Scripting Traffic, Aligning Power: A 5G Testbed Integrating gNB Activity and Commercial UE Power

Haochen Wang[1], Aaron Tartz[1], Adel Heidari[1], Kotturi Sai Prasanth[2], Agrim Gupta[1], Radha Krishna Ganti[2], Dinesh Bharadia[1]
[1]University of California, San Diego [2]Indian Institute of Technology Madras
{haw057,atartz,adheidari,agg003,dbharadia}@ucsd.edu,ksaiprasanth7@smail.iitm.ac.in,rganti@ee.iitm.ac.in

Abstract

On a modern smartphone, the cellular subsystem is one of the largest energy consumers, with power jointly determined by application-layer IP traffic and the gNB's configuration and scheduling decisions. Studying this relationship on real hardware is challenging because the application server, radio activity, and UE are distributed across separate systems, making it difficult to instrument and control them within a single experimental environment. We present a configurable 5G testbed that integrates these components into a unified experimental platform. The platform drives either real Internet applications or arbitrary scripted traffic over the same radio path and links per-slot gNB activity to hardware-measured UE power. As one demonstration of what the resulting dataset enables, we build gNB-Power, an XGBoost estimator that predicts commercial UE power from gNB-observable features alone, roughly halving the error of the standardized TR 38.840 baseline and transferring from synthetic to real applications on this testbed. The platform offers a flexible measurement primitive for UE power-saving studies, enabling controlled investigation of how both application-layer and network-side decisions affect UE energy consumption.

CCS Concepts

• **Hardware** → **Power and energy**; **Hardware test**; **Wireless devices**; • **Networks** → **Network components**; • **Computing methodologies** → **Machine learning**.

Keywords

5G NR, measurement testbed, srsRAN, OTA traffic generation, UE power, XGBoost

WiNTECH '26, Austin, TX, USA

ACM ISBN 979-8-4007-2879-2/26/10
https://doi.org/10.1145/3831662.3844174

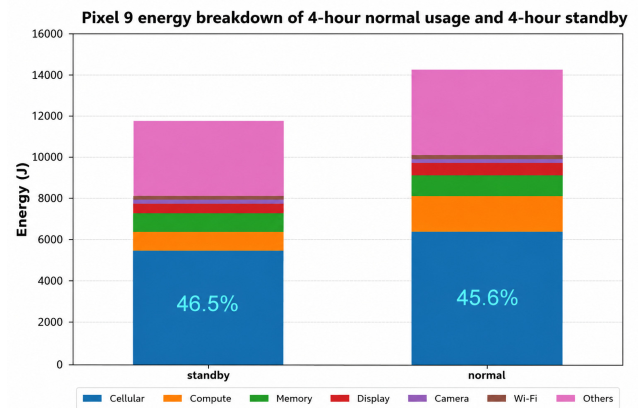

Figure 1: Energy breakdown of a Google Pixel 9 during four hours of normal usage and four hours of standby. Wi-Fi was disabled during the measurements.

1 Introduction

5G New Radio (NR) has dramatically increased the flexibility of cellular communication through wider bandwidths, multi-antenna transmission, and dense control signaling. While these capabilities improve throughput and latency, they also substantially increase the computational and RF workload of the user equipment (UE). Every scheduling decision made by the gNB translates into corresponding processing inside the UE modem, including synchronization, OFDM demodulation, channel estimation, MIMO detection, and channel decoding, all of which must be completed under strict slot-level timing constraints. At the same time, application traffic determines when and how much data is exchanged, shaping the sequence of uplink and downlink transmissions presented to the radio stack. Together, higher-layer traffic demand and network-side scheduling decisions determine the workload executed by the UE modem and ultimately its power consumption.

This processing overhead has made the cellular subsystem one of the largest energy consumers in modern smartphones. Figure 1 shows that on a Google Pixel 9, cellular communication accounts for approximately 46% of total device energy during both normal usage and standby. Here, standby refers

Paper website: https://wcsng.ucsd.edu/scripting-power/

to a screen-off, cellular-attached idle state with background application activity enabled, while normal usage refers to typical outdoor use with occasional social-media browsing and message checking. Cellular communication therefore consumes more energy than compute, memory, display, and other major subsystems, motivating extensive research on discontinuous reception (DRX), bandwidth adaptation, antenna adaptation, and other UE power-saving mechanisms [5, 7, 10].

Despite the importance of this problem, studying network-side strategies for UE power saving remains challenging because it is difficult to relate the highly dynamic traffic and radio activities generated by the 5G network to the real-time power consumption of a commercial UE's proprietary cellular hardware. UE energy is not a simple function of average traffic volume: traffic shapes with similar throughput can induce markedly different power draw (Figure 4), so static throughput-to-power intuition is insufficient. Consequently, most studies instead rely on analytical power models such as the widely used 3GPP TR 38.840, which assigns relative power units to slot-level activities such as PDCCH/PDSCH reception, PUCCH/PUSCH transmission, and sounding reference signal (SRS) transmission [1, 9]. While its transparency makes experiments comparable, it is not a calibrated replica of any commercial modem. Implementation-dependent state transitions, traffic patterns, and RF configurations introduce inaccuracies that a simple table of additive activity powers cannot capture.

What is therefore missing is a way to observe both sides simultaneously on real hardware: to drive arbitrary traffic through a configurable gNB while recording the connected UE's true power consumption at slot granularity. Such a platform would not only enable the design and evaluation of UE power-saving methods using real measurements, but also support the development of UE power estimators tailored to specific devices, environments, and gNB configurations.

To address this gap, we design such a platform and demonstrate its utility for gNB-side UE power estimation. This paper makes two contributions:

- **A slot-synchronized 5G UE power measurement platform.** We develop a configurable srsRAN-based 5G testbed with a commercial Qualcomm modem-based UE that supports both real Internet applications and programmable synthetic traffic. The platform aligns per-slot gNB scheduling activity with hardware-measured UE power using an efficient TR 38.840-based timing-alignment method.
- **A case study: gNB-side UE power estimation.** Using the synchronized dataset, we build gNB-Power, an XGBoost estimator that predicts commercial UE power using only causal gNB-observable features. It outperforms the standardized TR 38.840 baseline and transfers from synthetic traffic to unseen real applications without requiring any labeled target data.

2 Background and Related Work

2.1 Traffic and gNB Configuration-induced Causes of UE Power

With scheduled DL/UL IP traffic, a 5G UE can monitor PDCCH, receive PDSCH, transmit PUCCH or PUSCH, process CSI-RS, or transmit SRS. The gNB sets the allocation geometry for both DL and UL, including resource blocks, symbols, modulation, transport block size, retransmission context, and power control. These events determine whether the receive and transmit chains are active and how much baseband work is required. There are also UE-controlled factors that affect power, such as the UE's design of receive and transmit chains, the number of antennas, and the UE's power control and scheduling decisions. These are usually product-specific and proprietary, so our platform cannot observe or control these.

2.2 The 3GPP Reference Model

TR 38.840 defines relative power states and scaling rules for studying the NR UE power saving [1]. Based on the scheduled link activities such as PDCCH, PDSCH, PUCCH, PUSCH and SRS/CSI-RS in slot t, an activity classifier selects a relative value $P_{\mathrm{rel}}(t)$, scaled by bandwidth of the bandwidth part, number of carriers, number of antennas, and UE transmission-power factors. The model also accounts for the sleep states of the UE, namely micro-sleep, light-sleep, and deep-sleep, along with the power cost of transitions among these states. The model deliberately specifies relative rather than device-specific absolute power. Translating P_{rel} to milliwatts therefore requires mapping a selected baseline relative power state to real device power in the state.

2.3 Related Work and Gap

Analytical work has long designed and evaluated UE power-saving mechanisms using fixed power levels assigned to protocol states. Since the publication of TR 38.840, many NR studies have adopted its common activity-to-power mapping to evaluate measurement relaxation, traffic assistance, and other power-saving procedures [3, 8]. This practice improves comparability but also causes conclusions to inherit the abstraction error of the reference model. Recent work has introduced device-calibrated affine transformation of the TR 38.840 [6], but an affine-transformed analytical model remains constrained by its predefined state structure.

However, real measurements provided by our setup (Section 3) do not remove the need for accurate simulation. An accurate simulator that maps radio activity to UE power would not only let researchers evaluate scheduling decisions without repeatedly measuring every candidate policy on a non-trivial hardware setup, but may also enable real-time

Table 1: Configuration used for the measurements.

Component	Configuration
RAN	srsRAN 5G standalone gNB
Radio	USRP B210, OTA link
Carrier	Band n41, 20 MHz, 30 kHz SCS
UE	Telit FN990A28, Snapdragon X65
Measured quantity	Modem-supply power via INA219
gNB observation grain	One row per UE and 0.5 ms slot
Workload	Five application traces; scripted bidirectional UDP

gNB-side UE power estimation and UE power-aware decision making. The paired data from our setup enables such a simulator through supervised learning, which can use richer link-activity features than heuristic models typically include, while capturing nonlinear interactions and temporal correlations among them. We choose XGBoost because it is well suited to tabular data and offers efficient inference [4].

3 A Slot-Aligned 5G Measurement Platform

3.1 5G Testbed and Power Measurement

Figure 2 summarizes the data-collection stage of the platform. A Linux host runs an Open5GS core and an srsRAN gNB [11, 13], which serves a private 5G standalone cell through a USRP B210 software-defined radio. The experiments use band n41, 20 MHz channel bandwidth, and 30 kHz common subcarrier spacing. The UE is a commercially available Telit FN990A28 data card built around Qualcomm's Snapdragon X65 modem-RF system [12, 14], with four RF ports connected to a 4-by-1 PCB antenna assembly for over-the-air operation.

The UE modem is installed on a PCIe-to-M.2 4G/5G HAT connected to a Raspberry Pi 5 [15]. An INA219 high-side current sensor [2] measures the modem supply and records timestamped power samples on the Raspberry Pi, as shown in Figure 3. This trace provides the measured UE power P_t, which excludes the Raspberry Pi, display, application processor, and other components of a complete smartphone. Together, the gNB log and the power trace form the two raw streams that the next stage aligns into a per-slot dataset.

3.2 One Radio Path, Two Workload Modes

The platform supports two mutually exclusive traffic modes while preserving the same 5G radio path. For real-application captures, a browser runs inside a dedicated Linux network namespace and exchanges traffic with the public Internet through the private 5G standalone network. Downlink packets therefore originate from the Internet and traverse Open5GS, the srsRAN gNB, and the OTA link before reaching the commercial UE, while uplink packets generated by the UE return to the host over the same data bearer. For synthetic captures, the browser is replaced by a programmable traffic generator running in a separate host network namespace. Instead of forwarding Internet traffic, the host injects scripted downlink and uplink IP traffic directly through the same Open5GS–gNB–UE path. Thus, both traffic modes exercise the identical radio stack and differ only in the source of downlink traffic, while the UE can generate uplink traffic in either mode. Section 3.3 describes the programmable traffic generator in detail.

In both modes, Open5GS forwards all packets through the standalone core, and the gNB exports per-slot PHY, FAPI, and scheduler logs containing wall-clock timestamps, canonical slot indices, scheduled and decoded physical channels, resource allocations, link measurements, and RRC-derived configuration. These records constitute the only radio-side observations retained by the platform. The parser converts them into one canonical feature row for each UE and unwrapped global slot, forming the radio-side input to the alignment stage.

3.3 Programmable Bidirectional Traffic

The programmable generator acts as a rate controller for UDP traffic over the same 5G data bearer used by real applications. A schedule is a timeline of short intervals, each specifying the target downlink and uplink rates. In our experiments, we set the target to change every 10 ms, while the smallest supported rate-change interval is 1 ms. To realize a target rate, each sender computes how many 1200-byte UDP payloads should be sent in each 1 ms pacing slice and emits that many packets. When a scheduled rate is zero, the sender transmits no packets, preserving true idle gaps. The downlink and uplink schedules start from synchronized clocks, so offered rates and measured byte counts can be aligned with the UE power trace and gNB logs.

Figure 4 shows five canonical probes over the same testbed, with measured UE power beneath the offered traffic. Even simple probes reveal interesting comparisons. For example, the UL burst train and UL triangle deliver the same aggregate UL throughput over 60 s but consume substantially different energy. One plausible explanation is that the bursts push the UE power amplifier closer to saturation, distorting OFDM peaks and triggering additional retransmissions. This illustrates the core difficulty that motivates our platform: UE energy is not determined by average traffic volume alone. Even with similar aggregate throughput, distinct traffic patterns activate different combinations of uplink transmission, downlink reception, control monitoring, retransmissions, and modem state transitions.

3.4 Slot-Level Clock Alignment

The two streams are recorded on independent clocks, so they must be placed on a common time axis before per-slot activity can be joined with the power it induced. Canonical FAPI slot timestamps provide a coarse epoch alignment. We refine it

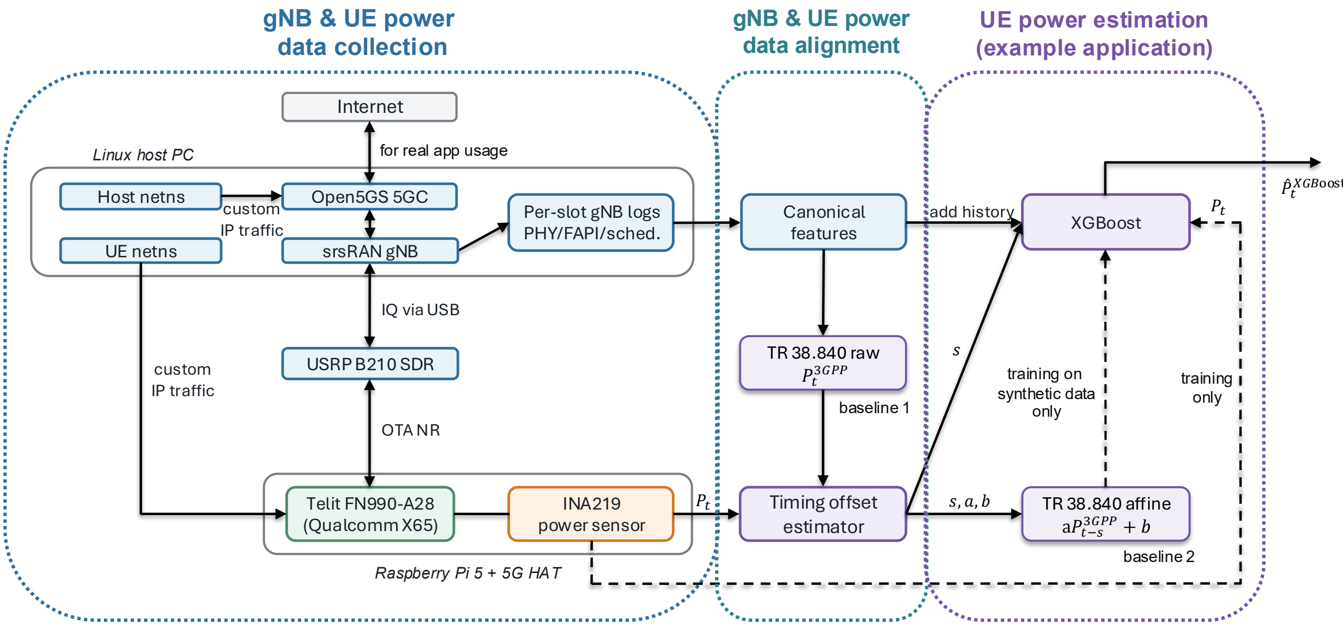

Figure 2: Overview of the measurement platform. The workflow consists of (left) data collection, (middle) gNB–UE power log alignment, and (right) gNB-side UE power estimation.

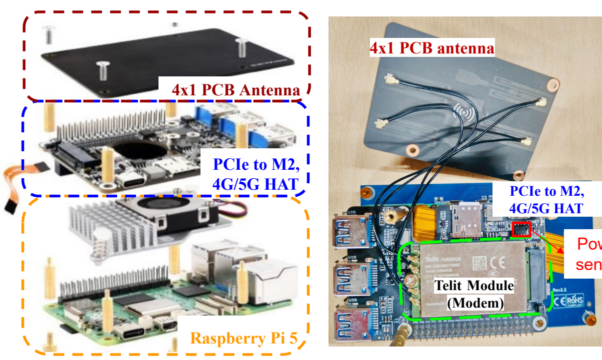

Figure 3: Commercial-modem measurement assembly. The Telit data card and current sensor mount on a PCIe-to-M.2 HAT connected to a Raspberry Pi 5; four coaxial leads feed the PCB antenna.

with an efficient TR 38.840-based search. After interpolating the meter trace onto the slot grid, we compute a TR 38.840 trace $\hat{P}_t^{3\text{GPP}}$ from the gNB activity and jointly choose an integer slot shift s and affine calibration parameters a, b:

$$(s^*, a^*, b^*) = \arg\min_{\substack{s\in\mathbb{Z} \\ a,b\in\mathbb{R}}} \sum_t \left(P_t - \left(a\hat{P}_{t+s}^{3\text{GPP}} + b\right)\right)^2. \quad (1)$$

Here P_t is the measured UE power on the slot grid, and $\hat{P}_t^{3\text{GPP}}$ is the TR 38.840 per-slot activity value computed from the gNB log (Section 2). In implementation, we enumerate candidate integer shifts s; for each one, the best a, b are obtained by a closed-form least-squares fit. Thus the only search dimension is the slot shift. The retained shift is the one with the lowest objective value.

Alignment acts as a data-cleaning step on the time axis rather than a power-modeling fit. In other words, it decides *when* a power sample corresponds to a slot instead of how power should be estimated from it. Its output is thus a per-slot table linking each gNB feature row with the induced UE power consumption, forming the core dataset for the remainder of this paper.

4 Case study: UE Power Estimation

4.1 Problem Formulation

As a demonstration of what the platform's slot-aligned dataset (Section 3) enables, we ask whether the joined per-slot data supports estimating UE power from gNB-observable features alone. Let $\mathbf{x}_t$ denote the gNB-observable feature vector for slot t. We estimate the measured UE power P_t as $\widehat{P}_t = f_\theta(\mathbf{x}_{t-H}, \ldots, \mathbf{x}_t)$, where H is a finite, causal history. Features from future slots are never used. The target timestamp comes from the slot-level clock alignment of Section 3.4. We realize f_θ with XGBoost, an additive ensemble of decision trees suited to heterogeneous tabular features with efficient inference [4].

4.2 Canonical Per-Slot Features

Table 2 groups the model inputs. PHY records are preferred for actual channel occurrence and allocation geometry; RRC and configuration records define BWP, search-space, TDD,

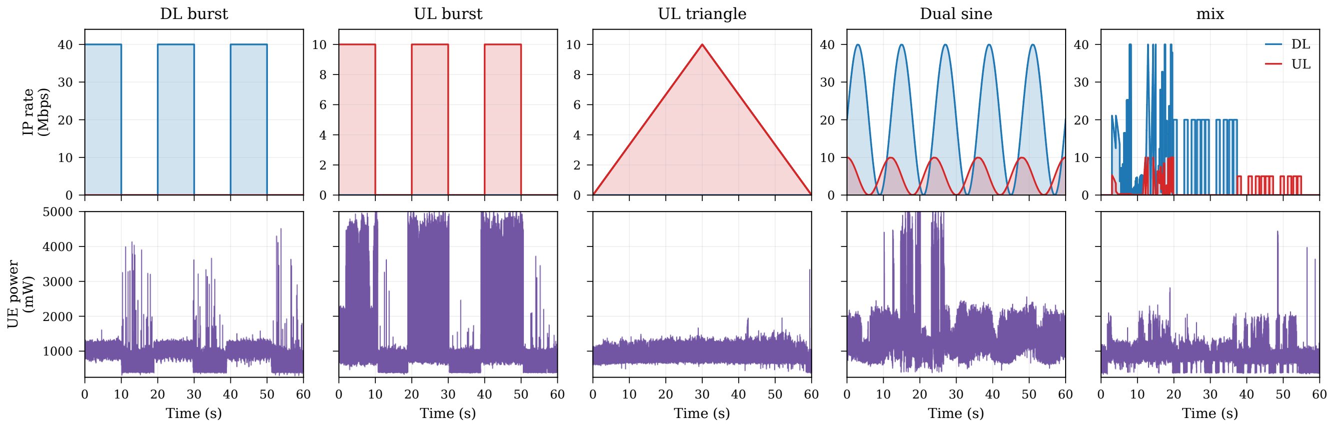

Figure 4: Representative programmed traffic probes. The top row shows offered downlink and uplink IP rates, and the bottom row shows the corresponding measured UE power.

Table 2: Representative feature families.

Family	Representative features
Timing/context	slot, SCS, TDD direction, active BWP, DRX
DL control/data	PDCCH count and aggregation; PDSCH PRBs, symbols, MCS, TBS, layers, HARQ
UL control/data	PUCCH format and bits; PUSCH PRBs, symbols, MCS, TBS, SINR, HARQ
Reference signals	SRS/CSI-RS occurrence and bandwidth
Scheduler state	buffer occupancy, PHR, retransmissions
Temporal context	causal lags and rolling mean/max of activity

and DRX context; FAPI supplies events absent from PHY, such as CSI-RS occurrence; scheduler and MAC records contribute nonduplicate buffer, HARQ, CSI, and power-headroom context.

Temporal features capture state persistence and small residual timing effects. For selected activity and load variables, gNB-Power adds lags at $\{1, 2, 4, 8, 16, 40, 80, 160, 320\}$ slots and rolling statistics over $\{4, 10, 20, 40, 80, 160\}$ slots. This supports online inference without target leakage.

4.3 Analytical Baselines

We use the 3GPP TR 38.840 power model (Section 2.2) as a baseline UE power estimator. We cannot observe the modem's internal sleep state directly, but across captures the lowest stable UE power floor is around 400–420 mW. We therefore map this floor to the TR 38.840 micro-sleep state (relative power 45, compared with 20 for light sleep and 1 for deep sleep), which gave lower error than treating the floor as light or deep sleep. This produces the raw trace $\hat{P}_t^{\mathrm{3GPP}}$. We also evaluate the shifted-affine calibration found during clock alignment (Section 3.4):

$$\widehat{P}_t^{\mathrm{3GPP-aff}} = a^* \hat{P}_{t+s^*}^{\mathrm{3GPP}} + b^*, \tag{2}$$

where s^*, a^*, b^* are the shift and affine parameters from Eq. (1). Comparing against both raw and calibrated variants separates simple timing/scale mismatch from structural error in the analytical model.

5 Results of UE Power Estimation

These experiments demonstrate what the platform's slot-aligned dataset supports: that gNB-observable activity, joined to hardware power, yields an accurate and deployable UE power estimator. We organize the evaluation around two research questions. **RQ1**: when a labeled power trace is available for the target application, can gNB-Power estimate instantaneous UE power accurately, and does that accuracy remain stable as the frozen model runs forward from its training window (Section 5.1)? **RQ2**: can a model trained *only* on synthetic OTA traffic generalize to unseen real applications (Section 5.2)? Throughout, the analytical baselines are the raw TR 38.840 estimate and its affine-calibrated variant $\widehat{P}_t^{\mathrm{3GPP-aff}}$ (Equation (2)). Power errors are in milliwatts, and interval energy errors are percentages on the chronological held-out interval.

The first question uses real Internet application traffic carried over the UE data path, so it evaluates estimation under naturally generated browser workloads. RQ2 instead uses the platform's traffic programmability: we train on scripted OTA traffic patterns whose power labels come from the same measurement setup, then test whether that synthetic corpus transfers to the real applications.

For evaluation metrics, we report the mean absolute error (MAE), mean absolute percentage error (MAPE), and interval

Table 3: Per-capture power estimation on the final 10 s holdout of each 60 s application capture. Each cell reports MAE / MAPE / interval energy error (mW / % / %).

Application	TR 38.840 raw	TR 38.840 aff.	gNB-Power
YouTube	201/27.4/10.9	182/27.5/6.8	**93/13.5/1.5**
Instagram	198/27.8/10.0	251/49.6/29.9	**105/15.8/0.9**
slither.io	256/30.9/25.7	107/12.2/3.8	**86/10.0/1.6**
X	260/29.5/15.6	176/24.3/4.5	**86/10.3/0.9**
Zoom	280/31.5/17.6	162/19.6/5.2	**133/15.4/0.7**
Macro	239/29.4/16.0	176/26.6/10.1	**101/13.0/1.1**

energy error:

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N} |P_t - \hat{P}_t|, \qquad \mathrm{MAPE} = \frac{100}{N}\sum_{t=1}^{N} \frac{|P_t - \hat{P}_t|}{P_t},$$

$$\mathrm{EnergyErr} = 100 \cdot \frac{\left|\sum_{t=1}^{N} \hat{P}_t - \sum_{t=1}^{N} P_t\right|}{\sum_{t=1}^{N} P_t}.$$

5.1 RQ1: In-Capture Accuracy and Temporal Stability

For each application, we collect a 60 s capture: the first 50 s are used for affine calibration and model training, and the final 10 s are held out for one-pass evaluation with no fitting or realignment. Table 3 reports accuracy on this final 10 s interval, and Figure 5 displays randomly selected 3-second windows from the same held-out interval. The five browser applications span distinct traffic regimes: stable video streaming (YouTube), shorts scrolling (Instagram), browsing a mixed social feed of text, images, and shorts (X), a lightweight but latency-sensitive multiplayer web game (slither.io), and video calling (Zoom). Across all five, gNB-Power predicts instantaneous UE power to **86–133 mW MAE** (10–16% MAPE), beating both raw and calibrated TR 38.840 on every workload. Its mean interval-energy error is 1.1%, compared with 10.1% for calibrated TR 38.840 and 16.0% for raw TR 38.840. Calibration helps substantially on some applications but is not sufficient on the higher-error apps: Instagram and X remain difficult for the affine baseline, whereas gNB-Power adapts to their bursty, less predictable load distributions. Conversely, when traffic is smoother or close to stationary, as in slither.io, the affine TR 38.840 baseline is already strong, and the XGBoost advantage is smaller.

A deployable estimator must remain accurate as it runs forward from the training window, without re-calibration. To quantify drift separately from the 60 s accuracy captures, we collect an additional 250 s capture for each application: the first 50 s are used for the same training procedure, and the remaining 200 s are sliced into twenty consecutive 10 s chunks. Each slice probes the frozen model farther from the training window (Figure 6). Accuracy remains stable for YouTube, slither.io, X, and Zoom. Instagram is the challenging case, with its bursty, higher-throughput traffic showing the largest variation and the clearest degradation, although it does not exhibit runaway drift.

Inference is also cheap enough to fit the slot budget. Replaying held-out slots on one CPU core, batched XGBoost inference costs ≈5 μs per slot (about 90× below the 500 μs slot period at 30 kHz SCS), and the single-slot path stays under budget at ≈350 μs. This timing covers model evaluation only, not log generation, feature resolution, or an integrated scheduling decision, so end-to-end online deployment remains future work.

5.2 RQ2: Generalizing to Unseen Applications via Synthetic Training

The per-capture model still requires a labeled power trace from the target application. Rather than labeling arbitrary usage of every app, we ask whether the platform can manufacture a principled training set that generalizes across applications. We therefore train *only* on programmable synthetic traffic and evaluate on the same final 10 s held-out windows from the 60 s real-application captures above.

Synthetic corpus. We script the UE traffic generator across five profiles (streaming, social-scroll, web-browse, video-call, and a rate sweep) with multiple rate/seed settings, yielding 18 synthetic captures whose measured power forms the only training labels (Figure 7 shows a representative schedule per profile). The profiles are designed at the traffic-category level, and their rate ranges are selected to overlap the gNB-observable ranges of the application workloads; they are not replayed application packet traces. Since no real-application data enters training, every application in Table 4 is a true held-out target. The estimator is trained on the pooled features of all 18 synthetic captures with a single global affine calibration, then frozen onto each application's final 10 s real-application test window.

3GPP-affine inputs. Training only on synthetic traffic removes the need for labeled target-application power, but it also makes transfer sensitive to absolute power-scale mismatch between synthetic and real captures. To give the model a physical anchor, gNB-Power gives XGBoost the raw and affine TR 38.840 traces as inputs rather than relying on traffic features alone:

$$\widehat{P}_t = g_\theta\left(\mathbf{x}_{t-H}, \ldots, \mathbf{x}_t, \widehat{P}_t^{\mathrm{3GPP}}, \widehat{P}_t^{\mathrm{3GPP-aff}}\right), \tag{3}$$

where g_θ predicts absolute UE power and $\widehat{P}_t^{\mathrm{3GPP-aff}}$ is the affine-calibrated TR 38.840 baseline of Equation (2). The affine parameters are learned on the synthetic training corpus and frozen for real-application evaluation; only the target capture's gNB activity changes the resulting TR 38.840 trace.

Findings. Trained only on synthetic traffic, the synthetic-trained gNB-Power reaches a macro-averaged **150 mW**

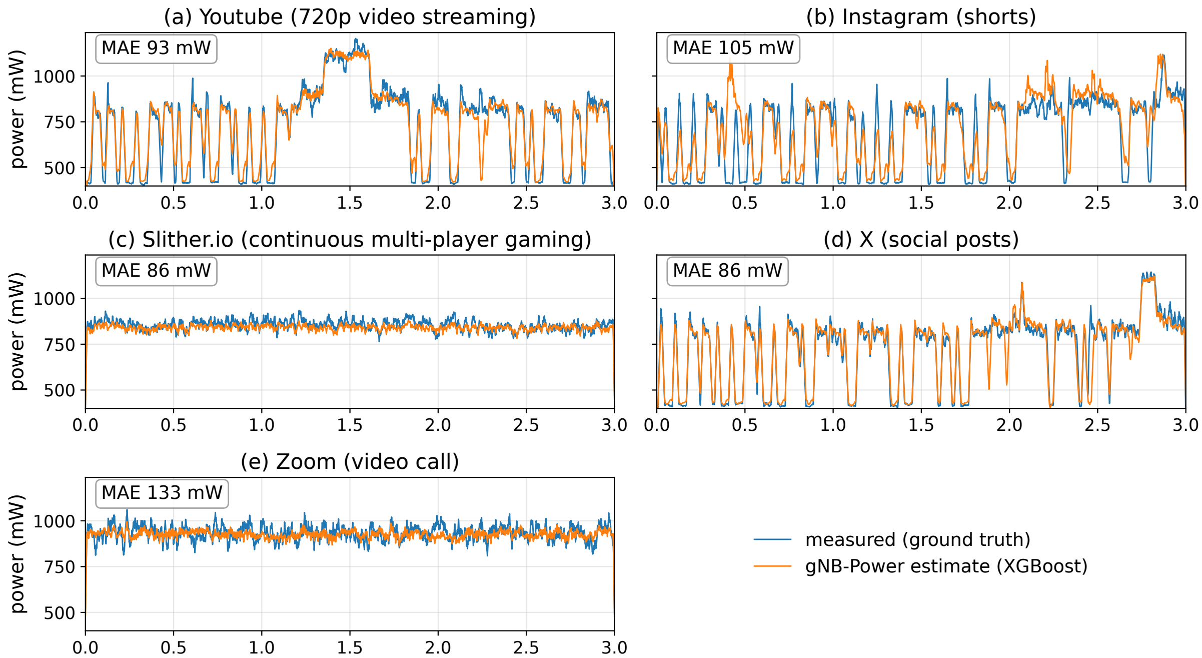

Figure 5: Randomly selected 3 s windows from each application's held-out interval. Measured UE power vs. the gNB-Power estimate. A 15 ms moving average is for visualization only; metrics use unsmoothed traces.

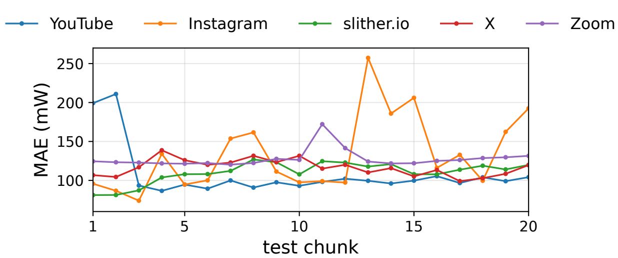

Figure 6: MAE on consecutive 10s chunks from separate 250s drift captures. Models are trained on the first 50s and evaluated on the remaining 200s.

MAE (21.5% MAPE and 4.3% interval-energy error) and beats the synth-fit and target-raw TR 38.840 baselines on every application (Table 4). It also outperforms a *target-fit* TR 38.840 calibration whose affine fit is granted access to the real held-out power (176 mW macro) on three of the five apps and in aggregate. The gap to the same-application in-capture estimator is roughly 1.5×.

6 Limitations and Future Work

This paper builds and validates a platform, but exercises only one application of it, a network-side power estimator (Section 4). It does not yet use the platform for the broader study the setup is designed to enable, and we regard that unexplored capability as the main limitation of this work.

Table 4: Synthetic-to-real generalization. MAE / MAPE / interval energy error (mW / % / %). Target-fit TR 38.840: affine calibration on held-out target power.

App	In-cap.† (ref.)	TR 38.840 synth-fit	TR 38.840 target-raw	TR 38.840 target-fit	XGBoost (synth-trained)
YouTube	93/13.5/1.5	203/35.4/16.8	192/26.2/5.6	182/27.5/6.8	**148/24.0/3.6**
Instagram	105/15.8/0.9	201/35.7/17.1	188/**26.4/4.7**	251/49.6/29.9	**164**/28.5/8.3
slither.io	86/10.0/1.6	165/20.1/12.4	229/27.2/19.8	**107/12.2/3.8**	119/14.5/4.3
X	86/10.3/0.9	194/26.9/8.1	245/27.7/10.1	176/24.3/4.5	**152/21.6/4.0**
Zoom	133/15.4/0.7	195/22.0/4.1	265/29.1/12.0	**162**/19.6/5.2	167/**18.8/1.2**
Macro	101/13.0/1.1	192/28.0/11.7	224/27.3/10.4	176/26.6/10.1	**150/21.5/4.3**

†Reference lower bound from Table 3; the synthetic-trained estimator uses no real-application power labels.

Using the platform to relate application-layer and gNB decisions to UE energy. Because the platform aligns realized traffic, gNB activity, and measured modem-supply power at slot granularity, it is positioned to ask how each side drives UE energy, from application-layer choices such as burst shape, direction, and timing to gNB decisions such as grant timing, allocation size, MCS, DRX opportunity, and retransmission handling. This paper only hints at that potential, for instance by showing that two workloads with similar aggregate throughput consume different energy (Section 3.3).

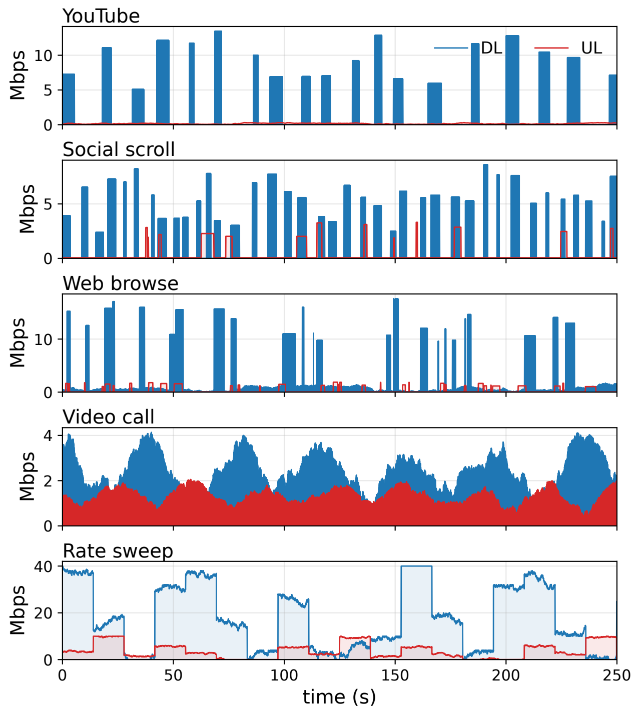

Figure 7: Representative scripted downlink and uplink IP-rate schedules from the synthetic training corpus.

It does not yet run the controlled studies, sweeping one side while holding the other fixed, that would turn the platform into a tool for attributing UE energy to its application-layer and network-side causes. Enabling that study, and eventually comparing candidate policies without measuring each on hardware, is the direction the platform is built to support.

Measurement scope. Results come from one configuration: one Snapdragon X65-based data card, one band n41 carrier, 20 MHz bandwidth, 30 kHz SCS, and one laboratory OTA environment. The sensor measures the modem's supply rail, not whole-device energy. The model and error values are therefore testbed-specific, and broader claims about absolute UE energy would require additional modems, bands, bandwidths, and radio conditions.

7 Conclusion

We presented a configurable 5G standalone testbed that jointly drives two sides of a UE's cellular energy—the device's application-layer IP traffic and the gNB's scheduling and configuration decisions—while measuring the resulting commercial UE power, aligned in one per-slot dataset. As one demonstration, our gNB-power estimator predicts UE power from gNB-observable features alone, roughly halving the TR 38.840 baseline error and transferring from synthetic to real applications on this testbed. Attributing UE energy to its application-layer and network-side causes remains the broader study this measurement primitive supports.

Acknowledgment

This work was supported in part by the U.S. National Science Foundation (NSF) under Award No. 2312979. Further, the authors acknowledge the anonymous reviewers.

References

[1] 3rd Generation Partnership Project. 2019. *Study on User Equipment Power Saving in NR*. Technical Report TR 38.840. 3GPP.

[2] Adafruit Industries. 2026. INA219 High-Side DC Current Sensor Breakout. https://www.adafruit.com/product/904. Accessed June 2026.

[3] Anum Ali, Yuqiang Heng, Vutha Va, Priyabrata Parida, Boon Loong Ng, and Jianzhong Charlie Zhang. 2023. UE Power Saving With Traffic Classification and UE Assistance. *IEEE Access* 11 (2023), 136869–136884. doi:10.1109/ACCESS.2023.3339327

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794. doi:10.1145/2939672.2939785

[5] Agrim Gupta, Adel Heidari, Avyakta Kalipattapu, Ish Kumar Jain, and Dinesh Bharadia. 2024. 3 W's of Smartphone Power Consumption: Who, Where and How Much Is Draining My Battery?. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 2248–2250. doi:10.1145/3636534.3698862

[6] Marvin Illian, Björn Luchterhandt, and Lin Wang. 2025. Band Switching for Mobile Energy Optimization in 5G Networks and Beyond. In *Proceedings of the 20th Workshop on Mobility in the Evolving Internet Architecture*. 44–50.

[7] Taehyoung Kim, Younsun Kim, Qiongjie Lin, Feifei Sun, Jingxing Fu, Youngbum Kim, Aris Papasakellariou, Hyoungju Ji, and Juho Lee. 2020. Evolution of Power Saving Technologies for 5G New Radio. *IEEE Access* 8 (2020), 198912–198924. doi:10.1109/ACCESS.2020.3033452

[8] Daniela Laselva, Laura L. Sanchez, Faranaz Sabouri-S, Qiyang Zhao, Jorma Kaikkonen, Lars Dalsgaard, and Pasi Kinnunen. 2021. UE Measurements Relaxation for UE Power Saving in 5G New Radio. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. 1–6. doi:10.1109/VTC2021-Fall52928.2021.9625315

[9] Mads Lauridsen, Daniela Laselva, Frank Frederiksen, and Jorma Kaikkonen. 2019. 5G New Radio User Equipment Power Modeling and Potential Energy Savings. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. 1–6. doi:10.1109/VTCFall.2019.8891144

[10] Kuang-Hsun Lin, He-Hsuan Liu, Kai-Hsin Hu, An Huang, and Hung-Yu Wei. 2023. A Survey on DRX Mechanism: Device Power Saving from LTE and 5G New Radio to 6G Communication Systems. *IEEE Communications Surveys & Tutorials* 25, 1 (2023), 156–183. doi:10.1109/COMST.2022.3215731

[11] Open5GS. 2026. Open5GS: Open Source Project of 5G Mobile Core Network. https://open5gs.org/. Accessed June 2026.

[12] Qualcomm. 2026. Snapdragon X65 5G Modem-RF System. https://www.qualcomm.com/products/technology/modems/snapdragon-x65-5g-modem-rf-system. Accessed June 2026.

[13] Software Radio Systems. 2026. srsRAN Project. https://www.srsran.com/. Accessed June 2026.

[14] Telit Cinterion. 2026. FN990 Series 5G Data Card. https://www.telit.com/devices/fn990-series/. Accessed June 2026.

[15] Waveshare. 2026. PCIe to M.2 4G/5G HAT for Raspberry Pi 5. https://www.waveshare.com/pcie-to-m.2-4g-5g-hat.htm. Accessed June 2026.