8GB/s 32GB 256-bit LPDDR5
NVIDIA Orin NX	CPU: 8 Cortex-A78 v8.2; GPU: 157-TOPS CUDA cores; DRAM: 102.4GB/s 16GB 128-bit LPDDR5

Table 1: The configuration of edge devices.

- *Two-phase prediction*: For prefill energy (Figure 4(c)), the GNN extracts graph features from the LLM request, deriving global stats (op count, layer count, dimensions, memory, ...). These combine through a linear layer to estimate prefill energy. For total energy (prefill + decode), another GNN/linear layer pair is used, incorporating prefill energy into the global features.

### Embodied Carbon Footprint Modeling

To compute the embodied carbon of each computing unit (e.g., CPU, GPU, or NPU) within an edge SoC, we propose a unit-level embodied carbon model:

$$\text{embodied\_carbon}_{SoC} = \sum_{i=1}^n \text{area}_i \cdot CPA, \quad (7)$$

where  $\text{area}_i$  denotes the area of unit  $i$ ,  $CPA$  is the carbon emission per unit area (ReCollect 2024), and  $n$  is the number of computing units. This model enables identification of the dominant contributors to the SoC total embodied carbon.

### Experimental Methodology

**Energy Measurement.** We develop a methodology for measuring the operational energy of peripherals and inference on edge devices (Figure 5). Inputs from WiFi, Bluetooth, cameras, or microphones are processed by the CPU, multimedia unit, NPU, or GPU before inference on the NPU/GPU, with outputs directed to displays or speakers. Energy consumption is recorded using an ODR0ID SmartPower3 meter (ODR0ID 2024) at 200 Hz. To ensure consistency and reliability across LLMs and devices, we follow:

- *Operation-Specific Measurement*: Energy is computed as the difference in power between active and idle states for each operation, with all other conditions held constant.
- *Environmental Control*: Measurements are conducted at 25 °C with a 10s cooldown between tests.

**LLMs.** We adopted the LLM-based virtual reality assistant application (Min and Jeong 2024) to generate Q&A inference requests using selected lightweight LLMs, suitable for edge deployment due to resource constraints. 4

models ranging from 0.5 to 1.8 billion parameters were used: internlm2-chat-1.8b (INT) (Cai et al. 2024), qwen1.5-0.5b (Q1.5) (Bai et al. 2023), tinyllama-1.1b-chat-v1.0 (LAM) (Zhang et al. 2024), and qwen2-1.5b (Q2) (Bai et al. 2023). Q2 was used specifically to evaluate CO2-Meter’s generalization to unseen LLM configurations. Prompt and output token length distributions were assumed to follow cloud-based inference patterns (Microsoft 2024), with all requests executed at batch size 1.

**Edge and Peripheral Devices.** We evaluated Rockchip- and NVIDIA-based edge devices (Table 1). Rockchip devices use an NPU for LLM inference, while NVIDIA devices rely on a GPU. CO2-Meter was tested mainly on Rockchip rk3588 (rk) (Rockchip 2024) and NVIDIA AGX Orin (orin) (NVIDIA 2024), with rk3568 and Orin NX used to test generalization. Peripherals included a HAMTYSAN 7-inch 800 × 480 LCD, Manhattan 2600 speakers, and a Logitech QuickCam Pro 9000 webcam.

**Energy Dataset.** We collected energy data for ~ 200 peripheral configurations to fit and validate equation-based energy models. For CO2-Meter’s GNN-based LLM inference energy predictor, we built a dataset of 40K measurements spanning LLMs (INT, Q1.5, LAM), request parameters, and SoCs (rk, orin), split into 32K/4K/4K for training/validation/testing. An extra 8K samples for Q2 on rk3568 and Orin NX evaluated generalization to unseen LLMs and hardware.

**Schemes.** To evaluate the accuracy of our equation-based peripheral energy models, we compared their predictions against real-world measurements. To assess the effectiveness of our GNN-based LLM inference energy predictor, we compared it against the following baselines:

- *RF*: A random forest model (Zhang et al. 2021) trained on global features such as total operations, transformer layer count, layer dimensions, and memory footprint.
- *NNLQP*: A GNN-based predictor (Liu et al. 2023) using 2 GIN (Xu et al. 2019) layers, and trained on total inference energy without phase separation.
- *2P*: The same as *NNLQP*, except it is trained on a dataset separating LLM inference into prefill and decode phases.
- *CO2-Meter*: The same as *2P*, except it uses the GNN shown in Figure 4(c).

**Setup.** All neural networks were implemented in PyTorch and trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 32. Experiments were conducted on an NVIDIA A100 GPU.

### Validation

#### Peripheral Operation Energy Validation

**WiFi & Bluetooth.** We evaluated the download and upload energy models on rk3588 (rk) and AGX Orin (orin), as shown in Figures 6 and 7. Equation 1 closely matches real-world measurements. For WiFi, energy per bit decreases exponentially with increasing bandwidth, as static power is amortized over a larger data volume. Orin, with more efficient antennas and support circuits, consistently outperforms rk in energy efficiency. The model achieves mean absolute errors of  $2.72 \times 10^{-9}$  J (rk) and  $4.02 \times 10^{-8}$  J (orin) for

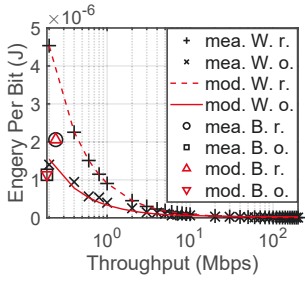


Figure 6: Download validation (W: WIFI; B: Bluetooth; o: Orin; r: rk3588; mea: measured; mod: model).

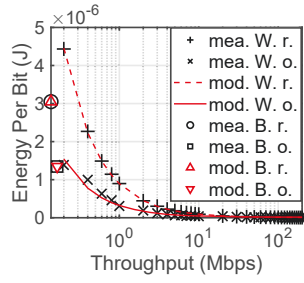


Figure 7: Upload validation (W: WIFI; B: Bluetooth; o: Orin; r: rk3588; mea: measured; mod: model).

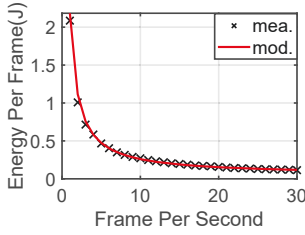


Figure 8: Camera validation (mea: measured; mod: model).

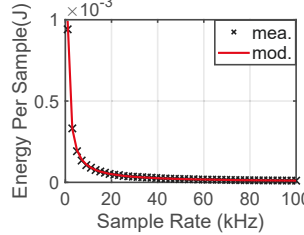


Figure 9: Microphone validation (mea: measured; mod: model).

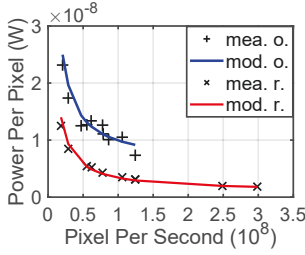


Figure 10: Video output val. (mea: measured; mod: model; o: Orin; r: rk3588).

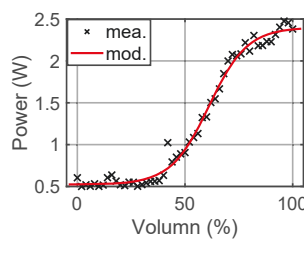


Figure 11: Speaker validation (mea: measured; mod: model).

download, and  $6.04 \times 10^{-9}$  J (rk) and  $3.42 \times 10^{-8}$  J (orin) for upload, across bandwidths from 0 to 200 Mbps. For Bluetooth, which operates at a fixed bandwidth, the model yields mean absolute errors of  $2.71/2.32 \times 10^{-8}$  J (rk) and  $4.26/4.85 \times 10^{-8}$  J (orin) for download/upload, respectively.

**Camera & Microphone.** Figures 8 and 9 validate our camera and microphone energy models. For the camera, energy per frame decreases with higher frame rates as static power is amortized across more frames. Equation 2 yields a mean absolute error of  $1.18 \times 10^{-2}$  J. Similarly, the microphone model shows reduced energy per sample at higher sampling rates, with a mean absolute error of  $1.80 \times 10^{-6}$  J. These results demonstrate strong consistency with real-world measurements.

**Video Output, Speaker, and Display.** We validated the power models for video output, speaker, and display, as

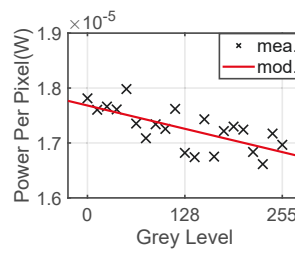


Figure 12: TFT LCD val. (mea: measured; and mod: model).

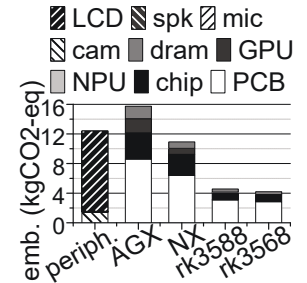


Figure 13: Embod. carbon of edge & periph. devices.

scheme	rk3588 (%)				AGX Orin (%)			
	INT	Q1.5	LAM	avg	INT	Q1.5	LAM	avg
RF	98.1	158.2	63.6	106.6	49.1	68.8	69.1	62.3
NNLQP	23.4	26.4	49.4	33.1	29.8	29.7	24.9	28.1
2P	11.9	11.4	12.3	11.8	16.3	22	17.9	18.7
CO2-Meter	10.8	10.1	10	<b>10.3</b>	15.2	20.3	18.5	<b>18</b>

Table 2: The MAPE comparison.

shown in Figures 10, 11, and 12, respectively. For video output, rk3588 (rk) consumes less power than AGX Orin (orin) at the same resolution, due to its energy efficient multimedia unit. Power per pixel decreases with increasing pixel rate due to amortization of static power. The video output model (Equation 4) achieves mean absolute errors of  $3.69 \times 10^{-10}$  W (rk) and  $1.21 \times 10^{-9}$  W (orin). Speaker power increases nonlinearly with volume, and the speaker model (Equation 5) yields a mean absolute error of  $5.39 \times 10^{-2}$  W. The TFT LCD power model, fitted using three parameters, shows power reduction with increasing pixel gray level and achieves a mean absolute error of  $4.71 \times 10^{-7}$  W.

**Image/Voice-to-Text Conversions.** Following the methodology in (Tu et al. 2024), we collected 14K energy samples for kernels in OpenOCR, RealtimeSTT, and TTS, partitioned into 10K for training, 1K for validation, and 3K for testing. A GNN-based energy predictor (Tu et al. 2024) was trained to model the energy consumption of image/voice-to-text and text-to-voice conversion tasks. On the test set, the predictor achieved 82% accuracy within a 10% deviation from actual values, consistent with previously reported results.

**Background Energy.** The model for background energy closely aligns with measured data, yielding negligible error.

### LLM Inference Energy Validation

**Seen Configuration.** We trained and evaluated the LLM inference energy predictor using data from LLMs (INT, Q1.5, LAM) on rk3588 and AGX Orin. Accuracy was measured by mean absolute percentage error (MAPE) and the share of predictions within 10% error bounds (10% EB)(Tables 2,3). A  $B\%$  at an  $N\%$  bound means  $B\%$  of predictions are within  $N\%$  of ground truth. Results are stronger on rk3588 due to higher absolute energy values, which reduce relative error risk. While RF scores well under error bounds, its MAPE is high from large outliers. Phase-specific modeling in 2P

scheme	rk3588 (%)				AGX Orin (%)			
	INT	Q1.5	LAM	avg	INT	Q1.5	LAM	avg
RF	40.7	46.3	57.9	48.3	25.1	27	22	24.7
NNLQP	34.2	42.5	15.2	30.6	11.8	21.3	15.3	16.1
2P	67.9	69.2	67.4	68.2	48.3	29.5	45.1	41
CO2-Meter	65.8	72.5	72.7	<b>70.3</b>	55.9	40.1	49.9	<b>48.6</b>

Table 3: The 10% EB comparison.

LLM	rk3568 (%)			Orin NX (%)		
	RF	NNLQP	CO2-Meter	RF	NNLQP	CO2-Meter
Q2	28.5	30.1	<b>69.2</b>	17.5	25.7	<b>45.1</b>

Table 4: The 10% EB comparison on unseen configurations.

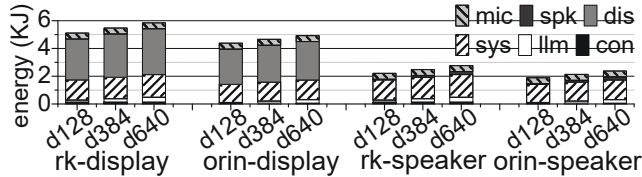


Figure 14: The operational energy of the LLM-based virtual reality assistant with a microphone input ( $dn$ : generated token length  $n$ ; mic: microphone; spk: speaker; dis: display; sys: system background; llm: LLM; con: conversion).

boosts 10% EB accuracy by 123–155% over NNLQP. Our GNN further improves these bounds by 3% on rk3588 and 19% on AGX Orin, underscoring CO2-Meter’s advantage on high-end SoCs.

**Unseen Configuration.** To assess generalization, we evaluated all schemes using 8K samples from Q2 inferences on rk3568 and AGX Orin—configurations not seen during training. Table 4 reports the 10% EB comparison across methods. Accuracy drops significantly for RF under unseen LLM and SoC settings. Compared to NNLQP, CO2-Meter improves 10% EB accuracy by 129% on rk3568 and 75% on AGX Orin, demonstrating its superior generalization to previously unseen configurations.

### Embodied Carbon Validation and Calculation

Figure 13 compares the embodied carbon of SoCs and peripherals, reported in CO<sub>2</sub>-eq to standardize greenhouse gas impacts. Our unit-level model (Equation 7) shows 10%–20% deviation from reported values, aligning with prior work (Gupta et al. 2022; Faiz et al. 2024). By modeling unit-level contributions without altering total chip area, it preserves chip-level accuracy. The detailed breakdowns:

- *Peripheral Devices.* Embodied carbon estimates include 1.43 kgCO<sub>2</sub>-eq for the camera, 0.04 for the microphone, and 0.08 for the speaker (Pirson and Bol 2021). The 7-inch TFT LCD dominates peripherals at 10.85 kgCO<sub>2</sub>-eq (Dell 2013).
- *Rockchip Edge SoCs.* For rk3588, the PCB (43.5 cm<sup>2</sup>, CPA: 0.071 kgCO<sub>2</sub>-eq/cm<sup>2</sup> (ReCollect 2024)) contributes 3.08 kgCO<sub>2</sub>-eq, and the 8 nm SoC (89 mm<sup>2</sup>, CPA: 1.2 kgCO<sub>2</sub>-eq/cm<sup>2</sup>) adds 1.07 kgCO<sub>2</sub>-eq, with its

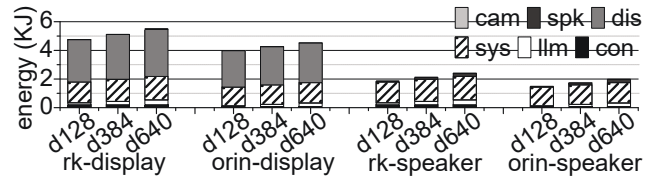


Figure 15: The operational energy of the LLM-based virtual reality assistant with a camera input ( $dn$ : generated token length  $n$ ; cam: camera; spk: speaker; dis: display; sys: system background; llm: LLM; con: conversion).

NPU (5% area) at 0.053 kgCO<sub>2</sub>-eq. LPDDR5 DRAM adds 0.42 kgCO<sub>2</sub>-eq (Gupta et al. 2022; Jones 2023), with 10.4% of rk3588’s embodied carbon tied to LLM inference. As a comparison, The PCB, SoC, NPU, and DRAM of rk3568 (fewer CPU cores and a lower-throughput NPU) contribute 2.84, 0.94, 0.03, and 0.38 kgCO<sub>2</sub>-eq, respectively (9.9% for LLM inference).

- *NVIDIA Edge SoCs.* For AGX Orin, the PCB (121 cm<sup>2</sup>) and SoC (455 mm<sup>2</sup>) contribute 8.6 and 5.46 kgCO<sub>2</sub>-eq, with its GPU (35% area) at 1.91 kgCO<sub>2</sub>-eq and LPDDR5 DRAM at 1.68 kgCO<sub>2</sub>-eq; 22.8% of its embodied carbon supports LLM inference. AGX Orin’s footprint is 3.44× rk3588’s. As a comparison, for Orin NX (fewer CPU cores, a lower-throughput GPU and a narrower-bandwidth DRAM), PCB, SoC, GPU, and DRAM add 6.4, 2.8, 0.81, and 0.88 kgCO<sub>2</sub>-eq, with 15.4% linked to LLM inference.

### Use Cases

We showcase CO2-Meter through three use cases: (1) analyzing operational energy of an LLM-based edge application, (2) examining operational vs. embodied carbon trade-offs, and (3) assessing LLM inference performance against embodied carbon across edge platforms. These cases highlight CO2-Meter’s role in quantifying and optimizing LLM deployment impacts, focusing on flagship SoCs—NVIDIA AGX Orin and Rockchip rk3588.

### Operational Energy Analysis

Considering the suitable application scenarios of an edge device, we evaluated the energy use of an LLM-based virtual reality assistant (Min and Jeong 2024) that processes 1.5K-token questions via camera or microphone, converts inputs with image-to-text or voice-to-text, runs inference with Q1.5, and outputs responses through a display or speaker. With microphone input, energy consumption scales with output length (128–640 tokens, Fig. 14). The TFT LCD dominates (>55% of total energy), and replacing it with speakers cuts energy usage by over 50%. Background energy from idle components (e.g., CPU) is the next largest share, suggesting low-power states could save more. LLM inference accounts for only 2–12% of total energy, so faster NPUs/GPUs can improve efficiency. Despite higher power draw, AGX Orin uses 14–15% less energy than rk3588 for the same task at all output lengths.

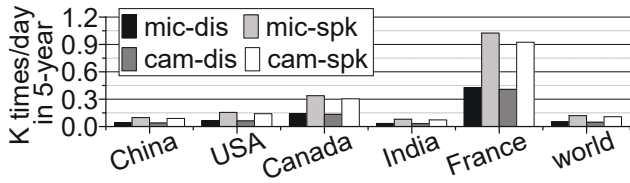


Figure 16: The number of requests for the virtual reality assistant making operational and embodied carbon equal (mic: microphone; dis: display; cam: camera; spk: speaker).

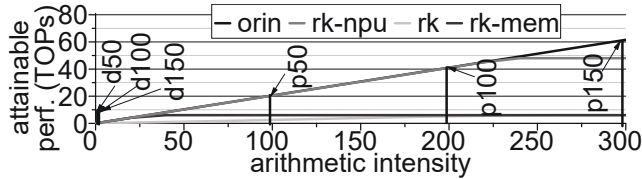


Figure 17: The Roofline model of Q1.5 inference (perf.: performance; d: decode; p: prefill; the numbers after *d* and *p* are prompt lengths; the generated token number is 1; rk: rk3588; orin: AGX Orin; mem: memory; npu: NPU).

Switching to camera input (Fig. 15) cuts the virtual reality assistant’s total energy use by 7–21%, since the camera captures just 2–3 frames instead of recording 7 minutes of audio. With camera input, AGX Orin uses 17–20% less energy than rk3588 across all answer lengths.

### Operational and Embodied Emissions

AGX Orin exhibits higher embodied carbon but lower operational energy consumption than rk3588, raising the question of how long its energy savings must persist to offset embodied emissions. Unlike data centers strategically placed in low-carbon regions, edge devices depend on the carbon intensity of local electricity grids, which varies widely across countries (Ritchie and Rosado 2020)—from below 0.1 kgCO<sub>2</sub>/kWh in France to about 0.7 kgCO<sub>2</sub>/kWh in India, with a global average near 0.48 kgCO<sub>2</sub>/kWh.

Figure 16 shows how often the LLM-based virtual reality assistant must be used over a 5-year lifespan for AGX Orin’s operational savings to offset its higher embodied carbon. Low-carbon regions (e.g., France) require more daily invocations to reach this break-even point. Among input–output settings, the camera–speaker (cam–spk) mode delivers the least operational carbon savings, demanding the highest usage frequency to offset AGX Orin’s embodied carbon versus rk3588.

### Performance and Embodied Carbon

Given that embodied carbon often dominates total emissions in edge SoCs, LLM performance upgrades must be weighed against their embodied impact. Key findings include:

- **Decode Phase Bottleneck:** Edge SoCs like rk3588 and AGX Orin are not optimized for the decode phase of LLM inference due to limited LPDDR bandwidth. As shown in Figure 17, Q1.5’s decode phase sustains low arithmetic intensity ( $\sim 2$ ) across prompt lengths

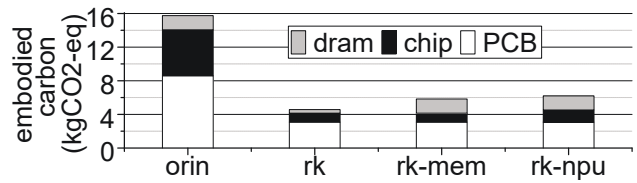


Figure 18: Embodied carbon comparison between various hardware configurations (rk: rk3588; orin: AGX Orin; mem: memory; npu: NPU).

(d50–d150), indicating a memory-bound regime. While server GPUs use HBM to overcome this, such solutions are impractical for edge devices with tight power budgets ( $\leq 20$  W), and viable decode-optimized hardware remains unclear.

- **Prefill Gains from LPDDR:** Boosting LPDDR bandwidth improves prefill performance. Replacing rk3588’s DRAM with AGX Orin’s LPDDR5 (rk-mem) yields a  $2.3\times$  speedup for a 50-token prompt (Figure 17) with only a 27.5% increase in embodied carbon (Figure 18).
- **Enhancing Low-End Devices:** On rk3588, both faster memory and more compute are needed. Scaling the NPU  $8\times$  and adopting LPDDR5 (rk-npu) delivers  $6.8\times$  and  $8\times$  speedups for 100- and 150-token prompts (Figure 17), at a 35.7% embodied carbon cost (Figure 18).

## Conclusion

This work introduces *CO2-Meter*, a unified framework for quantifying the end-to-end carbon footprint of LLM inference on edge devices, encompassing both operational and embodied emissions. By modeling peripheral energy via equations, capturing phase-specific inference energy with a GNN-based predictor, and formulating a unit-level embodied carbon model for SoCs, *CO2-Meter* bridges key gaps in existing estimators. Validation confirms its accuracy, and case studies highlight its utility in identifying carbon bottlenecks. *CO2-Meter* enables precise carbon assessment for sustainable LLM deployment, advancing greener hardware–software co-design, carbon-aware policy, and accountability in AI’s environmental impact.

## Acknowledgments

This work was supported in part by NSF CCF-2105972, OAC-2417589, and CAREER AWARD CNS-2143120.

## References

- Adekanye, O. A. M. 2024. LLM-Powered Synthetic Environments for Self-Driving Scenarios. *AAAI Conference on Artificial Intelligence*.
- Ale, L.; Zhang, N.; King, S. A.; and Chen, D. 2024. Empowering generative AI through mobile edge computing. *Nature Reviews Electrical Engineering*, 0: 1–9.
- Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; Hui, B.; Ji, L.; Li, M.; Lin, J.; Lin, R.; Liu, D.; Liu, G.; Lu, C.; Lu, K.; Ma, J.; Men,

- R.; Ren, X.; Ren, X.; Tan, C.; Tan, S.; Tu, J.; Wang, P.; Wang, S.; Wang, W.; Wu, S.; Xu, B.; Xu, J.; Yang, A.; Yang, H.; Yang, J.; Yang, S.; Yao, Y.; Yu, B.; Yuan, H.; Yuan, Z.; Zhang, J.; Zhang, X.; Zhang, Y.; Zhang, Z.; Zhou, C.; Zhou, J.; Zhou, X.; and Zhu, T. 2023. Qwen Technical Report. arXiv:2309.16609.
- Beigel, K. 2024. RealtimeSTT. <https://github.com/KoljaB/RealtimeSTT>.
- Cai, Z.; Cao, M.; Chen, H.; Chen, K.; Chen, K.; Chen, X.; Chen, X.; Chen, Z.; Chen, Z.; Chu, P.; Dong, X.; Duan, H.; Fan, Q.; Fei, Z.; Gao, Y.; Ge, J.; Gu, C.; Gu, Y.; Gui, T.; Guo, A.; Guo, Q.; He, C.; Hu, Y.; Huang, T.; Jiang, T.; Jiao, P.; Jin, Z.; Lei, Z.; Li, J.; Li, J.; Li, L.; Li, S.; Li, W.; Li, Y.; Liu, H.; Liu, J.; Hong, J.; Liu, K.; Liu, K.; Liu, X.; Lv, C.; Lv, H.; Lv, K.; Ma, L.; Ma, R.; Ma, Z.; Ning, W.; Ouyang, L.; Qiu, J.; Qu, Y.; Shang, F.; Shao, Y.; Song, D.; Song, Z.; Sui, Z.; Sun, P.; Sun, Y.; Tang, H.; Wang, B.; Wang, G.; Wang, J.; Wang, J.; Wang, R.; Wang, Y.; Wang, Z.; Wei, X.; Weng, Q.; Wu, F.; Xiong, Y.; Xu, C.; Xu, R.; Yan, H.; Yan, Y.; Yang, X.; Ye, H.; Ying, H.; Yu, J.; Yu, J.; Zang, Y.; Zhang, C.; Zhang, L.; Zhang, P.; Zhang, P.; Zhang, R.; Zhang, S.; Zhang, S.; Zhang, W.; Zhang, W.; Zhang, X.; Zhang, X.; Zhao, H.; Zhao, Q.; Zhao, X.; Zhou, F.; Zhou, Z.; Zhuo, J.; Zou, Y.; Qiu, X.; Qiao, Y.; and Lin, D. 2024. InternLM2 Technical Report. arXiv:2403.17297.
- Chen, F.; Attari, S.; Buck, G.; and Jiang, L. 2024. IoTCO2: Assessing the End-To-End Carbon Footprint of Internet-of-Things-Enabled Deep Learning. arXiv:2403.10984.
- Cheng, W.-C.; and Pedram, M. 2004. Power minimization in a backlit TFT-LCD display by concurrent brightness and contrast scaling. *IEEE Transactions on Consumer Electronics*.
- coqui.ai. 2024. a deep learning toolkit for Text-to-Speech, battle-tested in research and production. <https://github.com/coqui-ai/TTS>.
- Dell. 2013. Carbon Footprint of a Typical 19" Business Monitor. [https://i.dell.com/sites/csdocuments/Corporate\\_corp-Comm\\_Documents/en/display-white-paper.pdf](https://i.dell.com/sites/csdocuments/Corporate_corp-Comm_Documents/en/display-white-paper.pdf).
- Du, Y.; Chen, Z.; Xie, H.; Jia, C.; and Jiang, Y.-G. 2024. SVTRv2: CTC Beats Encoder-Decoder Models in Scene Text Recognition. *CoRR*, abs/2411.15858.
- Faiz, A.; Kaneda, S.; Wang, R.; Osi, R. C.; Sharma, P.; Chen, F.; and Jiang, L. 2024. LLMCarbon: Modeling the End-to-End Carbon Footprint of Large Language Models. In *The Twelfth International Conference on Learning Representations*.
- Fu, Z.; Chen, F.; Zhou, S.; Li, H.; and Jiang, L. 2024. LLMCO2: Advancing Accurate Carbon Footprint Prediction for LLM Inferences. arXiv:2410.02950.
- Gupta, U.; Elgamal, M.; Hills, G.; Wei, G.-Y.; Lee, H.-H. S.; Brooks, D.; and Wu, C.-J. 2022. ACT: designing sustainable computer systems with an architectural carbon modeling tool. In *IEEE/ACM International Symposium on Computer Architecture*.
- Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*.
- Hu, H.; Su, J.; Zhao, J.; Peng, Y.; Zhu, Y.; Lin, H.; and Wu, C. 2024. CDMPP: A Device-Model Agnostic Framework for Latency Prediction of Tensor Programs. In *European Conference on Computer Systems*.
- Jones, S. W. 2023. Modeling 300mm Wafer Fab Carbon Emissions. In *International Electron Devices Meeting*.
- Kasioulis, M.; Symeonides, M.; Ioannou, G.; Pallis, G.; and Dikaiakos, M. D. 2024. Energy modeling of inference workloads with AI accelerators at the Edge: A benchmarking study. In *IEEE International Conference on Cloud Engineering*.
- Kim, C. Y.; Lee, C. P.; and Mutlu, B. 2024. Understanding large-language model (llm)-powered human-robot interaction. In *ACM/IEEE International Conference on Human-Robot Interaction*.
- Kim, J. P. 2022. Sound Activity Monitor Circuit for Low Power Consumption of Always-On Microphone Applications. *Applied Sciences*.
- Liu, L.; Shen, M.; Gong, R.; Yu, F.; and Yang, H. 2023.>NNLQP: A Multi-Platform Neural Network Latency Query and Prediction System with An Evolving Database. In *International Conference on Parallel Processing*.
- Luccioni, S.; Jernite, Y.; and Strubell, E. 2024. Power hungry processing: Watts driving the cost of AI deployment? In *ACM Conference on Fairness, Accountability, and Transparency*.
- Microsoft. 2024. Azure Public Dataset. <https://github.com/Azure/AzurePublicDataset/>.
- Min, Y.; and Jeong, J.-W. 2024. Public Speaking Q&A Practice with LLM-Generated Personas in Virtual Reality. In *IEEE International Symposium on Mixed and Augmented Reality Adjunct*.
- Negri, L.; Beutel, J.; and Dyer, M. 2006. The power consumption of Bluetooth scatternets. In *IEEE Consumer Communications and Networking Conference*, volume 1.
- NVIDIA. 2024. Jetson Modules.
- ODROID. 2024. SmartPower3. [https://wiki.odroid.com/accessory/power\\_supply\\_battery/smartpower3](https://wiki.odroid.com/accessory/power_supply_battery/smartpower3).
- Pirson, T.; and Bol, D. 2021. Assessing the embodied carbon footprint of IoT edge devices with a bottom-up life-cycle approach. *Journal of Cleaner Production*, 322: 128966.
- ReCollect. 2024. Efficient Manufacturing of Recyclable Composite Laminates for Electrical Goods.
- Ritchie, H.; and Rosado, P. 2020. Electricity Mix. *Our World in Data*. <https://ourworldindata.org/electricity-mix>.
- Rockchip. 2024. RK3588.
- Song, J.; Cho, Y.; Park, J.-S.; Jang, J.-W.; Lee, S.; Song, J.-H.; Lee, J.-G.; and Kang, I. 2019. An 11.5 TOPS/W 1024-MAC butterfly structure dual-core sparsity-aware neural processing unit in 8nm flagship mobile SoC. In *IEEE international solid-state circuits conference*.
- Sparks, P. 2017. The route to a trillion devices, ARM White Paper.

Sun, L.; Sheshadri, R. K.; Zheng, W.; and Koutsonikolas, D. 2014. Modeling WiFi Active Power/Energy Consumption in Smartphones. In *IEEE International Conference on Distributed Computing Systems*.

Süzen, A. A.; Duman, B.; and Şen, B. 2020. Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN. In *IEEE International Congress on Human-Computer Interaction, Optimization and Robotic Applications*.

Tu, X.; Mallik, A.; Chen, D.; Han, K.; Altintas, O.; Wang, H.; and Xie, J. 2024. Unveiling Energy Efficiency in Deep Learning: Measurement, Prediction, and Scoring across Edge Devices. In *ACM/IEEE Symposium on Edge Computing*.

Ukarande, A.; Basaklar, T.; Cao, M.; and Ogras, U. 2024. PACT: Accurate Power Analysis and Carbon Emission Tracking for Sustainability. In *ACM/IEEE International Symposium on Low Power Electronics and Design*.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Venkataswamy, R.; Janamala, V.; and Cherukuri, R. C. 2024. Realization of humanoid doctor and real-time diagnostics of disease using internet of things, edge impulse platform, and ChatGPT. *Annals of Biomedical Engineering*.

Wu, C.-J.; Raghavendra, R.; Gupta, U.; Acun, B.; Ardalani, N.; Maeng, K.; Chang, G.; Aga, F.; Huang, J.; Bai, C.; et al. 2022. Sustainable AI: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Yi, Y.; Zhang, H.; Xiao, R.; Wang, N.; and Wang, X. 2023. NAR-Former V2: rethinking transformer for universal neural network representation learning. *Advances in Neural Information Processing Systems*.

Zhang, L. L.; Han, S.; Wei, J.; Zheng, N.; Cao, T.; Yang, Y.; and Liu, Y. 2021. nn-Meter: towards accurate latency prediction of deep-learning model inference on diverse edge devices. In *International Conference on Mobile Systems, Applications, and Services*.

Zhang, P.; Zeng, G.; Wang, T.; and Lu, W. 2024. TinyLlama: An Open-Source Small Language Model. arXiv:2401.02385.