How much battery does dark mode save?

An Accurate OLED Display Power Profiler for Modern Smartphones

Pranab Dash
Purdue University
dashp@purdue.edu

Y. Charlie Hu
Purdue University
ychu@purdue.edu

ABSTRACT

By omitting external lighting, OLED display significantly reduces the power draw compared to its predecessor LCD and has gained wide adoption in modern smartphones. The real potential of OLED in saving phone battery drain lies in exploiting app UI color design, *i.e.*, how to design app UI to use pixel colors that result in low OLED display power draw. In this paper, we design and implement an accurate per-frame OLED display power profiler, PFOP, that helps developers to gain insight into the impact of different app UI design on its OLED power draw, and an enhanced Android Battery that helps phone users to understand and manage phone display energy drain, for example, from different app and display configurations such as dark mode and screen brightness. A major challenge in designing both tools is to develop an accurate and robust OLED display power model. We experimentally show that linear-regression-based OLED power models developed in the past decade cannot capture the unique behavior of OLED display hardware in modern smartphones which have a large color space and propose a new piecewise power model that achieves much better modeling accuracy than the prior-art by applying linear regression in each small regions of the vast color space. Using the two tools, we performed to our knowledge the first power saving measurement of the emerging dark mode for a set of popular Google Android apps.

CCS CONCEPTS

• **Hardware** → **Power estimation and optimization**; • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**.

KEYWORDS

OLED Display; Power Modeling; Dark Mode; Display Power Profiler

ACM Reference Format:

Pranab Dash and Y. Charlie Hu. 2021. *How much battery does dark mode save? An Accurate OLED Display Power Profiler for Modern Smartphones*. In *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21)*, June 24–July 2, 2021, Virtual, WI, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3458864.3467682>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiSys '21, June 24–July 2, 2021, Virtual, WI, USA

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8443-8/21/06...\$15.00
<https://doi.org/10.1145/3458864.3467682>

and Services (MobiSys '21), June 24–July 2, 2021, Virtual, WI, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3458864.3467682>

1 INTRODUCTION

Optimizing the battery drain of mobile apps helps to extend the mobile device battery life which is critical to the mobile experience of the 5+ billion phone users (over half are smartphones) [6]. Optimizing the battery drain of mobile apps requires optimizing the battery drain of all power-hungry device components, including CPU, GPU, display, WiFi/LTE, GPS, and hardware decoder. After over a decade of phone evolution, display has remained as one of the major power-draining components in modern smartphones [12–14, 23, 25, 39, 41].

The latest display technology, Organic light-emitting diode (OLED) [26, 38], uses light-emitting diode as pixels and therefore omits the need for external backlight used in its predecessor LCD, and in doing so provides much better power efficiency than LCD. Due to its power efficiency and several other advantages (thinner, lighter, and more flexible) over LCD and LED, OLED has seen wide adoption in high-end and mid-range smartphones.

Since the OLED power draw is directly related to the displayed content, the real potential of OLED in terms of power savings lies in exploiting the app UI color design, *i.e.*, how to design the app UI to use pixel colors that result in less OLED display power draw. These include manual designs by app developers as well as automatic color transformations [24, 25, 39]. One of the extreme examples of color transformation to save display energy of mobile devices is the recent wide adoption of dark-themed color contents, known as dark mode, by both Android and iOS which added dark mode as one of major features in their 2019 OS update [7, 42], and app vendors who quickly rolled out dark mode UI options in their latest app releases [11, 16, 17, 21, 27, 29, 37].

Despite the industry's major push for dark mode, it remains unclear how much power and energy savings dark mode will bring to the apps, as the OLED display power saving from switching from normal (or light) mode to dark mode depends on the displayed content, which can vary significantly from one app to another, and from one app activity to the next. More generally, the industry is lacking an OLED display power profiling tool that can accurately estimate the OLED display power draw and attribute it to the individual UI components. Such a tool will enable an app developer to gain instant insight into how different UI designs affect the OLED display power draw in running the app on specific mobile devices and optimize the app UI design accordingly.

In this paper, we present the design and implementation of two OLED power management tools: an accurate per-frame OLED display power profiler, PFOP, that helps *developers* to gain insight into the impact of different app UI designs on its OLED power draw, and an enhanced Android Battery called Android Battery+ (or Battery+ for short) that helps *phone users* to understand and manage phone display energy drain, for example, from different app and display configurations such as dark mode and screen brightness. A major challenge in designing both tools lie in designing an accurate and robust OLED power model that can accurately estimate the display power draw as a function of the displayed content on any device.

Since power modeling of OLED display has been studied over the past decade, first on external display then earlier smartphones [23, 28, 33], we first study *if the prior-art power models of OLED display proposed in the past decade can accurately capture the power behavior of OLED displays on modern smartphones*. To answer this question, we conduct a measurement study to characterize the power behavior of OLED displays on four representative smartphones from four recent generations, Pixel 2 (2017), Moto Z3 (2018), Pixel 4 (2019) and Pixel 5 (2020). Our measurement study reveals unique power draw behavior of modern phone OLED displays: (1) The OLED display power draw violates the superposition property observed in [23], *e.g.*, the power draw in displaying the white color can be far less than the simple summation of those in displaying the three base colors: red, green and blue (after removing base power). (2) The OLED display power often violates the monotonicity property, that the power draw in displaying canonically larger RGB values in the color space should be higher. The two findings suggest that previously proposed linear and non-linear OLED power models are unlikely to achieve high prediction accuracy, as confirmed by our experiments.

Second, we study *how to develop new practical OLED power models that can accurately and robustly capture the power behavior of modern smartphone OLED displays*. To develop such a power model, we first unearth the root cause for why previous models fail: linear regression of subpixel power functions (which are linear in linear RGB values) cannot capture the OLED power behavior of modern OLED displays because any linear combinations of linear functions can only fit a flat plane, while the OLED power values across the color space do not fall into a flat plane.

Our new OLED power model is based on a key observation that though the pixel power values in the color space do not fall in a plane, they form a reasonably smooth surface. Therefore, if we cut the 3-D color space into small subgrids, the pixel power values within each subgrid are likely to fit closely in a flat plane and thus can be fitted well using linear regressions of the (linear) subpixel power functions. We experimentally show that our new piecewise OLED power model reduces the OLED prediction error of the prior art [33] by 5.1 $\times$, 4.4 $\times$, 5.6 $\times$ and 4.9 $\times$ for dark and 4.1 $\times$, 2.5 $\times$, 2.3 $\times$ and 5.8 $\times$ for light mode for the four phones, respectively, for a set of activities across 6 popular Google apps.

Third, we design and implement the two tools on top of our new OLED power model. PFOP accurately breaks down the OLED display power in displaying a frame among its UI components and faces the challenge of identifying all the UI components of a displayed frame, *e.g.*, an app activity. We design PFOP to use

only standard Android services so it is portable to any unmodified Android versions. Battery+ replaces the simple screen energy accounting of Android Battery with an accurate, per-app screen energy estimator. Like Android Battery, it runs in real time in the background and thus needs to have low overhead. We carefully apply pixel sampling to both screen recording and model application to control its average CPU utilization to be only 1.1%, 0.9%, 0.6% and 1.6% on Pixel 2, Moto Z3, Pixel 4, and Pixel 5, respectively, while minimally affecting the power estimation accuracy.

Finally, we showcase the usage of the two tools by performing a case study of the dark mode power saving that answers *how much does dark mode reduce the OLED display power draw and energy drain for real apps?* In particular, we study the power consumption of a set of 6 Google apps (each with 1B+ downloads) in two ways. First, *from the phone user's perspective*, we analyze the power draw of each Google app under a typical user interaction using Battery+. Battery+ shows that (1) switching from light mode to dark mode significantly reduces the average OLED power draw across all apps, though the reduction varies significantly among the apps and the phones, ranging between 15% - 36% (average 27%), 14% - 38% (average 28%), 11% - 25% (average 19%) and 14% - 32% (average 24%) at the 50% brightness level, on the 4 phones, respectively. (2) The significant OLED power reduction translates into moderate reduction in the total phone power draw, ranging between 4% - 16% (average 9%), 5% - 15% (average 9%), 2% - 11% (average 7%), and 0% - 18% (average 9%) at the 50% brightness level across the apps on the 4 phones, respectively.

Second, *from the app developer's perspective*, we use PFOP to analyze how much power, switching the UI design of each activity to dark mode saves in the 6 Google apps. Our study shows how PFOP accurately quantifies and displays the power saving for each UI component in switching to the new color scheme in dark mode, allowing app developers to easily experiment with the design-power tradeoff of alternative dark mode designs.

We summarize the contributions of our study as follows:

- We experimentally characterize the unique power draw behavior of OLED displays on 4 representative modern smartphones and show how they violate the underlying assumptions made in prior OLED power models proposed in the past decade.
- We propose a new piecewise OLED power model and experimentally show that it reduces the prediction error of the prior art by 5.1 $\times$, 4.4 $\times$, 5.6 $\times$ and 4.9 $\times$ for dark mode and 4.1 $\times$, 2.5 $\times$, 2.3 $\times$ and 5.8 $\times$ for light mode on Pixel 2, Moto Z3, Pixel 4 and Pixel 5, respectively.
- Using our new model, we develop a per-frame OLED power profiler that enables developers to instantly understand the OLED power draw breakdowns of activity UI designs to guide UI power optimization, and an enhanced Android Battery that enables phone users to understand and manage phone display energy drain, for example, from different app and display configurations such as dark mode and screen brightness.
- Using the two tools, we present to our knowledge the first public study of the OLED display power and phone power savings in switching from light mode to dark mode for a set of Google apps (each with 1B+ downloads).

Figure 1: Frame rendering in modern smartphones.

2 BACKGROUND

Frame rendering. Figure 1 illustrates the frame rendering process in a typical modern smartphone [4]. After the GPU renders all the layers for the next frame (for status bar, navigation bar, and app UI), e.g., every 16.7ms at 60 FPS, the bitmap of the frame content is stored in separate buffer queues. Each unit of the bitmap describes the color of a pixel in the standard RGB (sRGB) color space, (R, G, B) , which specifies the intensities of the red, green, and blue base colors. Modern displays come with 24-bit (8, 8, 8) RGB color setting, i.e., each intensity ranges between 0 and 255. SurfaceFlinger provides HardwareComposer HAL with a list of layers and instructs it to compose the final frame sent to the display. HardwareComposer HAL may perform additional color transformation to generate the final frame.

If the Android screenrecord command is issued, SurfaceFlinger composes the multiple layers from buffer queues into a frame buffer [5] sent to an additional virtual display for screen recording. Note that such recorded frame buffers would not include color transformation happening in HardwareComposer below.

OLED display. Each pixel of an OLED display consists of three subpixels, corresponding to red, green, and blue colors, which can differ significantly in their luminance efficacies [26, 38]. This is the reason why the color of a pixel determines its power consumption and thus different display content (e.g., from different app UI design) can result in very different OLED display power draw.

In addition to the panel of addressable pixels, the total OLED display power consumption is affected by the control circuitry that generates the control and data signals for the panel based on display contents. Different circuitry implementation, e.g., by different manufacturers, can potentially result in different power draw.

OLED power modeling. Since an OLED display consists of N pixels that emit lights independently of each other, the total power draw of an OLED display equals the sum of the power draw P_i by individual pixels (e.g., [23, 28]):

$$P_{OLED} = C + \sum_{i=1}^n P_i \quad (1)$$

Table 1: OLED displays in the 4 phones used in our study.

Phone	OLED display	Resolution	Base power (mA)	Dark Screen power (mA)	Android Version
Pixel 2	AMOLED	1920 x 1080	22.9 mA	43.2 mA	R (11)
Moto Z3	Super AMOLED	2160 x 1080	21.4 mA	57.6 mA	Pie (9)
Pixel 4	P-OLED	2280 x 1080	23.6 mA	78.8 mA	R (11)
Pixel 5	OLED HDR10+	2340 x 1080	19.3 mA	67.8 mA	R (11)

where constant C accounts for the power draw by the non-pixel component of the OLED display, denoted as *dark screen power*, and individual pixel power draw P_i depends on the pixel color value.

To use an OLED power model, the screen needs to be recorded and fed into the model. Ideally, we want to record the final frame sent to the display. However, Android only provides interfaces that record the frame buffer (via virtual display) which will miss possible color transformation in the hardware composer. We discuss the implications on OLED power modeling in §3.3 and §3.4.

3 CHARACTERIZING OLED POWER DRAW ON MODERN PHONES

An accurate OLED power model for modern smartphones must capture any unique OLED power draw behavior on such devices. Therefore, we start with a measurement study to characterize the OLED power draw behavior on four modern smartphones.

Methodology. We conduct our study on four smartphones from four recent generations, Pixel 2 (2017), Moto Z3 (2018), Pixel 4 (2019), and Pixel 5 (2020) as listed in Table 1.

To accurately measure the phone power draw in our experiments, we directly connect each of the four phones to a Monsoon power monitor [36] which has a current sampling rate of 5 KHz.

To facilitate measurements of OLED display power for different colored pixels, we created a simple Android app that displays a static image at a time on the screen, while the phone power draw is being measured. The images used will have the same dimension as the screen, e.g., 1920x1080 pixels on Pixel 2. For each of the static colored images, we generated 5 measurements, out of which we chose 3 that had the least standard deviation and took the average. First, we measured the CPU idle power as the power draw when the screen is off and the app is running using a wakelock. Second, we measured the OLED dark screen power by displaying a dark screen with all the pixel values set to zero and then subtract the CPU idle power (from Step 1) from the measured phone power. Finally, to measure the power draw by all the colored pixels, i.e., $\sum_{i=0}^n P_i$ in Eqn. 1, in displaying a static image, we measure the phone power and subtracted from it both the dark screen power and the CPU idle power. The above process is accurate as in all three steps, the CPU utilization remained steady at 2.5%, 4.9%, 1.7% and 2.6% on Pixel 2, Moto Z3, Pixel 4, and Pixel5, respectively.

3.1 Superposition Property

The first question we ask is whether the power draw of the three subpixels are independent, i.e., whether they satisfy the *superposition property* which states that the power consumed by three subpixels are additive.

Methodology. To validate the superposition property, we measure the OLED power consumed by 7 monochromatic images containing 3 base colors (red, green, blue), 3 2-channel colors (yellow, magenta

Figure 2: OLED display power in displaying one-color images on four phones (with default color mode). (LM: linear model as in Eqn. 2, LMLR: linear model with linear regression, NLM: Non-Linear Model).

and cyan) and the 3-channel color white, with varying intensities (sRGB values from 0 to 255).

Findings. Figures 2(a)–2(d) show the measured OLED display power. Due to the gamma-correction in sRGB standard, the subpixel power draw is known to be nonlinear in the sRGB values. Thus in this paper, we transform the sRGB format into the linear format, known as Linear RGB [8] ($LinearRGB = (sRGB/255)^{2.2}$), and plot all OLED power functions w.r.t. Linear RGB values. Figures 2(a)–2(d) show that (1) the subpixel power draw for the 3 base colors are linear in the Linear RGB values; (2) the subpixel power draw are not additive – the gap between their sum (without duplicating base power) and the power draw for white varies across the color intensity, reaching a maximum of 92%, 66%, 43% and 27% at linear RGB (sRGB) values of 0.002 (14), 0.04 (59), 0.01 (25) and 0.02 (39) for the 4 phones, respectively.

We performed the same analysis for the 2-channel color magenta (figures not shown due to page limit). Again, we observed a large deviation between the sum of the power individually observed for blue and red compared to the actual measurement; the gap varies across the color intensity, reaching 55%, 45%, 26% and 50% at linear RGB (sRGB) values of 0.05 (64), 0.05 (67), 0.01 (27) and 0.02 (39) for the four phones, respectively. Similar gaps are also observed for yellow, of 60%, 16%, 18% and 14% at linear RGB (sRGB) values of 0.05 (68), 0.10 (91), 0.02 (43) and 0.02 (39), and for cyan, of 44%, 17%, 21% and 11% at linear RGB (sRGB) values of 0.003 (19), 0.23 (130), 0.01 (34) and 0.01 (32), for the four phones, respectively. These results show that *subpixel powers are not additive*. The likely reason is that the subpixels utilize common driving circuits within the display hardware module.

3.2 Monotonicity Property

We next study whether a more basic property holds in the OLED power behavior on modern phones. The *monotonicity property* is stated as follows:

Monotonicity property: The OLED pixel power draw for RGB value (x_1, y_1, z_1) is less than that for (x_2, y_2, z_2) iff vector (x_1, y_1, z_1) is canonically less than (x_2, y_2, z_2) , i.e., $x_1 \leq x_2$, $y_1 \leq y_2$, and $z_1 \leq z_2$.

Methodology. To validate the monotonicity property, we measure the OLED power draw in displaying monochromatic images generated by sweeping the 3-D RGB color space, with each base color intensity varied from 0 to 255 (in the sRGB space) in steps of 16 (and

including value 255) for a total of 17^3 images. We then compare the display power draw for each pair of adjacent RGB values in the $17 \times 17 \times 17$ grid of discretized 3-D color space.

Findings. On all four phones, we found many instances of power measurements that violate the monotonicity property. Table 2 lists a few samples for each phone, where each row lists the power draw for two color vectors, the second being canonically larger yet drawing less power than the first. For example, on Pixel 4, the OLED power draw for RGB value (32, 16, 16) should be higher than that for (32, 16, 0). Instead, our measured value for the former, 89mA, is 1.09X lower than that for the later, 97mA. Since this pattern happens on all four phones, it is unlikely due to manufacturing defect. We conjecture¹ there are several possible reasons for the above non-monotonicity behavior: (1) the subpixels utilize common driving circuits within the display hardware module differently at different color intensities; (2) there are handset OEM-specific color transformations performed in the hardware composer (Figure 1) that alter the canonical order of sRGB values; and (3) there are other handset OEM-specific customizations of the circuits that control the pixels.

3.3 Impact of Brightness Level

Modern smartphones come with a hardware button or software brightness slider that allows users to directly adjust the display brightness level, e.g., between 0% – 100%. We found that *brightness level adjustment by either scheme does not change frame pixel values recorded in the frame buffer*, suggesting that some color transformation caused by the slider brightness level happens after the frame buffer, i.e., in the hardware composer (Figure 1). Hence an OLED power model based on the frame pixel values needs to be adjusted for the display brightness level. However, our measurements also showed that the OLED display power draw does not scale linearly in displaying a same image with the slider brightness level.

We searched the kernel exported hardware info (e.g., `/sys/class/backlight`) and found that for Android Pie and later versions, the brightness slider value (0% to 100%) is transformed to the system screen brightness value (0 to 255), which can be read from system variable `/sys/class/leds/lcd-backlight/brightness` on Pixel 2 and Moto

¹We could not find any white papers from the OLED vendors or the driver source code which are all proprietary.

Table 2: Sample RGB value pairs for which the OLED display power violates the monotonicity property.

Type	Color 1	Current (mA)	Color 2	Current (mA)
Pixel 2				
1 chl to	(176, 0, 0)	91	(176, 0, 32)	74
2 chl	(0, 16, 0)	56	(0, 16, 16)	48
	(0, 0, 48)	56	(32, 0, 48)	43
2 chl to	(112, 80, 0)	82	(112, 80, 16)	68
3 chl	(112, 0, 192)	146	(112, 32, 192)	131
	(0, 48, 48)	71	(48, 48, 48)	53
Moto Z3				
1 chl to	(240, 0, 0)	230	(240, 0, 32)	220
2 chl	(0, 192, 0)	135	(16, 192, 0)	133
	(0, 0, 240)	288	(32, 0, 240)	273
2 chl to	(192, 224, 0)	246	(192, 224, 48)	220
3 chl	(255, 0, 255)	429	(255, 48, 255)	409
	(0, 224, 160)	242	(64, 224, 160)	215
Pixel 4				
1 chl to	(80, 0, 0)	105	(80, 0, 16)	97
2 chl	(0, 48, 0)	94	(0, 48, 16)	91
	(0, 0, 240)	233	(32, 0, 240)	230
2 chl to	(32, 16, 0)	97	(32, 16, 16)	89
3 chl	(128, 0, 16)	113	(128, 16, 16)	109
	(0, 192, 176)	234	(16, 192, 176)	229
Pixel 5				
1 chl to	(64, 0, 0)	76	(64, 0, 32)	72
2 chl	(0, 96, 0)	83	(0, 96, 32)	80
	(0, 0, 48)	82	(16, 0, 48)	73
2 chl to	(64, 255, 0)	200	(64, 255, 32)	191
3 chl	(48, 0, 16)	79	(48, 32, 16)	71
	(0, 16, 48)	84	(32, 16, 48)	74

Z3 and /sys/class/backlight/panel0-backlight/brightness on Pixel 4 and Pixel 5.² Second, we experimentally found that the slider brightness level is mapped to the system brightness level via a logarithmic function, as shown in Figure 3. Finally, we experimentally confirmed that the OLED display current scales linearly with the system brightness value in displaying the 7 monochromatic colored images.

Our above findings have two implications. First, when recording a frame as an input to the OLED power model, we also need to record the system brightness level so that the output of the pixel value-based power prediction can be scaled properly. Second, the OLED display power at a reduced slider brightness level (e.g., 30%) actually drains much less power (e.g., $9/255 = 3.5\%$ for Pixel 4 as shown in Figure 3), due to the logarithmic relationship between the two (slider and system) brightness levels.

3.4 Impact of Color Mode and Color Correction

In addition to the brightness level, we found two other forms of color transformation can happen in recent versions of Android. First, Android Lollipop introduced *color correction* for visually impaired users [19]. Second, Android 8.1 Oreo added templates for adding new *color modes*. OEMs can use this template mechanism to display vibrant colors by utilizing the wide color gamut.

²The system variable on Pixel 5 takes values from 0 - 1023, which can be mapped to 0 - 255 linearly via division by approximately 4.

Figure 3: Relationship between the slider brightness level and the system brightness value.

Both color change features are implemented by passing a color matrix [20] to the hardware composer, which performs the color transformation on the frame pixels before sending them to the display. In doing so, they alter the OLED power draw without changing the frame buffer recorded, similarly as with brightness level adjustment. For example, our measurements have shown that on Pixel 2, displaying the same magenta-color image under Natural and Saturated Color Modes draws 7.1% less and 19.7% more power than under the default color mode. Hence, *OLED power models using frame buffer as input need to be derived for each color mode or correction*.

4 LIMITATION OF PREVIOUS MODELS

Since directly measuring the power draw of every pixel color is too costly (e.g., it would take $256^3 \times 2$ seconds or 1.06 years following our methodology of displaying each image for 2 seconds), power modeling has been proposed to estimate the OLED power draw of any pixel color based on measurements of a few pixel colors.

Simple linear model. Dong *et al.* [23] proposed one of the first pixel-level OLED current models, which models the pixel power draw as the simple sum of those of subpixels:

$$P_i(R_i, G_i, B_i) = f(R_i) + g(G_i) + h(B_i) \quad (2)$$

and showed that the model achieves an average error of 1% in displaying 300 benchmark images on an external OLED QVGA display module with a standard 16-bit (5, 6, 5) RGB setting. The simple linear model (denoted as LM) cannot model the OLED power on modern phones which violate the superposition property.

Adding Linear Regression. Violation of the superposition property has been observed previously [28, 32] and suggests the power draw of three subpixels are not independent. The simplest idea to accommodate the dependence is to introduce liner regression, as suggested by Kim *et al.* [28], who showed extending the simple summation into a linear formula with the coefficients derived via linear regression reduces the average error from 8.0% to 1.4% on Galaxy S3 for a set of screen images of 10 Android apps.

Since the power model should capture the pixel power draw for any combinations of RGB values, to assess the feasibility of the above model, we derived linear regression of $f(R_i)$, $g(G_i)$, and

$h(B_i)$ to fit the power draw of 4 colors, yellow, magenta and cyan, and white, as the intensity varies from 0 to 255³. The derived models for the 4 phones are:

$$P_i^{Pixel2} = 0.700 * f(R_i) + 0.928 * g(G_i) + 0.868 * h(B_i) \quad (3)$$

$$P_i^{MotoZ3} = 0.665 * f(R_i) + 0.928 * g(G_i) + 0.708 * h(B_i) \quad (4)$$

$$P_i^{Pixel4} = 0.779 * f(R_i) + 1.047 * g(G_i) + 0.929 * h(B_i) \quad (5)$$

$$P_i^{Pixel5} = 0.784 * f(R_i) + 1.280 * g(G_i) + 0.979 * h(B_i) \quad (6)$$

The models do not have constant terms because the base power is already accounted in Eqn. 1.

We then plot the predicted power with the new models for white in Figure 2, labeled as LMLR. We see that the new model improves the accuracy of the simple superposition model LM, but still incurs an average (over linear RGB values) prediction error of 19%, 23%, 15% and 6% for white (and 11%, 11%, 7% and 13% for magenta which are not shown) on the 4 phones, respectively.

One reason that such linear regression based models remain inaccurate is that they are linear in the subpixel powers (with positive coefficients) and hence still assume the monotonicity property which does not hold as shown in §3.2.

Using non-linear models. Another measurement study [33] on an external AMOLED module suggested adding non-linear terms, in particular, pairwise two-color coupling terms, to reduce the modeling error of linear models. We experimented with such a non-linear model (denoted as NLM), by adding pairwise two-color coupling terms and a three-color coupling term, and applying linear regression to derive the coefficients of the resulting polynomial model to fit the power draw of the same 4 colors, yellow, magenta and cyan, and white:

$$P = V \cdot [f(R_i) \ g(G_i) \ h(B_i) \ f(R_i) \cdot g(G_i) \quad (7)$$

$$f(R_i) \cdot h(B_i) \ g(G_i) \cdot h(B_i) \ f(R_i) \cdot g(G_i) \cdot h(B_i)]^T$$

$$V_{Pixel2} = [0.330 \ 0.611 \ 0.652 \ 0.011 \ 0.007 \ 0.005 \ -0.000]$$

$$V_{MotoZ3} = [0.416 \ 1.224 \ 0.535 \ 0.001 \ 0.003 \ 0.001 \ -0.000]$$

$$V_{Pixel4} = [0.286 \ 0.926 \ 0.859 \ 0.008 \ 0.005 \ 0.002 \ -0.000]$$

$$V_{Pixel5} = [0.226 \ 1.611 \ 0.914 \ 0.001 \ 0.005 \ -0.002 \ -0.000]$$

Note that the coefficients for the nonlinear terms are very small (close to but not equal zero) due to the larger magnitudes of the non-linear terms. We again plot the predicted power using the derived non-linear model for magenta and white for varying intensities in Figure 2, labeled as NLM. We see that compared to LMLR, the NLM model sometimes over-predicts (for small linear RGB values) and sometimes under-predicts (for large linear RGB values) the power for the white color, but overall exhibits similar accuracy as the LMLR model. The average prediction error is 12%, 18%, 11% and 4% for white and 6%, 6%, 5% and 11% for magenta on the 4 phones, respectively.

Model error across the color space. Since prior-art models cannot even approximate 4 monochromatic colors well, we can expect that they will have high prediction errors in the entire 3-D color space. We further measured the prediction error of the LMLR model at all RGB values of (x, y, z) where x, y, z span all multiples of 16.

³We use 4-color monochromatic images to train these models because they cover the sRGB space and give us $256 \times 4 = 1024$ equations. Training the models using real images may not cover the entire color space or provide enough diversity.

The results, shown in Figure 4, confirm that the LMLR model exhibits high error across the color space on all four phones, with a mean absolute prediction error of 6.90%, 10.35%, 7.10% and 6.11% and 90th percentile absolute error of 15.65%, 22.07%, 16.19% and 15.71%, for the four phones, respectively. Interestingly, the error is lower on Pixel 2, Pixel 4 and Pixel 5, which can be explained by their less violation of the superposition property as shown in Figure 2. Compared to LMLR, we observed that the NLM model (not shown) exhibits similarly high error across the color space on all four phones, with a mean absolute prediction error of 6.79%, 12.94%, 6.02% and 10.92% and 90th percentile absolute error of 15.45%, 28.75%, 13.38% and 28.09% respectively, for the four phones.

5 A NEW OLED POWER MODEL

In this section, we present a practical OLED pixel power model that is more accurate than prior arts and robust to device diversities.

5.1 Key Insight

Since the non-linear model (NLM) achieves similar accuracy as the linear model with linear regression (LMLR), we discuss our key insight using the LMLR model. Recall that LMLR models the pixel power for any given RGB value (R_i, G_i, B_i) as a linear combination of the subpixel power $f(R_i)$, $g(G_i)$, and $h(B_i)$ (Eqn. 6). Since the subpixel power $f(R_i)$, $g(G_i)$, and $h(B_i)$ are linear with respect to the subpixel linear RGB values (Figure 2), their linear combination should be able to and can only fit any plane in the 3-D coordinate space of (R_i, G_i, B_i) . The question then becomes: do the pixel power values $P_i(R_i, G_i, B_i)$ fall on a plane?

Since it is hard to visualize the measured $P_i(R_i, G_i, B_i)$ in the 3-D RGB space, we select a dissection of the 3-D color space with a fixed G value (at 112), and plot the measured power value for the 2-D R-B space (in linear RGB values) in Figures 5(a)-(d) for the four phones, where the height of the top surface shows the display power draw values. We see that the surface is not flat, *i.e.*, it does not fall into a plane! This is the key reason why any particular linear combination of subpixel power functions (which are linear in the subpixel color) cannot approximate the pixel power behavior well across the entire color space. In addition, our experiments in §4 showed that adding non-linear terms cannot fit the nonlinear pixel power behavior well either.

We make a key observation that *although the pixel power values do not fall in a plane across the entire color space, they correspond to a reasonably smooth surface, e.g.*, as shown in the 2-D dissection example in Figure 5. Thus if we cut the 3-D color space into small subgrids, the pixel power values within each subgrid are likely to fit closely in a flat plane and thus can be fitted well using linear combinations of the subpixel (linear) power functions $f(R_i)$, $g(G_i)$, and $h(B_i)$.

Motivated by the above insight, we propose a new pixel power model that can estimate the OLED display power with much better accuracy than the prior art. The basic idea of the new model is to divide the 3-D color space (of dimension $256 \times 256 \times 256$) into small subgrids and derive a piecewise LMLR or NLR model for each subgrid.

Figure 4: Prediction error of the LMLR model in the linear RGB color space, for all sRGB values (r, g, b), r, g, b are multiples of 16.

Figure 5: Measured pixel power values for a 2-D dissection of the 3-D color space on the four phones do not fall into a plane. All phones use default color modes: Boosted for Pixel 2, Vibrant Neutral for Moto Z3, and Adaptive for Pixel 4 and Pixel 5.

Figure 6: A $16 \times 16 \times 16$ subgrid in the 3-D sRGB color space. The power numbers are from Moto Z3.

5.2 Model Derivation

Deriving the above piecewise pixel power model consists of three steps: decomposition of the color space, data collection, and model derivation for each sub color space.

First, we decompose the $256 \times 256 \times 256$ RGB color space into subgrids of size $s \times s \times s$, where s is the granularity of the piecewise pixel power model. The choice of s determines the tradeoff between accuracy and the model size. The model size in turn determines the memory and lookup overheads. Figure 6 shows an example grid of size $16 \times 16 \times 16$ resulted from decomposing the 3-D color space into $16 \times 16 \times 16$ subgrids.

Second, we experimentally measure the pixel power value as before (§3.1) by displaying monochromatic images one at a time corresponding to the color of each grid point in the decomposition. For $16 \times 16 \times 16$ subgrids, there will be $17^3 = 4913$ grid points.

Finally, for each subgrid, we derive a pixel power model by applying linear regression to derive the coefficients of the model, *i.e.*, by fitting the power values at its 8 corner grid points (see Figure 6). We evaluate two model types for each subgrid, LMLR

(Eqn. 8) and NLM (Eqn. 9), and denote the new models as P-LMLR (piecewise LMLR) and P-NLM (piecewise NLM):

$$P_i^{grid_j} = e_1^j \cdot f(R_i) + e_2^j \cdot g(G_i) + e_3^j \cdot h(B_i) + e_4^j \quad (8)$$

$$P_i^{grid_j} = e_1^j \cdot f(R_i) + e_2^j \cdot g(G_i) + e_3^j \cdot h(B_i) + e_4^j \cdot f(R_i)g(G_i) + e_5^j \cdot f(R_i)h(B_i) + e_6^j \cdot g(G_i)h(B_i) + e_7^j \cdot f(R_i)g(G_i)h(B_i) + e_8^j \quad (9)$$

Handling color transformation. For the cases with color transformations, *i.e.*, due to color mode change or color correction, if the color transformation, *i.e.*, the color matrix, is known, the new power model can be derived from the default power model; otherwise, the new power model needs to be experimentally generated.

5.3 Using the Model in Real Time

Using the new OLED power model in real time includes four simple steps: (1) Record a pixel frame, *e.g.*, via a virtual display; (2) For each pixel in the frame, find the sRGB subgrid it belongs to; (3) Apply the piecewise pixel power model of that subgrid to estimate the power draw for the pixel; (4) Sum up the estimated power for all the pixels (Eqn. 1).

On the same four phones, Steps 1–4 take 54ms, 58ms, 46ms and 52ms respectively. This prevents the model from being used in real time, *e.g.*, by enhanced Android Battery (§6.2). To control the execution time of Steps 1–4 to be within a fraction of 16.7ms, *i.e.*, the per-frame rendering time, we apply uniform pixel sampling (experimentally found to be faster than random sampling yet gives similar modeling accuracy) similar as in [23]. We define *sampling ratio* K as the factor of reduction: one pixel is chosen from the center of each grid of $\sqrt{K} \times \sqrt{K}$ pixels. The choice of the sampling ratio

Figure 7: Additional prediction error of and runtime in applying the P-LMLR model under various sampling ratios.

determines the tradeoff between runtime reduction and modeling error. We experimentally evaluate the tradeoff in the next section.

5.4 Model Evaluation

To evaluate the new model, we implemented a power estimator consisting of a sampling-based frame recorder by modifying the Android screenrecord command code, and a power calculator that implements Steps 2–4 in a C++ program with 1 KLOC. It takes a pixel sampling ratio and the phone display power model as input, records a frame, and outputs the estimated display power consumption. Below, we evaluate the accuracy and runtime of the power estimator and how sampling affects their tradeoff.

Methodology. We evaluate the accuracy of the new pixel power model with a set of screenshots for 5 activities each for 6 popular Google apps, Calculator, Calendar, Maps, Phone, News, and YouTube, under both dark and light modes. We measure the ground truth OLED power following the same methodology as in §3. To evaluate the impact of pixel sampling, we vary the sampling ratio from 1 (no sampling) to 4096.

Impact of subgrid size. We first measure the prediction error of P-LMLR and P-NLM models across the activity screenshots from the 6 Google apps under subgrid sizes 16 and 32, on the four phones, respectively. The results show that using subgrid sizes 16 and 32 result in very similar prediction errors. Since using subgrid size 16 only needs 64KB to store the 4096 piecewise models (of 4 coefficients each), which is insignificant, and incurs negligible application time (see below), but can better cope with potential OLED power fluctuations (*e.g.*, on other phones), we use subgrid size $16 \times 16 \times 16$ in our model design.

Impact of sampling ratio. Figure 7 left shows the additional prediction error of P-LMLR when increasing the pixel sampling ratio compared to that of P-LMLR without sampling, for the set of activity screenshots from the 6 Google apps on the 4 phones. We see that increasing the sampling ratio increases the prediction error slowly. The additional error predicted is less than 1% for the 4 phones at the sampling ratio of 100.

Frame recording and power calculation time. Figure 7 right shows how the sampling ratio affects the total time (frame recording time (T1) plus power calculation time (T2)) in applying the P-LMLR model to calculate the power for a given frame. We observe that for all four phones, the total time, as well as the two individual times not shown, are inversely proportional to the sampling ratio. In particular, without sampling, the frame recording time are 10ms,

Table 3: Mean (median) display power prediction error for the 6 app activity screenshots.

Phone	Mode	Average Percentage Error			
		LMLR	NLM	P-LMLR	P-NLM
Pixel 2	dark	21.6 (23.3)	12.8 (13.5)	4.2 (3.9)	3.5 (2.3)
	light	7.2 (7.5)	5.4 (5.5)	1.7 (1.5)	1.7 (1.5)
Moto Z3	dark	18.9 (17.0)	13.1 (11.8)	4.3 (4.5)	4.6 (4.6)
	light	6.5 (7.4)	7.5 (7.9)	2.6 (1.0)	2.7 (0.9)
Pixel 4	dark	17.3 (20.1)	11.8 (13.7)	3.1 (3.1)	3.5 (3.6)
	light	5.0 (5.6)	4.8 (5.5)	2.2 (2.0)	2.3 (2.1)
Pixel 5	dark	7.0 (6.2)	4.6 (3.4)	1.4 (1.2)	1.4 (1.2)
	light	4.1 (4.4)	1.4 (1.6)	0.7 (0.7)	0.7 (0.7)

4ms, 4ms and 10ms and the power calculation time are 44ms, 48ms, 42ms and 42ms on the four phones, while with a sampling ratio of 100, the recording time are reduced to 1ms, 1ms, 1ms and 2ms, and power calculation time are reduced to 4ms, 3ms, 2ms and 1ms, respectively. We note the long frame recording time is due to the delay in pulling data from frame buffer and does not keep CPU busy. As we will show in §6.2, the average CPU utilization of power calculation is only 1.1%, 0.9%, 0.6% and 1.6% on the 4 phones, respectively. We thus conclude that a sampling ratio of 100 strikes a good balance between minimal prediction accuracy loss and sufficient model application runtime reduction.

Accuracy. Table 3 summarizes the average prediction error by LMR, NLM, P-LMLR, and P-NLM for the activity screenshots with 100x sampling on the four phones. The error for LMLR and NLM varies across the phones, because their error for individual pixel colors vary in different regions of the color space on different phones (Figure 2) and different images have different distribution of pixel colors. In contrast, the prediction errors of P-NLM and P-LMLR are almost identical and consistent across the 4 phones, *i.e.*, the two models are robust to phone diversities, and the errors are significantly lower than that of LMLR and NLM. In particular, P-LMLR reduces the average prediction error of LMLR by 5.1×, 4.4×, 5.6× and 4.9× for dark mode and 4.1×, 2.5×, 2.3× and 5.8× for light mode for the four phones, respectively.

6 OLED POWER PROFILING TOOLS

Using the new OLED power model presented in §5, we have developed two OLED power profiling tools.

6.1 Tool 1: Per-Frame OLED Power Profiler

PFOP is an offline per-frame OLED power profiler designed to enable an app developer to profile and optimize the OLED power draw of all the UI components known as views in Android in each app activity design. The objects in these views fall into two broad categories: dynamic objects and static objects. Since dynamic objects such as ads or videos will change in real time, the developer typically focuses on optimizing the power draw of static objects. A second design goal of PFOP is to make it easy-to-use and portable to any unmodified Android versions.

The main challenge of the PFOP profiler design is to identify all the UI components of the displayed frame, *e.g.*, separating dynamic objects from static objects, in order to calculate their corresponding OLED power draw. We also need to remove system components on the screen such as the status bar and the navigational bar.

Figure 8: The PFOP profiler screen shots on Pixel 4.

Table 4: PFOP profiler runtime breakdown (in seconds).

Tool component	Pixel 2	Moto Z3	Pixel 4	Pixel 5
Screen shot	0.41	0.58	0.42	0.32
Gathering Bounding box	0.38	1.08	1.84	1.40
Per-UI component current	1.47	1.32	1.84	1.10
Generating Histogram	0.14	0.15	0.17	0.14
Total frame current	0.02	0.02	0.03	0.02
Total	2.43	3.15	4.30	2.99

The PFOP profiler breaks down the OLED power draw by a frame in three steps. First, to identify all the UI components in a frame, it uses the Android tool “uiautomator” to obtain a dump of the current app screen in xml. Analyzing the xml file gives information about the various views including all the system components on the app screen and their bounding boxes. Because the system UI components are outside the app view hierarchy, PFOP uses the Android command “dumpsys window windows” to gather their bounding boxes. Second, PFOP uses the OpenCV library to extract the pixels within each bounding box. Finally, it applies the P-LMLR pixel power model (Eqn. 8) to derive the OLED power draw for each bounding box and hence the UI component.

To make it easy to use and portable (second design goal), we implemented PFOP as an Android app that only uses built-in tools uiautomator, dumpsys and screencap.

Figure 8 shows the screen shots of the profiler app in action. The app displays a floating icon which is an Android service that overlays on top of the app that is running, as shown in Figure 8a. Whenever the icon is pressed, it sends a message to the background process to capture the screen and screen layout via uiautomator. For screen analysis, the app can toggle between two outputs. The first is a histogram showing the number of color peaks in the frame, as shown in Figure 8b. The second is the bounding box of each selected view and the current per view at 100% brightness, as well as the total screen current at 30%, 50% and 100% brightness, as shown in Figure 8c.

Evaluation. Since PFOP is an offline profiler, we analyze the time it takes to analyze the OLED power draw of a frame. We break down the time due to its 5 processing steps: taking a screen shot,

gathering the bounding box data, generating per-UI component current, screen histogram generation, and calculating the total current for the screen. The results for the four phones are shown in Table 4. We see that taking screen shot, gathering bounding boxes, and histogram generation can take a few seconds. This is the price we pay for designing PFOP to be portable, *i.e.*, only using system commands on unmodified Android versions which do not use sampling. Nonetheless, the total time for analyzing a frame is less than 4.3 seconds on all four phones.

6.2 Tool 2: Android Battery+

Android Battery is a built-in Android utility that helps phone users to manage the phone battery life by monitoring and exposing the battery drain per app running on the phone by various components including the CPU, Wake Lock, Cellular, WiFi, Bluetooth, Sensor, and Camera, but not the screen. Instead, it simply gives the total screen energy used across all apps.

Android Battery is implemented in the Battery Stats system service using simple power models (power_profile.xml [2]) provided by the manufacturers of various phone components. For screen, the model just contains the base (at brightness level 0) and the maximum (at brightness level 255, or 1023 on Pixel 5) power consumption. Android Battery then estimates per-frame display power as a linear function of the brightness level. Its screen power estimation is thus frame-content oblivious and can be highly inaccurate.

Android Battery+. We enhanced Android Battery by replacing its simple screen energy calculation with a new screen energy estimator that uses our new content-based OLED power model and further accounts the screen energy to individual apps. We call the new version Android Battery+, or Battery+ for short.

Like Android Battery, Battery+ estimates the screen energy in real time. We first implemented the new screen energy module by modifying ‘screenrecord’ command code, as a 1000 LOC C++ program. The module runs as a background process that continuously calculates the screen power draw for every frame displayed.

We then modified the Activity manager, Battery Stats, and the Settings app in the Android framework. In particular, we modified the Activity manager to inform the modified Battery Stats of each app running on the screen. The modified Battery Stats then logs the app on-screen time and estimates the OLED power draw and energy drain per app by polling the screen energy background process. In total, Battery+ is implemented by adding 1200 LOC to the Android framework and 20 LOC in the Settings app. We could not implement Battery+ on Moto Z3 as its framework is not open-sourced, and thus we ran the screen energy module directly via the ‘adb shell’ interface in our evaluation in §7.

Evaluation. We first measured the overhead of Battery+’s screen energy module while running the camera app on screen for 10 minutes. As reported in §5.3, at a pixel sampling ratio of 100, the latency of per frame recording is 4ms, 3ms, 2ms and 1ms on Pixel 2, Moto Z3, Pixel 4, and Pixel 5, respectively, while incurring less than 1% additional prediction error. We then isolated the modified Battery Stats process and measured its CPU usage; the average CPU utilization is only 1.12%, 0.92%, 0.59% and 1.55% for the four phones, respectively.

Figure 9: Screenshot of Android Settings. The red box contains new entries added in Battery+.

Figure 10: Comparing Battery+ and Android Battery screen energy for the Google Calculator app.

Next, we compared the screen energy accuracy by Battery+ and Android Battery. We ran the calculator app for 10 minutes twice, first in light mode and then in dark mode, under full screen brightness. Figure 9 shows a screenshot of Battery+ on Pixel 2, where it shows the total screen time and screen energy for the app. Figure 10 plots the total phone energy drain measured using the Monsoon power monitor and the screen energy estimated by Android Battery and Battery+, under dark and light modes, respectively. We observe that (1) Android Battery is oblivious to screen content and incorrectly reports the same screen energy under the two modes; in particular, it over-estimates the OLED display energy drain in dark mode to be 4.62 $\times$ (50.39 mAh vs. 10.91 mAh), 2.15 $\times$ (27.59 mAh vs. 12.82 mAh), 3.75 $\times$ (73.46 mAh vs. 19.57 mAh), 5.22 $\times$ (86.88 mAh vs. 16.63 mAh) higher than the total phone energy drain, on the four phones, respectively. (2) In contrast, Battery+ correctly estimates the impact of the dark mode, as shown by the same non-OLED energy drain, *i.e.*, the same difference between the total phone energy drain and the OLED display energy drain, under light and dark modes.

7 CASE STUDY: POWER SAVING OF DARK MODE

In this section, we present a case study of how the tools help developers and phone users to quantify and manage the power saving

of the dark mode for a set of popular Google apps (each with 1B+ downloads).

7.1 Power Saving for Typical App Usage Scenarios

We first show how Battery+ helps phone users to easily learn the impact of dark mode on OLED and the whole phone battery drain.

Methodology. Table 5 lists the 6 popular Google apps in our experiment which all have recently introduced support for dark mode. We run each app under a typical user interaction (listed in Table 5) that lasts about 60 seconds. To ensure the identical user interactions are performed under both light mode and dark mode, we use UIAutomator 2 and Appium [3] to drive precoded user interactions for the apps and adb to poll the screen energy background process.

Impact of dark mode. Figure 11 shows the total phone power draw and the OLED display power draw portion reported by Battery+ running on the 4 phones for the 6 Google apps at 100% brightness. We make the following observations. (1) In light mode, the OLED display draws a significant portion of the total phone power, ranging between 44% - 71%, 48% - 73%, 49% - 73% and 46% - 63% across the 6 Android apps on Pixel 2, Moto Z3, Pixel 4, and Pixel 5 respectively. (2) Switching from light to dark mode significantly reduces the OLED power draw across all apps, by 40%-81% (average 67%), 34%-82% (average 64%), 36%-76% (average 60%), and 40%-83% (average 66%) across the apps on the 4 phones, respectively. (3) This large OLED power reduction translates into major reduction in the total phone power, ranging between 19%-57% (average 40%), 19%-58% (average 39%), 21%-59% (average 42%), and 20%-69% (average 47%) across the apps on the 4 phones. Interestingly, the overall phone power saving comes out to be similar across the 4 phones. While we used the screen energy background process to measure the precise power reduction in switching from light to dark mode for the same apps, a phone user can easily use Battery+ to get the approximate power reduction value by manually repeating the same app run, *e.g.*, for 1 minute.

Impact of screen brightness. As discussed in §3.3, changing the brightness level affects the OLED power draw in complicated ways on modern smartphones. Table 6 compares the OLED/whole phone power saving from switching to dark mode for the same experiment above under brightness levels 30%, 50% and 100% on the 4 phones.

We make two observations. (1) As expected, the lower the brightness level, the less power saving from switching to dark mode. In particular, the average OLED power saving across the apps on the 4 phones in switching to dark mode is reduced from 67%, 64%, 60% and 66% under 100% brightness level to 27%, 28%, 19% and 24% under 50% brightness level and 15%, 16%, 10% and 13% under 30% brightness level, and the average phone power saving across the apps on the 4 phones is significantly diminished from 40%, 39%, 42% and 47% under 100% brightness level to 9%, 9%, 7% and 9% under 50% brightness level and 3%, 6%, 6% and 4% under 30% brightness level, respectively. (2) The OLED power reductions under dark mode for Google Maps, News and YouTube are lower than those for the other apps. This is because these 3 apps display static (images) or dynamic (videos) embedded objects which are not affected by dark mode. We note that the small negative reduction in total phone power for some apps under 30% brightness happens because the display

Table 5: List of apps and test scenarios used in our dark mode energy saving study.

Apps (version)	Test scenario	Test duration	Actions
Calculator	Various calculations	60 s	Compute the first 20 Fibonacci numbers.
Google Phone	Create a new contact and dial a number	60 s	Create 2 new contacts. Check the call history and dial a number.
Google Calendar	Check, create and delete appointment	60 s	Create a new appointment. Delete the appointment. Check for appointment in the Day, Week and Month views.
Google Maps	Search and navigate	60 s	Search for nearby gas station and start navigation.
Google News	Reading News	60 s	Open a section and scroll in it.
YouTube	Search and watch a video	60 s	Open the app. Search keywords in the search. Pick the first video and watch for 10 secs. Leave the video. Repeat 2 times.

Figure 11: Total phone and OLED power draw under light mode and dark mode on the four phones.

power at this brightness level is already a small portion of the total power and slight perturbation in the experiment causes the total power to go up or down.

7.2 Power Saving for Major App Activities

From app developers' perspective, an app consists of multiple activities, each implementing a user interface for interacting with the app. Hence, to add dark mode to an app, the developer's effort largely boils down to implementing dark mode for each activity. In this section, we show how to use the PFOP profiler to gain insight into the power saving for major activities in the set of Google apps.

Methodology. We manually navigate each of the apps to stop at all its major activities when running under dark mode and light mode, and run the PFOP profiler to analyze the display power draw. In all, we studied 34 activities of the 6 apps, with many app-specific findings.

One expected general finding is that the power saving from switching to dark mode correlates with the amount of embedded dynamic objects (e.g., videos, ads): the more pixels dynamic objects occupy, the less power savings, as they stay the same in light and dark modes. For brevity, we will only present the case study for Google News.

Google News represents the class of apps whose main activities are rich in pre-generated graphical objects such as images (static) or videos (dynamic). Since the graphical objects typically stay the same in light and dark modes, the OLED power saving of such activities depends on the portion of the screen occupied by such embedded objects.

Figure 12 shows the results for the main activity of Google News. The screen is dominated by topic (part 2), news section (part 3),

and embedded images (part 4 and part 5). The remaining sections, Status bar (part 1) and Navigation bar (part 6), are small. Figure 12d shows that the major OLED power saving in switching to dark mode comes from part 2 and part 3 which switch from white color to near-dark color (with RGB values around 50), as shown in the corresponding color intensity histograms in Figures 12(a)-(b).

As expected, the embedded images stay the same in the two modes and contributes to no OLED power saving. The reason that we do not see any obvious peak in the histogram that corresponds to the part-4 image (25% of the screen size) is that the image has continuous textures which make the histogram spread over a wide range of RGB values, as shown in the enlarged histogram of the image only in Figure 12c.

8 RELATED WORK

We already discussed prior work on OLED display power modeling in §4. Below, we discuss previous work on optimizing OLED display power draw via changing screen colors.

Color remapping. Several studies explore the simple idea of selectively dimming portions of the OLED display panel according to the user's focus of attention (FOA) [15, 40] which are generally hard to predict accurately. More recent work focus on color remappings that exploit approximation of original color content that have lower OLED display power draw. In particular, application-specific color remapping modifies individual applications such as games [1], web browsers [24, 31] and web servers [30] to carry out pre-coded color remappings, while system-level color remapping [39] transparently generates color mappings (with controllable quality degradation) for dynamically generated images before displaying them.

Table 6: Power reduction when switching from light to dark mode at different screen brightness levels. X/Y: display power reduction / total phone power reduction. The Phone app is not supported on Moto Z3.

App	Brightness Level											
	Pixel 2			Moto Z3			Pixel 4			Pixel 5		
	30%	50%	100%	30%	50%	100%	30%	50%	100%	30%	50%	100%
Calculator	21%/ 7%	36%/ 12%	81%/ 54%	24%/ 10%	38%/ 12%	82%/ 58%	13%/ 6%	23%/ 10%	70%/ 51%	16%/ 2%	30%/ 11%	77%/ 58%
Google Phone	18%/ 4%	34%/ 9%	81%/ 50%	-	-	-	12%/ 5%	25%/ 11%	76%/ 56%	16%/ 8%	32%/ 14%	83%/ 63%
Google Calendar	21%/ 6%	34%/ 16%	79%/ 57%	21%/ 0%	37%/ 15%	80%/ 58%	13%/ 9%	25%/ 11%	73%/ 59%	19%/ 10%	32%/ 18%	80%/ 69%
Google Maps	8%/ -1%	15%/ 4%	40%/ 19%	8%/ 4%	14%/ 5%	34%/ 19%	6%/ 10%	11%/ 2%	36%/ 21%	7%/ 0%	14%/ 0%	40%/ 20%
Google News	11%/ -1%	22%/ 5%	64%/ 29%	15%/ 9%	27%/ 6%	63%/ 34%	8%/ 4%	15%/ 6%	56%/ 36%	9%/ 4%	20%/ 5%	63%/ 40%
YouTube	12%/ -1%	21%/ 6%	56%/ 28%	13%/ 6%	24%/ 8%	62%/ 28%	8%/ 1%	15%/ 3%	49%/ 27%	10%/ 2%	18%/ 6%	52%/ 31%
Average	15%/ 3%	27%/ 9%	67%/ 40%	16%/ 6%	28%/ 9%	64%/ 39%	10%/ 6%	19%/ 7%	60%/ 42%	13%/ 4%	24%/ 9%	66%/ 47%

(a) PFOP output for light mode

(b) PFOP output for dark mode

(c) Pixel color histogram of the penguin image.

Part	Power Reduction (mA)			
	Pixel 2	Moto Z3	Pixel 4	Pixel 5
1	9	12	11	13
2	64	86	76	94
3	100	135	120	147
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
Total	173	233	207	254

(d) OLED power saving from PFOP per-component output

Figure 12: Google News: Headlines activity.

Dark mode. Dark mode can be viewed as a special case of color transformation that reduces screen power by explicitly flipping white-themed background colors to dark-themed colors. Both Google with Android Q [10] and Apple with iOS 13 [22, 35] have added support for dark mode. Google is also working on Force Dark Mode [9] which will apply dark mode on apps that do not support the mode yet. Dark mode is also being adopted in non-mobile platforms such as laptops and in web browsers [18] which perform color inversion as most web pages have light background with black text. Chrome [43] has added flag settings to force dark mode and intelligent implementation to find and not invert image content.

9 CONCLUSION

In this work, we first presented a measurement study of the power behavior of OLED displays on modern smartphones. Our study shows they violate both the superposition property and the monotonicity property and explains the accuracy limitations of prior OLED power models. We then presented a novel OLED power model based on a key insight that the OLED power draw is relatively smooth within small subgrids of the large 3-D sRGB color space, and hence each can be fitted well via linear regression on the linear subpixel power functions. Our new OLED power model reduces the prediction error of prior models by 5.1 $\times$, 4.4 $\times$, 5.6 $\times$ and 4.9 $\times$ for dark mode and 4.1 $\times$, 2.5 $\times$, 2.3 $\times$ and 5.8 $\times$ for light mode across 4 recent generations of Android phones, respectively. Using the new OLED power model, we developed two OLED power profiling tools: a per-frame OLED power profiler and an enhanced Android Battery. Further, we performed to our knowledge the first power saving measurement of the dark mode for a set of popular Google apps to showcase how the two tools can help developers to optimize the battery drain of different app GUI designs and phone users to understand and manage phone display energy drain. The PFOP [34] app has been released in Google Play, and we plan to open source PFOP in github and Battery+ into AOSP to facilitate further research and development.

REFERENCES

- [1] B. Anand, K. Thirugnanam, J. Sebastian, P. G. Kannan, A. L. Ananda, M. C. Chan, and R. K. Balan. Adaptive display power management for mobile games. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, MobiSys '11*, pages 57–70, New York, NY, USA, 2011. ACM.
- [2] Measuring power values. <https://source.android.com/devices/tech/power/values>. Online; December 6, 2019.
- [3] Appium: Mobile app automation made awesome. <http://appium.io>. Online; August 1, 2019.
- [4] <https://source.android.com/devices/graphics>.
- [5] <https://source.android.com/devices/graphics/arch-sf-hwc.html>.
- [6] G. Association. The mobile economy 2019. <https://www.gsma.com/r/mobileeconomy/>. Online; August 7, 2019.
- [7] T. Bedford. Apple announces dark mode for your iphone – a new feature of ios 13. <https://www.techradar.com/news/apple-announces-dark-mode-for-your-iphone>. Online; June 3, 2019.
- [8] J. Bell. Programmer's guide to gamma correction. https://medium.com/@Jacob_Bell/programmers-guide-to-gamma-correction-4c1d3a1724fb. Online; December 6, 2019.
- [9] K. Bradshaw. Android q beta 5: 'force dark' mode setting works, no longer darkens some google apps. <https://9to5google.com/2019/07/10/android-q-beta-5-force-dark-mode-returns/>. Online; August 7, 2019.
- [10] D. Burke. Final beta update, official android q coming soon! <https://android-developers.googleblog.com/2019/08/final-beta-update-official-android-q.html>. Online; August 7, 2019.
- [11] E. Burrage. Outlook mobile is rolling out dark mode and exemplifies microsoft 365 design. <https://techcommunity.microsoft.com/t5/Outlook->

- Blog/Outlook-mobile-is-rolling-out-Dark-Mode-and-exemplifies/ba-p/813738. Online; December 6, 2019.
- [12] T. Chang, S. S. Xu, and S. Su. Ssim-based quality-on-demand energy-saving schemes for OLED displays. *IEEE Trans. Systems, Man, and Cybernetics: Systems*, 46(5):623–635, 2016.
 - [13] X. Chen, Y. Chen, Z. Ma, and F. C. A. Fernandes. How is energy consumed in smartphone display applications? In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, HotMobile '13, pages 3:1–3:6, New York, NY, USA, 2013. ACM.
 - [14] X. Chen, J. Mao, J. Gao, K. W. Nixon, and Y. Chen. Morph: Mobile oled-friendly recording and playback system for low power video streaming. In *Proceedings of the 53rd Annual Design Automation Conference*, DAC '16, pages 153:1–153:6, New York, NY, USA, 2016. ACM.
 - [15] X. Chen, K. W. Nixon, H. Zhou, Y. Liu, and Y. Chen. Fingershadow: An oled power optimization based on smartphone touch interactions. In *Proceedings of the 6th USENIX Conference on Power-Aware Computing and Systems*, HotPower'14, pages 6–6, Berkeley, CA, USA, 2014. USENIX Association.
 - [16] J. Cipriani. 11 apps that show off ios 13's new dark mode. <https://www.cnet.com/how-to/11-apps-that-show-off-ios-13s-new-dark-mode/>. Online; December 6, 2019.
 - [17] J. Cipriani. After you use dark mode in these 13 android apps, you'll never go back. <https://www.cnet.com/how-to/after-you-use-dark-mode-in-these-13-android-apps-youll-never-go-back/>. Online; December 6, 2019.
 - [18] J. Cohen. How to enable dark mode on your web browser. <https://www.pcmag.com/feature/369568/how-to-enable-dark-mode-on-your-web-browser>. Online; August 7, 2019.
 - [19] <https://support.google.com/accessibility/android/answer/6151850?hl=en>.
 - [20] <https://developer.android.com/reference/android/graphics/ColorMatrix>.
 - [21] G suite updates. <https://gsuiteupdates.googleblog.com/2019/09/dark-theme-android-ios.html>. Online; December 6, 2019.
 - [22] New features available with ios 13. <https://www.apple.com/ios/ios-13-preview/features/>. Online; August 7, 2019.
 - [23] M. Dong, Y.-S. K. Choi, and L. Zhong. Power modeling of graphical user interfaces on oled displays. In *Proceedings of the 46th Annual Design Automation Conference*, pages 652–657. ACM, 2009.
 - [24] M. Dong, Y.-S. K. Choi, and L. Zhong. Power-saving color transformation of mobile graphical user interfaces on oled-based displays. In *Proceedings of the 2009 ACM/IEEE International Symposium on Low Power Electronics and Design*, ISLPED '09, pages 339–342, New York, NY, USA, 2009. ACM.
 - [25] M. Dong and L. Zhong. Chameleon: A color-adaptive web browser for mobile oled displays. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 375–376, New York, NY, USA, 2011. ACM.
 - [26] S. R. Forrest. The road to high efficiency organic light emitting devices. *Organic Electronics*, 4:45–489, 2003.
 - [27] M. Jain. Introducing dark mode on microsoft intune company portal for ios. <https://techcommunity.microsoft.com/t5/Enterprise-Mobility-Security/Introducing-dark-mode-on-Microsoft-Intune-Company-Portal-for-iOS/ba-p/918453>. Online; December 6, 2019.
 - [28] D. Kim, W. Jung, and H. Cha. Runtime power estimation of mobile amoled displays. In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 61–64. IEEE, 2013.
 - [29] A. Li. The ultimate guide to dark mode for email marketers. <https://litmus.com/blog/the-ultimate-guide-to-dark-mode-for-email-marketers>. Online; December 6, 2019.
 - [30] D. Li, A. H. Tran, and W. G. J. Halfond. Making web applications more energy efficient for oled smartphones. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE 2014, pages 527–538, New York, NY, USA, 2014. ACM.
 - [31] D. Li, A. H. Tran, and W. G. J. Halfond. Nyx: A display energy optimizer for mobile web apps. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2015, pages 958–961, New York, NY, USA, 2015. ACM.
 - [32] R. Mittal, A. Kansal, and R. Chandra. Empowering developers to estimate app energy consumption. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 317–328. ACM, 2012.
 - [33] J.-H. Park and Y.-J. Kim. Accurate power model for mobile amoled displays. *Electronics Letters*, 51(7):553–555, 2015.
 - [34] PFOF in Google Play. <https://play.google.com/store/apps/details?id=com.pdeveloper.pcav5>. Online; December 4, 2020.
 - [35] M. Potuck. How to use dark mode on iphone in ios 13. <https://9to5mac.com/2019/06/04/how-to-use-dark-mode-iphone/>. Online; August 7, 2019.
 - [36] Monsoon power monitor. <http://www.msoon.com/LabEquipment/PowerMonitor/>.
 - [37] B. Schoon. Here's every google app with a dark mode, and how to enable it. <https://9to5google.com/2019/10/10/google-dark-mode-apps-gallery/>. Online; December 6, 2019.
 - [38] J. Shinar. *Organic Light-Emitting Devices: A Survey*. Springer, 2004.
 - [39] P. Stanley-Marbell, V. Estellers, and M. Rinard. Crayon: Saving power through shape and color approximation on next-generation displays. In *Proceedings of the Eleventh European Conference on Computer Systems*, EuroSys '16, pages 11:1–11:17, New York, NY, USA, 2016. ACM.
 - [40] K. W. Tan, T. Okoshi, A. Misra, and R. K. Balan. Focus: A usable & effective approach to oled display power management. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 573–582, New York, NY, USA, 2013. ACM.
 - [41] M. Wan, Y. Jin, D. Li, J. Gui, S. Mahajan, and W. G. J. Halfond. Detecting display energy hotspots in android apps. *Journal of Software: Testing, Verification, and Reliability*, 27(6):1–15, 2017.
 - [42] C. Welch. Google confirms dark mode is a huge help for battery life on android. <https://www.theverge.com/2018/11/8/18076502/google-dark-mode-android-battery-life>. Online; November 8, 2018.
 - [43] D. Wilde. Chrome canary 77 tweaks dark mode on web pages, no longer inverts image colors. <https://9to5google.com/2019/07/16/chrome-canary-77-dark-mode/>. Online; August 7, 2019.