Network Externalities and App Store Fees in Mobile Platforms

Amandeep Singh, Kartik Hosanagar, Aviv Nevo*
The Wharton School, University of Pennsylvania, Philadelphia, PA 19104
{amansin, kartikh, anevo}@wharton.upenn.edu

The benefits consumers get from a mobile phone depend on the software apps available to use on that phone. This creates an interdependence between the demand for handsets and the demand for software apps. To quantify the importance of these indirect network externalities, we develop a model of consumer demand for handsets and apps, as well as a model of developer entry decisions. We estimate the model using a unique dataset of mobile phone sales and app downloads. We find that a 1% increase in the number of apps results in a 1.5% increase in smartphone sales, with significant heterogeneity in this network externality between apps on platforms. To illustrate the applicability of the model, we compute the equilibrium platform fees for Apple and Android. We find that the complementarity between the handset sales and app demand implies lower platform fees with Apple's tight vertical integration. We estimate that increasing the strength of indirect network externalities by 10% results in a decrease in optimal fees by 2.5% for Apple and 0.5% for Android indicating a tight coupling of the two sides of the market for Apple, compared to Android.

Key words: Platform Competition; Indirect Network Effects; Online Marketplace; Two-sided Markets.

1. Introduction

Mobile platforms bring together consumers, who use the platform to access apps, and app developers, who use the platform to distribute their apps to consumers. The two-sided nature of mobile, or smartphone, platforms creates interdependent relationships between consumers and app developers, as the value of the platform to one user group depends on the size and engagement of the other. The interdependence creates a cycle where more consumers attract more app developers, and having more and better apps attracts consumers to the platform.

In this paper, we quantify the indirect network externalities, thereby determining the economic value of apps, an aspect of the market that is under explored despite the economic significance of the mobile apps market. We then illustrate the implications of our estimates for the equilibrium fees

* Aviv Nevo is currently on leave from University of Pennsylvania serving as the Director of the Bureau of Economics at the Federal Trade Commission. The views expressed in this paper do not necessarily reflect the views of the Commission or any individual commissioner.

that the platforms charge app developers. To do so, we develop and estimate a structural model of consumer demand for both handsets and apps, as well as a model of developer entry decisions. Our model captures the dynamic and interdependent nature of the smartphone market. This allows us to conduct counterfactual analyses — a key objective of our study — to understand how changes in the fees charged to developers by platforms like Apple and Android would impact the demand for apps and handsets. Consequently, this allows us to compute the equilibrium platform fees for Apple and Android, shedding light on the viability of the current fee structures.

Our research context has two significant complexities. First, the demand for both handsets and apps is dynamic. Handsets, being durable goods, are used by consumers for extended periods, while new product releases frequently occur in this rapidly innovating market. We therefore follow Gowrisankaran and Rysman (2012) and Lee (2013) and model consumers as forward-looking taking into account current and future prices, features, and the "quality" of the apps available on the relevant app store. Similarly, demand for apps is dynamic since once an app is downloaded a consumer is less likely to download it again. As shown in our results later, not accounting for this dynamic nature could lead to misleading results. Second, the externality exerted by apps on handset demand could be highly heterogeneous. Using the size of the complementary market is not only a poor approximation when there is significant heterogeneity in network effects but it also does not allow us to simulate the effect of platform fees on handset demand. We model the demand for handsets as a function of the quality of the available apps, which is derived directly from the app demand model. This allows us to model, and estimate, heterogeneity in network externalities across apps and simulate the effect of varying app quality levels on handset demand. On the developer side, we develop a model of developer entry and use an inequality based approach developed in Pakes et al. (2015) to estimate the entry cost across the two platforms.

We find that indirect network externalities are significant in app markets for apps that appeared at least once in the top 500 list during the two-year period of our study. On average, increasing the number of these apps by 1% leads to an increase in sales of smartphones by 1.5%. Further, we find evidence for significant heterogeneity in externalities exerted by these top apps. For instance, we find that the externality exerted by the top app is around 2700 and 5600 times that of the median app among those that appeared at least once in the top 500 on iOS and Google Play platforms respectively. We observe that Social Media apps like Facebook have the highest estimated impact on both platforms. Specifically, during the period that our dataset spans, removing Facebook from Apple or Android would have reduced the demand for respective handsets by 5.9% and 5.6%. Using our model we find that one month of access to the Facebook app was worth to be around $13 (or $9) to Apple (or Android) users during this time period.

The interdependence in demand, generated by the indirect network effects, has implications for the optimal platform fees. The optimal platform fees balance several effects, which we illustrate by examining the optimal fee of Apple and Android. The optimal fees need to balance the standard trade-off between margin and volume: a lower fee attracts more developers but implies lower revenues from each app download. Note, that this trade-off has dynamic implications because of the structure of demand in this industry, and will likely mean a different price for Apple and Google. The interdependence of demand creates an additional trade-off, that mostly impacts Apple's optimal fee. Apple profits from handset sales to a far greater extent than Google and therefore internalizes the inter-dependency of the demand for handsets, in a different way than Google.[1]

We find that the equilibrium fees for both Apple and Google are similar, with Apple's at 16% and Google's slightly higher at 19%.

Interestingly, the fact that Google's optimal fee is slightly higher showcases the crucial role of Apple's integration of handset and the app store. Apple's substantial handset sales exert downward pressure on the developer fees it should charge, creating a complementary relationship that Google lacks or experiences to a lesser extent. This differential also illustrates that Google's incentive to lower fees to attract developers may not be as significant as one might initially assume. Finally, our analysis indicates that increasing the strength of indirect network externalities by 10% results in a decrease in optimal fees by 2.5% for Apple and 0.5% for Android. This pattern further indicates that vertical integration within the Apple platform results in much tighter coupling of the two sides of the market, compared to Android.

Our paper makes two main contributions. First, we quantify the indirect network externalities and the economic value of apps in smartphone platforms. This analysis is motivated, in part, by legal interest in two-sided markets and the importance of indirect network externalities. In Ohio v. American Express Co. the Supreme Court discusses the importance of "pronounced indirect network effects and interconnected pricing and demand" (United States Supreme Court (2018) page 13). While the exact definition of "pronounced" remains ambiguous, our paper offers a way to measure the effects. Second, we apply our model to run several counterfactual analyses of the mobile market, including an analysis of how indirect network effects impact the fees platforms charge developers. The fees imposed on developers have been contentious and at the center of multiple industry debates and lawsuits. Initially, platforms levied a 30% fee, a figure that many developers argued was excessive, especially in light of the value they bring to the platforms. In

[1] At the time of our study Google did not sell Pixel phones and had only a small share in the Android handset market.

recent years, platforms have revised these fees downwards.[2] We investigate how sensitive the fees are to indirect network effects.

The issues we study in this paper extend beyond mobile apps and are relevant to multiple platform markets such as those for smart TVs (such as Amazon's Fire Stick, and Roku TV), emerging home assistance platforms (such as Amazon's Alexa and Google Home), etc. Accordingly, our results and insights have implications in those markets as well.

2. Related Literature

Our research is connected to multiple streams of literature both theoretical and empirical both of which are highly relevant in the current digital landscape. Firstly, our work is connected to the work on the measurement of indirect network effects of apps on handset demand. In exploring indirect network effects in platform markets, our work stands on the shoulders of seminal research such as Nair et al. (2004), Dubé et al. (2010), Corts and Lederman (2009), Lee (2013), and Chintagunta et al. (2018). This existing body of research opened the door to the study of network externalities. We extend this foundation by providing empirical evidence of these externalities in app markets, noting the significance of the existence of a significant amount of heterogeneity across apps. Prior studies such as Brynjolfsson et al. (2018) and Brynjolfsson et al. (2019) have sought to estimate the economic value that consumers attach to many influential apps, primarily relying on survey data. By utilizing a structural model of consumer demand and purchase decisions we provide an alternate way to measure this economic value more accurately and address the limitations of survey methods highlighted by Hausman (2012).

In line with our focus on the network effects on handset demand, we also take inspiration from early work in telecommunications and informatics, specifically studies on network effects on the iPhone platform (Garcia-Swartz and Garcia-Vicente (2015); Lee (2014)). We extend this exploration by accounting for the inherent endogeneity in the growth of two-sided markets, thereby enabling a causal estimation of the impact of apps on handset demand.

Simultaneously, our study connects to the burgeoning literature regarding the adoption and drivers of apps and app marketplaces. While Ghose and Han (2014), Li et al. (2016), and Bresnahan et al. (2014) have studied various aspects of app demand, none has explicitly quantified the role of indirect network effects, a gap we intend to fill.

The second strand of literature we contribute concerns the platform fees charged by Apple and Android to app developers. The theoretical foundations laid by Rysman (2004), Rysman

[2] As of 2023, in the United States, both Apple's App Store and Google's Play Store primarily charge a 30% fee on apps and in-app purchases but reduce the fee to 15% for subscriptions, small businesses, and certain other qualifying criteria (see AppleInsider (2023), Help (2023), and Verge (2023).

(2009), and Hagiu (2006) paved the way for understanding pricing and platform competition in two-sided markets. Our work also relates to empirical work in this area like Chintagunta et al. (2018) and Conlon (2012). Recently, app platform fees have become a contentious issue, with accusations of anti-competitive behavior. We enrich this discourse by identifying empirically the market equilibrium fees that Apple and Android should charge and how it is affected by the strength of indirect network externalities, bringing a data-driven perspective to the debate.

In conclusion, this study contributes significantly to the literature on indirect network effects and platform competition. Our insights into the economic value of apps, heterogeneity in network externalities, and the determination of platform fees provide novel dimensions to these research domains. Our findings are set to inform future discourse around digital platform regulation, benefitting policymakers, regulators, and legal professionals alike.

3. Data and Descriptive Analysis

In this section, we describe our data and provide an initial descriptive analysis to motivate the model that follows.

3.1. Data

We use panel data on monthly smartphone sales and daily app downloads in the U.S. from January 2013 to December 2014. The smartphone data and the apps data come from two different sources.

The smartphone sales data are from Counterpoint Research, a marketing research firm that tracks smartphones sold in the U.S. We observe the number of smartphones sold and the average price of each handset sold every month. We combine these sales data with handset characteristics. After dropping handsets that report either zero sales or price, we are left with data for 17 Apple and 53 Android-based handsets. In Table 1, we display the summary statistics of the characteristics of these handsets. Overall, 40% of the sales observed in our dataset are of Apple handsets, with the rest of sales of Android devices. iPhone 5S and Samsung S4 are the highest selling Apple and Android devices, respectively. The average price of Apple handsets is almost double the average price of Android handsets, despite the fact that Android handsets on average have larger displays, more RAM, more storage, and a longer battery life.

The app data come from App Annie, a firm that collects data from both Google Play and the Apple app store. We observe the list of top 500 apps downloaded (across all genres) on the Apple and Google Play U.S. store each day from 2013-2014, the same two-year period for which we observe handset sales.[3] This process results in $6,950$ and $13,731$ unique Google Play and Apple apps,

[3]The data for Android apps are only from Google Play and not other app marketplaces. The same is true for Apple, but unlike the Android ecosystem, there are no other app stores on Apple devices. This is not a major limitation given that Google Play has the lion's share of the Android app market in the U.S. We focus on the top 500 apps downloaded each day since we believe that apps downloaded less frequently are not likely to have a significant impact on handset demand.

Table 1. Handset Characteristics

	Apple		Android		Overall	
	Mean	Std	Mean	Std	Mean	Std
Price ($)	559	137	305	134	365	172
Monthly Sales (million units)	0.36	0.39	0.17	0.28	0.22	0.32
Display Size (inches)	4.0	0.4	4.6	0.6	4.5	0.7
Primary Camera (Megapixels)	8.0	0	8.0	3.7	8.0	3.2
RAM (Megabytes)	985	135	1526	630	1398	600
Storage (GB)	35	26	18	17	22	22
Battery (milliamp hours mAh)	1594	336	2276	531	2114	571

Note. The summary statistics are computed across 70, 17 Apple, and 53 Android, handsets over the 24-month period. Display size refers to the diagonal length of the screen measured in inches. Primary camera refers to the resolution of the back camera measured in megapixels. RAM is the random access memory of the handset measured in MB. Storage is the physical capacity available in the handset, measured in GB. Battery is measured in milliamp hours mAh.

respectively.[4] For each of these apps, we observe the genre, daily price and sales rank, number and prices of its in-app purchase assortment, daily review activity, app developers' (firm) characteristics, and versioning activity. We want readers to note that due to our data collection process if an app drops below rank 500 it is still tracked and not dropped from the dataset. We convert the daily sales rank into an estimate of the daily downloads for each app by building on methods proposed by Garg and Telang (2013). For details please see Appendix E.

In this paper, we aim to investigate the heterogeneous effect of apps on handset demand. This raises the question of whether app assortments are substantially different across the two platforms. In Table 2, we display the summary statistics of the characteristics of apps in our sample. The paid apps available on the Google Play store are priced higher, on average, than the Apple iOS store. As we previously noted, the Apple Store has more apps, a fact we need to account for when comparing averages. Indeed, the price difference is driven by the mix of apps: when we compare prices for the same set of paid apps we find that the difference between the stores is not statistically significant(i.e., $2.81 versus $3.09 (p-value = 0.15) on Apple vs Google Play store, respectively). We find that customers on Android leave many more reviews (and higher ratings) than on Apple iOS. This difference persists even when we hold the set of apps constant across the two app stores (1331 versus 3608 monthly reviews (p-value < 0.01) and 3.89 versus 3.96 average ratings (p$-$value $= 0.03$) on Apple vs Google Play store, respectively). This might be driven by the subtle differences in their review system policy: Google Play displays apps' reviews over its entire history (i.e., across versions) whereas Apple restarts apps' reviews to cover only their most recent version released.

[4]One potential reason for sampling more unique apps from the Apple app store compared to Google Play, could be more developers choosing to launch their apps on Apple first. Since developers not able to profitably operate their apps on Apple might not choose to launch on Google Play later, it could create a potential imbalance across the two app stores.

We also find that Apple developers tend to iterate over versions quicker than Android for even the same app (2.21 versus 4.34 months (p-value < 0.01)). This could reflect the more demanding and competitive nature of the Apple marketplace. Finally, we also observe that developers on Google Play tend to maintain bigger portfolios than Apple developers.

These subtle differences in app characteristics across the platforms can create varying levels of quality perceptions, even among the same apps on Apple iOS and Google Play. This difference in quality levels, even among the same apps serves as a source of variation for identifying apps' externalities on handset demand.

Table 2. App Characteristics.

	iOS		Google Play		Overall	
	Mean	Std	Mean	Std	Mean	Std
Price	3.14	4.30	3.91	3.15	3.40	5.35
Downloads (Daily)	353	1425	259	1292	326	1389
# of Monthly Reviews	783	3018	2091	5799	1234	4235
Average Customer Review Rating (Quality)	3.95	0.70	4.03	0.50	3.97	0.63
Average Version Age (months)	2.20	2.90	4.50	6.01	2.98	4.36
# of Versions	12	10	13	15	13	12
Firm Size	17	32	25	47	20	38

Note. The summary statistics are computed across our entire sample of 13,731 ios and 6,950 Google Play apps over the 2-year period. For price, we only report statistics for paid mobile apps. # of Monthly Reviews refers to the volume of customer reviews received in the focal month. # of Versions refers to the number of versions released for each app as observed in December 2014. Average Customer Review Rating refers to the average valence (out of 5) of customer reviews received by the app in the focal month. Firm Size refers to the portfolio size of the apps' developers in our sample. 59% of Apple apps have an In-App Purchase option vs 56% for Google Play apps.

3.2. Evolution over time

We now turn to the evolution of characteristics over time, which will motivate our modeling approach below. In Figure 3 and 4, we illustrate the trends related to the handset and app prices, app quality as a function of age, and handset sales and app downloads over time. As we can see in panel 3a, retailers tend to discount handsets as they age. This pattern is found in many durable products and is usually attributed to intertemporal price discrimination: retailers set a higher price early to extract surplus from the high-valuation consumers and then discount their prices to attract the lower-valuation consumers.[5] Another observation is to look at the pricing of iPhones around

[5]Apple never puts explicit discounts on its handsets until the launch of a newer version. However, the average sales price of an Apple handset can still show a slight downward pattern due to the various promotions run by retailers. Counterpoint Research takes into account these retail promotions to calculate the average selling price of Apple handsets.

the release of new models. In Figure 5, we present a comparative analysis of average prices for the iPhone 4 and its predecessors two months before and after the release of the iPhone 5 and its variants. The figure distinctly reveals a significant discounting of older iPhone models post the release of newer ones. This pattern, consistent with practices observed in many durable goods markets, suggests that manufacturers employ intertemporal price discrimination strategies to maximize their profits over the product lifecycle. These strategies may create incentives for consumers to anticipate future discounts and adjust their purchasing decisions accordingly, thereby exhibiting forward-looking behavior.

In panel 3b, we show a slightly reverse pattern in apps' pricing: apps' prices slightly increase as they age. In Panel 3c, we show that app quality, as measured by the app's average rating received in the focal month, shows an upward trend as the app ages. This indicates that as the apps mature their quality levels (as perceived by customers) improve on average.

In Figure 6, we examine how handset characteristics (camera, display size, battery, ram, storage) are evolving over time across Apple and Android. Overall, we find that the industry has been steadily improving in each respective handset feature. As with prices, customers can also time their purchases in anticipation of better handset characteristics. As we discuss in the next section, static demand models, which do not account for such anticipatory consumer behavior, may lead to biased estimates of demand elasticities.

3.3. Overlap in Apps offering

In this section, we elaborate on the decisions made by developers that have chosen to offer their apps across both platforms. Due to the interconnected nature of smartphone and app demand, decisions made by developers affect consumer choices. In our sample, 2103 apps launched on both Apple and Google Play (as observed up to two years after the period of our study, i.e., up until Jan 2017).

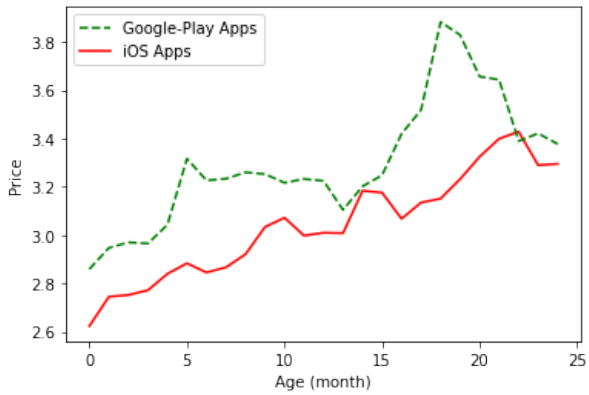
In Table 7, we report entry decisions for apps with a presence on both platforms in our sample. We find developers have a strong preference for the Apple platform. For instance, the majority (64.6%) of app developers prefer deploying an app on iOS first vs. Android. Specifically, apps are launched on the Apple App Store an average of 6 months before Google Play. This could affect the Android platform's attractiveness to customers and could be the reason for concerted efforts[6] carried out by Android to attract developers towards its platform. Across the various genres, the gap between Apple and Android launches is the longest for apps requiring more development resources (for instance music, photo and video, and social networking) and the least for generic apps (like utilities).

[6]See https://medium.com/jay-tillu/flutter-googles-most-ambitious-framework-e3b36ca7091f and https://stablekernel.com/article/the-business-case-for-google-flutter/ for more discussion.

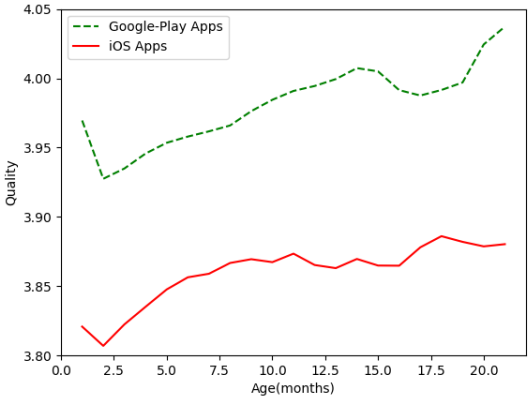
Figure 3. Handset and App Characteristics over Time



(a) Handset Prices



(b) App Prices



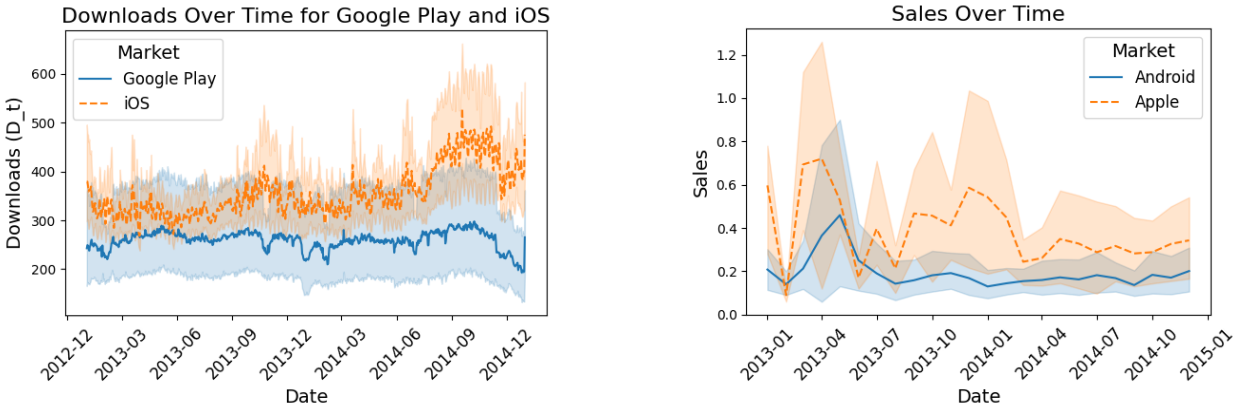
(c) App Ratings

Note: The figures present the evolution of handset and smartphone prices and quality (average customer rating) as they age. In Panel 3a, we present the evolution of smartphone prices as they age. The average is calculated over the mix of all handsets of a specific age. In panel 3b, we present the evolution of apps' prices (only paid apps) as they age. For this purpose, we only average over apps that were launched in and after January 2013. In panel 3c we present the evolution of app quality measured by the average customer ratings they received in the focal month as they mature.

4. Model

In this section, we present a model that examines the influence of indirect network externalities on mobile platforms. On the demand side, our model accounts for consumers' decisions to purchase a handset and download an app from the App Store. As seen in the previous section, the price and attributes of handsets and apps evolve over time, which incentivizes consumers to strategically time their choices. To capture consumer demand for both handsets and apps, we employ a dynamic model. We also provide empirical evidence demonstrating how neglecting consumer forward-looking behavior could result in inaccurate estimates for both the handset and app models.

Figure 4. Evolution of Handset Sales and App Downloads over time

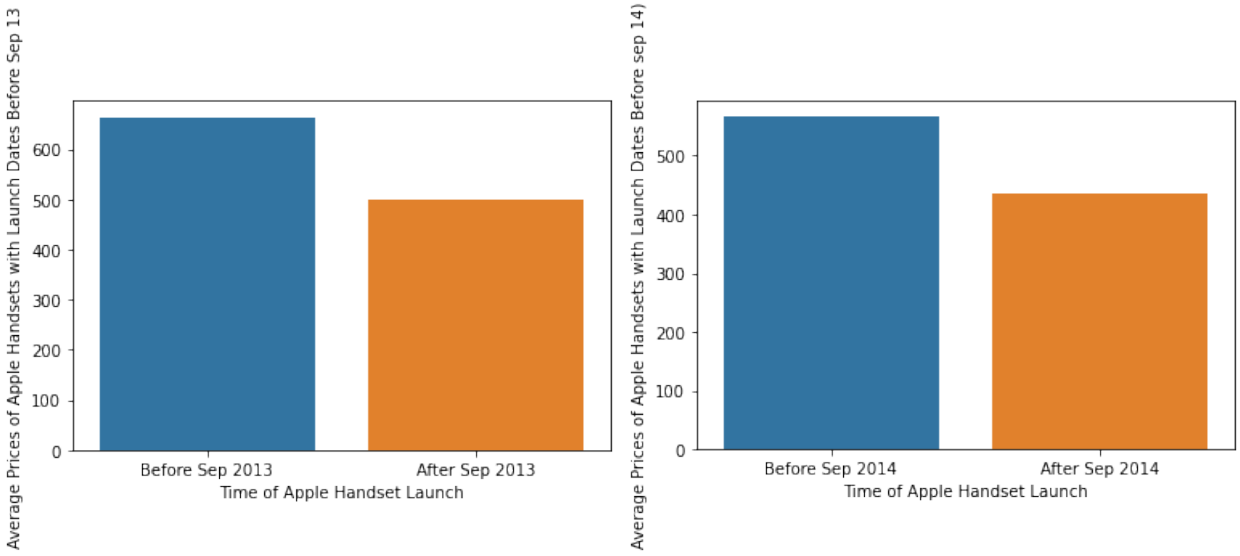


(a) App Downloads

(b) Handset Sales

Note. The figures report how handset sales and app downloads evolve over time for both Android and Apple handsets. The handset sales are measured in million units. The shaded region in both figures indicates the 95th percentile region for the measured metric.

Figure 5. Apple handset prices two months before and two after the launch of a new model.

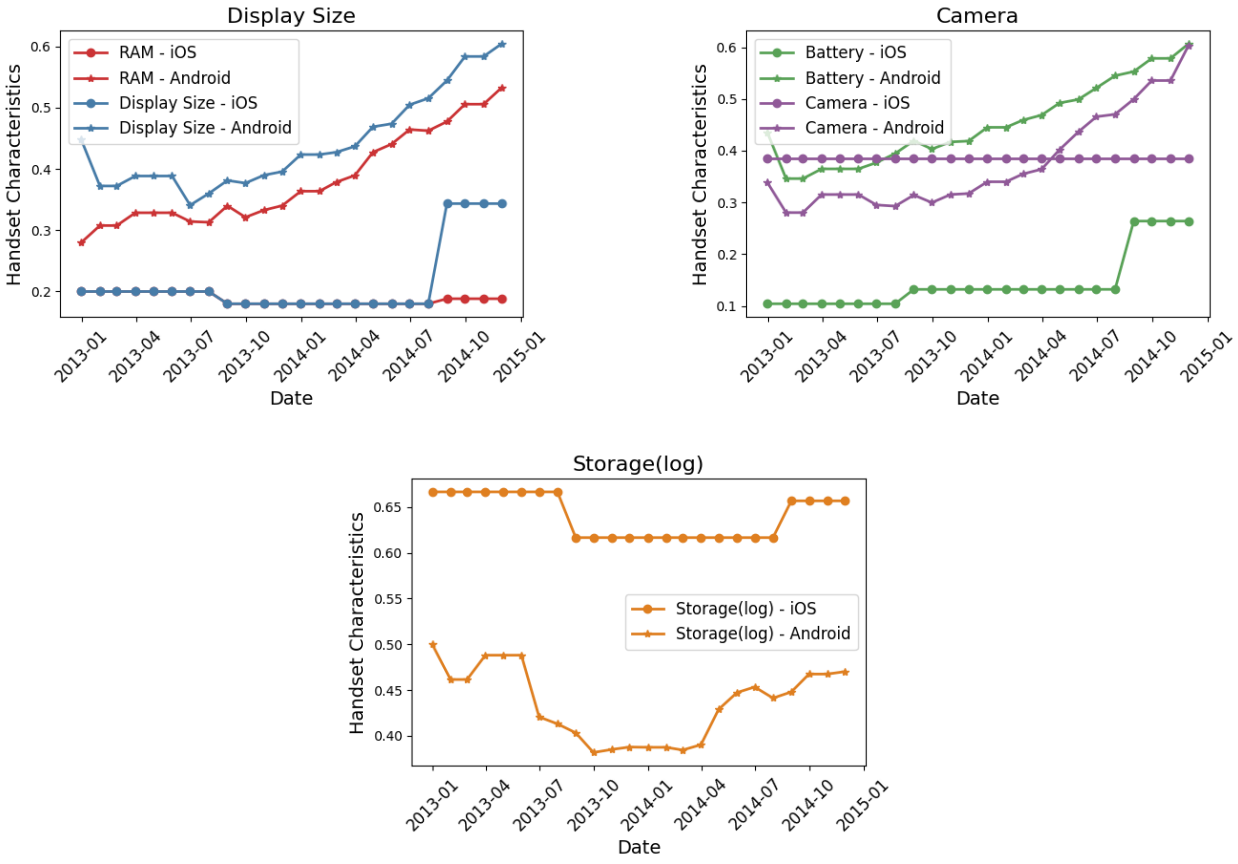


(a) Effect After Launch Event Sep 13

(b) Effect After Launch Event Sep 14

Our model closely follows Lee (2013), which investigates the decisions of video game platforms and games. On the supply side, we model app developers' choices to offer their apps on two platforms. Specifically, after obtaining demand estimates, we analyze app developers' observed entry

Figure 6. Evolution of Handset Characteristics.



Note. The figures report how the handset features for the entire mix of handsets in the market evolve over time. The lines with the circle marker denote the average value of Apple handset characteristics over time and the star marker lines indicate the average value of Android handset characteristics over time. For the sake of convenience, each handset characteristic is normalized to a value between 0 and 1. Time period 1 denotes January 2013 and Time period 24 denotes December 2014.

patterns to estimate entry costs. This sequential estimation strategy ensures that any potential misspecification in the app developer entry model does not affect the demand estimates.

4.1. Demand for Handsets

We first consider a consumer's handset purchase decision. Handsets are durable goods that are bought roughly every 3 years.[7] Consumers, in general, can time their purchases, and indeed, when faced with declining prices for a given model (Figure 3) and improving attributes across models (Figure 6), they have incentives to do so. For instance, price-sensitive consumers may wait for the price of handsets to decline. Similarly, the less price-sensitive consumers may purchase early and leave the market, making the mix of consumers in later periods more price-sensitive. Therefore, to

[7]See https://www.itsworthmore.com/blog/post/years-a-new-smartphone-last.

Table 7. Developer Entry Decisions

Overall Statistics

Apple First	64.6%
Google Play First	12.2%
Both	23.2%

Statistics by Genre

Genre	Days
Entertainment	196
Games	151
Music	305
Photo and Video	292
Social Networking	238
Utilities	103
Others	232

Note: In this table, we report summary statistics for entry decisions of app developers. "Apple First" denotes the share of app developers that chose to launch on iOS first. "Google Play First" denotes the share of app developers that chose to launch on Google Play first. "Both" denotes the share of app developers that launched on both platforms at the same time. We also report the difference between Android and Apple launch dates (in days) across various genres.

consistently estimate demand elasticities we need to account for this dynamic behavior.[8] In Section 8, we compare the results from a dynamic model of consumer demand to a static model and find, how not accounting for the dynamic behavior of customers can lead to misleading estimates.

In each period t (month) the consumer decides whether to buy one of H_t handsets available at time t or wait until the next period. If consumer i purchases handset $j \in H_t$ at time t they get utility

$$u_{i,j,t} = \underbrace{\boldsymbol{\alpha}_i^x \vec{x}_{j,t} + \alpha_i^{p,hs} \log p_{j,t} + \alpha^{\Gamma} \Gamma_{g(j),t}(\alpha_i^{p,app}, \boldsymbol{\alpha}_i^w) + \xi_{j,t}}_{\delta_{i,j,t}} + \epsilon_{i,j,t}^{(hs)} \tag{1}$$

where, $\vec{x}_{j,t}$ refers to the observable characteristics of the handset at t, including camera resolution, display size, ram, storage, and handset fixed effects. $\boldsymbol{\alpha}_i^x$ denotes consumer i's tastes for these characteristics; $\alpha_i^{p,hs}$ captures consumer i's (handset) price sensitivity; $\xi_{j,t}$ captures (time varying) characteristics of handset j that are unobserved by the econometrician but are valued, and observed, by the consumers. Unobserved characteristics, for instance, could include unobserved

[8] See Melnikov (2013), Gowrisankaran and Rysman (2012), Conlon (2012), Aguirregabiria and Nevo (2013) for further discussion.

promotional activities, unobserved handset equity, or systematic demand shocks. Some components of the unobserved characteristics are fixed over time and can be captured using dummy variables. Therefore, we model $\xi_{j,t} = \bar{\xi}_j + \Delta\xi_{j,t}$. The error term $\Delta\xi_{j,t}$ is the time varying component of the unobserved attributes. The term $\Gamma_{g(j),t}(\cdot)$ captures the utility the consumer gets from the apps (software) available at the app store $g(j)$. This term varies between Apple and Android phones. For now, we will assume that it is known, and in the next section discuss how we model it. Finally, $\epsilon^{(hs)}_{i,j,t}$ captures random shocks that are assumed to be independently and identically distributed according to the type I extreme value distribution.

We assume that if the consumer buys a handset they exit the market for handsets. Therefore, $u_{i,j,t}$ can be viewed as the lifetime value from the purchase of product j. As we noted above, on average consumers replace their handset roughly every 3 years. We observe two years of data and therefore, modeling repeat purchases is not a first order concern (or clearly identified using our data). Furthermore, such a simplification significantly reduces the complexity of the problem and makes identification of primary effects of interest (i.e., network effects) easier. We also assume that there is no resale of handsets.

Consumers have the option of not purchasing in period t, in which case they can consider purchasing in the future. The consumer i who does not purchase a new handset at time t, i.e. chooses the outside option (denoted as $j = 0$), gets the one-period utility

$$u_{i,0,t} = \epsilon^{(hs)}_{i,0,t} \tag{2}$$

where $\epsilon^{(hs)}_{i,0,t}$ is assumed to be independently and identically distributed according to the type I extreme value distribution. In principle, consumers might own an older handset. This will in part be captured by $\epsilon^{(hs)}_{i,0,t}$.

At time t, the consumer is faced with $|H_t| + 1$ choices, where H_t denotes the set of handsets available at time t. The consumer chooses the option that gives her the maximum expected value, given her information set at t. At the time of choice, consumers know prices and attributes (including those unobserved by us) of all products. They also observe all (current) $\epsilon_{i,j,t}$. Let Ω_t be the state variable that captures all relevant information used by consumers to make their decision. This state will include the set of handsets, their prices, attributes, and associated app store utilities ($\Gamma_{g(j),t}(\cdot)$), and will change over time. Consumers form expectations about the evolution of these variables, as we discuss below, but do not know future values $\vec{\epsilon_i}$.

The Bellman equation for the consumer's problem can be written as

$$V(\Omega_{it}, \epsilon_{it}) = \max_{A_t \in (0,1)} \begin{cases} \epsilon^{(hs)}_{i0t} + \beta\, \mathbb{E}(V(\Omega_{it+1})|\Omega_{it}), & A_t = 0 \text{ if no purchase} \\ \max_{j \in H_t}\{u_{ijt}\}, & A_t = 1 \text{ if buys a handset} \end{cases} \tag{3}$$

where, $V(\Omega_{it}) = \int V(\Omega_{it}, \epsilon_{it}) dF_\epsilon(\epsilon_{it})$ is the integrated value function.

The dimension of the state space in the above problem is very high, and therefore the problem is impractical to solve (and take to the data). We therefore follow the literature to reduce the dimensions of the problem in two steps.

First, following Rust (1994), we integrate the logit error ϵ and rewrite (3) using the integrated value function

$$V(\Omega_t) = \ln(\exp(\beta \mathbb{E}(V(\Omega_{t+1})|\Omega_t)) + \exp(\zeta(\Omega_t))) \tag{4}$$

where $\zeta_i(\Omega_t) = \ln(\sum_{j \in H_t} \exp \delta_{ijt}))$ is the so-called inclusive value;[9] and $\delta_{ijt} = \boldsymbol{\alpha}_i^x \vec{x}_{j,t} + \alpha_i^{p,hs} p_{j,t} + \alpha^\Gamma \Gamma_{g(j),t}(\alpha_i^{p,app}, \boldsymbol{\alpha}_i^w) + \xi_{j,t}$. As in Rust (1994) we can now solve for the integrated value function, which is only a function of Ω. In other words, by focusing on the integrated value function we significantly reduce the dimension of the state space.

Furthermore, as in Hendel and Nevo (2006), Melnikov (2013), and Gowrisankaran and Rysman (2012) we assume that

$$G_i(\Omega_{t+1}|\Omega_t) = G_i(\zeta_{i,t+1}(\Omega_{t+1})|\zeta_{i,t}(\Omega_t)) \tag{5}$$

where $G()$ is the (exogenous) transition process of the state variables. As Hendel and Nevo (2006) note, $\zeta(\Omega)$ can be viewed as a price index. Therefore, this assumption means that the consumer does not need to know or keep track, of Ω_t. Nor does the consumer need to predict the future Ω_{t+1}. Instead, the consumer just needs to keep track of the lower dimension $\zeta(\Omega)$.

This assumption implies that in order to predict the distribution of future states consumers only use the lower dimensional price index ζ_t and not the higher dimensional Ω_t. This can be taken as a statement about how the actual process in which Ω_t evolves or as a behavioral statement about how consumers form expectations. The key is that consumers only track the current value of ζ_t to predict the future distribution of ζ_{t+1} instead of the whole information set Ω_t. This reduces the dimension of the state space to a more tractable single dimension and makes the computation of V numerically feasible. Specifically, it allows us to rewrite equation (4) as follows

$$V(\zeta(\Omega_t)) = \ln(\exp(\beta \mathbb{E}(V(\zeta(\Omega_{t+1}))|\zeta(\Omega_t))) + \exp(\zeta(\Omega_t))) \tag{6}$$

We further assume an AR(1) specification for $\mathbb{E}[\zeta_{i,t+1}|\zeta_{i,t}]$ as follows

$$\zeta_{i,t+1} = \phi_{i1} + \phi_{i2}\zeta_{i,t} + \mu_{it+1} \tag{7}$$

where, ϕ_{i1} and ϕ_{i2} are parameters, and μ_{t+1} is identically and independently distributed with mean 0.

[9] We omit the Euler constant as it does not affect the decision-making process.

Using the model we can construct an expression for the probability of buying each handset. The probability of consumer i purchasing a handset j at time t can be written as the probability of a purchase happening at time t times the probability of purchasing handset j conditional on a purchase happening, which is given by

$$s_{ijt}^{hs} = \frac{\exp\left(\zeta_{i,t}\right)}{\exp\left(\zeta_{i,t}\right) + \exp\left(\beta\,\mathbb{E}(V(\zeta_{i,t+1})|\zeta_{i,t})\right))} * \frac{\exp\left(\delta_{i,j,t}\right)}{\exp\left(\zeta_{i,t}\right)} \tag{8}$$

The aggregate market share of handset j in market t, which will be used in the estimation, can be computed by integrating the distribution of consumers

$$s_{jt}^{hs} = \int s_{ijt}^{hs} d\mathcal{P}^{*}$$

where $d\mathcal{P}^{*}$ denotes the distribution of consumer preferences. As we discuss below, this market share can be mapped to the observed market shares of handsets to estimate the model parameters. Finally, we want to highlight a key limitation of our current dataset: its constraint on accurately modeling brand loyalty. Our access is limited to aggregate-level consumer demand data, which doesn't allow us to account for individual-level purchasing behaviors or track consumers' consecutive buying decisions. This limitation may result in an incomplete representation of brand loyalty, particularly for consumers who consistently prefer a specific brand like Apple. In our model, we've attempted to approximate consumer heterogeneity through the term $u_{i,0,t}$, representing the outside utility of each customer. Furthermore, we acknowledge that disregarding brand loyalty effects could lead to an underestimation of both price and platform (Γ) elasticities. To understand this potential downward bias, consider the evolving patterns of mobile handset prices and corresponding consumer demand. For simplicity, let's momentarily assume there is no price endogeneity. The variation in demand corresponding to the variation in prices identifies price elasticity. In the presence of brand loyalty effects, customers might be less affected by price variations. Consequently, a model ignoring brand loyalty effects might attribute this to consumers having low sensitivity to price variations. A similar principle applies to the effect of platform utility. Ignoring brand loyalty effects could lead to an underestimation of indirect network effects, similar to the underestimation of price elasticity. To better understand the magnitude of potential bias our estimates may have due to a lack of granular loyalty data, we conducted a simulation study (see Appendix L). Our simulations indicate that without extreme brand loyalty levels, the additional bias incurred by omitting loyalty is limited.

4.2. Demand for Apps

Once a consumer purchases a handset, depending on the operating system, they get access to either the Apple app store or Google Play store.[10] In the previous section we denoted the benefits from the relevant app store, $g(j)$, by $\Gamma_{g(j)}(\cdot)$. We now explain how we model, and later estimate these benefits.

At a high level, we need to deal with two main issues. First, consumers typically download multiple apps in a given period.[11] We deal with this by treating each app download as an independent decision.

Second, we need to deal with the fact that apps are durable products: once consumers download an app, they are not going to download it again. Ignoring this can lead to mistakes in inference. Consider the following example: suppose there are two apps, A and B. Assume that A is launched in period t and is valued more by consumers than B, which is released in period $t+1$. In period t, assume almost all consumers download A. Now, in period $t+1$, there will be very few consumers left who have not downloaded A; however, the entire market has the option to download B. Thus, it is possible to observe a greater number of downloads for B than for A. If we ignore this dynamic and simply use the relative market shares each period to infer the value, we would get biased estimates of customer preferences. We deal with this issue by modeling the consumers' dynamic decision.

We assume that consumer i, compares the perceived lifetime utility, namely, the expected utility over all future periods, for every app l with their reservation utility (and continuation value over future periods) of not downloading the app. If the consumer perceives the (lifetime) utility of an app to be greater than the reservation utility, they download it, otherwise, they wait. We assume that a consumer can download an app only once because the data only accounts for the first download of an app by a user. The (lifetime) utility a consumer i gets by downloading an app l at time t (day) on platform $g(j)$ is

$$u^{app}_{i,g(j),l,t} = \underbrace{\boldsymbol{\alpha}^{w}_{i}\vec{w}_{g(j),l,t} + \alpha^{p,app}_{i} p_{g(j),l,t} + \eta_{g(j),l,t}}_{\psi_{i,g(j),l,t}} + \epsilon^{(app)}_{i,g(j),l,t} \tag{9}$$

where, $\vec{w}_{j,t}$ refers to the observable characteristics of the app at time t; $\boldsymbol{\alpha}^{w}_{i}$ denotes consumer i's taste towards these characteristics; $\alpha^{p,app}_{i}$ captures i's (app) price sensitivity; $\eta_{g(j),l,t}$ captures

[10] Some Android manufacturers like Samsung provide their users with access to an exclusive set of apps through their private app stores. The fixed part of these effects gets absorbed into the handset-specific fixed effects. The time-varying components are captured by the unobservable handset characteristic $\xi_{j,t}$.

[11] Consumers who choose to download apps, download on average approximately 4 apps per month. See, for example, https://www.comscore.com/Insights/Presentations-and-Whitepapers/2017/The-2017-US-Mobile-App-Report?cs_edgescape_cc=US)

(time-varying) characteristics of app l that are unobserved by the econometrician but are valued, and observed, by the consumers. Some components of $\eta_{g(j),l,t}$ will be fixed over time, and therefore we model $\eta_{g(j),l,t} = \bar{\eta}_{g(j),l} + \Delta\eta^{app}_{g(j),l,t}$. The error term $\Delta\eta^{app}_{g(j),l,t}$ is the time-varying component of the unobserved attributes. The unobserved characteristics could include unobserved promotional shocks, app fixed effects, or systemic shocks in-app demand. Finally, $\epsilon^{(app)}_{i,g(j),l,t}$ captures random shocks that are assumed to be independently and identically distributed according to the type I extreme value distribution.

The observable characteristics include the app's current version, current version's age, current period's quality (rating), firm (app developer) size, in-app purchases, and time-invariant attributes. App developers continuously update their apps. The current version number of an app captures this development and serves as a good indicator of app maturity. To account for the potential changes in quality over time and the app's current quality, we employ two separate measures: version age and consumer ratings in the focal period. The version age helps us understand the extent to which an app's age may influence its quality trajectory, while the current consumer ratings act as a proxy for the app's present quality. Finally, the app fixed effects capture any attributes, observed and unobserved by us, that remain constant for an app over time and represent perceived quality differences across apps.

If the consumer i decides not to download an app l on platform $g(j)$ at time t they get the one-period utility $\epsilon^{(app)}_{i,g(j),0,t}$, where $\epsilon^{(app)}_{i,g(j),0,t}$ is assumed to be independently and identically distributed according to the type I extreme value distribution.

We assume that consumers make the decision to download an app independently of other apps. So, for example, a consumer's decision of whether and when to download Spotify is independent from her decision on whether and when to download Facebook. As we will see, this assumption makes the model more tractable and allows us to deal with dynamics. However, it does not account for substitution across apps. Unlike the handset market, where consumers exit after buying a handset, allowing for both substitution and consumer dynamics in the apps market is computationally difficult, as it involves keeping track of all consumption sets over the entire set of apps. Lee (2013) shows that in markets with many products, the loss in the predictive power of a model based on independence assumptions is minimal for most products and almost negligible for top products. Since we only include top apps in our analysis, and top apps (for instance Facebook, Instagram, etc) in our context are highly differentiated, this assumption seems reasonable for our application and should not have a significant impact on our analysis. We discuss the robustness of our results to this assumption in Appendix D.

As noted earlier, the consumer decides whether to download an app or wait for each period. If they wait, they have the option to download the app in the future, which makes this a dynamic

(stopping) problem. Let Ω_t be the state variables that capture all relevant information used by consumers to make their decisions. The Bellman equation can be written as

$$W_{i,g(j),l}(\Omega_{i,l,t},\epsilon_{i,g(j),l,t}) = \max_{a_t\in\{0,1\}} \begin{cases} u^{app}_{i,g(j),l,t}, & a_t = 0 \text{ if downloads} \\ \epsilon_{i,g(j),0,t} + \beta E[W_i(\Omega_{i,l,t+1})|\Omega_{i,l,t}], & a_t = 1 \text{ if waits} \end{cases} \tag{10}$$

where, $W(\Omega_t) = \int W(\Omega_t,\epsilon_t)dF_{\epsilon_t}(\epsilon_t)$.

As we did in the demand for handsets, we follow Rust (1994) and integrate out the logit error ϵ to rewrite equation (10) as

$$W(\Omega_t) = \ln\left(\exp\left(\beta\,\mathbb{E}(W(\Omega_{t+1})|\Omega_t)\right) + \exp\left(\psi(\Omega_t)\right)\right) \tag{11}$$

where, $\psi_t = \boldsymbol{\alpha}^w_i \vec{w}_{g(j),l,t} + \alpha^{p,app}_i p_{g(j),l,t} + \eta_{g(j),l,t}$. Further, as in Hendel and Nevo (2006), note we can interpret $\psi(\Omega)$ as a price index. Next, similar to handsets, to reduce the dimension of the state space, we assume that

$$F_{i,g(j)}(\psi_{i,t+1}(\Omega_{t+1})|\Omega_t)) = F_{i,g(j)}(\psi(\Omega_{t+1})|\psi(\Omega_t)), \tag{12}$$

where $F()$ is the (exogenous) transition process of the state variables.

As before, we assume that with regards to the distribution of future states, the consumer only uses the lower dimensional price index ψ_t and not the higher dimensional Ω_t. This can be interpreted as a statement about how the actual process Ω_t evolves or as a behavioral statement about how consumers form expectations. The key is that consumers only use the current value of ψ_t to predict the future distribution of ψ_{t+1} instead of the whole information set Ω_t and make download decisions. This reduces the dimension of the state space to a single dimension and makes the computation of W numerically feasible. Specifically, it allows us to rewrite equation (11) as follows

$$W(\psi(\Omega_t)) = \ln(\exp\left(\beta\,\mathbb{E}(W(\psi(\Omega_{t+1}))|\Omega_t)\right) + \exp\left(\psi_t(\Omega_t)\right)) \tag{13}$$

As we did for handsets, we use a AR(1) specification for $\mathbb{E}[\psi_{i,t+1}|\psi_{i,t}]$ as follows:

$$\psi_{i,t+1} = \phi^{app,g(j)}_{i1} + \phi^{app,g(j)}_{i2}\psi_{i,t} + \epsilon_{t+1} \tag{14}$$

where, ϵ_{t+1} is identically and independently distributed with mean 0. ϕ_{i1} and ϕ_{i2} are the incidental parameters.

Using the model, we can write an expression for the probability of consumer i to download app l in period t, which is given by

$$s^{app}_{ijt} = \frac{\exp\left(\psi_{i,l,g(j),t}\right)}{\exp\left(\psi_{i,l,g(j),t}\right) + \exp\left(\beta\,\mathbb{E}(W(\psi_{i,l,g(j),t+1}))|\psi_{i,l,g(j),t})\right)} \tag{15}$$

The aggregate share of downloads of app l on platform $g(j)$ at time t can computed by integrating over the distribution of consumers

$$s_{jt}^{app} = \int s_{ijt}^{app} d\mathcal{P}^*$$

where $d\mathcal{P}^*$ denotes the population distribution of consumer preferences. To estimate the parameters of the model, we will map this probability to the observed share of app downloads in the data.

Using this model, we can also derive the utility a consumer gets from having access to a certain app store. The value is the sum of the utility from apps currently on the platform and the discounted utility of apps that will be available on the app store in the future. Formally, the utility from access to the app store $\Gamma(\cdot)$ is given by

$$\Gamma_{g(j),t}(\alpha_i^p, \boldsymbol{\alpha}_i^w) = \underbrace{[\sum_{l \in \cup_{\tau=0}^{\tau=t} L_{g(j),\tau}} W_{i,g(j),l}(\Omega_{i,t})]}_{\text{Current app utility} \equiv \Delta^c_{i,g(j),t}} + \underbrace{E[\sum_{\tau=1}^{\tau=\infty} (\beta)^\tau (\sum_{l \in L_{g(j),t+\tau}} \mathbf{E}[W_{i,g(j),l}(\Omega_{i,t+\tau})]) | \Omega_{i,t}]}_{\text{Expected future app utility} \equiv \Delta^f_{i,g(j),t}} \quad (16)$$

where, $L_{g(j),\tau}$ denotes the set of apps launched in period τ. Similarly, $\Delta^f_{i,g(j),t}$ denotes the expected values of apps that are expected to be launched in the future given the current information set $\Omega_{i,t}$. $\Delta^c_{i,g(j),t}$ captures the current app utility attributed by the apps currently present on the platform.

4.3. Developers' Entry Decisions

We now consider the developers' entry decisions on the two platforms. Precisely, we will now model the entry costs that affect developers' ability to offer an app across the two platforms. We assume that an app not present on any platform would commit k_{entry} months in advance to enter either or both app stores. Such an app would incur a platform-specific cost θ_{apple} and/or $\theta_{android}$. The entry decision of the developer will depend on the trade-off between these costs and the expected profit from entry –

$$\Pi_{l,t}(d_t \equiv \{\} \rightarrow \{p\}; \theta) = \mathbb{E}\Big[\sum_{\tau \geq t + k_{entry}} (\beta^{\tau - t} D_{l,\tau}^p) | \Omega_{l,t}\Big] - \theta_{l,p}^{entry} \quad (17)$$

where $D_{l,\tau}^p$ denote the download of app l in time period τ on platform p; $\Omega_{l,t}$ is the app developer's information set at time t; θ denotes the model parameters; and $d_t \equiv \{\} \rightarrow \{p\}$ denotes the decision at time t to do the launch on just platform p. The above expression is the net present value of the total number of downloads the developer is expecting to receive from the app l and platform p. Ideally, we would need data on the net present value of the app revenues. An app makes revenue through three sources – (i) upfront (if paid), (ii) in-app purchases, and (iii) ad revenue. One limitation of our dataset is that we do not have data on ad revenues. Thus, we normalize expected profits per download to 1 and thus the expected present value of app profit can be expressed as

discounted value over the future stream of downloads. The normalizing assumption might initially seem to disregard app revenue models that depend on the installed base, given that ad revenues and subscription fees accrue on the entire base, not just new downloads. In Appendix J, we demonstrate that the expected lifetime revenue for an app developer can always be expressed as a discounted value over the future stream of just downloads multiplied by a normalizing constant, regardless of revenue models.

By normalizing the profits, all apps are treated equally with respect to their means of monetizing users, whether by upfront price (free versus paid), in-app, or advertising. The consequence of our normalization is that we can not recover the true dollar value of entry costs but only costs relative to download numbers. Previous literature (Allon et al. (2019)) has shown that the assumption of treating all apps equally with respect to their means of monetization can be a useful approximation for capturing the true data-generating process of developers. For example, Allon et al. (2019) found that this assumption can accurately capture the revenue-generating behavior of mobile app developers, even when the app market is characterized by significant heterogeneity in terms of app quality and monetization strategies. This suggests that, while the assumption may not capture all of the complexity of the app market, it can still provide valuable insights into the behavior of app developers.

Next, the profit function if the app developer chooses to do a simultaneous launch on both platforms at once –

$$\Pi_{l,t}(d_t \equiv \{\} \rightarrow \{p,p'\};\theta) = \mathbb{E}\Big[\sum_{\tau>t+k_{entry}} (\beta^{\tau-t}[D^{p}_{l,\tau} + D^{p'}_{l,\tau}])|\Omega_{l,t}\Big] - \theta^{entry}_{l,p} - \theta^{entry}_{l,p'} \tag{18}$$

Now let's consider the expected profit for an app l operational on platform p that elects to enter platform p'. Once the app is deployed on the other platform it gets another stream of downloads $\{D^{p'}_{t+k_{port}+1}, D^{p'}_{t+k_{port}+2}, \dots\}$ from platform p', where k_{enter} refers to the time required to enter the other platform. Then, the expected normalized profits from app l, if developer chooses to enter (i.e., $d_t \equiv \{p\} \rightarrow \{p,p'\}$) platform p' at time t is given by

$$\begin{aligned}\Pi_{l,t}(d_t \equiv \{p\} \rightarrow \{p,p'\};\theta) = \mathbb{E}\Big[\sum_{t<\tau\leq t+k_{port}} (\beta^{\tau-t} D^{p}_{l,\tau})|\Omega_{l,t}\Big] + \\ \mathbb{E}\Big[\sum_{\tau>t+k_{port}} (\beta^{\tau-t}(D^{p'}_{l,\tau} + D^{p}_{l,\tau}))|\Omega_{l,t}\}\Big] - \theta^{port}_{l,p'}\end{aligned} \tag{19}$$

Note that the stream of downloads from platform p' starts from period $t + k_{enter} + 1$. We can now use the observed empirical patterns of developer entry to infer the economic cost of entry of an app to a platform (Apple or Android). Specifically, by observing when a developer chooses to enter a platform, we can construct estimates of entry costs.

If we observe the initial entry of an app (denoted by decision d) at time $\mathcal{T}$, this implies that the developer would have found it profitable to do so $k+1$ periods before over not entering (denoted by d^c). That is

$$\Pi_{l,\mathcal{T}-k-1}(d_{\mathcal{T}-k-1};\theta) - \Pi_{l,\mathcal{T}-k-1}(d^c_{\mathcal{T}-k-1};\theta) \geq 0 \tag{20}$$

Similarly, for all time periods $\tau < \mathcal{T}-k-1$, developer would have not found action d profitable

$$\Pi_{l,\tau}(d_{\tau};\theta) - \Pi_{l,\tau}(d^c_{\tau};\theta) \leq 0 \tag{21}$$

5. Estimation

In this section, we discuss the estimation of the model(s) presented in the previous section using the data described in Section 3. We estimate the different parts of the model separately.

5.1. Estimation of Demand

In our model, the unobserved demand shocks in handset and app demand, ξ_{jt} and η_{lt}, may be correlated with various observed handset and app features. For instance, handset prices and the platform utility term (Γ) could be correlated with the unobserved handset demand shock ξ_{jt}. Similarly, on the app side, app prices, review ratings, app age, and versioning behavior could all be correlated with unobserved demand shocks in app demand η_{lt}. Failing to account for these correlations during parameter estimation could lead to biased estimates. To address this issue, we estimate both handset and app demand using a generalized method of moments (GMM) approach. We follow the methodology used in static models, such as those by Berry et al. (1995) and Nevo (2001), and in dynamic models by Gowrisankaran and Rysman (2012) and Melnikov (2013). The core idea is to estimate model parameters by minimizing a GMM objective function, which is a sample analog of moments between the structural error term (a function of parameters) and instruments. This approach allows us to address the endogeneity of certain variables and estimate the parameters governing the distribution of random coefficients. In this section, we discuss key aspects of our estimation strategy. More detailed information is provided in Appendix A. We begin by explaining our choice of error term for constructing GMM moment conditions. Next, we describe our instrumental variables and outline the computational steps involved in the estimation process.

To estimate our model and construct the GMM objective function, we build on the following moment restrictions

$$\mathbb{E}\left[\nu^{hs}_{j,t}(\theta^{hs})|\mathbf{Z}^{hs}_t\right] = 0 \qquad \mathbb{E}\left[\nu^{app}_{g(j),l,t}(\theta^{app})|\mathbf{Z}^{app}_t\right] = 0 \tag{22}$$

where $\mathbf{Z}^{hs}_t$ and $\mathbf{Z}^{app}_t$ are instrumental variables, and θ^{hs} and θ^{app} include all model parameters of handset and app models, respectively. The econometric error terms ν^{hs} and ν^{app} are defined using the first difference of the unobservable attribute

$$\nu^{hs}_{j,t} = \xi_{j,t} - \xi_{j,t-1}, \quad \forall j \in H_{t-1}$$
$$\nu^{app}_{g(j),l,t} = \eta_{g(j),l,t} - \eta_{g(j),l,t-1}, \quad \forall l \in L_{g(j),t-1} \tag{23}$$

A few remarks are in order. First, the estimation is at the aggregate market level, and therefore the individual choice level errors $\epsilon^{(hs)}_{i,j,t}$ and $\epsilon^{(app)}_{i,g(j),l,t}$ that affect consumers' choices of handsets and apps are integrated out. Second, by defining the error term as the first difference, we difference away a headset-specific constant (and an app-specific constant in the demand for apps). This significantly simplifies the GMM optimization problem, which can now be carried out by searching only for a handful of model parameters. With the model parameters in hand, we can recover the fixed effects. Third, we have two sets of conditional moment restrictions: one set coming from the handset demand model and another from the app demand model. To estimate our demand model, we stack both sets of moment restrictions –

$$\mathbb{E}\left[error(\theta^{hs}, \theta^{app})|\mathbf{Z}\right] = 0, \tag{24}$$

where

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}^{hs} & 0 \\ 0 & \mathbf{Z}^{app} \end{bmatrix}, \quad error = \begin{bmatrix} \nu_{hs} \\ \nu_{app} \end{bmatrix},$$

Using the above moment conditions, one can construct the GMM objective function as follows –

$$\underset{\{\theta^{hs},\theta^{app}\}}{\arg\min}\, (error(\theta^{hs}, \theta^{app})^T\mathbf{Z})\mathbf{W}(error(\theta^{hs}, error^{app})^T\mathbf{Z})^T, \tag{25}$$

where $W = (\mathbf{Z}^T\mathbf{Z})^{-1}$. This process is computationally intensive, so we propose a simplification to the optimization problem. We can rewrite the optimization problem as follows:

$$\underset{\{\theta^{hs},\theta^{app}\}}{\arg\min}\, \underbrace{(\nu^{hs}_{j,t}(\theta^{hs}, \theta^{app})^T\mathbf{Z}^{hs}_t)\mathbf{W}^{hs}(\nu^{hs}_{j,t}(\theta^{hs})^T\mathbf{Z}^{hs}_t)^T}_{GMM_{hs}} + \underbrace{(\nu^{app}_{g(j),l,t}(\theta^{hs}, \theta^{app})^T\mathbf{Z}^{app}_t)\mathbf{W}^{app}(\nu^{app}_{g(j),l,t}(\theta^{hs}, \theta^{app})^T\mathbf{Z}^{app}_t)^T}_{GMM_{app}},$$

where $\mathbf{W}^{hs} = \left(\mathbf{Z}^{hs'}\mathbf{Z}^{hs}\right)^{-1}$ and $\mathbf{W}^{app} = \left(\mathbf{Z}^{app'}\mathbf{Z}^{app}\right)^{-1}$.[12] For our estimation, we have two sets of parameters, one pertinent to the handset model (θ^{hs}) and a set of parameters for the app model (θ^{app}). The app model parameters (θ^{app}) are dependent on handset model parameters (θ^{hs}) and vice-versa. Instead of optimizing both parameters in a joint fashion, we proposed an iterative two-step optimization routine. Also, note that for our estimation, we assume monthly discounting at a

[12] A two-step approach to calculate GMM weighting matrices can also be carried out to obtain more precise estimates. However, given the size of our dataset and the complexity of the model, the computational burden of performing two-step GMM estimation was significantly high.

0.97 rate. For standard errors, we use the stacked GMM expression and the standard formula of variance for a GMM estimator (for reference, see Newey and McFadden (1994)). We provide more computational details at the end of this Section and in Appendix A.

Instruments: To address the endogeneity issue, we make a structural assumption[13]. We assume that the evolution of the unobserved demand shocks ξ_{jt} and η_{jt} from period $t-1$ to t could not be predicted in advance when observable handset and app features were set at period $t-2$. This structural assumption allows us to use all past values of observable features of the handset and app as instruments to interact with the first differences (ν_t^{hs} or ν_t^{app}). Note that this assumption does not preclude the scenario that merchants can partially predict the values of future demand shocks (ξ_{t-1} and ξ_t or η_{t-1} and η_t) but not their first differences (ν_t^{hs} and ν_t^{app}). Thus, for instruments $\mathbf{Z}_t^{hs}$ and $\mathbf{Z}_t^{app}$ we can use (lagged) values of observable features of the handsets and apps, respectively.

Specifically, the instrument set Z_t^{hs} includes, two- period lagged values of handset price, current- and one- period lagged values of platform utility $\{\Delta^c_{i,g(j),t},\ \Delta^c_{i,g(j),t-1}\}$, and current period values of BLP-style instruments[14] on handset exogenous characteristics – RAM, display size, battery, storage, camera.

From our model, handset prices could be correlated with the shocks in handset demand (ν^{hs}) and hence could be endogenous. We assume that the two-lagged values of prices do not correlate with ν^{hs} but do correlate with level differences in handset prices and hence could help identify the price coefficient. With regards to the platform utility, the exclusion of current- and one-period lagged values of platform utility $\{\Delta^c_{i,g(j),t},\ \Delta^c_{i,g(j),t-1}\}$ relies on a timing assumption, namely that software firms cannot immediately respond to shocks in individual handsets' sales or predict them in advance. We test the robustness of our results against the possibility that some app developers might be able to track individual handset sales and respond rapidly in Appendix C and do not find evidence for the same. For instance, using the two-period lagged value of current platform utility $\{\Delta^c_{i,g(j),t-2}\}$ gives almost similar model estimates.

The instrument set Z_t^{app} includes two-period lagged values of app prices, two-period lagged value of version number and version age, two-period lagged value of app's rating, and current period values of BLP-style instrument on app-age.

[13] As a robustness we estimate the model with cost-based instruments and without any such structural assumption. We get qualitatively very similar results. For details see Appendix C.

[14] We follow the suggestion of Berry et al. (1995), and for each product j and its exogenous observable characteristic k we construct the sum of product characteristic k of other products controlled by the same firm f and of those controlled by competing firms within the same market t.

$$\mathcal{H}_j(\vec{x}_t^k) = \Big\{\{\sum_{i\in P_{ft}\setminus\{j\}} x_{it}^k,\ \sum_{i\notin P_{ft}} x_{it}^k\}\Big\} \tag{26}$$

where P_{ft} refers to the set of products offered by firm f at time t. For handsets, a firm refers to the manufacturer of the handset like Apple, Samsung, etc., and for apps, a firm refers to the company that developed the app.

Similar to handset prices, app prices might also be correlated with app demand shocks ν^{app}, and be endogenous. We assume that the two-lagged values of app prices do not correlate with ν^{app} but do correlate with level differences in app prices, and hence could help identify the price coefficient. Also, the app's versioning behavior (and thus version age and version #) and app's current period rating may also be correlated with sudden demand shocks. We assume that two-period lagged value of app's versioning related variables and two-period lagged value of app's current period rating does not correlate with demand shocks, but does correlate with level differences in app's version age and version #, and current period rating, respectively.

To test if our instruments are weak, we consider the first stage regressions in level differences of the endogenous variables onto their respective instrument set. We carry out the weak instrument test as laid out in Cragg and Donald (1993). Since, for both handsets and apps, we have multiple endogenous variables, we compute the Cragg-Donald statistic i.e., the minimum eigenvalue of the matrix analog of F-statistic from the first stage regression. We calculate the critical value for weak instruments using the procedure laid out in Stock and Yogo (2002). Across all specifications we consider, we are able to reject the hypothesis that our instruments are weak.

Our identification argument assumes that the handset and app specific demand shocks (ν_t^{hs},ν_t^{app}) are serially uncorrelated. To test the validity of this assumption, we use the estimated first differences ($\hat{\nu}_t^{hs}$, $\hat{\nu}_t^{app}$) (for more details see Wooldridge (2010)) from our estimated model and estimate the following regression:

$$\hat{\nu}_t = \kappa \hat{\nu}_{t-1} + error_t \tag{27}$$

Estimating the above regression, we are unable to reject the null that $\kappa = 0$ for both handset and app models.

5.2. Identification

We now delve into determining the key parameters in the demand model. Our identification arguments closely follow those of Lee (2013) who estimate a similar model with a similar set of parameters. The price coefficients for mobile handsets, denoted by $\alpha^{p,hs}$, and for smartphone apps, represented by $\alpha^{p,app}$, are discerned through the fluctuations in prices and sales across various handset models, diverse apps, and over time. The fixed effects for handsets and apps are identified by the average sales levels of smartphones and apps across different options.

The identification of average price sensitivities, $\alpha^{p,l}_{l \in hs,app}$, depends on the variations in prices and sales over time and among the two platforms. Instead of imposing the constraint $\alpha^{p,hs} = \Gamma \alpha^{p,app}$, we separately estimate the coefficients for handsets and apps, so Γ is not identified by the differences in price responsiveness between the two. Rather, Γ is primarily identified as handset

sales reacting to changes in app utility (both within and across platforms) due to fluctuations in app quality/availability and sales over time.

Generally, consumer preference variance is identified from changes in product features. The panel data also offer an additional identification source through the endogenous shift in consumer valuations distribution over time. If the heterogeneity in α^p is significant, consumer reactions to price alterations over time will vary.

For example, initial platform or app buyers will be less responsive to price changes compared to later purchasers. If there is no observable change in consumer purchase behavior concerning price alterations throughout a product's lifespan, it could imply $\sigma_p = 0$. However, any meaningful heterogeneity in price sensitivities would result in distinct customer reactions to price changes.

Lastly, the scale parameter Γ, connecting the utility derived from the app store to the utility gained from handsets, is identified by the changes in mobile handset sales as a response to variations in the anticipated utility from the app store. To demonstrate this, consider that a decrease in expected app store utility (due to price increases or a decline in app quality) leads to a reduced demand for handsets, implying a positive Γ since consumers perceive the two products as complementary. Conversely, if the changes in app quality/assortment or prices have no effect on app sales, it suggests that Γ is nearly zero.

5.3. Estimation of Developers' Entry Costs

We carry out the estimation of entry cost parameters using methods developed in Pakes et al. (2015). The method is similar to the generalized method of moments (Hansen (1982)) method with a slightly different objective function. We first construct sample analogs of the moment conditions $m_{l,t}$, specified in Section 4 as a function of the model parameters. To do so, we use $h(\cdot)$ a set of positive exogenous functions to aggregate moments to construct population moments for each genre g and platform p as follows.

$$\hat{m}_g^p = \frac{1}{n} \sum_{l \in \mathcal{S}_{g,p}} \sum_t h(\Omega_{t,l,p}) \otimes m_{l,t} \tag{28}$$

where $\mathcal{S}_{g,p}$ is the set of apps on platform p, with genre g and $\otimes$ representing the Kronecker product.

To compute the values of moments ($m_{l,t}$), we need to estimate the present value of app profits i.e., $\mathbb{E}[D^p_{l,t'}|\Omega_{l,t}, d_t]$ for $t' > t$. To simulate the demand for apps in future periods, we use the estimated demand processes F and G. App developers know that their launch decisions only affect the levels $\{\delta_{i,j,t}\}$ and not the evolution processes F and G. This allows us to simulate the downloads for each app in the future independent of other app developers' decisions. For our estimation, we assume

app developers take on an average 1[15] month[16] to port an app with an already developed back-end infrastructure. With regards to entry cost, we assume the specification for entry cost $C_l(\theta) = \theta_p^g$, where $\theta \equiv \{\theta_p^g\}$, and g is the genre of the title l. Formally, the set estimator $\hat{\Theta}$ that consistently identifies the set Θ_0 is given as follows

$$\hat{\Theta} = \arg\min_{\theta} ||\hat{\Sigma}_{ii}^{-\frac{1}{2}} \min(0, \hat{m}(\theta))|| \tag{29}$$

where, $\hat{\Sigma} = \mathbb{V}(\hat{m}(\Theta_0))$, i.e., the variance-covariance matrix of moment conditions. The key idea behind the above objective function is that it penalizes all the violations in the moment inequalities. All we know is that the moments are positive at the true parameters. So the moment inequality estimator penalizes values of θ that lead to negative values of moment and tries to minimize the number of violations. Note that unlike methods that utilize moment equalities, estimating model parameters using inequalities can lead to set estimates, rather than a point estimate. Thus, if multiple values of θ satisfy the moment inequalities (i.e., minimize the equation 29 to 0), all are accepted, however, if none satisfies, then θ that minimizes the absolute value of deviation in inequalities is admitted. We normalize the moments by the square root of the variance of the moments. To compute the variance, we carry out the estimation in two steps. In the first step, we estimate Θ_0 using an identity matrix as the normalizing factor. Using the estimated set $\hat{\Theta}_0$, we then compute $\hat{\Sigma}$. To compute $\hat{\Sigma}$, we use centroid of the identified set $\hat{\Theta}_0$. Next, we re-estimate Θ, using the estimated $\hat{\Sigma}$ as the normalizing factor.

With regards to inference, literature has proposed multiple characterization and methods for defining and estimating confidence intervals. For our analysis, we define confidence interval as the set that includes each element of the identified set with fixed probability (see Imbens and Manski (2004) and Romano and Shaikh (2008)). We detail the procedure to compute the confidence interval in Appendix A.3.

Finally, as a robustness test for moment inequalities, we run a misspecification test as described in Ho and Pakes (2014), to test if the moment inequalities are correctly specified (see Appendix F for details). We were unable to reject the null that our model is correctly specified.

6. Results

Demand Estimates. We first present the estimation results for the handset and app demand models. In Table 8, we present results from three different specifications[17] of the demand for

[15] We run our analysis for values of $k = 1, 3,$ and 5 months, and find slightly different cost estimates however similar counterfactual implications.

[16] According to many business reports (see for instance Existek (2020)) the average to time to develop the front-end of an app is around $\sim$ 6 weeks.

[17] In Appendix K, we also estimate our model without accounting for endogeneity, treating all explanatory variables as exogenous. The estimates for both the handset and app models are reported in Tables 27 and 28, respectively.

handsets. In column (I), we present the results for a model where $\Gamma(\cdot)$ is approximated by the total number of apps (i.e., we assume that each app contributes equally to platform utility). To estimate the model in column (I), we use the set of instruments described in Section 5 except instead of lagged values of platform quality, we use current and one-period lagged values of platform size (i.e., the number of apps) as an instrumental variable. We find that platform size has a statistically significant and positive effect on handset demand.

In column (II), we present the results for a static model, where we assume that in each period, consumers choose to either buy one of the available handsets or exit the market without purchasing. They cannot choose to wait in anticipation of a better option in the future. We present more details of the static model of handset demand in Appendix B. In the static model, the effect of price on handset demand is not statistically significant and the estimated coefficient is smaller, compared to the estimates in columns I and III, which account for dynamics in consumer behavior. This reinforces the point we made in Sections 3 and 4 that ignoring consumer dynamics could lead to an underestimation of demand elasticities.

Finally, in column (III), we report results for a model where we allow for both consumer dynamics and heterogeneity across apps in the impact on platform utility. We estimate the complete model as described in Section 4. We find that platform utility $\Gamma(\cdot)$ has a positive and significant impact on handset demand.

To better interpret the model estimates we compute elasticities with respect to each handset characteristic. To compute these elasticities we change the respective characteristic by 1% and then use the model estimates to compute the corresponding change in handset demand. We assume consumers expect the specific characteristic to align back next period, and hence the artificial intervention does not affect consumers' expectations over future states. This procedure translates to simulating a temporary change in handset characteristics. Another way to interpret the model estimates would be to simulate a permanent change in handset characteristics wherein the specific characteristic does not align back in the next period. In general, the permanent elasticity numbers are a little lower than the temporary ones. For the sake of brevity, we will only discuss the temporary elasticity numbers here.

We summarize the aggregate elasticity numbers with respect to handset characteristics in Table 9. To compute the aggregate elasticities, we use the estimates from column (III) and the average across handsets. For instance, the demand for a particular handset increases by 1.90% with a 1% increase in display size. With respect to RAM, we do not find a significant impact on handset demand.

Across both models, we find that the price coefficient is positively biased compared to the model using instrumental variables.

Table 8. Estimated Parameters (Handsets)

	I	II	III
$log(prices)$	-2.082***	-0.093	-2.705***
	(0.453)	(0.266)	(0.722)
ram (megabytes)	-0.575***	-0.093	0.010
	(0.134)	(0.079)	(0.124)
$display\text{-}size$ (inches)	0.633***	0.401***	0.412***
	(0.149)	(0.088)	(0.138)
$battery$ (mAh)	0.001***	0.0004***	0.001***
	(0.000)	(0.000)	(0.000)
$log(storage)$ (GB)	0.792***	-0.021	0.739***
	(0.054)	(0.032)	(0.050)
$camera$ (megapixels)	0.232***	0.073***	0.233***
	(0.018)	(0.011)	(0.016)
$log(\#apps)$	1.595***	-	
	(0.349)		
Γ	-	0.124***	0.257***
		(0.017)	(0.031)
$Handset\ FE$	✓	✓	✓
$\sigma^{p,hs}$	0.003	0.090	0.008***
	(0.881)	0.238	(0.000)
N	895	895	895
GMM Obj	4.76e-4	1.171e-05	1.41e-4

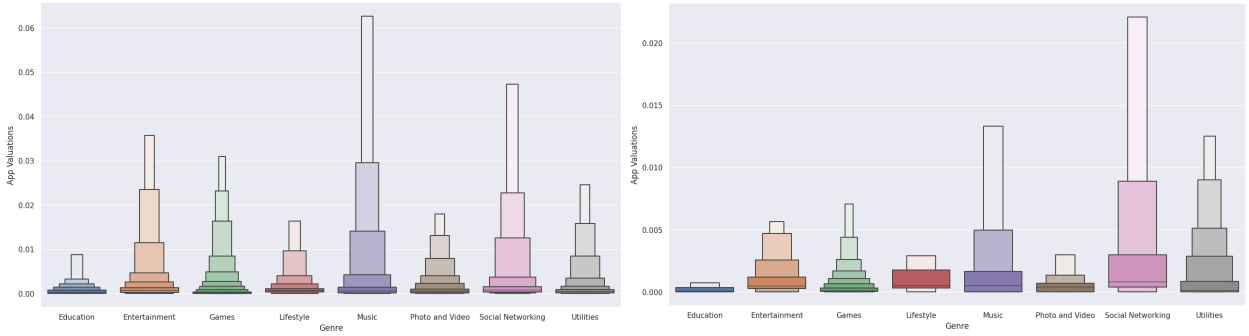
Standard errors in parentheses. *p<0.1; **p<0.05; ***p<0.01. For specification (I) we estimate a model where $\Gamma(\cdot)$ is approximated by the total number of apps (i.e., we assume that each app contributes equally to platform utility). For specification (II) we estimate a model that does not account for dynamics in consumer buying behavior. For specification (III) we estimate our full model, which allows for heterogeneity in app externality and consumer dynamics. The parameters for time-invariant characteristics are estimated by projecting the estimated handset fixed effects onto the handset time-invariant characteristics using ordinary least square regression.

Table 9. Handset Elasticities

log(price)	-2.64%
display size	1.90%
battery	2.26%
log(storage)	0.73%
camera	2.08%
Price Elasticity Across Handsets	
iPhone 5	-2.65%
Samsung Galaxy S4	-2.60%
LG G2	-2.63%

Note. The numbers report the percentage change in handset sales following a temporary change in respective handset characteristics by 1%. Temporary change implies that customers' belief over future evolution remains unchanged following a price reduction.

Figure 11. Estimated App Valuations (W) across Genres



(a) iOS Apps

(b) Google Play Apps

App valuations are the log scores of normalized estimated app lifetime value (W). For every app we first average (W_{it}) across the length of our entire sample and then normalized the scores across all apps in their respective platform.

To compute the price elasticity of handset demand we averaged across all handsets and find that a 1% price increase leads to a decline in sales of the particular handset by 2.64%. In Table 9, we report price elasticities for a few handsets. For instance, consider the own-price elasticity of the iPhone 5, which is among the top purchased handsets. We estimate sales will fall by 2.65% following a 1% price increase. Similarly, for Samsung Galaxy S4, we find that its sales will drop by

Table 10. Estimated Parameters (Apps)

(a)

	(I)
price	-0.540***
	(0.078)
version age	-0.007***
	(0.0014)
version #	-0.083***
	(0.019)
ratings	6.355***
	(1.304)
age	0.002***
	(0.000)
App FE	✓
$\sigma^{p,app}$	0.145***
	(0.005)
N	1,801,815
GMM Obj	6.52e-11

Standard errors in parentheses.
* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

(b)

	App Fixed Effects
	(1)
in-app-purchases	-0.129***
	(0.004)
paid	-2.967***
	(0.004)
$log(firmsize)$	0.232***
	(0.001)
Entertainment	0.757***
	(0.016)
Games	-1.848***
	(0.015)
Lifestyle	0.215***
	(0.017)
Music	0.692***
	(0.017)
Others	0.053***
	(0.015)
Photo and Video	0.586***
	(0.016)
Social Networking	1.781***
	(0.016)
Utilities	0.156***
	(0.016)
ios	-0.667***
	(0.004)
Intercept	-17.72***
	(0.016)
Observations	1801814.0
R2	0.381
Note:	*p<0.1; **p<0.05; ***p<0.01

The estimation is carried out on our entire sample of 13,731 iOS and 6,950 Google Play apps. The parameters for time-invariant characteristics are estimated by projecting the estimated residual $\eta_{g(j),l,t}$ onto the app's time-invariant characteristics using ordinary least squares regression. The residual term $\eta_{g(j),l,t}$ is the sum of a time-invariant and time-varying component ($\eta_{g(j),l,t} = \bar{\eta}_{g(j),l} + \Delta\eta^{app}_{g(j),l,t}$). The feature *prices* indicates the price of the paid app and is set to 0 for free apps; *version age* indicates the days since the last version was released; *version* indicates the current iteration of the app; *ratings* is set to 1 if the average current period's rating is above 3, and 0 otherwise; *age* indicates the days since the release of the app; *in-app-purchases* is an indicator variable indicating whether there are in-app purchases, and *paid* indicates whether the app is paid.

2.60%, with a price increase of 1%. We do not find much heterogeneity in price sensitivities across the consumer population, as indicated by a small but statistically significant σ_{hs} parameter.

In Table 10, we report estimates of app demand parameters, and in Table 12 we present the aggregate elasticities with respect to app characteristics implied by these results. We find a positive and significant effect of app age on app demand i.e., consumers drive higher lifetime utility from apps that are mature in age. This could be driven by the fact that as apps mature, they undergo a series of experiments and development rounds by its developers, which improve its quality and make it more competitive in the marketplace. Further, we also find a negative and significant effect of version age on app demand i.e., as the current version of an app gets old (i.e., the app is not updated in time), it becomes less attractive to consumers. This indicates that customers prefer apps by developers who regularly update their apps and engage in quality versioning. However, we also find evidence that by controlling for overall age and versioning age, apps having a relatively low frequency of versioning are preferred. We find a negative and significant effect of version-# on the app's demand. A very high number of versioning rounds could indicate lower quality levels and error proneness on the developer side. Hence, a preferred app developer is one that engages optimally in quality versioning with regular updates to cater to evolving consumer needs. Finally, we estimate that paid Apple and Google Play apps demonstrate price elasticities of -1.51% and -2.03% respectively. Note that, we found higher price elasticities for iPhones versus Android models in Table 9 and here we find higher price elasticities for Google Play apps compared to Apple apps. These results may appear contradictory at first glance. We believe that the difference in elasticity between Apple and Android apps can be attributed to several factors. One such factor is the difference in the average price and assortment of apps in our sample as we reported average elasticities for Apple and Android apps. In our data, Android apps are priced higher on average compared to Apple apps, which may contribute to the higher average elasticity observed for Google Play apps. Regarding the higher price elasticity for iPhones compared to Android models, this could be primarily driven by the fact that Apple devices are generally more expensive, making consumers more price-conscious when purchasing a handset.

In Table 10b, we report how time-invariant app characteristics affect app demand (and indirectly handset demand). We regress the estimated app fixed effects on the various time invariant app characteristics, e.g., if an app has in-app purchases (*in-app-purchases*), if the app is paid (*paid*), and genre of the app. We find that social networking apps tend to attract the highest demand. Also, paid apps tend to attract much lower demand (and thus, exert lower network externalities) than free apps. Thus, platform operators should vary the platform fees they charge from the apps, as a higher fee could mean high-priced apps and could reduce demand for their handsets. Finally,

Table 12. App Elasticities

price	-1.41%
log(firmsize)	0.23%
version age	-0.15%
version #	-0.55%
age	0.79%

Note. The numbers report the percentage change in app downloads following a temporary reduction in respective app characteristics by 1%. A temporary price reduction implies that customers' belief over future evolution remains unchanged following a price reduction. We only report numbers for app characteristics that have continuous values.

apps that make consumers pay for certain functionalities through in-app purchases also attract lower demand.

We now turn to the question of the impact of platform size (or the number of apps that appeared at least once in the top 500 list during the two-year period of our study) on handset demand. For this, we again look at columns I and III in Table 8. Averaging across both platforms, increasing the number of these apps by 1% increases the sales of a particular platform on an average of 1.5%. In column (III), we account for heterogeneity in app externalities and estimate the complete model detailed in Section 4. We estimate that a 1% increase in platform utilities increases the sales of a particular platform by 1.3%. We also find evidence for significant heterogeneity (see figure 11) in externalities these top apps exert. In figure 11, we display the variation in app lifetime valuations across various genres in both Apple iOS and Google Play platforms for apps that appeared at least once in the top 500 list during the two-year period of our study. The variation in app lifetime values indicates the heterogeneity in consumers' preferences for these apps. Regarding the effect on handset demand, we find that the externality exerted by the top app is around 2700 and 5600 times that of the median app among those that appeared at least once in the top 500 on iOS and Google Play platforms, respectively. Overall, qualitative evidence for Social Networking and Music apps has the highest variation.

Finally, we look at the impact of top apps on handset demand. Understanding the impact of apps on handset demand could prove crucial to platform operators. [18] In Table 13, we display the apps on Apple and Android platforms that have the largest estimated impact on their platform sales. These numbers are computed by removing each app from the available set of apps on the platform. We assume our intervention does not affect consumers' belief ($\hat{G}$) over the evolution of handset qualities and hold it constant. We observe that the Social Media apps like Facebook have the highest estimated impact on both platforms. Removing Facebook from Apple or Android can reduce the demand of respective handsets by -5.9% and -5.6%. We also observe that apps owned by Apple's competitor Google like Youtube (-3.5%) have a significantly high impact on Apple's handset sales.

Our model allows us to factor in the two-sided nature of the market and measure the dollar values that consumers attach to free apps (or goods) like Facebook or Instagram. Prior literature has struggled to measure the economic value of free goods or services like Facebook and has mainly relied on customer surveys (for instance Brynjolfsson et al. (2019)) to measure their dollar value. However, our model allows us to infer these values from purchase decisions made by consumers. In Table 13, we report the dollar value for one month of access to top apps perceived by both Apple and Android customers.[19] Using our model, we estimate that one month of access to the Facebook app is worth around \$13 (or \$9) to Apple (or Android) users[20]. These numbers are a little lower than \$48 for one month of access as reported by Brynjolfsson et al. (2019) using survey data. This is in line with existing literature on survey methods, which has found that consumers tend to overstate their value for products in surveys (Hausman 2012).

Entry Cost Estimates In Table 14, we show our estimated entry cost to Android and iOS, respectively. The estimated costs are normalized to the number of downloads. Note that instead of point estimates, we recover set estimates of developers' entry costs. On average, the development costs for Android are much higher than for Apple iOS. This is in line with existing literature (e.g. He et al. (2018)), that points out how dealing with multiple prevailing operating system versions of Android, increases development costs for developers. Further, many business reports (for instance

[18] For instance, many countries have recently considered putting sanctions on certain apps. The United States banned WeChat (see https://www.usatoday.com/story/tech/2020/08/07/what-is-wechat-why-trump-wants-ban-tencent/3319217001/ for United States users and the ban extended to Google and Apple app stores offering the app.

[19] One limitation of our analysis is that customers can still access apps through other channels, for example, their computers. However, since the majority of traffic, especially to social media websites, comes through apps, we can still get very reasonable estimates of their value to consumers. For instance, 80% of users access Facebook only through mobile, and Instagram was "app only" during the time of our study.

[20] We want to acknowledge that, given we do not account for brand loyalty effects in our model, the economic value of apps could be underestimated, and the actual value could be significantly larger.

Table 13. Apps with largest estimated impact and value

		(Apple)	Value($)
Apple	Facebook	-5.940	13.04
	Instagram	-5.195	11.40
	Youtube	-3.436	7.54
	Facebook-Messenger	-3.012	6.61
		(Android)	Value ($)
Google Play	Facebook	-5.626	8.58
	Facebook-Messenger	-4.840	7.38
	Instagram	-3.250	4.95
	Snapchat	-1.630	2.48

Note. The table reports the top four apps with the highest estimated impact on both Apple and Google Play. The numbers indicate the percentage drop in handset sales following the removal of the app from platform. The reports are estimated by simulating a permanent removal of apps from either platform. The second column shows the estimated impact on the competing platform. The second column report the dollar value as perceived by customers on Apple and Android for one month of app availability.

see Car (2015)) have also indicated how Android programming requires more written code for the same functionality than Apple and can add to development costs. We estimate the entry cost to Android to be on average 33% higher[21]. Our estimates are in the same ranges as reported by many media reports (for instance see Car (2015)). We also find that on average "Photo and Video" and "Social Networking" apps incur much higher entry costs consistently across both iOS and Android. This is probably due to the additional complexity that comes with developing "Photo and Video" and "Social Networking" apps. Both "Photo and Video" and "Social Networking" apps involve dealing with camera applications. Since developing *optimized* camera applications might involve explicitly accounting for various camera hardware available across Android and Apple handsets, it might add additional cost to developer effort. Further, social media apps have an additional component of handling social networks, which could add to developers' costs.

Effect of Developers' Delayed Entry Many businesses do not find it profitable to launch apps for both iOS and Android at once, so they start with one platform. For instance, the Android version of Airbnb was launched 14 months after the iOS version. Similarly, Clubhouse is only available on iOS at the time of this writing. This is partly due to resource constraints and partly

[21] We use the center of the identified set for comparison purposes.

Table 14. Estimated Entry Costs

	Android Entry Cost	Apple Entry Cost
Entertainment	16,714	4,606
	[12,714 18,114]	[800 6,700]
Games	17,891	(12,886 12,972)
	[14,391 19,391]	[2,937 19,837]
Music	17,906	(8,433 8,947)
	[8,606 20,706]	[3,200 13,700]
Photo and Video	(27,356 28,001)	(7,908 7,976)
	[17,750 33,150]	[4,300 10,000]
Social Networking	(21,301 22,349)	(25,106 25,358)
	[11,750 29,150]	[19,325 30,725]
Utilities	14,161	5,352
	[8,961 16,161]	[2,000 6,900]
Others	17054	(3550 3678)
	[13,554 8,754]	[200 6,300]

Note. We report in round brackets, the identified set of costs to develop an app for Android or Apple across the various genres. All values that satisfy the moment inequalities (i.e., minimize the equation 29 to 0), belong to the identified set, however, if none satisfies, then value that minimizes the absolute value of deviation in inequalities is used. The estimated costs are normalized to the number of downloads. The 95% confidence intervals are reported in square brackets. We compute the 95% confidence interval as the set that includes each element of the identified set with fixed probability (see Imbens and Manski (2004) and Romano and Shaikh (2008)). We detail the procedure to compute the confidence interval in Appendix A.3.

because businesses often begin with an experimental version to gauge customers' preferences and tend to use only one platform for this initial experimentation.

In our sample, 2103 apps launched on both Apple and Google Play (up to the period two years after the end of the data we use for estimation, i.e., up to Jan 2017). However, the vast majority of app developers launch on Apple first. To understand the economic impact of developers' preference for Apple iOS over the Android platform, we estimate how much more sales Android handsets could have generated (over the period of our study) if apps that launched on Apple first were instead launched on Android at the same time. We then compare the counterfactual demand to the

actual realized demand of Android handsets in 2013-2014 to ascertain the losses due to developer's sequential entry.

To simulate the counterfactual demand for Android handsets, we artificially add new apps to the Android platform as they arrive on the Apple platform and compute the corresponding demand. One caveat in artificially adding apps to the Android platform is ascertaining the quality levels of new apps to be added. For this purpose, we non-parametrically estimate the Google Play quality (ψ^t_{play}) at any time t, as a function of iOS quality (ψ^t_{ios}), genre and time. Using this we simulate the quality of an app if it was released at an earlier time on Android.

We estimate that Android suffers significant losses due to developers preferring Apple for their initial launch. Specifically, we estimate revenue losses from handset sales to be around \$208M in 2013 and \$260M in 2014.

7. Platform Fees and Market Equilibrium

In this section, we use our demand and developer entry cost estimates to estimate the platform fees Apple and Google should charge in equilibrium. Our interest in this question is motivated by the recent debates and legal battles surrounding these fees. One example is the legal battle between Epic Games and Apple. Epic, the creator of the popular video game Fortnite, attempted to bypass Apple's in-app purchasing system and was subsequently removed from the App Store. Epic argues that the 30% platform fee that Apple charges for in-app purchases on the App Store is excessive, stifles competition, and does not suitably acknowledge the value of apps to the platform. Similar arguments against this so-called Apple tax have been made by scores of other developers such as Spotify. To estimate the equilibrium platform fees, we the model the system as a dynamic game. We assume the following timing of actions –

1. At the beginning of the game both Apple and Google decide on their respective platform fees.
2. At every time period t –
   (a) $\{\delta_{i,j,t}\}$ and $\{\psi_{i,g(j),l,t}\}$ for all handsets and apps is determined;
   (b) Apple and Android OEMs set handset prices;
   (c) Consumers make decisions on handset purchases;
   (d) App developers make their entry decisions;
   (e) Consumers download and purchase apps;
   (f) Profits are realized for Apple, Google, and Android OEMs.

Outside of developers, there are three key agents in the smartphone markets – (1) Apple, (2) Google, and (3) Android OEMs (Original equipment manufacturers)[22]. The profit made by each of these in period t can be expressed as follows –

[22]For the sake of simplicity we are clubbing all OEMs together.

$$\Pi_{Apple,t}(\vec{p}, fee_{Apple}) = \underbrace{\sum_{J \in J_t^{Apple}} ((p_{j,t} - mc_{j,t}))Q_{j,t}}_{\textbf{Profit from Apple Handset Sales}} + \underbrace{\left(\sum_{i \in L_{Apple,t}} Rev_{i,t}(fee_{Apple}) \right) - Cost_{Apple,t}}_{\textbf{Profit from iOS App Store}}$$

$$\Pi_{Google,t}(fee_{\text{Google-Play}}) = \underbrace{\left(\sum_{i \in L_{\text{Google-Play},t}} Rev_{i,t}(fee_{\text{Google-Play}}) \right) - Cost_{\text{Google-Play},t}}_{\textbf{Profit from Google-Play App Store}}$$

$$\Pi_{\text{Android-OEM},t}(\vec{p}) = \underbrace{\sum_{J \in J_t^{Android}} ((p_{j,t} - mc_{j,t}))Q_{j,t}}_{\textbf{Profit from Android Handset Sales}}$$

where $p_{j,t}$ denotes the price of an Apple or Android handset j at time period t; $mc_{j,t}$ denotes the marginal cost of the handset j during time period t; $Q_{j,t}$ denotes the quantity of handset j sold in time period t; $Rev_{i,t}$ denotes the revenue made by app i in time period t on its host platform, it includes revenue from both new and existing users; $Cost_{.,t}$ denotes the cost for managing the app platform by Apple or Google in time period t; fee_{Apple} and $fee_{\text{Google-Play}}$ denote the fees charged by Apple and Google to app developers on Apple iOS and Google Play app store, respectively. In the case of Apple, the profit term includes two terms – (1) profit from Apple handset sales, and (2) profit from the app store. On the other hand, the profit term for Android only includes profit from the Google Play store. The profit Android-OEM includes the profit from the handset sales. Note that for the sake of simplicity, we assume that the marginal cost of managing an app on the app store is 0 and the app platforms only incur some fixed cost. We believe this is a reasonable assumption for software products.

A Markov Perfect Equilibrium in this game is a vector of strategy functions σ such that each player's (Apple, Google, or Android-OEM) strategy maximizes the value of the agent for each possible state and taking as given the other agents' strategies.

$$\sigma^i(\Omega_t) = \begin{cases} \{p_t(\Omega_t), fee_{Apple}(\Omega_t)\} \text{ if i = Apple} \\ \{fee_{\text{Google-Play}}(\Omega_t)\} \text{ if } i = \text{Google-Play} \\ \{p_t(\Omega_t)\} \text{ if } i = \text{Android-OEM} \end{cases} \tag{30}$$

Let $U_i^{\sigma}(\Omega_t)$, represent the expected present value of the agent i's current and future profits given that the other agent behave according to their respective strategies and that agent i chooses its current and future actions optimally. The value for each of the agents' is then implicitly defined as the solution to the following bellman equation -

$$U_i^\sigma(\Omega_t) = \begin{cases} \max_{\vec{p}, fee_{Apple}} \Big\{ \Pi_{Apple,t}(\Omega_t, p_t, fee_{Apple}) + \beta\, \mathbb{E}[U_i^\sigma(\Omega_{t+1}) | \Omega_t, p_t, fee_{Apple}] \Big\} \text{ if i = Apple} \\ \max_{fee_{\text{Google-Play}}} \Big\{ \Pi_{Google,t}(fee_{\text{Google-Play}}) + \beta\, \mathbb{E}[U_i^\sigma(\Omega_{t+1}) | \Omega_t, fee_{\text{Google-Play}}] \Big\} \text{ if } i = \text{Google-Play} \\ \max_{\vec{p}} \Big\{ \Pi_{Android-OEM,t}(\Omega_t, p_t) + \beta\, \mathbb{E}[U_i^\sigma(\Omega_{t+1}) | \Omega_t, p_t] \Big\} \text{ if } i = \text{Android-OEM} \end{cases} \tag{31}$$

Unlike prices, it is very hard operationally for the platforms to set fees every period and thus they are set constant for a long period of time. Therefore, we assume that these fees are time-invariant and not dependent on the information set Ω_t. Since our focus is on platform fees rather than handset pricing, we make some simplifying assumptions to simplify the dynamic game. We assume that the prices of Apple and Android handsets evolve according to an AR(1) process, which allows us to estimate a static game where players optimize based on the present value of price and platform fees, as future profits become a deterministic function of present choices. For further simplicity, we assume that the profit margin on Apple phones is fixed at 40%, consistent with information from public sources Laricchia (2023).

To simulate revenue from apps and handsets in future periods, we use the estimated demand processes F and G. We assume that changes in platform fees affect only the levels $\{\delta_{i,j,t}, \psi_{i,g(j),l,t}\}$ and not the evolution processes F and G. This enables us to simulate sales and downloads (and therefore revenue) for handsets and apps in the future.

To estimate the equilibrium, we need to determine two things. First, we need to estimate how revenue from existing apps changes as a function of the app store commission rate. To do this, we estimate how a change in platform fees affects app downloads using our app demand model, which captures the relationship between app price, in-app purchases, and app downloads. Second, we need to estimate how revenue changes from entry/exit of new apps as a function of the app store commission rate. To do this, we consider the term $\digamma_t = \sum_{l \in \Lambda_t} W_{l,t}$, where Λ_t denotes the set of apps launched in period t. $\digamma_t$ denotes the utility the consumer expects to get from apps launched in period t. Note that the term $\digamma_t$ is platform-specific, and for the sake of brevity, we have dropped the platform subscript. Consider the term $\digamma_t^g$ that denotes utility from apps of genre g launched in period t. We construct a non-parametric estimator $\hat{b}$ such that $\digamma_t^g = b(\mathbb{E}[D_t|\Omega_{t-k}], \theta_g)$, where $\mathbb{E}[D_t|\Omega_{t-k}]$ denotes the downloads an app developer of genre g expects to get if they chose to launch in period t and θ_g denotes the entry cost for genre g. Using the estimator $\hat{b}$, we simulate the effect of $\Delta\,\mathbb{E}[D_t|\Omega_{t-k}]$ on $\digamma_t^g$. To calculate the total revenue (i.e., $\sum_{i \in L_{Apple/Play,t}} Rev_{i,t}$) from app downloads, we follow a step-by-step process. The $L_{Apple/Play,t}$ indicates all the apps or Apple (or Google Play) platform at time t. However, in our dataset, we only observe a subsample of apps. We first determine the revenue each of the apps in our sample has made. Then we multiply this

amount by a calibration factor to get a good estimate of the total app store revenue. The specifics of how we figure out an app's revenue based on the number of its downloads are explained in an additional section, referred to as Appendix H.

In Figure 15, we report platform equilibrium fees for two values of pass-through. Pass-through refers to the extent to which changes in platform fees are reflected in the final prices paid by consumers for apps and in-app purchases. For example, if an app's price is $100 and the platform fee increases from 30% to 35%, a 100% pass-through would result in the entire $5 tax increase being passed on to consumers, leading to a new price of $105. Note that, in our model, we do not explicitly model the pricing decisions of app developers. Instead, we will use different values of pass-through to represent the equilibrium response of app developers to changes in platform fees and handset prices. This approach simplifies the estimation problem considerably while allowing us to focus on the main question. We estimate that Google Play's market equilibrium fees are a little higher than Apple's. This reflects the fact that Apple also has a source of revenue from handsets and thus has the incentive to keep its fees low as the higher fees create a negative externality on handset revenue.

To further delve into how the strength of indirect network externalities impacts equilibrium platform fees, we adjusted our model. We raised the coefficient on Γ, the measure of the value from the apps, by 10% and recalculated the platform equilibrium fees under a condition of 100% pass-through. We find that increasing the strength of indirect network externalities by 10% leads to a decrease in optimal platform fees by 2.5% for Apple and 0.5% for Android.[23] This pattern confirms that vertical integration within the Apple platform results in a much tighter coupling of the two sides of the market, compared to Android.

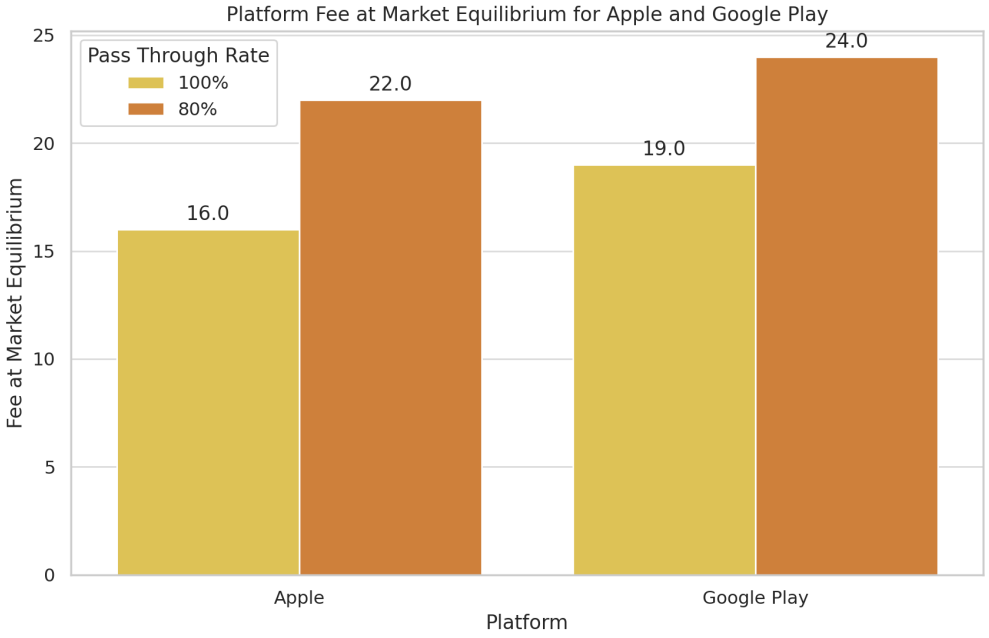
8. Concluding Remarks and Limitations

In this paper, our contributions are two-fold: (i) we quantify the heterogeneous indirect network externalities, thereby determining the economic value of apps, and (ii) we evaluate the implications of these indirect network effects on the fees charged by Apple and Android to app developers.

Our investigation into these phenomena comes at a crucial time, given the heightened significance of indirect network externalities in light of various antitrust lawsuits against Apple and Google. Understanding the two-sided nature of the market and the resulting network externalities becomes all the more critical given the recent legal precedent emphasizing the need to examine both sides of the market when indirect network effects are deemed substantial.

[23] Our model does not account for brand loyalty, which likely leads to an underestimation of indirect network effects. Consequently, the actual magnitude of optimal fees may be more pronounced than our estimates suggest. This implies that our reported values should be considered conservative lower bounds rather than definitive measurements.

Figure 15. Platform Fees



The plot indicates the estimated equilibrium fees for Apple and Google Play, assuming an 80% and 100% pass-through for app developers. Pass-through measures how changes in platform fees affect consumer prices for apps and in-app purchases. It reflects the extent to which fee increases are transferred to the end user. A 100% pass-through means the entire fee increase is added to the consumer price. For example, consider an app initially priced at $100 with a 30% platform fee. If the fee increases to 35%, representing a $5 increase, a 100% pass-through would result in the new consumer price rising to $105.

Our examination provides a much-needed empirical basis to ascertain if network externalities in smartphone markets are "substantial," giving a more balanced view of competition within smartphone platforms. We also address the contentious issue of platform fees imposed on developers, a subject that has led to intense debates and allegations of anti-competitive behavior. Our analysis offers unique insights into the role of network externalities within the fee structures.

To conduct this analysis, we develop a dynamic model of demand for handsets and apps and use a unique panel dataset of handset sales and app downloads to estimate the model. We estimate significant heterogeneity in externalities exerted by apps on handset sales. The model helps us quantify the very significant (negative) impact of delayed developer entry on Android.

Our findings have many implications for platform operators and policymakers. For platform operators, our findings show how important it is for platforms to detect trending apps that are exclusively available on competing platforms and to court those developers so as to minimize entry delays. This could explain why Microsoft decided to insource[24] development of the Facebook app for its mobile platform back in the day.

[24] See https://blogs.windows.com/devices/2013/03/22/facebook-apps-on-your-nokia-lumia/

Our modeling approach and specific findings are potentially also relevant to several issues that surround competition between app platforms. For example, our results demonstrate that there is significant heterogeneity in the externalities various apps exert on handset sales and therefore in their value to the platform ecosystem. Furthermore, our estimates of entry cost highlight the contrasting difference in entry costs across the two platforms.

Regarding platform fees, we estimate that the equilibrium levels of fees for both Apple and Android are somewhat similar, although Android's fees are slightly higher. The slightly higher optimal fee for Google underlines the importance of Apple's vertical integration in creating a complementary demand. This differential implies that Google's incentive to lower fees to attract developers might be less significant than previously thought. Additionally, our estimates suggest that increasing the strength of indirect network externalities has much less impact on Android fees compared to Apple. This pattern confirms that vertical integration within the Apple platform results in a much tighter coupling of the two sides of the market, compared to Android. These varied effects of network externalities on these platforms could help legal experts differentiate the specific circumstances of these two platforms.

Finally, we also want to point limitations of our analysis. In our model, we do not allow for substitution across apps and compromise it to account for consumer dynamics. However, we demonstrate that in our case, a dynamic model demand could lead to much better out-of-sample predictive performance than a model that only accounts for substitution. Incorporating both substitution and dynamics requires keeping track of all possible consumption sets over a large number of products, and hence is not computationally tractable. We believe, methods to tractably solve such combinatorially hard problems could be an interesting direction to work on in the future. Additionally, since we do not have access to individual-level data, we cannot explicitly model customers' brand loyalty. Incorporating this aspect in future studies could help obtain more precise estimates from our model. Furthermore, regarding the counterfactual analysis, we make several simplifying assumptions to keep the problem computationally tractable. For example, we do not separately identify the pass-through rate of platform fees and conduct our analysis for the two most plausible pass-through values. Given that our estimates of platform fees are closely aligned with the current charges imposed by Apple and Android, it makes us confident that our assumptions adequately capture the true data-generating process to a significant degree.

References

Aguirregabiria, V. and Nevo, A. (2013). Recent developments in empirical io: Dynamic demand and dynamic games. Advances in Economics and Econometrics, 3:53–122.

Allon, G., Askalidis, G., Berry, R., Immorlica, N., Moon, K., and Singh, A. (2019). When to be agile: Ratings and version updates in mobile apps. Available at SSRN 3476286.

AppleInsider (2023). Every apple app store fee, explained: How much, for what, and when. Accessed: 2023-06-07.

Berry, S., Levinsohn, J., and Pakes, A. (1995). Automobile prices in market equilibrium. Econometrica: Journal of the Econometric Society, pages 841–890.

Bresnahan, T., Orsini, J., and Yin, P.-L. (2014). Platform choice by mobile app developers. NBER working paper.

Brynjolfsson, E., Collis, A., and Eggers, F. (2019). Using massive online choice experiments to measure changes in well-being. Proceedings of the National Academy of Sciences, 116(15):7250–7255.

Brynjolfsson, E., Eggers, F., and Gannamaneni, A. (2018). Measuring welfare with massive online choice experiments: A brief introduction. In AEA Papers and Proceedings, volume 108, pages 473–76.

Car, T. (2015 (accessed October, 2015)). Android Development Is 30% More Expensive Than iOS. And We — Infinum.

Chintagunta, P. K., Qin, M. S., and Vitorino, M. A. (2018). Licensing and price competition in tied-goods markets: An application to the single-serve coffee system industry. Marketing Science, 37(6):883–911.

Conlon, C. T. (2012). A dynamic model of prices and margins in the lcd tv industry. unpublished, Columbia University.[818, 835] Daley, Brendan and Brett Green (2012),"Waiting for news in the market for lemons." Econometrica, 80:1433–1504.

Corts, K. S. and Lederman, M. (2009). Software exclusivity and the scope of indirect network effects in the us home video game market. International Journal of Industrial Organization, 27(2):121–136.

Cragg, J. G. and Donald, S. G. (1993). Testing identifiability and specification in instrumental variable models. Econometric Theory, pages 222–240.

Derdenger, T. and Kumar, V. (2019). Estimating dynamic discrete choice models with aggregate data: Properties of the inclusive value approximation. Quantitative Marketing and Economics, 17(4):359–384.

Dubé, J.-P. H., Hitsch, G. J., and Chintagunta, P. K. (2010). Tipping and concentration in markets with indirect network effects. Marketing Science, 29(2):216–249.

Existek (2020). App development timeline: How long does it take?: Existek blog. Existek Blog.

Garcia-Swartz, D. D. and Garcia-Vicente, F. (2015). Network effects on the iphone platform: An empirical examination. Telecommunications Policy, 39(10):877–895.

Garg, R. and Telang, R. (2013). Inferring app demand from publicly available data. MIS Quarterly, pages 1253–1264.

Ghose, A., Goldfarb, A., and Han, S. P. (2012). How is the mobile Internet different? search costs and local activities. Info. Systems Res., 24(3):613–631.

Ghose, A. and Han, S. P. (2014). Estimating demand for mobile applications in the new economy. Management Science, 60(6):1470–1488.

Gowrisankaran, G. and Rysman, M. (2012). Dynamics of consumer demand for new durable goods. J. Political Econom., 120(6):1173–1219.

Hagiu, A. (2006). Pricing and commitment by two-sided platforms. The RAND Journal of Economics, 37(3):720–737.

Hansen, L. P. (1982). Large sample properties of generalized method of moments estimators. Econometrica: Journal of the Econometric Society, pages 1029–1054.

Hausman, J. (2012). Contingent valuation: from dubious to hopeless. Journal of Economic Perspectives, 26(4):43–56.

He, D., Li, L., Wang, L., Zheng, H., Li, G., and Xue, J. (2018). Understanding and detecting evolution-induced compatibility issues in android apps. In 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE), pages 167–177. IEEE.

Help, G. P. C. (2023). Understanding google play's service fee. Accessed: 2023-06-07.

Hendel, I. and Nevo, A. (2006). Measuring the implications of sales and consumer inventory behavior. Econometrica, 74(6):1637–1673.

Ho, K. and Pakes, A. (2014). Hospital choices, hospital prices, and financial incentives to physicians. American Economic Review, 104(12):3841–84.

Imbens, G. W. and Manski, C. F. (2004). Confidence intervals for partially identified parameters. Econometrica, 72(6):1845–1857.

Laricchia (2023). Apple's gross margin 2005-2023. Accessed: 2023-05-25.

Lee, R. S. (2013). Vertical integration and exclusivity in platform and two-sided markets. American Economic Review, 103(7):2960–3000.

Lee, S. Y. (2014). Examining the factors that influence early adopters' smartphone adoption: The case of college students. Telematics and Informatics, 31(2):308–318.

Li, X., Bresnahan, T., and Yin, P.-L. (2016). Paying incumbents and customers to enter an industry: Buying downloads. Available at SSRN 2834564.

Melnikov, O. (2013). Demand for differentiated durable products: The case of the us computer printer market. Economic Inquiry, 51(2):1277–1298.

Nair, H., Chintagunta, P., and Dubé, J.-P. (2004). Empirical analysis of indirect network effects in the market for personal digital assistants. Quantitative Marketing and Economics, 2(1):23–58.

Nevo, A. (2001). Measuring market power in the ready-to-eat cereal industry. Econometrica, 69(2):307–342.

Newey, W. K. and McFadden, D. (1994). Large sample estimation and hypothesis testing. Handbook of econometrics, 4:2111–2245.

Pakes, A. (2012). Nber econometrics minicourse. National Bureau of Economic Research.

Pakes, A., Porter, J., Ho, K., and Ishii, J. (2015). Moment inequalities and their application. Econometrica, 83(1):315–334.

Research, A. (2023). Tablets will generate 35% of this year's $25 billion. https://www.abiresearch.com/press/tablets-will-generate-35-of-this-years-25-billion-/. Accessed: 2023-06-29.

Romano, J. P. and Shaikh, A. M. (2008). Inference for identifiable parameters in partially identified econometric models. Journal of Statistical Planning and Inference, 138(9):2786–2807.

Roodman, D. (2007). A short note on the theme of too many instruments. Center for Global Development Working Paper, 125.

Rust, J. (1994). Structural estimation of Markov decision processes. In Engle, R. and McFadden, D., editors, Handbook of Econometrics, volume 4, chapter 51, pages 3081–3143. North Holland.

Rysman, M. (2004). Competition between networks: A study of the market for yellow pages. The Review of Economic Studies, 71(2):483–512.

Rysman, M. (2009). The economics of two-sided markets. Journal of economic perspectives, 23(3):125–143.

Stock, J. H. and Yogo, M. (2002). Testing for weak instruments in linear iv regression. Technical report, National Bureau of Economic Research.

United States Supreme Court (2018). Ohio v. american express co. Supreme Court Reporter, 138:2274. Docket No. 16-1454.

Verge, T. (2023). Google will reduce play store cut to 15 percent for a developer's first $1m in annual revenue. Accessed: 2023-06-07.

Wooldridge, J. M. (2010). Econometric Analysis of Cross Section and Panel Data. MIT press.

Appendices

A. Estimation Details

A.1. Computation Details

We will briefly outline the steps we take for estimating our model parameters. To be able to estimate our model using moment conditions (24), we first need to compute demand shocks ν^{hs} and ν^{app} as a function of the data and model parameters. Broadly, we will follow a similar procedure as used in Berry et al. (1995), with some deviations. The key insight on which our estimation is based is that the unobservable product characteristics for handset and apps only appear in the expression of mean lifetime utilities of handset ($\delta_{j,t}$) and apps ($\psi_{g(j),l,t}$), respectively. Thus, we will first estimate the mean lifetime utilities (δ and ψ) from observed data as a function of model parameters. For this, we will use the analytical expressions for purchase probabilities using our model. As described in Section 4, the probability of purchasing a particular handset in a given period (see equation 8) and the probability of downloading an app in a given period (see equation 40) can be written down in terms of model parameters

$$s^{hs}_{ijt} = \frac{\exp\left(\delta_{i,j,t}\right)}{\exp\left(\zeta_{i,j,t}\right) + \exp\left(\beta\,\mathbb{E}(V(\zeta_{i,j,t+1})|\zeta_{i,j,t})\right))}$$

$$s^{app}_{i,l,g(j),t} = \frac{\exp\left(\psi_{i,l,g(j),t}\right)}{\exp\left(\psi_{i,l,g(j),t}\right) + \exp\left(\beta\,\mathbb{E}(W(\psi_{i,l,g(j),t+1}))|\psi_{i,l,g(j),t})\right))}$$

We can map these expressions to the observed shares and solve the resultant implicit system of equations to solve for the mean lifetime utilities terms δ and ψ

$$s^{hs}_{.t}(\cdot) = S^{hs}_{.t} \qquad\qquad s^{app}_{.t}(\cdot) = S^{app}_{.t}$$

where $S^{hs}_{.t}$ and $S^{app}_{.t}$ are observed aggregate market shares of particular handsets and apps.

In our case, since we cannot analytically integrate over the customer distribution, we numerically compute left hand side of the above equations i.e., the aggregate market shares $s_{.t}$ (see Appendix for details). To model consumer heterogeneity, we assume that price sensitivities of consumers are parameterized as follows: $\alpha^{p,l}_i = \alpha^{p,l} + \sigma^l \nu_i$, for $l \in \{hs, app\}$ and ν_i is drawn from standard normal distributions. To compute the market shares, we also need access to value functions $V(\delta)$ and $W(\psi)$. To compute them we first use an ordinary least square regression to estimate the incidental parameters (ϕ^{hs} and ϕ^{app}) using the currently guessed values of mean utilities δ and ψ. We then compute the value functions $V(\delta)$ and $W(\psi)$ using the imputed values of incidental parameters using a fine grid approximation over δ and ψ. We now carry out a contraction mapping similar to BLP to impute the values of mean utilities from the observed data

$$\delta^{h+1}_{.t} = \delta^{h}_{.t} + \lambda^{hs}(s^{hs}_{.t}(\delta^{h}_{.t}, \phi^{hs}, \sigma^{hs}) - S^{hs}_{.t})$$

$$\psi_{.t}^{h+1} = \psi_{.t}^{h} + \lambda^{app}(s_{.t}^{app}(\psi_{.t}^{h}, \phi^{app}, \sigma^{hs}, \sigma^{app}) - S_{.t}^{app})$$

Note that predicted market shares for each particular app also depends on the heterogeneity parameters of handsets (σ^{hs}). We take a two step estimation procedure where σ_{hs} is estimated first from the handset model. Adding tuning parameters λ^{hs} and λ^{app} allows for faster convergence of contraction mapping. We carry out the above iterations for each model until $|\delta_{.t}^{h+1} - \delta_{.t}^{h}|$ and $|\psi_{.t}^{h+1} - \psi_{.t}^{h}|$ are less than the error tolerance threshold. After the convergence on mean utility, the incidental parameters are updated using an ordinary least square regression on the updated mean utility. The contraction mapping iterations are again run to get updated mean utility. We stop once we get convergence on the incidental parameters. With the mean utility in hand, we will use the expressions in the equation to compute the unobservable product characteristics as a function of model parameters

$$\xi_{jt} = \hat{\delta}_{j,t} - \vec{x}_{jt}\theta^{hs}$$

$$\eta_{jt} = \hat{\psi}_{g(j),l,t} - \vec{x}_{g(j),l,t}\theta^{app}$$

Our econometric error terms can then simply by constructed by taking the first difference of ξ and η. Since we know all time invariant characteristics get canceled out after taking the first difference, we only include the time varying characteristics in $\vec{x}_{j,t}^{hs}$ and $\vec{x}_{g(j),l,t}^{app}$. Finally, the GMM objective function for the handset and app models can be constructed by interacting our econometric error terms with above specified instruments for each model respectively

$$\arg\min_{\theta^{hs}} (\nu_{j,t}^{hs}(\theta^{hs})^T \mathbf{Z}_t^{hs}) \mathbf{W}^{hs} (\nu_{j,t}^{hs}(\theta^{hs})^T \mathbf{Z}_t^{hs})^T$$

$$\arg\min_{\theta^{app}} (\nu_{g(j),l,t}^{app}(\theta^{app})^T \mathbf{Z}_t^{app}) \mathbf{W}^{app} (\nu_{g(j),l,t}^{app}(\theta^{app})^T \mathbf{Z}_t^{app})^T$$

where $\mathbf{W}^{hs}$ and $\mathbf{W}^{app}$ are the weighting matrices. To estimate the model parameters pertinent to time-invariant characteristics we use a second ordinary least square regression.

A.2. Estimation of Demand Model

As described in Section 5, we use the generalized method of moments to estimate our demand model for both handsets and apps. We construct the GMM objective function using equations (24) and (23). We use a **iterative**-nested estimation procedure, which simplifies the estimation somewhat. We use a **iterative**-nested estimation procedure because we need customer heterogeneity parameters of handsets to estimate apps demand parameters. This is because the distribution of consumers that can download apps is dependent on the type of customers who buy the handset. The steps of our estimation procedure are as follows.

1. **Initialization Phase**: In this stage, we get an initial estimate of customer heterogeneity ($\sigma^{p,hs}$) over handset prices. For initialization purpose, we will not take into account the platform utility Γ. The mean of the platform utility is captured by the handset fixed effects $\bar{\xi}_j$, and the deviations from mean are captured by the unobserved handset utility term $\Delta\xi_{j,t}$. To estimate the handset specific parameters, we minimize the following GMM objective function

$$\arg\min_{\theta^{hs}} (\nu^{hs}_{j,t}(\theta^{hs})^T \mathbf{Z}^{hs}_t) \mathbf{W}^{hs} (\nu^{hs}_{j,t}(\theta^{hs})^T \mathbf{Z}^{hs}_t)^T$$

where $\nu^{hs}_{j,t}(\theta) = \xi_{j,t} - \xi_{j,t-1}$. We use the instruments as detailed in Section 5, except the lagged values of platform utility (Γ). Next, to estimate $\xi_{j,t}$, we consider the following set of equations. Consider δ_{jt} as the *mean lifetime utilities* of handset for consumers with mean parameter values $\boldsymbol{\alpha}^{p,hs}$. By definition of mean lifetime utilities (see equation (1)) we have

$$\delta_{j,t}(\theta) = \boldsymbol{\alpha^x} x_{j,t} + \boldsymbol{\alpha^{p,hs}} \log(p)_{j,t} + \xi_{j,t}$$
$$\xi_{j,t} - \xi_{j,t-1} = \delta_{j,t}(\theta) - \delta_{j,t-1}(\theta) - \boldsymbol{\alpha^x}(x_{j,t} - x_{j,t-1}) -$$
$$\alpha^{p,hs}(\log(p)_{j,t} - \log(p)_{j,t-1})$$

Thus, we have

$$\nu^{hs}_{j,t}(\theta) = \delta_{j,t}(\theta) - \delta_{j,t-1}(\theta) - \boldsymbol{\alpha^x}(x_{j,t} - x_{j,t-1}) -$$
$$\alpha^{p,hs}(\log(p)_{j,t} - \log(p)_{j,t-1})$$

As one can note, the handset demand shocks are a function of handset mean parameters, handset observed time-varying characteristics and handset *mean lifetime utility* δ_{jt}.

Customer Heterogeneity: To simulate customer heterogeneity, we assume that customer price taste parameters, at the start of our study are distributed as independent normal random variables (i.e., $\boldsymbol{\alpha}^p_i - \boldsymbol{\alpha}^p \sim N(0, \Sigma)$),

$$\Sigma = \begin{bmatrix} \sigma_{hs} & 0 \\ 0 & \sigma_{app} \end{bmatrix} \tag{32}$$

Thus,

$$\boldsymbol{\alpha}^{p,hs}_i = \boldsymbol{\alpha}^{p,hs} + \nu^{hs}_i \sigma_{p,hs} \text{ and } \boldsymbol{\alpha}^{p,app}_i = \boldsymbol{\alpha}^{p,app} + \nu^{app}_i \sigma_{app}{}^{25} \tag{33}$$

Also, note that $\delta_{i,j,t}$ can also be written as follows

$$\delta_{i,j,t} = \delta_{j,t} + (\boldsymbol{\alpha}^{p,hs}_i - \boldsymbol{\alpha}^{p,hs})(\log(p)_{j,t}) \tag{34}$$

Also as discussed in Section 4, the logit inclusive value $\zeta_{i,j,t}$ and mean lifetime utility for handset, $\delta_{i,j,t}$ are sufficient statistics for determining if a consumer of type i will purchase a

[25]Since few paid apps were launched before our period of study, $\alpha^{\mathbf{p,app}}$ and σ_{app} could potentially vary across them. However, for numerical tractability we impose $\alpha^{\mathbf{l}}_{\mathbf{app}} \approx \alpha_{\mathbf{app}}$, and $\sigma^l_{app} \approx \sigma_{app}$ $\forall l$.

handset j at time t or not. The probability of consumer type i, to buy handset j in period t is given by

$$\hat{s}_{i,j,t}(\delta_{i,j,t}, \zeta_{i,t}) = \frac{\exp(\zeta_{i,t})}{\exp(\zeta_{i,t}) + \exp(\beta \mathbb{E}(V(\zeta_{i,t+1})|\zeta_{i,t}))} * \frac{\exp(\delta_{i,j,t})}{\exp(\zeta_{i,t})} \tag{35}$$

However, in our data we only observe aggregate shares for each handset. Thus, we numerically approximate the normally distributed customers as a finite distribution of consumer type i so that each type has a corresponding weight w_{it}. This also allows us to update the distribution of customers when they exit after making a purchase by adjusting the weights.

$$\hat{s}_{j,t}(\delta_{i,j,t}, \zeta_{i,t}) = \sum_i w_{it}^{hs} \hat{s}_{i,j,t}(\delta_{i,j,t}, \zeta_{i,t})$$
$$\sum_i w_{i0}^{hs} = 1$$
$$w_{it+1}^{hs} = w_{it}^{hs}(1 - \sum_{j \in H_t} s_{ijt})$$

For the initial consumer distribution, the weights sum upto 1. Ideally one can use initially weights to be $\frac{1}{n_I}$, where I is the number of individuals to be simulated. However, we use the points (i.e., (w_i, ν_i)) generated through gauss hermite quadrature rule to get better approximation of the consumer distribution and simulate customer type i. Note, that to estimate market shares we also need access to the term $\mathbb{E}(V(\zeta_{i,j,t+1})|\zeta_{i,j,t}))$. To estimate the expected value of the value function, we will first estimate the incidental parameters ϕ_{i1} and ϕ_{i2} using a first stage regression based on current guess of ζ

$$\zeta_{i,t+1} = \phi_{i1} + \phi_{i2}\zeta_{i,t} + \mu_{it+1} \tag{36}$$

We then compute the value functions $V(\zeta)$ using the imputed values of incidental parameters We use a fine grid approximation over ζ and use value iteration to estimate the value function V. Next, similar to Berry et al. (1995), we use contraction mapping to estimate $\vec{\delta}$ given $\sigma_{p,hs}$ from the observed market shares $S_{.t}^{hs}$.

$$\delta_{.t}^{h+1} = \delta_{.t}^{h} + \lambda^{hs}(s_{.t}^{hs}(\delta_{.t}^{h}, \phi^{hs}, \sigma^{hs}) - S_{.t}^{hs})$$

Adding tuning parameters λ^{hs} allows for faster convergence of contraction mapping. We carry out the above iterations until $|\delta_{.t}^{h+1} - \delta_{.t}^{h}|$ are less than the error tolerance threshold. After the convergence on mean utility, ζ is updated using the values of δ. Then, the incidental parameters are updated using ordinary least square regression in equation 36. The contraction mapping iterations are again run to get updated mean utility. We stop once we get convergence on the incidental parameters (ϕ_{i1} and ϕ_{i2}). With the mean utility in hand, we will use

the expressions in equation below to compute the unobservable product characteristics as a function of model parameters

$$\xi_{jt} = \hat{\delta}_{j,t} - \vec{x}_{jt}\theta^{hs}$$

Our econometric error terms can then simply be constructed by taking the first difference of ξ. Since we know all time invariant characteristics get cancelled out after taking the first difference, we only include the time varying characteristics in $\vec{x}^{hs}_{j,t}$. Finally, the GMM objective function for the handset and app models can be constructed by interacting our econometric error terms with the above-specified instruments for each model, respectively

$$\arg\min_{\theta^{hs}} (\nu^{hs}_{j,t}(\theta^{hs})^T \mathbf{Z}^{hs}_t) \mathbf{W}^{hs} (\nu^{hs}_{j,t}(\theta^{hs})^T \mathbf{Z}^{hs}_t)^T \tag{37}$$

where $\mathbf{W}^{hs}$ is the weighting matrix. We use $(\mathbf{Z}^{hs\prime}\mathbf{Z}^{hs})^{-1}$ as our weighting matrix. The model parameters for handset model can then be estimated by search over parameters that minimize the GMM objective functions. To estimate the model parameters pertinent to time-invariant characteristics we use a second ordinary least square regression with handset level fixed effects as the dependent variable and time-invariant characteristics as the independent variables. Finally, we update the σ^{hs} parameter, and re-estimate our model. We continue iterating until we find the parameter σ^{hs} that minimizes the GMM objective criterion.

2. **App Demand**: In this phase once we have estimated customer heterogeneity parameters for handset parameters, we move to estimate model parameters for apps. Similar to the handset side, we use generalized method of moments for estimating model parameters on the app side.

   We know $\nu^{app}_{g(j),l,t}(\theta) = \eta^{app}_{g(j),l,t}(\theta) - \eta^{app}_{g(j),l,t-1}(\theta)$. Thus, to estimate $\nu^{app}_{g(j),l,t}(\theta)$, we have

$$\nu^{app}_{g(j),l,t}(\theta) = \psi_{g(j),l,t}(\theta) - \psi_{g(j),l,t-1}(\theta) - \boldsymbol{\alpha}^{\boldsymbol{w}}(w_{g(j),l,t} - w_{g(j),l,t-1}) - \alpha^{p,app}(p_{g(j),l,t} - p_{g(j),l,t-1}) \tag{38}$$

where $\psi_{g(j),l,t}(\theta)$ denotes the *mean lifetime utilities* consumers get from downloading or purchasing app l on platform $g(j)$ at time t. Next, we describe how we accommodate for customer heterogeneity in the app's demand estimation. As described earlier, the initial distribution of customers taste parameters is

$$\boldsymbol{\alpha}^{p,app}_i = \boldsymbol{\alpha}^{p,app} + \nu^{app}_i \sigma_{app}$$

Also, we know

$$\psi_{i,g(j),l,t} = \psi_{g(j),l,t} + (\boldsymbol{\alpha}^{p,app}_i - \boldsymbol{\alpha}^{p,app})(p_{g(j),l,t}) \tag{39}$$

Similar to the handset demand, the mean *lifetime utility* of an app l as perceived by consumer of type i, $\psi_{i,l,g(j),t}$ is a sufficient statistic for determining if a consumer i on platform $g(j)$ will purchase/download app in period t or not. The share of consumer type i, who will download app l, in period t is given by

$$\hat{s}_{i,l,g(j),t}(\psi_{i,l,g(j),t}) = \frac{\exp\left(\psi_{i,l,g(j),t}\right)}{\exp\left(\psi_{i,l,g(j),t}\right) + \exp\left(\beta\,\mathbb{E}(W(\psi_{i,l,g(j),t+1}))|\psi_{i,l,g(j),t})\right)} \tag{40}$$

As before, for every app we only observe aggregate shares. Since, we had approximated the customer distribution using a finite distribution over weights w_i and consumer heterogeneity parameter ν_i, we can write the total share of consumers that download an app l in period t by:

$$\begin{aligned}
\hat{s}_{l,g(j),t}(\psi_{i,l,g(j),t}) &= \sum_i w_{it}^{l,g(j)} \hat{s}_{i,l,g(j),t}(\psi_{i,l,g(j),t}) \\
\sum_i w_{i0}^{l,g(j)} &= 1 \\
w_{it+1}^{l,g(j)} &= w_{it}^{l,g(j)}(1 - s_{ig(j)lt} + q_{it})
\end{aligned}$$

where q_{it} denotes the share of new consumers of type i, who gain access to the platform $g(j)$. The value of q_{it} is used from the handset estimation step (or initialization step). The initial weight parameters $w_{i0}^{l,g(j)}$, are all set to be the same across all apps. However, the future weights and thus the distribution of consumer types can evolve independently for each app across the two platforms. To compute the market shares, we also need access to value functions $W(\psi)$. Similar to the handset side, we will first estimate the incidental parameters $\phi_{i1}^{app,g(j)}$ and $\phi_{i2}^{app,g(j)}$ using a first stage regression based on current guess of ψ

$$\psi_{i,t+1} = \phi_{i1}^{app,g(j)} + \phi_{i2}^{app,g(j)} \psi_{i,t} + \epsilon_{t+1} \tag{41}$$

We compute the value functions $W(\psi)$ using the imputed values of incidental parameters. To compute the value function, we use a fine grid approximation over ψ and carry out value iteration to estimate W. We next carry out a contraction mapping similar to impute the values of mean utilities from the observed data

$$\psi_{.t}^{h+1} = \psi_{.t}^{h} + \lambda^{app}(s_{.t}^{app}(\psi_{.t}^{h}, \phi^{app}, \sigma^{hs}, \sigma^{app}) - S_{.t}^{app})$$

We add tuning parameter λ^{app} for faster convergence of contraction mapping. We carry out the above iterations until $|\psi_{.t}^{h+1} - \psi_{.t}^{h}|$ is less than the error tolerance threshold. After

the convergence on mean utility the incidental parameters are updated using a ordinary least square regression on the updated mean utility. The contraction mapping iterations are again run to get updated mean utility. We stop once we get convergence on the incidental parameters of apps. With the mean utility in hand, we will use the expressions in equation to compute the unobservable product characteristics as a function of model parameters

$$\eta_{jt} = \hat{\psi}_{g(j),l,t} - \vec{x}_{g(j),l,t}\theta^{app}$$

Econometeric error term for the apps model can then simply by constructed by taking the first difference of η. Since we know all time invariant characteristics get cancelled out after taking the first difference, we only include the time varying characteristics in $\vec{x}^{app}_{g(j),l,t}$. Finally, the GMM objective function for the model can be constructed by interacting our econometeric error terms with the specified instruments

$$\arg\min_{\theta^{app}} (\nu^{app}_{g(j),l,t}(\theta^{app})^T \mathbf{Z}^{app}_t) \mathbf{W}^{app} (\nu^{app}_{g(j),l,t}(\theta^{app})^T \mathbf{Z}^{app}_t)^T$$

where $\mathbf{W}^{app}$ is the weighting matrix. We use $(\mathbf{Z}^{app'}\mathbf{Z}^{app})^{-1}$ as our weighting matrix. The model parameters is estimated by search over parameters that minimize the GMM objective function. To estimate the model parameters pertinent to time-invariant characteristics we use a second ordinary least square regression. Finally, we update the σ^{app} parameter, and re-estimate our model. We continue iterating until we find the parameter σ^{app} that minimizes the GMM objective criterion.

3. **Handset Demand** After estimating the app demand model, we get an estimate for $\hat{\Gamma}_i$. Next, we carry out estimation of handset demand parameters in a similar fashion to the initialization phase, with a slight modification.

$$\begin{aligned}\nu^{hs}_{j,t}(\theta) = \delta_{j,t}(\theta) - \delta_{j,t-1}(\theta) - \boldsymbol{\alpha}^{\boldsymbol{x}}(x_{j,t} - x_{j,t-1}) - \\ \alpha^{\Gamma}(\hat{\Gamma}_{g(j),t}(\cdot) - \hat{\Gamma}_{g(j),t-1}(\cdot)) - \alpha^{p,hs}(\log p_{j,t} - \log p_{j,t-1})\end{aligned}$$

Also, the lifetime utility for customer i is given by

$$\delta_{i,j,t} = \delta_{j,t} + (\boldsymbol{\alpha}^{p,hs}_i - \boldsymbol{\alpha}^{p,hs})(\log p_{j,t}) + \alpha^{\Gamma}(\hat{\Gamma}_{g(j),t}(\boldsymbol{\alpha}^{p,app}_i) - \hat{\Gamma}_{g(j),t}(\boldsymbol{\alpha}^{p,app})) \tag{42}$$

We repeat steps 1 and 2, until convergence on model parameters. For standard errors, we use the stacked GMM expression and used the standard formula of variance for a GMM estimator (for reference see Newey and McFadden (1994)). To account for the uncertainty in the standard errors of the parameter estimates due to the first-stage estimation of downloads from ranks, we adjust our standard errors using the method described in Newey and McFadden (1994) (see equation 6.12).

A.3. Estimation of Moment Inequalities

We follow a procedure as detailed in Pakes (2012) and Ho and Pakes (2014). Consider the estimator as described in equation (29) i.e.,

$$\hat{\Theta} = \arg\min_{\theta} ||\hat{\Sigma}_{ii}^{-\frac{1}{2}} \min(0, \hat{m}(\theta))|| \tag{43}$$

Since, our moments our independent across the various genres (see Section **??**), we solve the above optimization problem independentally across the various genres and platforms (for notational ease, we have dropped the platform subscript), i.e.,

$$\hat{\Theta}^g = \arg\min_{\theta} ||{}^g\hat{\Sigma}_{ii}^{-\frac{1}{2}} \min(0, \hat{m}^g(\theta))|| \tag{44}$$

The first step entails estimating the variance covariance matrix to normalize the moments. As it is apparent the sample variance covariance matrix (${}^g\hat{\Sigma}$) is a function of Θ^g. Thus, we adopt the standard two step estimation procedure, just like the estimation of weighting matrix in GMM. In the first step, we estimate Θ_0^g using an identity matrix as the normalizing factor. Using the estimated set $\hat{\Theta}_0^g$, we then compute ${}^g\hat{\Sigma}$. To compute ${}^g\hat{\Sigma}$, we use centroid of the identified set $\hat{\Theta}_0^g$. Next, we re-estimate Θ^g, using the estimated ${}^g\hat{\Sigma}$ as the normalizing factor.

For set of exogenous functions $h(\cdot) \in \mathbb{R}_+^2$, we use the following specification:

$$h(\Omega_{p,l,t}) = \begin{cases} \mathbb{I}(t = \mathcal{T}_l - k) * |\mathbb{N}(0,1)|, \\ \mathbb{I}(t < \mathcal{T}_l - k) * |\mathbb{N}(0,1)|, \end{cases} \tag{45}$$

where $\mathcal{T}_l - k$ refers to the time when app developer decides to port.

Next, we describe how we estimate the 95% confidence intervals for our set estimator.

1. We start by defining a grid of potential values of entry costs for each genre and platform. We assume a coarse grid uniformly spaced points such that $\Theta_p \in [-30000, 50000]$. We use uniform spacing of 100 for our grid.
2. For each $\theta_p \in \Theta_p$, take 100 random draws from a normal distribution with mean zero and variance-covariance matrix $\mathbb{V}(\hat{\Sigma}(\theta_p)^{-\frac{1}{2}} \hat{m}^g(\theta_p))$. For each random draw, say $\tilde{m}(\theta_p)$, we compute the euclidean norm of its negative part (i.e., compute $||\tilde{m}(\theta_p)_-||$).
3. Compute the 95^{th} quantile of this empirical distribution and let this critical value of our test be $c(\theta_p)$.
4. Now, from the original sample, we compute the actual value of $||\hat{\Sigma}(\theta_p)^{-\frac{1}{2}} \hat{m}^g(\theta_p)||$.
5. If it is less than $c(\theta_p)$, then we accept the value θ_p in the 95% confidence interval, otherwise we reject it.

B. Static Model of Handset Demand

In this Section, we briefly outline the static model of handset demand we used as comparison with our main model in Section 4 of main text. We model the lifetime utility a consumer i gets by buying a handset j at time t as

$$u_{ijt} = \underbrace{x'_{jt}\beta_i + \xi_{jt} + \alpha_i \log p_{jt}}_{\delta_{ijt}} + \epsilon_{ijt}, \tag{46}$$

where $\vec{x}_{j,t}$ refers to the observable characteristics of the handset at time t; $\boldsymbol{\alpha}_i^x$ denotes consumer i's taste towards these characteristics; $\alpha^{p,hs}$ captures consumer's coefficient for price of handset; Finally we have the term $\xi_{j,t}$, that captures the time specific variations from the unobserved mean lifetime utility of app l, which are unobservable to the econometrician but are valued by the consumers;

As before, to allow for heterogeneous preferences among consumers we let

$$\alpha_i = \alpha + \sigma^{hs}\nu_i$$

$$\nu_i \sim P_\nu(\nu), \tag{47}$$

where P_ν is a standard normal. Σ allows for each component of ν_i to have a different variance. Also let $\theta = (\beta, \alpha, \sigma, \Sigma)$ be a vector containing all the model parameters. Each consumer at time t, chooses to buy/download an app that gives her the maximum lifetime utility. With this assumption the conditional probability that consumer i chooses product j in market t takes the following form:

$$s_{ijt}(\vec{\delta}_t, \theta) = \frac{\exp\left(\delta_{jt} + \log p_{jt}\Sigma\nu_i\right)}{1 + \sum_{k=1}^{k=J} \exp\left(\delta_{kt} + \log p_{kt}\Sigma\nu_i\right)}, \tag{48}$$

such that,

$$\delta_{jt} = x_{jt}\beta + \alpha \log p_{jt} + \xi_{jt}, \tag{49}$$

where N_g refers to the number of apps in genre g. The unconditional choice or aggregate market share of product j at time t is given by:

$$s_{jt}(\delta_t; \theta) = \int s_{ijt}(\cdot)\phi(\nu)d\nu, \tag{50}$$

We numerically approximate the above integral by taking Monte-Carlo draws from P_ν for sufficient number of individuals, such that

$$\hat{s}_{jt}(x_j; \delta, \sigma, \Sigma) \approx \frac{1}{ns}\sum_{i=1}^{ns} \frac{\exp\left(\delta_{jt} + \log p_{jt}\Sigma\nu_i\right)}{1 + \sum_{k=1}^{k=J} \exp\left(\delta_{kt} + \log p_{kt}\Sigma\nu_i\right)}, \tag{51}$$

To complete the demand estimation, we first compute observed market shares

$$s_{jt} = q_{jt}/M_t \tag{52}$$

where q_{jt} denotes downloads of app j in time period t, and M_t denotes the market size. For our estimation, we use a market size of 30M. We then equate the observed market shares s_{jt} to the market share imputed in equation (59) such that

$$s_{jt} = \hat{s}_{jt}(x_{\cdot t}, p_{\cdot t}, \delta_{\cdot t}; \sigma). \tag{53}$$

To estimate our model, we use generalized method of moments, similar to our main model. To construct the moment conditions for GMM, we use the unobserved error term $\xi_{jt} - \xi_{jt-1}$, and interact it with a set of instruments Z_{jt}. We use the same set of instruments as our main as described in Section 5.

C. Validity of Instruments

We used a timing assumption to argue for the exclusion of our instrument for the platform utility. The exclusion of our instrument relied on the fact developers might not be able to instantly respond to demand shocks in specific handset sales and hence both current period and one-period lagged value of platform utility can be used as valid instruments. However, it could be speculated that certain developers are able to better track the sales and are quick to respond to shocks in sales of handsets. Thus, to test the robustness of our results, we carry out a 'difference-in-Sargan'[26] test with just one-period lagged value of the platform utility (in specification IV). We are unable to reject the hypothesis that current value of platform utility is excluded from the instrument set. We report our results in Table 16. We also estimate our model with just two-period lagged value of the platform utility in specification V. Our results for both specifications are very similar to those we find in our main analysis. Further, prior literature has shown that BLP-style instruments might not provide enough variation to precisely identify demand model parameters.

We also note the limitation of the lagged instrument strategy as its validity depends on a structural assumption that cannot be empirically validated. To properly address this concern, we collected additional data on cost shifters that allow us to estimate the model without requiring such structural assumptions. In our revised manuscript, we empirically validate this structural assumption by re-estimating the model using cost-shifters. We obtained information on the country of origin for each app in our dataset and collected the corresponding daily exchange rate data, with the USD set as the base currency, from `https://openexchangerates.org`. For both the iOS App

[26] For more details see Roodman (2007)

Store and Google Play, we compute the following expression for every time period: $\frac{1}{|\mathcal{L}_t^p|}\sum_{i\in\mathcal{L}_t^p} r_{i,t}$, where i denotes an app on platform p (either iOS or Google Play), $\mathcal{L}_t^p$ denotes the set of apps on platform p at time t, and $r_{i,t}$ denotes the exchange rate for the country where i has its headquarters, with USD as the base rate. This expression calculates an average exchange rate over the assortment of all apps on the platform p at time t. We use this as a cost shifter that affects the utility app developers can provide to customers but does not directly affect customer utility.

Furthermore, since many parts of handsets are manufactured in China, we also obtained monthly exchange rate data for the Chinese yuan.

Thus, the instrument set Z_t^{hs} includes the current-period value of the Chinese yuan exchange rate, the current-period value of averaged exchange rates across all apps ($\frac{1}{|\mathcal{L}_t^p|}\sum_{i\in\mathcal{L}_t^p} r_{i,t}$), and current-period values of BLP-style instruments on handset exogenous characteristics – ram, display size, battery, storage, and camera.

Similarly, on the app demand model side, we use exchange rate depending on the headquarters of app instead of lagged prices as an instrumental variable. As demonstrated in Table (17 and 18), we find qualitatively similar results to our our model that used the structural assumption for identification. This, we believe, also somewhat validates this line of literature, indicating that this structural assumption can be used to recover reliable estimates.

D. Independence of Apps

As discussed in Section 4, one limitation of our model of demand for apps is – we do not account for substitution across apps. However, as we earlier mentioned extant literature has found that in many products markets, when products are highly differentiated, comprising substitution for dynamics can lead to much better predictive performance. Since, static model allow for the same user base making choice across the entire app portfolio, they inadvertently attribute much higher utility to new apps. To validate this, we estimate a static model of demand in a spirit similar to Berry et al. (1995) and Ghose et al. (2012).

To compare the out-of-sample prediction performance of both the models, we first randomly sample 1500 apps from both Apple App Store and Google Play. We make sure all apps existed in the first two quarters of 2013 (i.e., January 2013 – June 2013), and also in the last two quarters of 2014 (i.e., July 2014 – December 2014). Next, we estimate both a static model and a dynamic model on the first half of the data i.e., only using the data from the first two quarters of 2013. Next, using these estimates we make predictions on the latter half of the data i.e., last two quarters of 2014. In Table 19, we report the mean squared error of both the methods on the second half of the data. We find, that our dynamic model of demand provides much lower MSE, and hence, better predictions than a static model.

Table 16. Estimated Parameters (Handsets)

	Model IV	Model V
log(prices)	-2.322**	-2.546***
	(1.034)	(0.719)
ram	0.1457	0.173
	(0.115)	(0.117)
display-size	0.451***	0.415***
	(0.129)	(0.131)
battery	0.001***	0.001***
	(0.000)	(0.000)
log(storage)	0.489***	0.572***
	(0.047)	(0.047)
camera	0.214***	0.219***
	(0.015)	(0.016)
Γ	0.292**	0.296***
	(0.141)	(0.030)
Handset FE	✓	✓
$\sigma^{p,hs}$	0.008***	0.009***
	(0.002)	(0.002)
N	895	895
GMM Obj	1.17e-4	1.15e-4

Standard errors in parentheses.
Note: *p<0.1; **p<0.05; ***p<0.01

The parameters for time-invariant characteristics are estimating by projecting the estimated residuals onto the handset time-invariant characteristics using ordinary least square regression.

For completeness, we briefly outline the static model of demand for apps. We advise the readers to refer to Berry et al. (1995) for further details. We model the lifetime utility a consumer i gets by downloading an app j at time t from platform $g(j)$ as

$$u_{ijt} = \underbrace{x'_{jt}\beta_i + \eta_{jt} + \alpha_i p_{jt}}_{\psi_{ijt}} + \epsilon_{ijt}, \tag{54}$$

where $\vec{x}_{j,t}$ refers to the observable characteristics of the app at time t; $\boldsymbol{\alpha}_i^x$ denotes consumer i's taste towards these characteristics; $\alpha^{p,app}$ captures consumer's coefficient for price of app; Finally we have the term $\eta_{j,t}$, that captures the time specific variations from the unobserved mean lifetime utility of app l, which are unobservable to the econometrician but are valued by the consumers;

As before, to allow for heterogeneous preferences among consumers we let

$$\alpha_i = \alpha + \sigma^{hs}\nu_i$$

Table 17. Estimated Parameters (Handsets) Cost-Based Instruments

	Model VII
$log(prices)$	-2.390***
	(0.778)
ram	-0.208
	(0.128)
$display\text{-}size$	0.488***
	(0.143)
$battery$	0.0012***
	(0.000)
$log(storage)$	0.731***
	(0.052)
$camera$	0.238***
	(0.017)
Γ	0.208*
	(0.107)
$Handset\ FE$	✓
$\sigma^{p,hs}$	0.0085***
	(0.003)
N	895
GMM Obj	1.12e-4

Standard errors in parentheses.
Note: *p<0.1; **p<0.05; ***p<0.01

The parameters for time-invariant characteristics are estimating by projecting the estimated model residuals onto the handset time-invariant characteristics using ordinary least square regression.

$\nu_i \sim P_\nu(\nu), (55)$where P_ν is a standard normal. Σ allows for each component of ν_i to have a different variance. Also let $\theta = (\beta, \alpha, \sigma, \Sigma)$ be a vector containing all the model parameters. Each consumer at time t, chooses to buy/download an app that gives her the maximum lifetime utility. With this assumption the conditional probability that consumer i chooses product j in market t takes the following form:

$$s_{ijt}(\vec{\delta}_t, \theta) = \frac{\exp\left(\delta_{jt} + \log p_{jt}\Sigma\nu_i\right)}{1 + \sum_{k=1}^{k=J} \exp\left(\delta_{kt} + \log p_{kt}\Sigma\nu_i\right)}, \tag{56}$$

such that,

$$\delta_{jt} = x_{jt}\beta + \alpha p_{jt} + \xi_{jt}, \tag{57}$$

where N_g refers to the number of apps in genre g. The unconditional choice or aggregate market share of product j at time t is given by:

$$s_{jt}(\delta_t; \theta) = \int s_{ijt}(\cdot)\phi(\nu)d\nu, \tag{58}$$

Table 18. Estimated Parameters (Apps) (Cost-Based Instruments)

(a)

	(VI)
prices	-0.541***
	(0.076)
version age	-0.0079***
	(0.0015)
version #	-0.083***
	(0.019)
ratings	6.356***
	(0.822)
age	0.0025***
	(0.0004)
App FE	✓
N	1,801,438
GMM Obj	6.52e-11

Standard errors in parentheses.
* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

(b)

	App Fixed Effects
	(VI)
in-app-purchases	-0.648***
	(0.004)
paid	0.251***
	(0.004)
$log(firmsize)$	0.273***
	(0.001)
Entertainment	0.554***
	(0.015)
Games	-1.312***
	(0.014)
Lifestyle	-0.388***
	(0.016)
Music	0.417***
	(0.016)
Others	-0.148***
	(0.014)
Photo and Video	-0.115***
	(0.015)
Social Networking	0.413***
	(0.016)
Utilities	-0.051***
	(0.015)
ios	-0.030***
	(0.004)
Intercept	-18.054***
	(0.015)
Observations	1801438.0
R2	0.178
Note:	*p<0.1; **p<0.05; ***p<0.01

The estimation is carried out on our entire sample of 13,731 iOS and 6,950 Google Play apps. The parameters for time-invariant characteristics are estimated by projecting the estimated residual $\eta_{g(j),l,t}$ onto the app's time-invariant characteristics using ordinary least squares regression. The residual term $\eta_{g(j),l,t}$ is the sum of a time-invariant and time-varying component ($\eta_{g(j),l,t} = \bar{\eta}_{g(j),l} + \Delta\eta^{app}_{g(j),l,t}$). The feature *prices* indicates the price of the paid app and is set to 0 for free apps; *version age* indicates the days since the last version was released; *version* indicates the current iteration of the app; *ratings* is set to 1 if the average current period's rating is above 3, and 0 otherwise; *age* indicates the days since the release of the app; *in-app-purchases* is an indicator variable indicating whether there are in-app purchases, and *paid* indicates whether the app is paid.

Table 19. Out-of-Sample MSE for Static vs Dynamic Model of Apps

	iOS	Google-Play
Static	1066842.3	5636.1
Dynamic	714.5	1369.0

Note. The Table compares the out-of-sample prediction performance of a static model of app demand, that accounts for substitution across apps and a dynamic model of demand that compromises substitution to account for consumer dynamics. We first randomly sample 1500 apps from both Apple App Store and Google Play. We make sure all apps existed in the first two quarters of 2013 (i.e., January 2013 – June 2013), and also in the last two quarters of 2014 (i.e., July 2014 – December 2014). Next, we estimate both a static model and a dynamic model on the first half of the data i.e., only using the data from the first two quarters of 2013. Next, using these estimates we make predictions on the latter half of the data i.e., last two quarters of 2014.

We numerically approximate the above integral by taking Monte-Carlo draws from P_ν for sufficient number of individuals, such that

$$\hat{s}_{jt}(x_j;\delta,\sigma,\Sigma) \approx \frac{1}{ns}\sum_{i=1}^{ns} \frac{\exp\left(\delta_{jt}+\log p_{jt}\Sigma\nu_i\right)}{1+\sum_{k=1}^{k=J}\exp\left(\delta_{kt}+\log p_{kt}\Sigma\nu_i\right)}, \tag{59}$$

To complete the demand estimation, we first compute observed market shares

$$s_{jt} = q_{jt}/M_t \tag{60}$$

where q_{jt} denotes downloads of app j in time period t, and M_t denotes the market size. For our estimation, we use a market size of 30M. We then equate the observed market shares s_{jt} to the market share imputed in equation (59) such that

$$s_{jt} = \hat{s}_{jt}(x_{\cdot t}, p_{\cdot t}, \delta_{\cdot t};\sigma). \tag{61}$$

To estimate our model, we use generalized method of moments, similar to our main model. To construct the moment conditions for GMM, we use the unobserved error term $\eta_{jt} - \eta_{jt-1}$, and interact it with a set of instruments Z_{jt}. We use the same set of instruments as our main as described in Section 5.

E. Estimating App Downloads

We build on methods introduced by Garg and Telang (2013), to estimate app sales from app download rank data. Our data source contains daily download ranks across -63 and -56 app categories for iOS and Google Play respectively. However, we have multiple missing entries in the overall sales rank. This is mostly due to products having very low downloads and thus very high

Table 20. Handset Characteristics Description.

Data	**Description**
Display Size	The screen size varies from model to model and is the diagonal length of screen measured in inches.
Primary Camera	Smartphones have built in digital cameras, and we measure it through their resolution in Megapixels
RAM	RAM is the random access memory of the handsets and is measured in MB.
Storage	This is physical storage available in the phones, measured in GB.
Battery	Battery is measured in milliamp hours mAh.

overall download rank. To address the missing download rank issue, we use rank in other categories to impute the overall sales rank.

$$D_{c_1} = D_{c_2} \tag{62}$$

To estimate missing ranks in overall rank for apps, we use subcategory rankings and project them onto the overall rank. This procedure leaves us still with 3% and 4% missing ranks in iOS and Google Play respectively. We use linear interpolation to impute the missing rank data. Next, we use the pareto distribution to estimate app downloads as a function of download rank. To calibrate the parameters for the pareto distribution we source download data for 300 iOS and 300 Google Play apps from a market research firm that tracks app downloads in the united states. For a detailed discussion, we advise readers to refer to Garg and Telang (2013). To account for the uncertainty in the standard errors of the parameter estimates due to the first-stage estimation of downloads from ranks, we adjust our standard errors using the method described in Newey and McFadden (1994) (see equation 6.12).

F. Test of Model Misspecification

Ho and Pakes (2014) describe a procedure to test if the moment inequalities are correctly specified. We follow a similar procedure as theirs. Given, we estimate non-empty confidence interval sets we can not reject the null that the model is misspecified.

G. Robustness To Inclusive Value Sufficiency

Recent literature Derdenger and Kumar (2019), has pointed how the inclusive value sufficiency assumption could lead to significant bias if consumers instead track some multidimensional state variable instead of just the inclusive value. Given the high dimensionality of the consumer state space, one has to resort certain restrictions to make the problem numerically feasible to estimate. However, to test the robusntess of our results with respect the IVS assumption we estimate an alternate model of demand where consumers track one additional market variable – average value of handset storage (log) across all handsets and average value of app ratings for app demand model

Table 21. App Genre Descriptives

	iOS	Google Play
Games	42%	63%
Entertainment	7%	4%
Education	4%	2%
Lifestyle	5%	2%
Music	4%	3%
Photo and Video	8%	3%
Social Networking	4%	3%
Utilities	6%	7%
Others	19%	13%

Note. The genre statistics are calculated across our entire sample of 13,731 iOS and 6,950 Google Play apps, respectively.

Table 22. Smartphones with Highest Observed Market Share

		(Market Share)
Apple	Apple iPhone 5S	0.13
	Apple iPhone 5C	0.08
	Apple iPhone 5	0.06
	Apple iPhone 6	0.05
Android	Samsung Galaxy S4	0.10
	Samsung Galaxy S5	0.05
	Samsung Galaxy Note 3	0.04
	Samsung Galaxy S3	0.04

Note. The table reports the top smartphones with highest observed market shares for the period of our study.

respectively along with the respective inclusive values. Specifically, for the handset demand model we assume the following transition process of state variables –

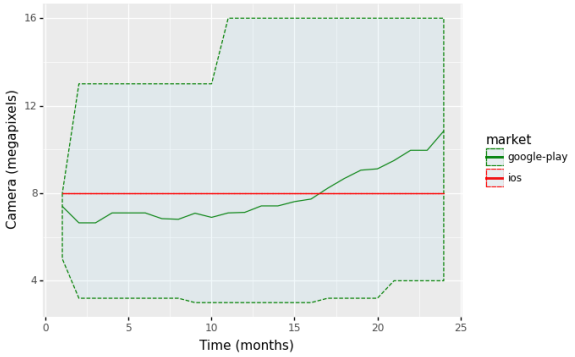
$$G_i(\Omega_{t+1}|\Omega_t) = G_i(\zeta_{i,t+1}(\Omega_{t+1}), \bar{X}_{i,t+1}|\zeta_{i,t}(\Omega_t), \bar{X}_{i,t}) \tag{63}$$

This assumptions means that the consumer does not need to know, or keep track, of Ω_t. Nor does the consumer need to predict the future Ω_{t+1}. Instead the consumer just need to keep track of the lower dimension $\zeta(\Omega)$ and $\bar{X}_{i,t}$ (average value of storage (log) across all handsets).

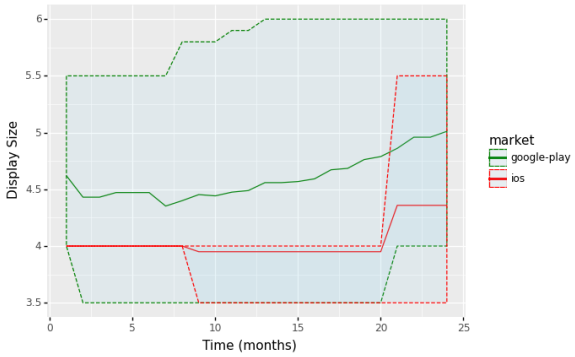
Similarly, for the app demand model we assume a similar change the app state transition process –

$$F_{i,g(j)}(\Omega_{t+1}|\Omega_t)) = F_{i,g(j)}(\psi(\Omega_{t+1}), \bar{X}_t|\psi(\Omega_t), \bar{X}_t), \tag{64}$$

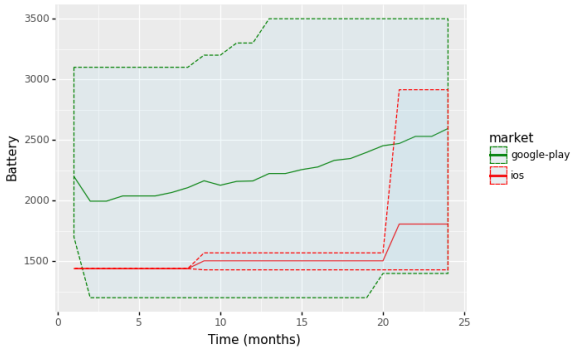
Figure 23. Handset Characteristics Evolution (Nonstandardized)



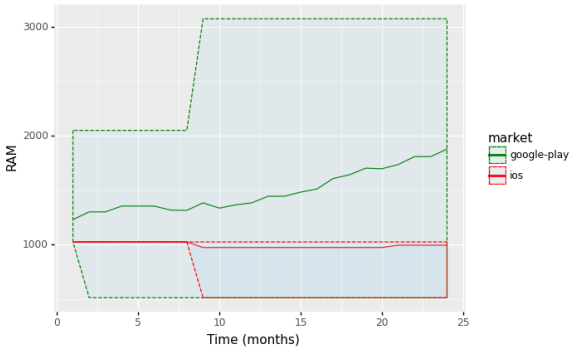
(a) Camera (Megapixels)



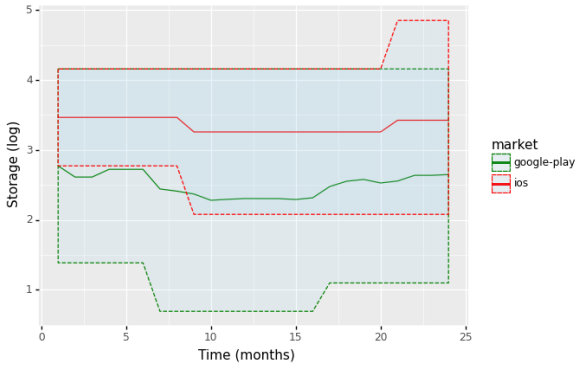
(b) Display Size



(c) Battery (mAh)



(d) RAM



(e) Storage (log(GB))

Note. The figures report how the handset features for the entire mix of handsets in the market evolve over time. The solid line denotes the average value of handset characteristics over time and the dashed line indicate the minimum and maximum value of handset characteristics over time. Time period 1 denotes January 2013 and Time period 24 denotes December 2014.

Table 24. Estimated Parameters (Handsets) (Robustness to Inclusive Value Sufficiency)

	Model VI
log(prices)	-2.223***
	(0.580)
ram	0.016
	(0.113)
display-size	0.447***
	(0.126)
battery	0.001***
	(0.000)
log(storage)	0.5659***
	(0.046)
camera	0.209***
	(0.015)
Γ	0.245***
	(0.027)
Handset FE	✓
N	895
GMM Obj	7.48e-5

Standard errors in parentheses.
Note: *p<0.1; **p<0.05; ***p<0.01

The parameters for time-invariant characteristics are estimating by projecting the estimated residuals onto the handset time-invariant characteristics using ordinary least square regression.

As before assume an AR(1) specification for $\mathbb{E}[\zeta_{i,t+1}, \bar{X}_{i,t+1}|\zeta_{i,t}, \bar{X}_{i,t}]$ as follows

$$\begin{bmatrix} \zeta_{i,t+1} \\ \bar{X}_{i,t+1} \end{bmatrix} = \Phi_{i1} + \Phi_{i2} \begin{bmatrix} \zeta_{i,t} \\ \bar{X}_{i,t} \end{bmatrix} + \mu_{it+1} \tag{65}$$

where, Φ_{i1} and Φ_{i2} are parameters, and μ_{t+1} is identically and independently distributed with mean $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Similarly, for $\mathbb{E}[\psi_{i,t+1}, \bar{X}_{i,t+1}|\psi_{i,t}, \bar{X}_{i,t}]$ we assume an AR(1) specification as follows:

$$\begin{bmatrix} \psi_{i,t+1} \\ \bar{X}_{i,t+1} \end{bmatrix} = \Phi_{i1}^{app,g(j)} + \Phi_{i2}^{app,g(j)} \begin{bmatrix} \psi_{i,t} \\ \bar{X}_{i,t} \end{bmatrix} + \epsilon_{t+1} \tag{66}$$

where, ϵ_{t+1} is identically and independently distributed with mean $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Φ_{i1}^{app} and Φ_{i2}^{app} are the incidental parameters. In Table 24 and 25, we report the results of our estimation and find very similar estimates as before. This gives us some confidence that the underlying true data-generating process might not be driven by a high-dimensional state vector and could be approximated just by using inclusive value in the state space.

Table 25. Estimated Parameters (Apps) (Robustness to Inclusive Value Sufficiency)

(a)

	(I)
price	-0.649***
	(0.092)
version age	-0.0063***
	(0.0020)
version #	-0.086***
	(0.019)
ratings	6.555***
	(0.977)
age	0.002***
	(0.000)
App FE	✓
N	1,801,438
GMM Obj	2.14e-12

Standard errors in parentheses.
* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

(b)

	App Fixed Effects
	(1)
in-app-purchases	-0.387***
	(0.004)
paid	-0.640***
	(0.004)
$log(firmsize)$	0.191***
	(0.001)
Entertainment	0.769***
	(0.015)
Games	-1.491***
	(0.014)
Lifestyle	0.321***
	(0.016)
Music	0.554***
	(0.016)
Others	0.185***
	(0.015)
Photo and Video	0.189***
	(0.015)
Social Networking	1.459***
	(0.015)
Utilities	0.209***
	(0.015)
ios	-0.667***
	(0.004)
Intercept	-18.335***
	(0.016)
Observations	1801438.0
R2	0.178
Note:	*p<0.1; **p<0.05; ***p<0.01

The estimation is carried out on our entire sample of 13,731 iOS and 6,950 Google Play apps. The parameters for time-invariant characteristics are estimated by projecting the estimated residual $\eta_{g(j),l,t}$ onto the app's time-invariant characteristics using ordinary least squares regression. The residual term $\eta_{g(j),l,t}$ is the sum of a time-invariant and time-varying component ($\eta_{g(j),l,t} = \bar{\eta}_{g(j),l} + \Delta\eta^{app}_{g(j),l,t}$). The feature *prices* indicates the price of the paid app and is set to 0 for free apps; *version age* indicates the days since the last version was released; *version* indicates the current iteration of the app; *ratings* is set to 1 if the average current period's rating is above 3, and 0 otherwise; *age* indicates the days since the release of the app; *in-app-purchases* is an indicator variable indicating whether there are in-app purchases, and *paid* indicates whether the app is paid.

H. Computing App Revenue

To compute the app revenue numbers, we first source another set of daily app data from a market research firm (`https://appfigures.com/`) on randomly sampled 40 apps – 20 iOS and 20 Google Play. The market research firm along with other app characteristics also tracks downloads and revenue made by each app on a daily basis. Ideally, app developers do not share revenue numbers publicly. The market research firm requires developers signing up on their platform to share the revenue numbers of their apps. In return, the market research firm gives developers access to revenue numbers of their competitors. This allows them to build a comprehensive dataset of app revenue numbers. Note, that the revenue numbers do not include revenue apps make from advertisements but only the revenue from in-app purchases or upfront payments if the app was paid. For our analysis, we consider three kinds of apps – (i) free with in-app purchases, (ii) paid without in-app purchases, and (iii) paid with in-app purchases. We do not consider free apps without in-app purchases because Apple or Google Play does not get any revenue from these apps. For each of these apps, we estimate a linear regression of observed revenue on current period downloads and cumulative lifetime downloads. For the paid apps, we first subtract the revenue through upfront payments. Note that this only provides us with the revenue numbers of apps currently on the platform. To obtain revenue numbers for the entire app store, we use the following approach to calibrate our model. We set Apple iOS revenue in 2013 worldwide at \$10.6B and Android revenue at \$4.4B, as suggested by Research (2023). The Apple numbers are similar to the official figures reported by Apple (see link). To calculate the United States' contribution to Apple iOS and Google Play revenue in 2013, we employ the following strategy: for each world region, we determine the share of Apple and Android smartphones (see link). We then weigh these shares by the GDP of the respective region to determine the percentage of Apple and Google Play revenue that comes from the United States. Our method implicitly assumes that the share of app revenues from the US compared to the rest of the world is proportional to the share of handset revenues. In the end, we compute \$6.14B revenue for Apple iOS and \$1.32B for Google Play from the United States.

I. Alternate Model of Developers' Entry Decisions

We present an alternate parameterization for the model of developers' entry decisions on the two platforms, where we decompose the app development cost into a base app development cost and a platform-specific app development cost. We assume that an app not present on a platform would commit k months in advance to enter the app stores. If the app was not present on any platform, it would incur a base app development cost θ^{base} and a platform-specific cost $\theta^{platform}_{apple}$ and/or $\theta^{platform}_{android}$, depending on the platform(s) it chooses to enter. The entry decision of the developer will depend on the trade-off between these costs and the expected profit from entry:

$$\Pi_{l,t}(d_t \equiv \{\} \rightarrow \{p\}; \theta) = \mathbb{E}\Big[\sum_{\tau \geq t+k} (\beta^{\tau-t} D^p_{l,\tau}) | \Omega_{l,t}\Big] - \theta^{platform}_{l,p} - \theta^{base}_{l}, \tag{67}$$

where $D^p_{l,\tau}$ denotes the download of app l in time period τ on platform p; $\Omega_{l,t}$ is the app developer's information set at time t; θ denotes the model parameters; and $d_t \equiv \{\} \rightarrow p$ denotes the decision at time t to launch on just platform p.

Next, the profit function for the app developer who chooses to do a simultaneous launch on both platforms at once is given by:

$$\Pi_{l,t}(d_t \equiv \{\} \rightarrow \{p, p'\}; \theta) = \mathbb{E}\Big[\sum_{\tau > t+k} (\beta^{\tau-t} [D^p_{l,\tau} + D^{p'}_{l,\tau}]) | \Omega_{l,t}\Big] - \theta^{platform}_{l,p} - \theta^{platform}_{l,p'} - \theta^{base}_{l}, \tag{68}$$

Now let's consider the expected profit for an app l operational on platform p that elects to enter platform p'. Once the app is deployed on the other platform, it gets another stream of downloads $\{D^{p'}_{t+k+1}, D^{p'}_{t+k+2}, \dots\}$ from platform p', where k refers to the time required to enter the other platform. Then, the expected normalized profits from app l, if developer chooses to enter (i.e., $d_t \equiv \{p\} \rightarrow \{p, p'\}$) platform p' at time t is given by

$$\begin{aligned} \Pi_{l,t}(d_t \equiv \{p\} \rightarrow \{p, p'\}; \theta) = \mathbb{E}\Big[\sum_{t < \tau \leq t+k} (\beta^{\tau-t} D^p_{l,\tau}) | \Omega_{l,t}\Big] + \\ \mathbb{E}\Big[\sum_{\tau > t+k} (\beta^{\tau-t} (D^{p'}_{l,\tau} + D^p_{l,\tau})) | \Omega_{l,t}\}\Big] - \theta^{platform}_{l,p'} \end{aligned} \tag{69}$$

Note that the stream of downloads from platform p' starts from period $t+k+1$. We can now use the observed empirical patterns of developer entry to infer the economic cost of entry of an app to a platform (Apple or Android). Specifically, by observing when a developer chooses to enter a platform, we can construct estimates of entry costs.

If we observe the initial entry of an app (denoted by decision d) at time $\mathcal{T}$, this implies that the developer would have found it profitable to do so $k+1$ periods before over not entering (denoted by d^c). That is

$$\Pi_{l,\mathcal{T}-k-1}(d_{\mathcal{T}-k-1}; \theta) - \Pi_{l,\mathcal{T}-k-1}(d^c_{\mathcal{T}-k-1}; \theta) \geq 0 \tag{70}$$

Similarly, for all time periods $\tau < \mathcal{T} - k - 1$, developer would have not found action d profitable

$$\Pi_{l,\tau}(d_\tau; \theta) - \Pi_{l,\tau}(d^c_\tau; \theta) \leq 0 \tag{71}$$

We estimate our model using the same estimation method as before. We report the results in Table 26. Overall, after comparison with the original results, the development costs appear to be of a similar order.

Table 26. Estimated App Development Costs

Genre	Port to Apple Cost	Base Development Cost	Port to Android Cost
Entertainment	270.54 [10.79, 3688.79]	4414.58 [897.71, 4744.04]	14272.08 [13991.14, 16206.64]
Games	(5082.32, 7446.04) [1519.39, 10756.09]	(5575.26, 8040.29) [2229.74, 11632.65]	(11546.23, 13795.30) [8583.10, 16266.08]
Music	171.48 [0.49, 1226.61]	9124.49 [7807.68, 9403.70]	10458.75 [9906.85, 11594.16]
Photo and Video	2315.22 [56.24, 7720.13]	5624.07 [232.94, 7962.11]	22715.81 [20607.21, 27184.85]
Social Networking	5620.00 [2988.75, 8314.61]	19615.56 [17165.72, 22497.08]	6948.15 [5700.34, 9038.13]
Others	718.20 [27.59, 1902.13]	2849.74 [1698.56, 3709.97]	14767.62 [14009.28, 15624.02]

Note. We report in round brackets, the identified set of costs to develop an app for Android or Apple across the various genres. The estimated costs are normalized to the number of downloads. The 95% confidence intervals are reported in square brackets. Confidence intervals are computed using a bootstrap procedure. To compute the confidence intervals, 50 bootstrap samples were constructed by randomly sampling apps with replacement, and then the parameter values were estimated on the bootstrapped samples. In brackets, we report the 5th and 95th percentile values of the estimator across the bootstrap samples.

J. App Lifetime Revenue

PROPOSITION 1. *Under the following assumptions about app revenue streams:*

1. *One-time purchase revenue:* $R_{upfront} = D \cdot p^{(upfront)}$
2. *Subscription revenue:* $R_{subscription} = IB \cdot p^{(subscription)}$
3. *Ad-supported revenue:* $R_{ad} = IB \cdot a$
4. *In-app purchase revenue:* $R_{iap} = IB \cdot iap$

where:

- D *is the number of new downloads/purchases in the current period*
- IB *is the cumulative downloads/purchases (installed base) in the current period*
- $p^{(upfront)}$ *is the observed one-time purchase price*
- $p^{(subscription)}$ *is the subscription fee per period*
- a *is the estimated ad revenue per installed app per period*
- iap *is the average in-app purchase revenue per user per period*

The expected lifetime revenue for an app can be expressed as a discounted sum of future downloads multiplied by a constant factor:

$$R_{\infty} = \eta \sum_{t=1}^{\infty} \beta^{t-1} D_t \tag{72}$$

where $\eta = p^{(upfront)} + \frac{\beta}{1-\beta}(p^{(subscription)} + a + iap)$ *and* β *is the discount factor.*

The lifetime revenue the app developer will accrue can be written as follows –

$$R_\infty = \sum_{\tau=1}^{\infty} \beta^{\tau-1} r_\tau \tag{73}$$

$$\begin{aligned} R_\infty = {} & D_1 p_{\text{upfront}} \\ & + \beta \left[D_2 p_{\text{upfront}} + D_1 p_{\text{subscription}} + D_1 a + D_1\, iap \right] \\ & + \beta^2 \left[D_3 p_{\text{upfront}} + D_1 p_{\text{subscription}} + D_2 p_{\text{subscription}} \right] \\ & + \beta^2 \left[D_1 a + D_2 a \right] + \beta^2 \left[D_1\, iap + D_2\, iap \right] \\ & + \cdots \end{aligned} \tag{74}$$

$$\begin{aligned} R_\infty = {} & D_1 \left[p_{\text{upfront}} + \beta(p_{\text{subscription}} + a + iap) + \beta^2(p_{\text{subscription}} + a + iap) + \cdots \right] \\ & + \beta D_2 \left[p_{\text{upfront}} + \beta(p_{\text{subscription}} + a + iap) + \beta^2(p_{\text{subscription}} + a + iap) + \cdots \right] \\ & + \beta^2 D_3 \left[p_{\text{upfront}} + \beta(p_{\text{subscription}} + a + iap) + \beta^2(p_{\text{subscription}} + a + iap) + \cdots \right] \\ & + \cdots \end{aligned} \tag{75}$$

$$R_\infty = \eta \sum_{n=0}^{\infty} \beta^n D_{n+1}, \tag{76}$$

where $\eta = [p_{\text{upfront}} + \beta(p_{\text{subscription}} + a + iap) + \beta^2(p_{\text{subscription}} + a + iap) + \cdots]$.

K. Estimation Without Accounting for Endogeneity

We also estimate our model without accounting for endogeneity, treating all explanatory variables as exogenous. The estimates for both the handset and app models are reported in Tables 27 and 28, respectively. Across both models, we find that the price coefficient is positively biased compared to the model using instrumental variables.

L. Brand Loyalty Effects

To understand the extent of bias due to not incorporating brand loyalty effects in our model, we conduct the following simulation analysis. We will simulate demand data for a set of consumers and will estimate demand models that account for brand loyalty and demand models that do not account for brand loyalty.

We assume that a consumer i in every time-period t can choose between two products, where the utility a consumer gets by buying a product j is given as follows:

$$u_{ijt} = \alpha price_{jt} + \gamma \mathbf{I}(C_{it-1} = j) + \epsilon_{it}, \tag{77}$$

where C_{it-1} indicates the choice made by consumer i in the previous period. The probability that a consumer i chooses, say, product A in time period t is given by:

Table 27. Estimated Parameters (Handsets) without Instrumental Variables

	Model
$log(prices)$	-0.6827***
	(0.1604)
ram	-0.1763
	(0.1190)
$display\text{-}size$	0.7360***
	(0.1330)
$battery$	0.0010***
	(0.0001)
$log(storage)$	-0.0554
	(0.0480)
$primary\text{-}camera$	0.1937***
	(0.0160)
Γ	0.2325***
	(0.0251)
$Handset\ FE$	✓
σ	~ 0
N	895
GMM Obj	0.000050

Standard errors in parentheses.
Note: *p<0.1; **p<0.05; ***p<0.01

The parameters for time-invariant characteristics are estimating by projecting the estimated residuals onto the handset time-invariant characteristics using ordinary least square regression.

$$P_{iAt} = \frac{\exp\left(\alpha price_{At} + \gamma \mathbf{I}(C_{it-1} = A)\right)}{\exp\left(\alpha price_{At} + \gamma \mathbf{I}(C_{it-1} = A)\right) + \exp\left(\alpha price_{Bt} + \gamma \mathbf{I}(C_{it-1} = B)\right)}, \tag{78}$$

For our simulation, we assume a market with 1000 consumers at time $t = 1$, where in every time period each consumer chooses to buy either product A or product B. Given estimates of α and γ, along with the observed choices between products A and B, the likelihood function for the model can be written as:

$$\ln L(\alpha, \gamma; \text{data}) = \sum_i \sum_t \Big[y_{it} \log Pr_{iAt} + (1 - y_{it}) \log(1 - Pr_{iBt}) \Big],$$

where $y_{it} = 1$ if consumer i chooses product A, and 0 otherwise. The likelihood sums the log probability of the observed choices across all consumers and time periods, conditional on the model parameters. We estimate the parameters α and γ by choosing the values that maximize the log-likelihood of the observed choices.

We simulate 1000 unique customers who choose between products A and B every period. For simulating product prices, we draw from $N(0.5, 0.5)$. We vary γ from 0 to 1, corresponding to a

Table 28. Estimated Parameters (Apps) without Instrumental Variables

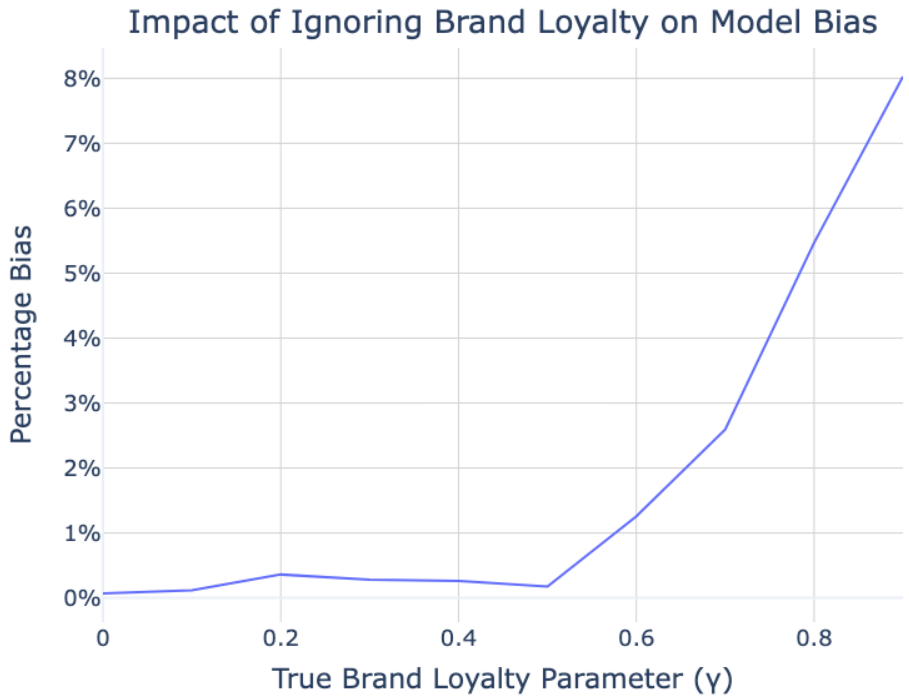
	Model
age	0.0005***
	(0.000)
version	0.013***
	(0.000)
version age	-0.000***
	(0.000)
R_t	-0.029***
	(0.010)
prices	-0.239***
	(0.009)
App FE	✓
N	1,801,438
GMM Obj	2.07e-38

Standard errors in parentheses.
* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

mean dollar value of brand loyalty ranging from 0% to 100%. Intuitively, this number indicates the percentage price discount required to offset the value from brand loyalty. For instance, in our setup, the γ value of 0.5 would mean that, on average, a price discount of 50% is required to offset the brand loyalty effect. We conduct our simulation across 100 simulated data samples.

In Figure 29, we report the percentage additional bias in the estimation of parameter α, when we do not account for consumer inertia or brand loyalty effects. We first estimate α from a model accounting for the brand loyalty effect in the utility function. We then compare this estimate to one from a misspecified model without brand loyalty but with brand fixed effects. We calculate the percentage increase in bias relative to the true value of α due to omitting brand loyalty. We find that without extreme brand loyalty levels, the additional bias incurred by omitting loyalty is limited.

Figure 29. Impact of Ignoring Brand Loyalty on Model Bias



The figure reports the average percentage bias in α across 100 simulated data samples. We first estimate α from a model accounting for the brand loyalty effect in the utility function. We then compare this estimate to one from a misspecified model without brand loyalty but with brand fixed effects. We calculate the percentage increase in bias relative to the true value of α due to omitting brand loyalty.