Firewall for Innovation

Jie Zhou*

December 12, 2024

Click here for an updated version.

Abstract

Do protectionist policies foster domestic growth and innovation in the digital economy, and if so, how? This paper investigates the impact of the Great Firewall (GFW) in China – the world's largest system of internet regulation – on the development of domestic mobile apps. By blocking foreign apps at times determined mostly by political considerations, the GFW prompted a 30% user base expansion for Chinese substitute apps (identified through their baseline text descriptions). Monthly data on these apps' underlying technologies, extracted from their compiled source code, reveal that Chinese substitute apps accelerated their innovation efforts, with in-house development increasing by 14% two years after the blockage. This technological progress spilled over broadly post-blockage, as both domestic and foreign apps adopted more Chinese technologies. I further show that increased access to data was one important driver. Chinese apps requested more types of sensitive data and were more likely to share user data access with outside firms after their foreign substitutes were blocked. These increased types of user data generate innovation; quasi-random variation in the introduction of new data access raises in-house technology development. Finally, using data-sharing networks between app developers, I show that in-house development also increased at the firms that user data was shared with. In summary, protectionist policies brought about through China's GFW boosted its app industry, potentially contributing to China's leadership role in this fast-growing industry.

Keywords: industrial policy, innovation, digital economy
JEL Classification: F13, L86, O33, O38

*I am deeply grateful to Daron Acemoglu, Ben Olken, and David Atkin for their support throughout this project. I also thank Martin Beraja, Tishara Garg, Quan Le, Dean Li, Tobias Salz, Advik Shreekumar, Yulu Tang, and Jean Tirole for many helpful suggestions. My appreciation goes to Meng Xiao, Jingyu Xie, Yuting Qiu, and Daxi Chen for their technical expertise and industry insights. I acknowledge generous support from Jerry Hausman Graduate Dissertation Fellowship, George and Obie Shultz Fund, ACES Fellowship. All errors are my own. Zhou: Massachusetts Institute of Technology; email: `jiezhou@mit.edu`.

1 Introduction

Technological protectionism has become an increasingly prevalent trend in the global digital economy, driven by an urgent desire to foster domestic innovation.[1] One of the theoretical rationales underlying this shift towards industrial policy is the need to address market failures arising from static and dynamic externalities, which often necessitate policy intervention to boost domestic technologies and the competitiveness of domestic firms (Hirschman, 1958; Baldwin, 1969; Krueger, 1990; Young, 1991; Hausmann and Rodrik, 2003; Melitz, 2005; Stiglitz, Lin and Patel, 2013; Juhász, Lane and Rodrik, 2023).

The nonrivalrous nature of digital products and the evolution of learning methods have further intensified the dynamics shaping industrial policy. Digital inputs like data and intermediaries such as software libraries[2] can be utilized simultaneously by multiple entities without depletion (Goldfarb and Tucker, 2019; Jones and Tonetti, 2020). This characteristic fosters external economies through shared library adoption and data exchange, which can amplify collective technological progress. Additionally, digital tools such as A/B testing[3] and machine learning enable data-driven decisions, broadening learning opportunities beyond traditional on-the-job methods (Farboodi, Mihet, Philippon and Veldkamp, 2019; Bajari, Chernozhukov, Hortaçsu and Suzuki, 2019) and opening new pathways for rapid knowledge acquisition.

This paper presents a comprehensive empirical analysis focusing on two major aspects of protectionist policies in the digital economy. First, it directly examines whether such policies, by generating positive demand shocks, spur innovation of domestic products and cultivate a domestic technology ecosystem. Second, it investigates an important mechanism through which these policies may function: the increase in data collection and sharing among domestic firms following protectionist measures, potentially enhancing productivity through both internal and external economies. This channel is particularly policy-relevant given the rapid rise in data localization policies – a central form of digital

[1]This is exemplified by policies like the United States' CHIPS and Science Act of 2022, which aims to boost domestic semiconductor production; the European Union's push for digital sovereignty through regulations such as the Digital Markets Act of 2020; and India's "Aatmanirbhar Bharat" (Self-Reliant India) program introduced in 2020 to promote local technology ecosystems.

[2]Software libraries are reusable collections of code that provide pre-built functions or tools, allowing developers to efficiently build applications by integrating existing solutions rather than creating code from scratch.

[3]In internet firms, A/B testing is an experimental approach that compares two versions of a product feature or service to determine which performs better, enabling data-driven decisions to optimize user engagement and outcomes.

protectionism.[4]

I estimate these effects by exploring how the Great Firewall (GFW) in China – the world's largest system of internet regulation – impacts innovation in domestic mobile applications (apps). This is a particularly appropriate setting for studying the effects of protectionist policies on a domestic digital economy for two reasons: the availability of precise technological tracking data and China's rapid app industry growth.

First, mobile apps, the cornerstone products of consumer-facing companies in the digital economy,[5] provide exceptionally granular data that allow for precise tracking of technological evolution. For example, each version of a mobile app includes detailed documentation of the technologies it incorporates, down to the functionality level, along with the corresponding providers within the raw code. This enables precise and comprehensive measurements of technological changes for a product at a monthly frequency, a task that has been a substantial challenge in prior research.

Second, China's digital economy, particularly the mobile apps sector, was in its infancy when the GFW was initiated in the early 2000s, but it has since achieved remarkable growth and competitiveness. Presently, China not only hosts the world's most localized app market, where nearly all of the most-downloaded apps in 2022 were developed domestically; it is also the only country that can rival the U.S. in regional and international markets of mobile apps (World Bank, 2024). Despite these advances, the lessons of China's rapid ascent in the digital realm remain unclear.

To investigate the dynamic effects of the blockage of foreign apps on innovation, I utilize an event-study design that incorporates three types of variation and three novel datasets.

The study leverages the staggered timing of foreign app blockages, which introduces both temporal variation and variation in blockage timing in my analysis. This allows me to compare the innovation *within* an app to control for potential industry-wide and firm-level confounders. Through extensive online research,[6] I compile the blockage status of 114 major foreign apps by the GFW from 2009 to 2023, providing detailed start and end

[4] The number of countries implementing data localization requirements nearly doubled from 35 in 2017 to 62 in 2021, with the total number of such policies (both explicit and *de facto*) increasing from 67 to 144 over the same period (Cory and Dascoli, 2021).

[5] In 2021, Meta reports that over 98% of their daily active users access their services via mobile devices, and over 90% of their advertising revenue comes from mobile platforms. Similarly, In Q4 2021, Netflix disclosed that more than 70% of its streaming in emerging markets occurs on mobile devices, highlighting the critical role of mobile apps in user engagement and market expansion.

[6] The main source is `greatfire.org` which monitoring the status of foreign websites in China since 2011. For foreign apps that we cannot find historical information from the website, I search for news articles and validate with user reviews that complain about inaccessibility of the apps.

dates and reasons for blockage when available. Notably, the primary intent of the GFW has been to control information flow and maintain stability; a significant proportion of these blockages, at least 55%, is triggered by sudden political events.[7] This helps to mitigate concerns about the selection of promising industries through favoritism – a common challenge in assessing the impacts of industrial policies.

I then further define cross-sectional variation by classifying foreign-domestic app pairs into substitutes, complements, and neutrals using large language models. I first classify 20,000 randomly selected foreign-domestic app pairs into these three categories using `gpt-4`. After ensuring labeling consistency of `gpt-4`[8] and alignment with human evaluations,[9] I use these data to train a logistic regression model that I use to classify pairs based on app descriptions before the blockage of the foreign app.

To measure innovation in mobile apps, I compile three novel datasets. Starting with the *Domestic Apps Dataset*, I collect detailed records of over 230,000 apps from 6,000 Chinese internet firms, covering the period from 2014 to 2023. I scrape *(i)* basic app information (including developer, description, reviews, and update logs for each version), *(ii)* data on user activity (e.g., monthly active users and time spent per user), and *(iii)* the complete package for each app version, which is a collection of libraries and data assets necessary for the app's functionality. Unlike manufactured products, where tracking product improvements can be challenging, digital products offer a transparent and detailed view of technological advancement by comparing raw code across versions. I decompile these packages to extract the libraries – collections of code at the functionality level – and user data requested by the apps. This process provides insights into the technologies utilized by these apps and their user privacy practices.

Additionally, I establish the *Foreign Apps Dataset* by following the same procedure as the one employed for domestic apps. This dataset includes the 114 foreign apps that were blocked and approximately 450,000 additional apps, selected by randomly choosing 1% of apps per country per year. Finally, I create the *Library Dataset*, which links libraries extracted from app packages to their respective companies and countries, providing a comprehensive view of technology sourcing and application.

[7] For example, foreign domain `instagram.com` was blocked in China during the 2014 Hong Kong pro-democracy protests. The platform was used by protesters to share images and information, which the Chinese authorities wanted to prevent from spreading.

[8] To test consistency, we randomly selected 20 app pairs and ran the model 10 times for each pair using identical inputs. We then calculated the average proportion of times the model produced the most frequent label across these runs, yielding a consistency rate of 0.94, which demonstrates the overall robustness of the classification.

[9] All substitutes and complements, along with a randomly selected set of neutrals (2,667 app pairs in total), were independently labeled by human annotators, achieving an 83% agreement rate with `gpt-4` labels.

The causal effects of blocking foreign substitutes are identified by comparing the mean differences in outcomes for domestic apps before and after the first blockage of their foreign substitutes to those for apps that have not yet been treated or will never be treated.

A month-to-month comparison of active users reveals that blocking foreign substitutes creates a positive demand shock for domestic apps, increasing their monthly active users by 30% two years after the blockage. Importantly, this increase is not observed prior to the blockage. On the other hand, the blockage has little impact on domestic apps using technologies developed by the blocked foreign apps, suggesting that the blockage primarily affects consumers' access to foreign domains but has minimal effects on Chinese firms' access to foreign technologies.[10] I then demonstrate that blocking foreign substitutes for Chinese apps leads to a significant increase in the quantity and quality of technologies developed by Chinese firms.

First, I find a significant increase in in-house technologies and notable shifts in the composition of libraries within Chinese apps following the blockage of foreign substitutes. To investigate the quantity of innovation, I categorize libraries into three groups based on their developers: in-house, domestic third-party, and foreign third-party. Notably, the number of in-house libraries increases by 14% two years after the blockage.[11] This surge in in-house libraries indicates a significant boost in internal development and innovation capabilities within Chinese firms.

Additionally, the adoption of domestic third-party libraries rises by 14.5%, highlighting increased collaboration among Chinese developers and their practice of leveraging local technologies. In contrast, the number of foreign third-party libraries decreases by 7%, suggesting a reduced reliance on foreign technological inputs. This divergence, driven by blockages of foreign substitutes, marks a significant strategic shift in Chinese app development. It highlights a move toward self-reliance, enhanced local collaboration, and the creation of a more resilient and innovative domestic tech ecosystem.

Second, I assess the quality of Chinese libraries by examining the adoption of Chinese firms' self-innovated libraries in foreign markets. I aggregate the treatment to the firm level by selecting the earliest time when the foreign substitutes of a firm's apps are blocked. I then compare the percentage of apps in each non-Chinese market that

[10]This outcome is due to the nature of the blockage, which operates at the domain level. The restriction of one domain (e.g., `chatgpt.com`) does not necessarily extend to related domains (e.g., `openai.com`). Additionally, even if a library is hosted on a blocked domain, the third-party library is integrated into the app package during the development stage. Users download the complete package directly from the App Store to their local devices rather than from the blocked domain. This integration process ensures that the blockage does not hinder the use of foreign technologies in Chinese apps.

[11]This increase is substantial, representing about half of the growth observed when Google integrated an AI-powered search engine into its app in February 2024.

adopts Chinese firms' libraries within a firm-country cell. The analysis reveals that Chinese firms' libraries are increasingly adopted internationally, especially in Asian markets, after the blockage of their foreign competitors. This trend highlights the competitive quality of Chinese technologies and their growing acceptance in the global market.

For both the quantity and quality of technologies, there is no effect when using the blockage of foreign neutrals (i.e., apps that are classified as neither substitutes nor complements of the domestic app) as a placebo test. This lack of impact suggests that, on average, the influence of confounding factors – such as spillovers from apps whose foreign substitutes are blocked – on the control group is negligible.

Lastly, I investigate whether the technological trajectory of domestic apps is driven by imitation or innovation. While adopting existing technologies (imitation) can stimulate short-term growth, long-term progress hinges on firms transitioning to innovation-driven strategies (Acemoglu, Aghion and Zilibotti, 2006; Benhabib, Perla and Tonetti, 2021; König, Storesletten, Song and Zilibotti, 2022). Determining whether technological growth arises from imitation or genuine innovation is important to understand the long-term implications of these blockages.

To assess imitation and innovation, I measure technological similarity between domestic and blocked foreign apps using update logs – textual summaries of newly added features and functionalities.[12] An increase in similarity scores suggests domestic apps are aligning their updates more closely with foreign apps, indicating imitation. Conversely, innovation is characterized by the introduction of unique features not present in foreign apps, which leads to a decrease in the similarity score.

By comparing the technological similarity of substitutable foreign-domestic app pairs before and after the blockage of the foreign app, the findings reveal a steady decrease in similarity scores following the blockage, while no effect is observed for neutral pairs. This pattern suggests that domestic apps begin to pursue divergent innovation paths, developing more distinct features over time.

The preceding analysis provides direct evidence of the positive impact of blockage on innovation. Next, I present three pieces of evidence showing one potential mechanism

[12]The construction of technological similarity involves a four-step process: First, each log is converted into a vector representing unique features weighted by their importance. Second, cosine similarity quantifies the alignment between any two vectors; a higher score indicates greater feature similarity between the apps. Third, each domestic app's monthly update is benchmarked against all foreign app updates from the preceding 12 months, which represent a potential pool of features for imitation. The cosine similarity between the domestic and each foreign update is calculated, with the maximum value designated as the similarity score for that update. Finally, averaging these maximum scores across all updates of the domestic app within that month yields an overall similarity score for each domestic-foreign app pair.

that may be an important contributor to this effect: the role of data. In particular, I show that *(i)* data collection and sharing efforts significantly increase following the blockage; *(ii)* the resulting growth in user data appears to lead to more internal technological development; and *(iii)* the blockage generates positive spillover effects on other apps via data-sharing networks.

Historical data on apps' privacy practices enable the analysis. I explore how apps collect user data by examining keys in the information property list file (`Info.plist` file) of app versions. These keys are unique identifiers that provide essential information to the device's operating system about the app's configuration and requirements. Specifically, a certain set of keys specifies the types of user data the app requires, enabling the operating system to grant access and transfer the data accordingly.[13] In alignment with Apple's privacy guidelines, I categorize the `Info.plist` keys into four categories based on data sensitivity – whether the data include identifiable information – and the extent of data sharing – whether it is shared with third parties. The categories are: sensitive and shared, sensitive but not shared, non-sensitive, and unrelated to user data.

By comparing the number of sensitive `Info.plist` keys within each domestic app, I find that, in terms of *data scope*, Chinese apps are collecting 22% more types of sensitive user data and sharing 9% more types of such data with third-party platforms *per user* following the blockage. Accounting for the growth in the user base – *the scale of data* – the overall scale of data collection has surged by approximately 50%, highlighting a significant expansion in data acquisition and sharing practices.

Given the expansion in user data collection especially the scope of data, the next question is how do these data contribute to innovation in apps? Answering this question is challenging because directly regressing innovation measures on the types of data collected by apps raises two endogeneity concerns. First, the choice of data is endogenous to an app's productivity. Second, causality may be reversed, as an app's decision to request more user data could be a response to innovations already implemented.

To identify the causal effect of data on technology development, I focus on data scopes that allow the use of shift-share instrumental variables (IVs) to capture changes in the types of data collected. Periodic updates to the iOS developer system introduce new `Info.plist` keys, which vary across data types and privacy levels. These introductions

[13] For instance, keys like `NSCameraUsageDescription` and `NSLocationWhenInUseUsageDescription` are required if an app needs to access the camera or use location services, respectively.

serve as natural experiments that exogenously enhance[14] or reduce[15] apps' capability to acquire additional user data – the "shift" – based on each app's existing collection of `Info.plist` keys optimized for its functionalities – the "share." The first stage of the IV regression shows that introducing a new `Info.plist` key correlates with 0.016 more user-data-related keys in an app if its privacy level is higher than existing settings under the same data type in the app; alternatively, the keys decrease by 0.043 if the privacy level is lower. Moreover, there is minimal impact on keys and iOS libraries not related to user data, mitigating concerns that the IVs might be correlated with other iOS system changes affecting innovation.

The IV regression of the change in the quantity of technologies on the change in data indicates that having more data significantly increases an app's technological innovation. Specifically, when examining apps within firms to control for internal spillover effects, adding one additional user-data-related `Info.plist` key to an app leads to an 11% increase in the number of in-house libraries *in the following month*. The short time span makes it less likely that adjustments in capital and labor are the primary factors.[16] Furthermore, there is no significant effect on the adoption of third-party libraries, implying that the boost in innovation stems from internal efforts rather than integrating external technologies.

The observed expansion in data collection following the blockage, combined with data's positive impact on technology development, collectively suggests the direct effect of the blockage on within-app innovation through data. Furthermore, if the blockage stimulates innovation via data, spillover effects should be observable in other apps through data-sharing linkages, as data are nonrivalrous and can be utilized by multiple entities simultaneously (Jones and Tonetti, 2020).

To test the hypothesis, I construct a specific data-sharing network by analyzing data transfers between apps and third parties over insecure networks. Since 2015, Apple has required apps to specify exceptions for insecure network communications in their `Info.plist` files. From 2015 to 2017, however, most data transfers in China occurred

[14]For example, in December 2016, Apple introduced the `NSLocationAlwaysAndWhenInUseUsageDescription` key, allowing developers to request both "When In Use" and "Always" location access in a single prompt. This streamlined the process, making it easier to obtain continuous background location data, thereby increasing the amount of user data that apps could collect.

[15]For example, in September 2015, Apple introduced the `LSApplicationQueriesSchemes` key, which limits an app's ability to query the presence of other installed apps via their URL schemes. Apps must now explicitly declare the schemes they want to query in their `Info.plist`. This change reduced apps' capability to collect data about other installed apps unless explicitly declared.

[16]On average, each app in the sample has 0.5 updates per month, suggesting that product iteration is extremely rapid in this industry. Therefore, the time span is sufficient for observing developments in an app's library.

over insecure channels, partly due to restrictions imposed by the GFW. This environment provides a unique opportunity to map app-to-firm data-sharing relationships by directly identifying the declared data receivers at the firm level from the `Info.plist` files of the data-provider apps.

By comparing the performance of data receivers before and after the blockage of foreign substitutes of data providers I find suggestive positive spillover effects on innovation through the data-sharing network. Specifically, following the blockage of data providers' foreign substitutes, there is an upward trend in the number of in-house libraries developed by data-receiver firms, along with a significant increase in the adoption of self-innovated libraries. Placebo tests, including blockages of foreign substitutes that occurred before the data-sharing relationships were established and blockages of foreign neutrals after the relationships formed, show minimal impacts on data receivers, reinforcing the causal link between the blockage and the observed innovation spillover.

In summary, revisiting the rationale for industrial policy in light of the evidence, a number of the results discussed above and features of the setting are strongly suggestive of the policy rationale for protectionist policies in this context: *(i)* the growth of an ecosystem and data-sharing network and the likelihood of knowledge spillovers; *(ii)* positive spillovers arising from expanding data-sharing networks, which, even when priced, may yield positive externalities when consumers retain property rights over their data (Jones and Tonetti, 2020); *(iii)* the dynamic learning as data increase, which, even if internalized, may justify policy intervention if firms are credit constrained or myopic.

Related Literature – The debates surrounding industrial policy primarily arise from practical limitations rather than theoretical ambiguities, as outlined in Juhász et al. (2023). A substantial theoretical framework supports the rationale for industrial policy, particularly regarding static and dynamic externalities, with foundational contributions from (Hirschman, 1958; Baldwin, 1969; Krugman, 1987; Young, 1991; Matsuyama, 1992; Melitz, 2005; Stiglitz, Lin and Patel, 2013). While existing empirical research has identified beneficial spillovers from industrial policies, whether broadly through external economies of scale (Bartelme et al., 2019; Garg, 2024), or via more specific mechanisms like input-output linkages (Liu, 2019; Manelici and Pantea, 2021; Lane, 2022), knowledge spillovers (Juhász, 2018; Juhász et al., 2023), and internal learning with financial frictions (Choi and Levchenko, 2021; Barwick et al., 2024a), direct empirical evidence connecting these positive externalities to the effectiveness of industrial policy is sparse, especially in reduced-form analyses. The primary contributions of this paper are twofold. First, I develop novel metrics for measuring innovation beyond aggregate sales (Juhász, 2018; Rotemberg, 2019;

Manelici and Pantea, 2021; Barwick et al., 2024b), total factor productivity (Lane, 2022), and patents (Barwick et al., 2024a), opening the black box of firm productivity changes and product-level upgrading. This includes directly examining changes in technology inputs,[17] assessing the competitiveness of self-developed technologies, and distinguishing between pure imitation and genuine innovation. Second, I provide new empirical evidence of positive spillover effects within technology and input-sharing networks.

This paper also contributes to literature on the impacts of international trade on innovation.[18] Prior studies have documented mechanisms – including market size effects (Lileeva and Trefler, 2010; Coelli et al., 2022), learning (Atkin, Khandelwal and Osman, 2017), and technology spillovers (Bai et al., 2020; Bergeaud et al., 2024) – that generally have positive impacts on innovation. However, evidence on the competition channel is more mixed.[19] This paper contributes to this debate by detailing the mechanism through which the exclusion of foreign competition influences domestic firms' data collection and data-sharing practices, which may in turn feed back into innovation.

Lastly, this paper contributes to the expanding literature on the economics of data, particularly regarding how data enhance firms' productivity functions; see Veldkamp and Chung (2024); Goldfarb and Tucker (2019) for an in-depth review. While prior empirical studies have predominantly examined the impact of data on within-firm productivity,[20] this paper offers complementary insights by demonstrating that, beyond driving internal efficiencies, data can also generate positive spillovers that extend beyond individual firms.[21]

Outline – The structure of this paper is outlined as follows: Section 2 provides a detailed description of the research setting. Section 3 elaborates on the data structure and the construction of variations. Section 4 assesses the impact of the blockage on innova-

[17]The use of library information in mobile app development to assess developers' technology choices is relatively new, though there are already studies in this area. For instance, Jin, Liu and Wagman (2024) examines how the EU's privacy law, GDPR, influenced app developers to use fewer third-party tools in their apps, especially in apps available in both the US and Europe.

[18]For comprehensive surveys on this topic, see Bloom, Van Reenen and Williams (2019) and Akcigit and Melitz (2022).

[19]Gorodnichenko et al. (2010) find robust positive increases in firm-level innovation measures due to increased import competition; however, Autor et al. (2020) argue that Chinese import competition has a negative impact on the patenting activity of U.S. manufacturing firms.

[20]Begenau et al. (2018) suggest that access to big data has lowered the cost of capital for large firms relative to smaller ones; Bajari et al. (2019) investigate how dataset size influences the performance of data-driven decision systems at Amazon, finding that while larger datasets improve forecasting accuracy, the marginal benefits taper off as data volumes become very large; Demirer et al. (2024) show the complementarity between data storage and computing.

[21]Beraja et al. (2023) also observe that access to government data via public security contracts substantially boosts the production of commercial AI software in China.

tion. Section 5 explores the most important underlying mechanism: the expansion in data collection and data sharing. Finally, Section 6 concludes by evaluating how the findings support or challenge the infant industry argument.

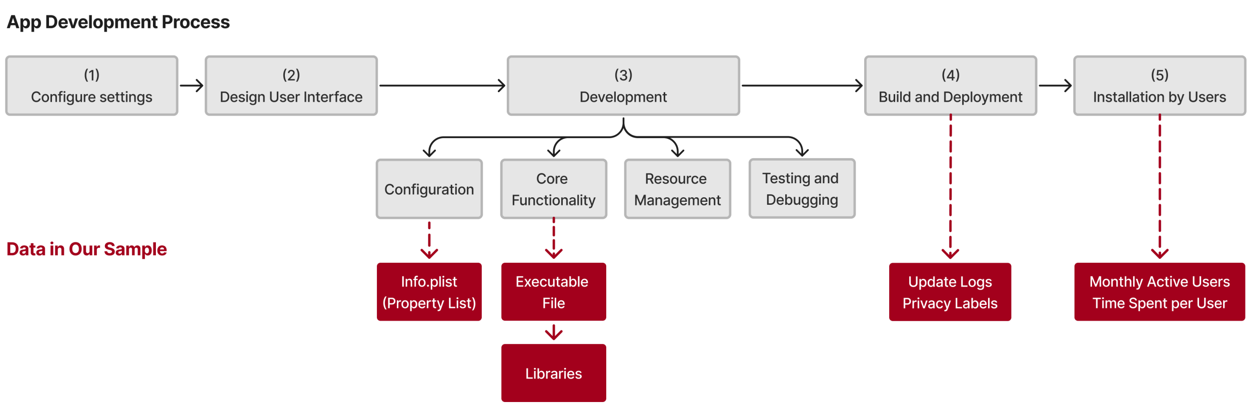
2 Context

2.1 The Ecosystem of Mobile Apps and Backend Technologies

In this section, I first outline the app development process to contextualize the key data used in the analysis, such as information property lists, libraries, and update logs. I then demonstrate how technology networks can be constructed based on the linkages formed through library adoption. Finally, I highlight how tech companies generate significant revenue by offering their advanced technologies as APIs[22] or SDKs[23], enabling other organizations to seamlessly integrate these tools into their products, thereby expanding the reach and functionality of these technologies.

App Development Process – The development process of a mobile application project involves several critical steps to ensure a robust and functional app. Figure 1 describe the pipeline for a mobile application development.

Figure 1: Conceptual Process of Mobile Application Development



After the development team configures essential project settings such as the app name and target devices (Step 1 in Figure 1), they proceed to design the user interface compo-

[22] An API (Application Programming Interface) is a set of rules and tools that allows different software applications to communicate with each other. It defines methods and data structures that developers can use to interact with the service.

[23] An SDK (Software Development Kit) is a collection of software development tools, libraries, and documentation that developers use to create applications for specific platforms or services. It often includes APIs, sample code, and development utilities.

nents and layout structure (Step 2). The next crucial step is the development phase (Step 3), where the app's core functionalities are implemented and also our major data come from.

Under the "Development-Configuration" step, developers must specify essential information (like version number, supported device) about the app in a file called the Information Property List (`Info.plist` file). A critical aspect of the `Info.plist` file is its role in detailing the app's data collection practices, particularly those related to user privacy. Developers must include key-value pairs in this file to declare the types of data their app collects, how this data is used, and whether it is shared with third parties. This transparency is part of Apple's stringent privacy requirements, designed to protect user data and ensure users are informed about how their information is handled. For example, if an app requests access to sensitive information such as the camera or location services, the `Info.plist` file must contain specific keys like `NSCameraUsageDescription` and `NSLocationWhenInUseUsageDescription` to provide a clear explanation to users.

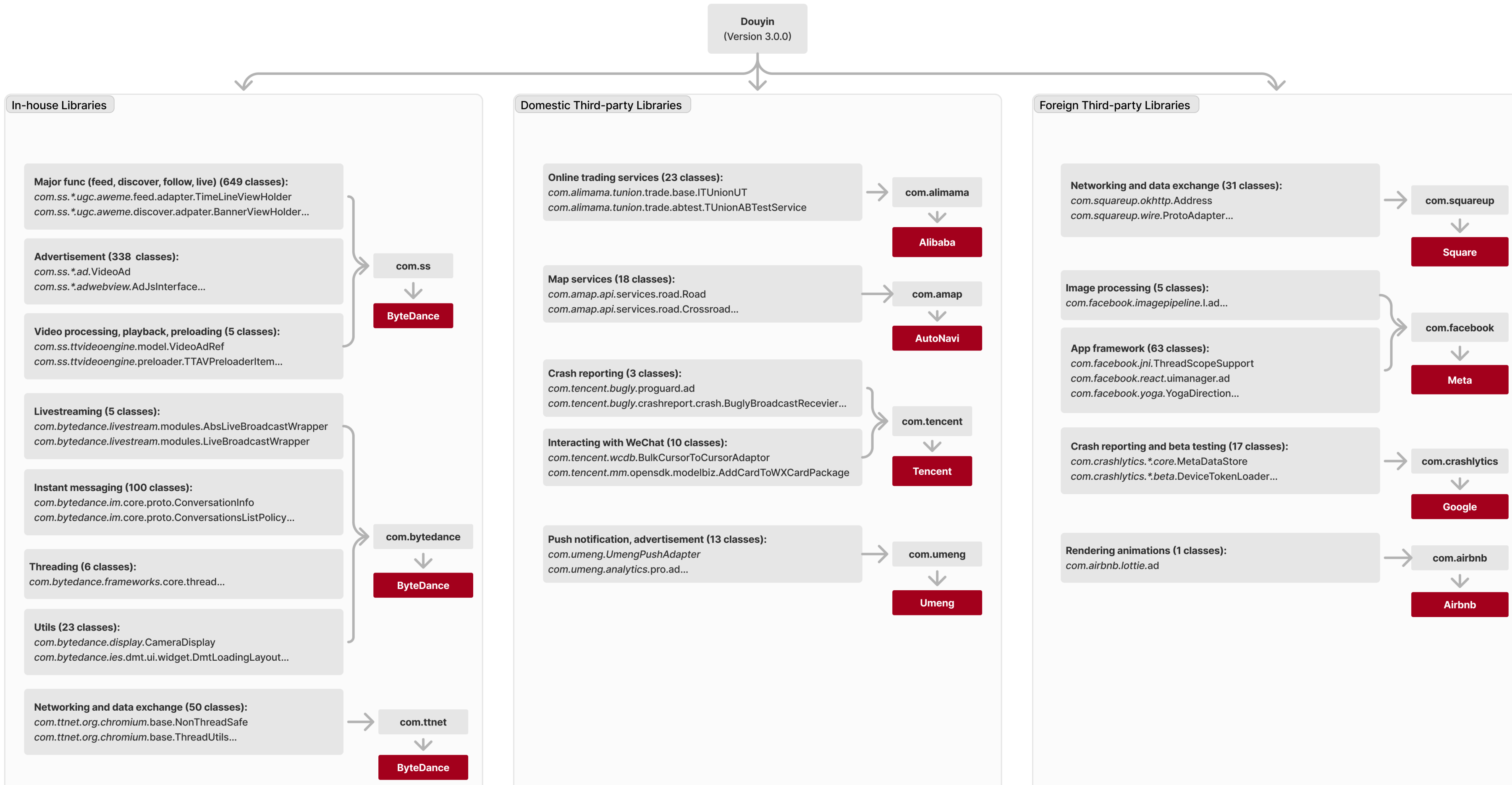
Under the "Development-Core Functionality" step, developers focus on implementing the primary features and functionalities of the app. The core functionality developed during this step is ultimately packaged as a main executable file. From this file, one can extract all classes—the most fundamental units in a computer program—used by the app. These classes follow a specific naming convention. For example, a class in *Douyin* (the Chinese version of *TikTok*) version 3.0.0 is structured as follows:

$$\underbrace{\underbrace{\underbrace{\underbrace{\text{com.bytedance.}}_{\text{Developer Name}}\,\text{livestream.}}_{\text{Library Name}}\,\text{modules.}}_{\text{Module Name}}\,\text{AbsLiveBroadcastWrapper}}_{\text{Class Name}}$$

This naming structure provides detailed information about the class's origin, making it possible to identify the developer, library, and module associated with each class, providing insights into the technologies utilized and the source of these technologies.

Libraries play a crucial role in mobile application development by providing reusable code, tools, and functionalities that simplify and accelerate the development process. They can be either external or developed in-house. In Figure 2, we decompile the full package of *Douyin* (the Chinese version of *TikTok*) version 3.0.0 to illustrate the composition of in-house and third-party libraries within a mobile app.

Figure 2: Libraries in *Douyin* (the Chinese version of *TikTok*) version 3.0.0



Notes: Grey boxes indicate the raw classes extracted from the *Douyin* 3.0.0 package. Library names within class names are italicized, and company names are highlighted in red boxes.

Two key points are highlighted in this figure: *(i)* the majority of the libraries are in-house, indicating a significant reliance on internally developed technology; *(ii)* despite foreign domains like `google.com` and `facebook.com` being blocked in China, Chinese firms can still access and adopt technologies from their companies (Google and Meta) as third-party libraries. This demonstrates the complex interplay between domestic and foreign technologies in Chinese apps, where even with restrictions, global technologies continue to play a role.

After the development phase is completed, apps typically undergo updates to introduce new features, improve performance, or fix bugs. Each update is accompanied by an update log, as illustrated in Step 4 of Figure 1, which reflects all the new libraries added to the app in a textual format. Additionally, privacy labels must be submitted and reviewed by Apple, detailing the data collection practices of the app (for more detailed background about privacy labels, please see Appendix C).

Users can finally access the app after the new version is successfully on the shelf. This step completes the lifecycle from development to deployment, ensuring that the app is functional, up-to-date, and compliant with privacy standards. This detailed development process allows us to track and analyze technological advancements within apps.

B2B Tech Integration – Tech companies are not only developing cutting-edge technologies for their own consumer-facing (B2C) services but also packaging these mature technologies as APIs and SDKs to provide business-facing (B2B) services. This strategy enables developers from other organizations to easily integrate advanced tools into their own products, thereby enhancing functionality and reducing development time.

For example, the Google Maps API,[24] launched in June 2005, allows developers to embed Google Maps into their websites or applications, making it one of the most popular online map services for third-party use. As of 2023, the Google Maps API contributes 18% of Google Maps' earnings.[25] Similarly, large language models (LLMs) are increasingly being offered through APIs or SDKs, with 72% of companies utilizing these models via such services rather than opting for self-hosting (Xu, 2024). Additionally, 27% of OpenAI's revenue is generated from its API services, despite its focus on B2C use cases.[26]

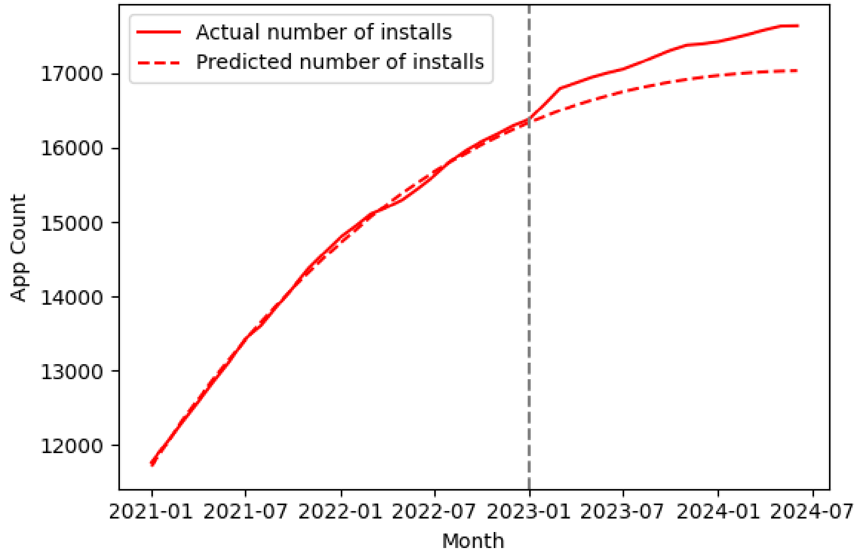
This approach not only extends the reach of the technology but also creates a significant revenue stream for these tech companies. Chinese tech firms are also focusing on this model, leveraging their expertise in various technologies to offer comprehensive plat-

[24] The Google Maps API allows developers to embed and customize Google Maps within their websites or applications, providing access to a variety of mapping features such as geolocation, route planning, and traffic information.

[25] Source: `https://shorturl.at/dFPR6`

[26] Source: `https://www.tanayj.com/p/openai-and-anthropic-revenue-breakdown`

Figure 3: Current Installs of Azure SDKs



Notes: This figure shows the current installs of Azure SDKs before and after the integration of GPT models in January 2023. The solid red line represents the actual number of current installs of Azure-related SDKs. The dashed line depicts the predicted trend based on data before January 2023. This trend was fit using a cubic polynomial regression model trained on pre-January 2023 data, which was then used to predict the app count across the entire dataset.

forms and development tools, thus competing with US tech giants in overseas markets (Huang, 2023). Companies like Tencent and Alibaba, for instance, provide a wide range of services for payment solutions to other businesses after establishing successful payment apps, facilitating rapid integration for businesses worldwide.

Technology Network Built on Libraries – The use of external libraries creates a network that measures technological dependencies among tech firms. In March 2024, OpenAI began adopting A/B testing, as announced by Statsig, a company specializing in A/B testing services.[27] This integration is explicitly evidenced in ChatGPT's version 1.2024.045 source file, uploaded to the Apple App Store in February 2024, where the `StatsigInternalObjC` library is imported.

The OpenAI-Statsig link is a rare case where the connection is announced publicly, highlighting the importance of detecting external libraries to reveal technological connections between firms. OpenAI and Azure, Microsoft's cloud computing service, have a significant partnership that leverages Azure's powerful computing infrastructure to host and scale OpenAI's GPT-3 models. In January 2023, OpenAI's GPT-3 models became generally available for business use cases through the Azure OpenAI Service.[28] As illustrated

[27]Source: https://statsig.com/customers
[28]Source: https://shorturl.at/5zXI0

in Figure 3, there was a marked increase in the installation of Azure-related libraries following this release. This surge indicates a growing adoption of Azure's infrastructure by various businesses, likely driven by the new accessibility and capabilities offered through the integration of OpenAI's models. This example highlights how tracking the use of external libraries can provide valuable insights into the technological ecosystems and strategic partnerships within the industry.

2.2 The Great Firewall

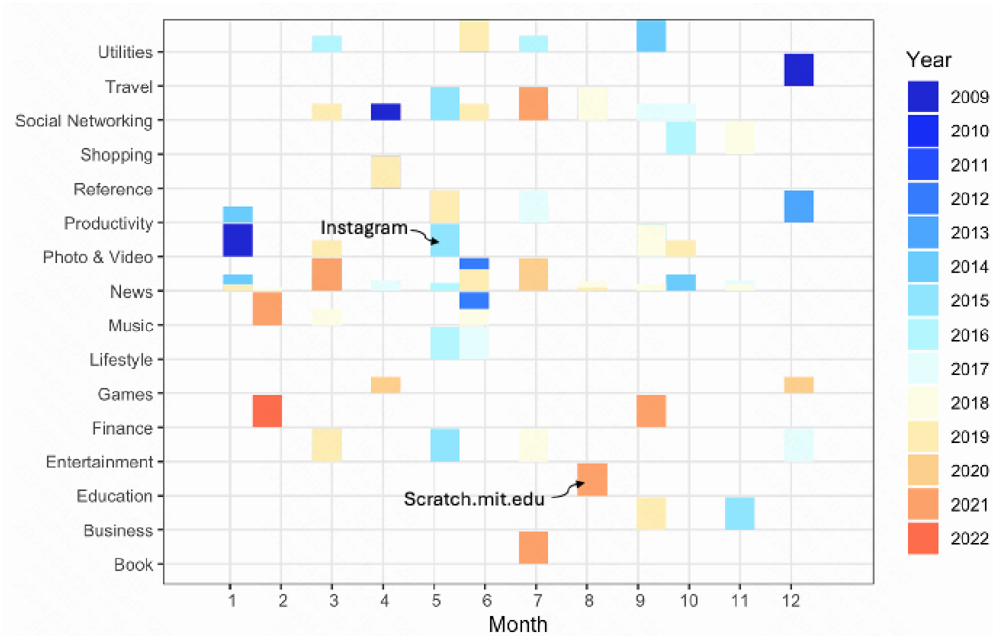
The Great Firewall (GFW) of China is the world's largest system of internet regulation, established while China's digital economy was still in its infancy. Initiated in 1998 and completed in 2006, the GFW has grown to become a key component of China's internet infrastructure. At its inception in 2006, China had only 131 million internet users, which accounted for 10.5% of its population and 18% of the global internet population. By 2023, this number had surged to nearly 1 billion users, representing 70.6% of China's population and 59.5% of internet users worldwide.

A comprehensive study by GFWatch (Hoang et al., 2021) highlighted the vast censorship capabilities of the Great Firewall (GFW). Over a nine-month period in 2020, daily tests of millions of domains uncovered 311,000 censored domains, showcasing the extensive reach of the GFW. By linking blocked domains to apps, as detailed in Appendix A.4, Figure 4 illustrates the distribution of blockages for 114 major foreign apps, categorized by app type and over time.

This system sets the stage for quasi-random experiments that are *(i)* predominantly politically driven, *(ii)* inherently unpredictable, and *(iii)* notably difficult for the general public to circumvent.

Primarily Political Blockages – The initial purpose of GFW was not to promote the domestic digital economy but to control information flows and maintain stability. Instances of blocked online content often align with the government's strategy of information control on sensitive topics, frequently driven by quasi-random political events. For example, `instagram.com` was blocked during the 2014 Hong Kong pro-democracy protests to prevent protesters from sharing images and information. Through manual investigation of the reasons behind major foreign app blockages, at least 55% of these blockages are found to be triggered by sudden political events. Figure A.11 highlights the temporal distribution of these politically motivated blockages, providing insight into their frequency and patterns over time.

Figure 4: The Distribution of Blockage across App Category and Time



Notes: This figure show the distribution of blocked foreign apps in our sample across app category (in iOS store) and time. The y-axis represents the app category, the x-axis shows the month when the app is blocked. The color of the represents year in which the app is blocked. And the height of each bar indicates the number of blocked apps.

Unpredictability – It is exceptionally challenging for users and domestic firms to predict which websites or applications will be blocked by the Great Firewall (GFW), especially at more granular time frames like months. Unlike censorship methods in other countries that redirect users to pages notifying them of censorship, the GFW uses a more subtle approach by sending misleading information. For example, when someone in China tries to access a blocked website, the GFW might provide incorrect directions, causing the user's browser to reach the wrong destination or no destination at all. This results in error messages or connection timeouts without any clear indication that the site is censored. It makes it seem as if the website is down due to technical issues rather than being deliberately blocked.[29] This tactic obscures the occurrence of censorship, making the anticipation of blockages nearly impossible.

Circumvention Disparities Between Consumers and Firms – The GFW's robust and evolving censorship techniques have made circumventing it increasingly difficult, posing

[29]Technically, utilizing a combination of DNS strategies and IP restrictions can efficiently isolate websites and servers that are subject to censorship, targeting them by both their domain names and IP addresses. For instance, DNS manipulation was employed not just to prevent access to controversial material but also to boost local enterprises. In a notable case from 2002, for a period of two months, attempts to visit google.cn were rerouted to Baidu, which is the dominant search engine in China.

distinct challenges for consumers and firms. Virtual Private Networks (VPNs) are among the most common tools developed to bypass these controls, allowing users to connect to servers outside China and access blocked content by masking their IP addresses. However, the Chinese government frequently updates its censorship mechanisms to detect and disrupt VPN usage, and by March 2018, it began to more strictly enforce regulations targeting both individuals and businesses using unauthorized VPNs.

While tech firms in China often establish their own VPN tools[30] for employees to access foreign resources, domestic firms can still access blocked foreign content, especially technological resources. However, this practice is costly and typically not feasible for individual consumers. As a result, circumvention remains significantly more challenging for consumers compared to firms. This disparity between consumers and firms makes the blockage of foreign products by the GFW resemble a protectionist policy, generating local positive demand shocks for domestic firms without reducing their access to foreign technologies.

3 Empirical Strategy

3.1 Data

To examine the impact of the GFW on domestic internet companies, I have compiled four databases: (1) The *Domestic Apps Database* contains detailed records on over 230,000 apps from 6,000 Chinese internet firms from 2014 to 2023, including data on user activity and updates. More importantly, I've extracted user permissions and libraries from the source code of each app version, providing insights into user privacy practices and the technologies utilized by these apps; (2) The *Foreign Apps Database* include 114 popular foreign apps that are blocked in China, as well as approximately 450,000 additional apps, selected by randomly choosing 1% of apps per country per year. (3) The *Library Database* links these libraries to their respective companies and provides detailed descriptions from their `GitHub` repositories, enhancing the understanding of technology sourcing and application; (4) The *Blockage Database* records the blockage status of 114 popular foreign products by the GFW, with detailed start and end dates of blockages, sourced from `GFWatch` (Hoang et al., 2021) and `greatFire.org`. Figure A.12 presents an overview of the data structure and data sources. More details about data construction are described in Appendix A.

Despite the GFW influencing competition from foreign apps, China's app industry

[30] Examples include Kit used by Kuaishou Technology and Feilian for ByteDance.

demonstrates notable variability in market concentration across different categories. The Herfindahl-Hirschman Index (HHI), calculated based on each app's usage share within its category, reveals a broad spectrum of concentration levels. In the Education category, the HHI is as low as 450, indicating a highly competitive market, while the Navigation and Shopping category reaches an HHI of 5,500, suggesting dominance by a few major players. The median HHI across all categories stands at 1,834.

Moreover, from 2014 to 2023, China's app industry has remained exceptionally dynamic and fast-paced. On average, each Chinese app is updated approximately 0.5 times per month, indicating that product iterations occur on a near-monthly basis. This high frequency of updates allows for analyses at the monthly level to effectively capture and reflect market shocks.

3.2 Mapping App Descriptions to App Relationship

To measure the extent to which the blockage of a foreign app influences domestic apps, I identify substitutable, complementary, and neutral pairs of apps. This is crucial for two reasons. First, existing market definitions, such as app categories, can be overly broad and subject to manipulation by developers.[31] Second, defining these relationships allows us to generate meaningful cross-sectional variation for economic interpretation.

Given the vast number of app pairs and the scarcity of labels of interest, such as substitute and complement, traditional approaches like consumer surveys are prohibitively costly and inefficient. To address this, I develop a cost-effective and replicable three-step methodology: (1) creating a training set by asking `gpt-4` to label app relationships based on their descriptions, (2) validating the labeling results of `gpt-4`, and (3) training a classifier with this labeled training set. This approach leverages the scalability and adaptability of language models to efficiently handle large-scale labeling tasks, building on the growing use of such models in market research (Li, Castelo, Katona and Sarvary, 2024; Brand, Israeli and Ngwe, 2023).

First, for the creation of the training set, I employ `gpt-4` to analyze the descriptions of pairs of applications. The labeling task presents two main challenges: on the one hand, the sample is highly imbalanced, with the most important label representing a very small fraction of the data; on the other hand, using human labeling to generate an effective training set is prohibitively costly due to this imbalance. To address these issues, I use large language models that are already trained on millions of human-labeled data points

[31] For example, Instagram is categorized under Photo & Video, while TikTok is listed under Entertainment, despite the two apps competing directly in the social media space. Developers may strategically choose their app category to influence their ranking in the Apple App Store.

for classification. Specifically, I use the gpt-4 model to classify each pair of app descriptions as substitutes, complements, or neutral, using a specific prompt detailed in Figure A.3. I randomly select 18,066 pairs of applications and label their relationships, thereby creating a substantial training dataset for. Examples for each category are shown in Figure A.4.

Second, I validate the labeling consistency of gpt-4 to ensure reproducibility emphasized by Dell (2024) and assess its alignment with human evaluations. To test consistency, I randomly selected 20 app pairs and ran the model 10 times for each pair using identical inputs. I then calculated the average proportion of times the model produced the most frequent label across these runs, yielding a consistency rate of 0.94, which demonstrates the overall robustness of the classification. Figure A.5 visualizes these results, showing that the model's classifications for substitutes and complements are generally robust, although complements exhibit slightly less consistency. Additionally, all instances of substitutes and complements, along with a randomly selected set of neutral labels (2,667 app pairs in total), were independently reviewed by human annotators, achieving an 83% agreement rate with gpt-4's labels.[32] More details on the human labeling process can be found in Appendix B. Figure A.6 presents the confusion matrix, revealing that human evaluations align closely with gpt-4 for neutral and substitute labels, while complements show a relatively higher degree of divergence.

Lastly, based on this training set, we encode the descriptions of each application within these pairs into vectors of size 768.[33] Using these vectors as inputs, I train a Logistic Regression model to predict the relationships as classified by gpt-4, achieving an accuracy rate of 0.86 on the test set.[34]

Applying the classifier to the full sample of 15,373,920 foreign-domestic app pairs, where we vectorized the app descriptions prior to the blockage of the foreign app in each pair, we find that 16.5% are classified as substitutes, 2.1% as complements, and the remaining 81.3% as neutral. To ensure that our treatment is not overly concentrated among a few domestic apps, which would indicate a highly selective treatment effect, we further analyze the distribution of treatment types across domestic apps. As illustrated in Figure A.7, 63.6% of the domestic apps have at least one foreign substitute that is blocked, indicating that the treatment is sufficiently widespread. Additionally, 24% of Chinese apps

[32]This result aligns with Sun (2024), who demonstrates that cosine similarity from large language model embeddings can effectively capture these substitution patterns.

[33]This encoding leverages the bert_multilingual_cased model, chosen for its robust performance across multiple languages and its effectiveness in capturing essential semantic nuances for our analysis.

[34]We also test other models such as Random Forest Classifier, Linear SVC, and Neural Networks, with accuracy rates shown in Table A.1.

have neither foreign substitutes nor complements blocked, forming the never-treated group in our empirical design. Finally, due to the small size and ambiguity results in consistency and alignment test, I omit complements in the rest of analysis.

4 Causal Impacts of Blockage on Innovation

In this section, I first examine the direct effect of foreign substitute blockages on the domestic demand for Chinese apps and the number of libraries in domestic apps sourced from blocked foreign apps. I demonstrate that the blockage mainly generates a positive demand shock while causing minimal reduction in knowledge spillover. Given this positive demand shock, I then explore the effect of the blockage on Chinese app innovation from three perspectives: quantity, quality, and originality.

4.1 Blockage as Demand Shock

Before presenting the results on innovation and other metrics, I begin by analyzing the average impact of foreign substitute blockages on domestic app demand over the past decade. To capture both the extensive and intensive margins of demand, I use two key measures. The first measure, monthly active users, reflects the number of unique users who engage with an app at least once during a given month. This metric provides a clear view of the app's reach, indicating how many users find it valuable enough to use regularly—capturing the extensive margin of demand. The second measure assesses the average time each active user spends on the app per month, offering deeper insights into the intensity of user engagement and the overall user experience—representing the intensive margin of demand.

To causally identify the effect of the blockage on domestic app demand, I compare the mean differences in demand measures for domestic apps before and after the blockage of their foreign substitutes, against those for apps that have not yet been treated or will never be treated. In order to simplify the estimation by lessening the concern about heterogeneous treatment effects, I start by only considering *the first blockage of foreign substitutes* that the app receives to benchmark the effect.

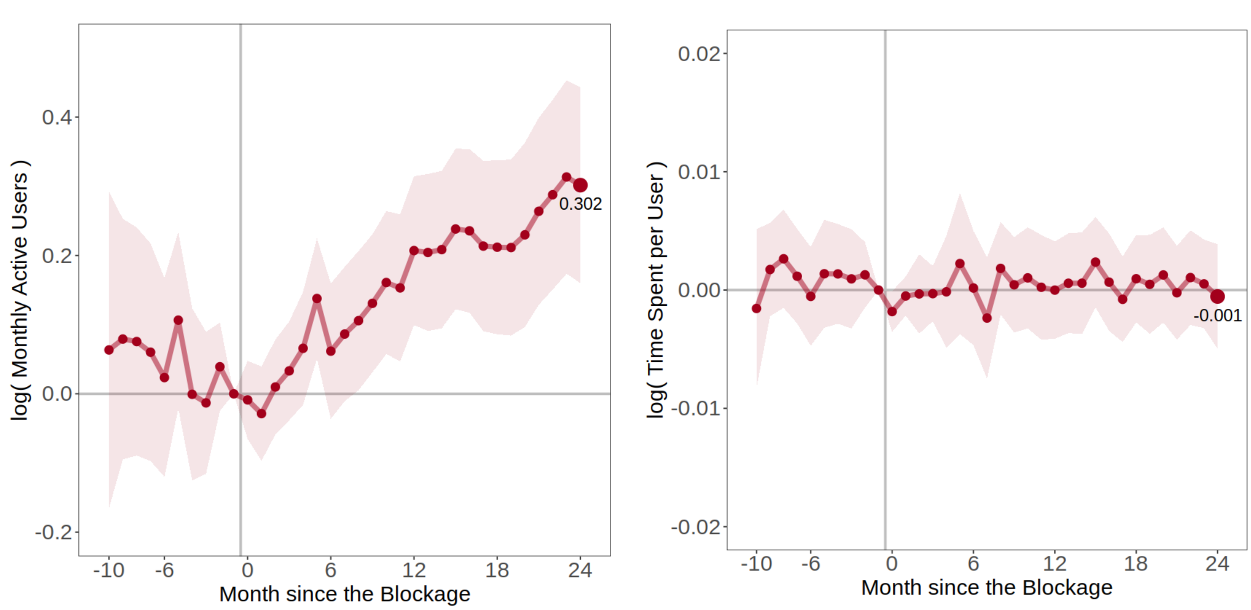
Specifically, I estimate the following two-way fixed-effect model:

$$Y_{it} = \sum_{l \neq -1} \beta_l SubstituteBlocked_{it}^{l} + \psi_{g(i)t} + \delta_i + X_{it}^{T}\xi + u_{it} \tag{1}$$

where $SubstituteBlocked_{it}^{l}$ is an indicator for domestic app i being l months away from the first blockage of its foreign substitute at time t; $\psi_{g(i)t}$ is a full set of app category-time fixed

Figure 5: The Effect of Blockage on Domestic App's Demand

(a) Extensive Margin: Monthly Active Users **(b)** Intensive Margin: Time Spent per User



Notes: These figures show the effect of blockage on monthly active users and time spent per user. They plot the estimates based on Equation . The red dots and lines shows the point estimates of β_l, and the bandwidth in grey shows the corresponding 95% confidence intervals of β_l. In Panel (a), the outcome variable is the log of monthly active users for app i in month t. In Panel (b), the outcome variable is the log of time spent per user of app i in month t. Standard errors are clustered at the app level.

effects, capturing variations in time trends across different categories of apps to account for broader market trends and category-specific shocks; δ_i denotes domestic app fixed effects, controlling for time-invariant characteristics of each app that might influence its performance; and X_{it} includes control variables including app age, which accounts for the maturity and lifecycle effects on app performance; Additionally, I include $Y_{i,t-1}$ to control for dynamic bias, as discussed by Klosin (2024), which arises when past outcomes influence current outcomes.

The identification assumption is the exogenous timing of the blockage events, meaning that the blockage is not correlated with unobserved factors that might simultaneously affect the demand for the domestic apps.

I present the estimated coefficients graphically in Figure 5, which plots the nonparametrically estimated β_l coefficients along with the corresponding 95% confidence intervals. The effect in the year prior to the blockage is normalized to zero.

Figure 5 Panel (a) illustrates a significant increase of approximately 30% in monthly active users – about 670,000 users – 24 months after the blockage of foreign substitutes, compared to a monthly growth rate of around 4.9% before the blockage. This effect is quantitatively substantial and underscores the impact of the blockage on user engagement. To put this into perspective, the global average cost per installation for iOS apps was $0.86 in 2017 and $3.50 in 2024.[35] Consequently, the surge in demand resulting from the blockage effectively saved the affected apps approximately $576,200 to $2,345,000 in user acquisition costs. This represents a significant cost saving and highlights the economic impact of the blockage on domestic app developers.

The blockage of foreign substitutes is unlikely to reduce access to foreign technologies for two reasons. First, the blockage operates at the domain level, meaning the restriction of one domain (e.g., `chatgpt.com`) does not necessarily extend to related domains (e.g., `openai.com`). For instance, Google Cloud services remain accessible in China, allowing Chinese developers to continue using these services despite other Google domains being blocked.[36] As I explained in Section 2.1, even if a library is hosted on a blocked domain, the third-party library is integrated into the app package during the development stage. Users download the complete package directly from the App Store to their local devices rather than from the blocked domain. This integration process ensures that the blockage does not hinder the use of foreign technologies in Chinese apps. Second, I directly examine the effect of the blockage of foreign substitutes on the number of libraries in domestic apps, specifically those developed by the blocked foreign firms. I present the results in Figure A.13. The analysis reveals that the blockage does not significantly impact the number of libraries in domestic apps.

Finally, while the primary objective of the GFW has been to control information flow and maintain political stability, concerns persist that the government may selectively favor certain promising sectors, as illustrated by the recent blockage of *OpenAI*. This raises the possibility that blockages may align with the government's strategic interests in promoting domestic alternatives or advancing key industries. To address this concern, I have divided the analysis in Figure A.14 to separately examine the effects of blockages driven by political events, as identified through news reports (Panel (a)), and those stemming from non-political or unspecified reasons (Panel (b)). The trends in both cases closely resemble those observed in Figure 5, suggesting that the demand patterns are consistent regardless of the political or non-political motivations behind the blockages.

4.2 Technology Development

Innovation is central to industrial policy. Proponents argue that temporary protection of selected sectors can permanently alter a country's pattern of comparative advantage,

[35]Source: `http://alturl.com/acdx7`; `http://alturl.com/ri9b2`
[36]Source: `https://shorturl.at/3V8vP`

fostering long-term competitiveness and growth (Krugman, 1987; Melitz, 2005). However, critics contend that such protection can shield firms from competitive pressures, potentially stifling innovation and hindering efficiency improvements (Akcigit, Ates and Impullitti, 2018).

To investigate the impact of blocking foreign substitutes on domestic firms' innovation behavior, I decompose my evaluation into three steps. First, I examine the number of domestic (both in-house and domestic third-party) and foreign libraries used within Chinese apps before and after the blockage. The findings indicate that Chinese apps are developing more in-house technologies, with the overall contribution of domestic technology increasing post-blockage, while reliance on foreign technology decreases. However, this divergence in technology adoption does not necessarily imply an improvement in Chinese technology in terms of quality. Therefore, in the second step, I analyze the adoption of Chinese libraries both in domestic and foreign markets to provide a clearer understanding of the quality and competitiveness of Chinese technology.

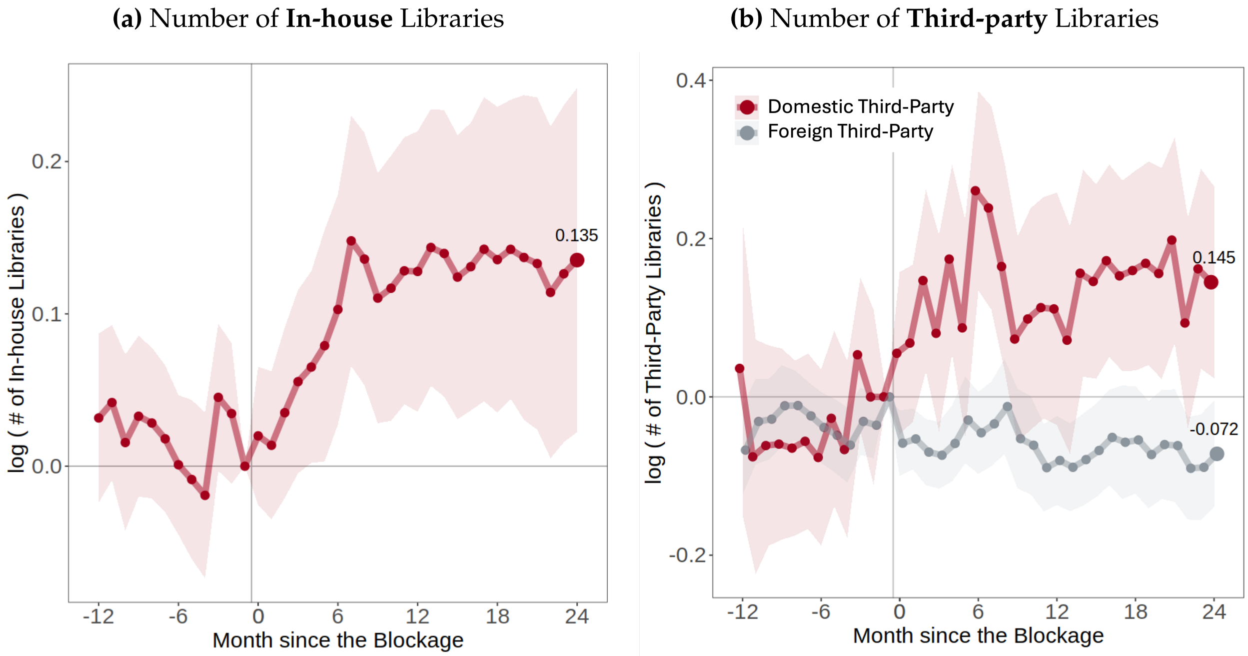
Quantity: Growth in Domestic Technologies – To measure the usage of domestic and foreign technologies within Chinese apps over time at the monthly level, I follow a systematic approach: *(i)* extract all libraries from the package of each version of a Chinese app, *(ii)* manually determine the developers of these libraries by linking the prefixes of the libraries to the corresponding companies, and *(iii)* categorize the libraries into three groups based on the developer's information: in-house libraries (i.e., libraries developed by the company of the app), domestic third-party libraries, and foreign third-party libraries. Figure 2 uses the example of *Douyin* (the Chinese version of *TikTok*) in July 2018 to illustrate this process. Technical details are further explained in Appendix A.1. From 2014 to 2024, I collected 789,076 full packages for 186,220 apps from which I extracted 10,788 libraries.

Using the first blockage of a foreign substitute that a Chinese app experienced during the decade as the treatment, I estimate the effect of the blockage of foreign substitutes separately on the number of in-house libraries using Equation 2. I follow Chen and Roth (2024) and use a numerically equivalent way to obtain the estimate of Equation 5 by using Poisson QMLE to estimate:[37]

$$Y_{it} = \exp\left(\sum_{l \neq -1} \beta_l SubstituteBlocked_{it}^{l} + \psi_{g(i)t} + \delta_i + X_{it}^{T}\xi\right) U_{it} \tag{2}$$

[37]This assumes a "ratio" parallel trends framework, as described in Wooldridge (2023), where, in the absence of treatment, the average percentage change in the mean outcome for the treated group mirrors that of the control group.

Figure 6: The Effect of Blockage on Technologies in Domestic Apps



Notes: These figures show the effect of blockage on the adoption of domestic and foreign libraries. The red dots and lines shows the point estimates of β_l in Equation 2, and the bandwidth in grey shows the corresponding 95% confidence intervals of β_l. In Panel (b), the effect of blockage on the number of domestic libraries are presented with solid lines, while the estimates for foreign libraries are presented with the dashed line. Standard errors are clustered at the app level.

Figure 6 shows the estimates for the effect of blockage on the number of in-house libraries with the solid line and the bandwidth in grey shows the corresponding 95% confidence intervals of the estimates.

There is a significant increase of approximately 14% in the number of in-house libraries in Figure 6 Panel (a). To put this effect into perspective, consider Google's introduction of the "search image with lens" functionality in October 2020, which similarly resulted in a 14% increase in in-house libraries for the app.[38,39] Another notable increase occurred in February 2024, when Google integrated an AI-powered search engine into its app, leading to a 26% increase in in-house libraries.

There is no significant increase in the number of in-house libraries before the blockage of foreign substitutes, and the number of in-house libraries takes off immediately after the blockage and persists for at least two years. This timing, with no pre-trends, is reassuring for the validity of my identification strategy. The absence of pre-trends suggests that there is little anticipation before the blockage of foreign substitutes and there is also no

[38] iOS app ID is 284815942, Android ID is com.google.android.googlequicksearchbox

[39] In contrast, the average monthly growth rate in in-house libraries for the app is around 4.7% from 2013 to 2024.

evidence that foreign apps are selected for blockage based on the number of in-house libraries owned by their Chinese substitutes.

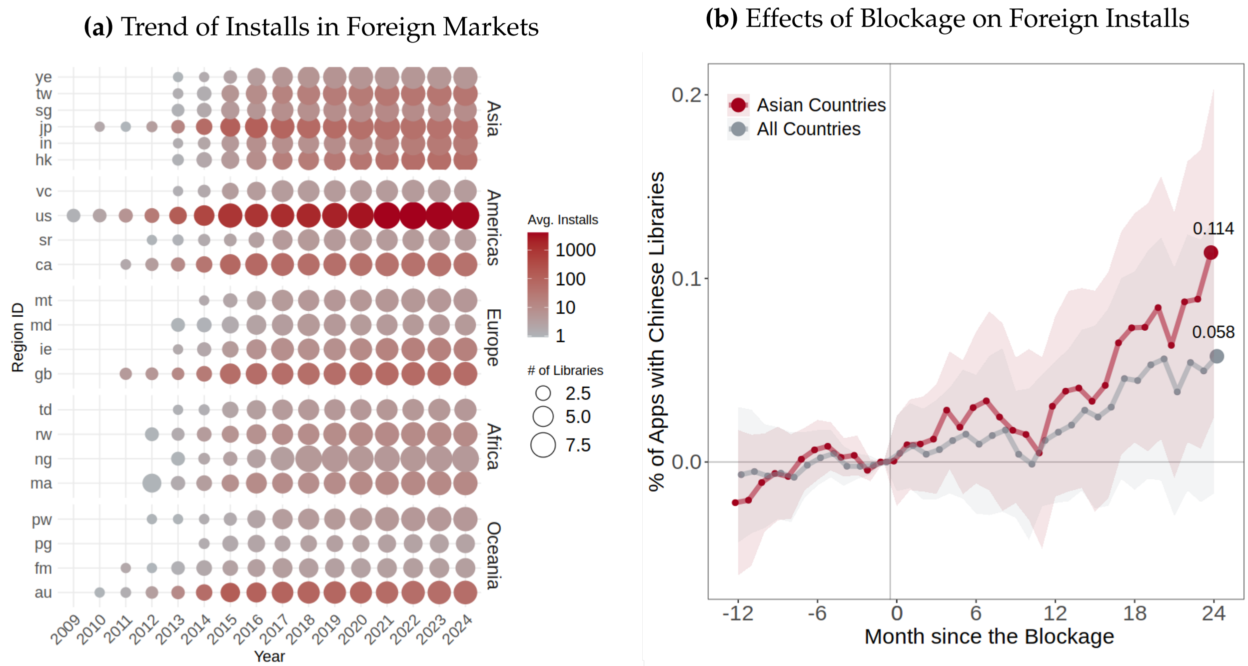
In Figure 6 Panel (b), I present the estimates for both domestic and foreign third-party libraries. The patterns depict a clear divergence in the adoption of domestic and foreign third-party technologies in Chinese apps after the blockage of foreign substitutes. Specifically, the adoption of domestic third-party libraries also rose by 14.5%, shown by the red lines/dots. In contrast, the number of foreign third-party libraries decreased by 7%, shown by the grey lines and dots. This trend suggests that Chinese developers are increasingly collaborating with other domestic firms, leveraging local technologies to enhance their apps' functionalities. This intra-national cooperation likely fosters a more robust domestic tech ecosystem, further driving innovation and reducing dependency on foreign technologies.

Overall, these changes reflect a broader strategic adjustment within Chinese app development, emphasizing self-reliance, enhanced local collaboration, and a shift towards building a more resilient and innovative domestic tech industry. The increased in-house development and domestic partnerships not only bolster the capacity for innovation but also pave the way for sustained growth and competitiveness in the global market.

Quality: The Adoption of Chinese Technologies – I have shown that Chinese apps are increasingly adopting domestic technologies over foreign ones. This trend can be attributed to several mechanisms beyond the mere improvement of Chinese technologies. One possible mechanism is mechanical: due to internet blockages, Chinese firms may have less knowledge about foreign technologies, leading them to rely more on domestic solutions. Another mechanism could be the distance advantage: even if Chinese technologies are comparable to foreign ones, domestic firms might find it less costly and more efficient to switch to local technologies due to closer connections and better integration within the domestic market.

To gain a better understanding of the quality of Chinese technologies, I examine their adoption in foreign markets. This analysis helps me determine whether the preference for domestic technologies is driven by their intrinsic quality or by external factors such as accessibility and connectivity. To track the adoption of Chinese technologies over time, I randomly select 1% of apps per country per year, ensuring the developers are from the respective countries. Following the same procedure as with Chinese apps, I collect the packages, extract the software libraries used by these apps, and link the libraries to their developer companies. Then, for each Chinese company, I aggregate the number of libraries adopted and the number of installs for each library at the country-month level.

Figure 7: The Adoption of Chinese Technologies in Foreign Markets



Notes: Panel (a) presents the trend of the average number of libraries adopted per company (represented by the size of the circle) and the average number of installs per library per company (represented by the shade of the circle) across countries (y-axis) and time (x-axis). Panel (b) exhibits the estimates of β_l in Equation 3. We conduct the estimation separately for Asian countries and all countries. The estimates for Asian countries are represented with red dots/lines, while the estimates for all countries are in grey. Standard errors are clustered at the firm level.

Figure 7, Panel (a) presents the trend of the average number of libraries adopted per company (represented by the size of the circle) and the average number of installs per library per company (represented by the shade of the circle) across countries (y-axis) and time (x-axis). We can clearly see from the figure that Chinese firms are exporting more libraries and achieving higher adoption rates in foreign apps over the years.

To more precisely identify the effect of blockage on the adoption of Chinese technologies in foreign markets, I construct the percentage of apps that adopt its libraries in country c in month t for each Chinese firm f. Then, by defining the earliest time point when a Chinese firm's app faces the blockage of its foreign substitute as the firm-level treatment, I compare the percentage of apps that adopt the firm's technologies before and after the firm experiences the first blockage of its foreign competitor. I run the following regression:

$$Y_{fct} = \sum_{l \neq -1} \beta_l SubstituteBlocked^l_{ft} + \psi_t + \delta_{fc} + X^T_{fct}\xi + u_{fct} \quad (3)$$

where Y_{fct} represents the percentage of apps in country c that include at least one library developed by Chinese firm f at time t. The term $SubstituteBlocked^{l}_{ft}$ is an indicator for Chinese firm f being l months away from the first blockage of its foreign competitor occurring at time t. The variable δ_{fc} is a set of Chinese firm-country fixed effects, controlling for unobserved characteristics that are constant within each firm-country pair, such as the firm's baseline technological capabilities and market strategies. The variable ψ_t is a set of time fixed effects that control for factors varying over time but are common across firms and countries, such as global technological trends and macroeconomic conditions. Lastly, I control for variables X_{fct}, which include firm age and the total number of on-shelf apps in country c at time t.

Figure 7 Panel (b) presents the estimates of β_l from Equation 3, estimated separately for the sample pooling all countries together and the sample including only Asian countries. The results show that the percentage of foreign apps adopting Chinese companies' technologies increases after the blockage of foreign competitors (grey dots/lines), with more pronounced effects in Asian countries (red dots/lines). As more productive firms select into exporting (Melitz, 2003), the increase in adoption of Chinese technologies by foreign apps may reflect the higher quality and competitiveness of Chinese technologies, particularly in regions with closer physical and cultural ties.

Placebos and Robustness – I carry out placebo exercises where I re-estimate the baseline specification (Equation 2), but for blockages of foreign neutrals. Specifically, for each domestic app, I randomly select an app pair such that the domestic app is in the pair and the app pair is classified as neutral. I use the blockage of the foreign app in the randomly selected pair as the treatment and estimate its impact on the number of domestic and foreign libraries. This placebo exercise allows me to assess whether the results are not driven by spurious measurement of technology and also enables me to distinguish the effects of blockage working via signals to all Chinese app developers that foreign technologies may not be easily accessible in the long run. The results of this exercise are presented in Figure A.15. The results shows no significant increase in both the number of libraries (Panel (a)) and foreign installs (Panel (b)), and the point estimates are quantitatively much smaller than my baseline estimates. This lack of impact suggests that, on average, confounding factors such as spillovers from apps whose foreign substitutes are blocked on the control group are negligible.

4.3 Imitation versus Innovation

While previous evidence shows that domestic apps are developing more and higher-quality technologies – suggesting growth in total factor productivity (TFP) – the evolution of the TFP distribution primarily hinges on firms' capacity for innovation, as discussed by Acemoglu et al. (2006) and König et al. (2022). While adopting existing technologies – referred to as imitation – can catalyze short-term growth, sustained advancement depends on firms embracing an innovation-centered strategy. Therefore, even though the current infant industry literature does not differentiate between imitation and innovation,[40] it is important to examine whether domestic firms are merely imitating existing technologies or genuinely innovating to fully understand the long-term implications for the blockages.

Measurement for Originality – To address this question, I measure the originality of the new features that Chinese apps are adding to their applications. In particular, I construct a similarity score based on the update logs of both blocked foreign apps and domestic apps. These logs textually reflect the functionalities of libraries newly added to an app's package, allowing me to assess the similarity between the technologies used in domestic and foreign apps.

Specifically, let $\mathcal{F}$ be the set of blocked foreign apps and $\mathcal{C}$ be the set of all Chinese apps. For each foreign-domestic app pair $(F, C) \in \mathcal{F} \times \mathcal{C}$, I denote the set of update logs published by the foreign app F in month t as $\mathcal{L}_t^F = \{l_{t1}^F, l_{t2}^F, \ldots, l_{tn}^F\}$. The set of update logs published by the Chinese app C in month t is denoted by $\mathcal{L}_t^C = \{l_{t1}^C, l_{t2}^C, \ldots, l_{tn}^C\}$.

For example, Google in December 2018 ($\mathcal{L}_t^{Google}, t = 2018/12$) has two update logs:

> $l_{t,1}^{Google}$: *Google Lens: **Search what you see and get stuff done using your camera.** Scan, search, and translate text, find clothing and products, identify plants and animals, and more. Tap the Lens icon in the search box to get started.*

> $l_{t,2}^{Google}$: *Collections: Keep track of the content you've visited and get back to it later...*

Baidu, a substitute app for Google in China, has six update logs from February 2021 ($\mathcal{L}_{t'}^{Baidu}, t' = 2021/02$), including:

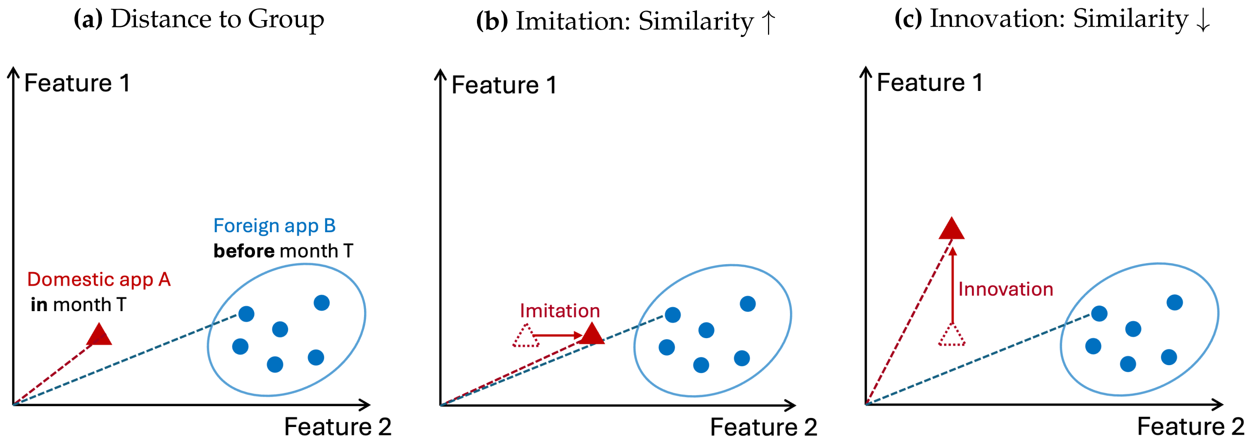
> $l_{t',1}^{Baidu}$: ***Take photos to recognize everything**: identify flowers in spring.*

I then convert the textual update logs (l^F or l^C) into word embeddings ($\mathbf{v}^F$ or $\mathbf{v}^C$) using Term Frequency-Inverse Document Frequency (TF-IDF), a statistical measure commonly used to assess the importance of a word within a document relative to a corpus. Intuitively, each update log is transformed into a vector where each dimension represents a unique word across all update logs, capturing distinct app features.[41] The vector values

[40] Juhász (2018) examines technology adoption, while Lane (2022) examines TFP in general.

[41] When preprocessing these logs for TF-IDF calculations, common stopwords and frequent but less informative phrases like 'fixing bugs' are removed to enhance the semantic relevance of the data. This preprocessing step helps to focus on more meaningful terms specific to app updates.

Figure 8: Illustration for Similarity Measurement



Notes: These figures illustrate how similarity between update logs is measured using cosine similarity (Panel a) and show changes in the magnitude of the similarity score when a domestic app imitates existing foreign features (Panel b) or innovates new features (Panel c).

reflect the importance of each feature in the log, increasing with the frequency of a word in the document and decreasing based on the word's prevalence in the entire corpus. Mathematically, for a feature i, its TF-IDF score is:

$$\text{TF-IDF}(i, l, \mathcal{D}) = \text{tf}(i, l) \times \text{idf}(i, \mathcal{D}),$$

where $\text{tf}(i, l)$ is the frequency of word i in update log l, and $\text{idf}(i, \mathcal{D})$ is the log of one over the share of update logs containing i in all update logs $\mathcal{D} = \left(\cup_t \mathcal{L}^C(t)\right) \cup \left(\cup_t \mathcal{L}^F(t)\right)$.

The similarity between a pair of update logs (l^F, l^C) is then calculated as the cosine similarity of their vectors, defined as

$$s(l^F, l^C) = \frac{\mathbf{v}^F \cdot \mathbf{v}^C}{\|\mathbf{v}^F\| \|\mathbf{v}^C\|}.$$

A higher score (up to 1) indicates greater functional similarity and alignment in technology use between the update logs, while a lower score (down to -1) suggests diverging functionalities, reflecting technological differentiation. For example, the similarity between $l_{t,1}^{Google}$ and $l_{t',1}^{Baidu}$ is $s(l_{t,1}^{Google}, l_{t',1}^{Baidu}) = 0.42$, which is higher than the similarity between $l_{t,2}^{Google}$ and $l_{t',1}^{Baidu}$, where $s(l_{t,2}^{Google}, l_{t',1}^{Baidu}) = 0.16$. This indicates closer functional alignment in the first pair.

Using this similarity measurement, each update log published by the Chinese app in month t can be compared to all the update logs published by the foreign app over the past twelve months, including month t. As illustrated in Panel (a) of Figure A.1, the similarity with a single domestic app's update log l^C to a set of foreign update logs is defined as the maximum cosine similarity[42] across all pairs in $\{l^C\} \times \mathcal{L}^F_{\leq t}$, where $\mathcal{L}^F_{\leq t} = \bigcup_{t-12 \leq k \leq t} \mathcal{L}^F(k)$:

$$S(\mathcal{L}^F_{\leq t}, l^C_t) = \max\{s(l^F, l^C) | l^F \in \mathcal{L}^F_{\leq t}\},$$

This approach uses existing technologies in foreign apps as a benchmark, allowing us to capture directional changes in domestic apps' update logs to distinguish between imitation and innovation behavior. As shown in Panel (b) of Figure A.1, when a domestic app *imitates* existing features of foreign apps (specifically, Feature 2 in the figure), the similarity $S(\mathcal{L}^F_{\leq t}, l^C_t)$ becomes *larger*. Conversely, if the domestic app begins to *innovate*, moving in the direction of Feature 1, as illustrated in Panel (c) of Figure A.1, the similarity $S(\mathcal{L}^F_{\leq t}, l^C_t)$ *decreases*, signaling a divergence from the original technological path.

Finally, since each domestic app C can have multiple update logs, denoted by $\mathcal{L}^C_t = \{l^C_{t1}, l^C_{t2}, \ldots, l^C_{tn}\}$ in month t, I define the average similarity score for a foreign-domestic app pair $p = (F, C)$ in month t to be:

$$Similarity_{pt} = \frac{1}{|\mathcal{L}^C_t|} \sum_{l^C \in \mathcal{L}^C_t} S(\mathcal{L}^F_{\leq t}, l^C). \tag{4}$$

The Effect of Blockage on Originality – For all domestic apps with at least one foreign substitute blocked, I investigate how the update trajectories of foreign-domestic app pairs classified as substitutes differ following the foreign app's blockage. Specifically, I examine whether these pairs exhibit changes in the direction of their updates, as reflected by the similarity of their update logs, in comparison to substitutable pairs that have not yet been blocked. To do this, I estimate the following empirical model:

$$\begin{aligned} Similarity_{pt} = & \sum_{l \neq -1} \beta_l Blocked^l_{pt} \times Substitute_p \\ & + \xi_{Ft} + \psi_{Ct} + \delta_p + u_{pt} \end{aligned} \tag{5}$$

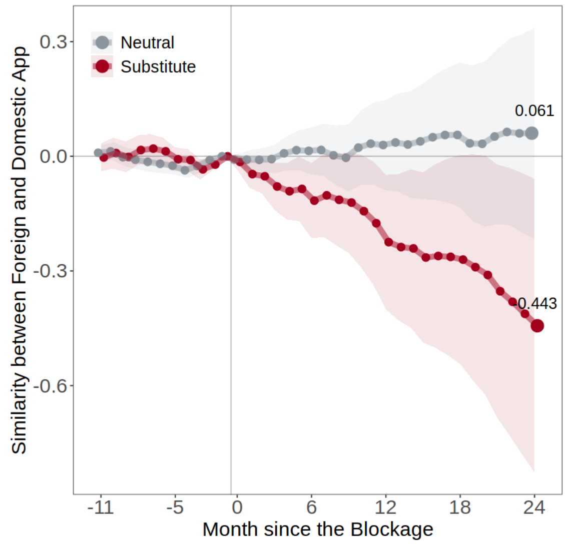
where $Similarity_{pt}$ is the similarity of foreign-domestic app pair $p = (F, C)$ in month t, calculated based on Equation 4. $Blocked^l_{pt}$ is a dummy indicating the app pair p is l month away from the blockage of the foreign app F in p occurring at time t. $Substitute_p$ indicates the foreign F and domestic C in app pair p are substitutes when equals to one, neutrals when equals to zero.

ξ_{Ft} denotes foreign app-month fixed effects, ensuring that similarity score comparisons are made between pairs sharing the same foreign app (e.g., between pairs such as $(Google, Baidu)$ versus $(Google, TikTok)$), rather than across foreign apps (e.g., $(Google, Baidu)$

[42]This is equivalent to defining the distance between an update l^C_t and its benchmark set $\mathcal{L}^F_{\leq t}$ as the minimum cosine distance (1- cosine similarity) from l^C_t to any point in $\mathcal{L}^F_{\leq t}$, following the convention in topology.

Figure 9: The Effect of Blockage on Similarity between Update Logs

(a) Average Similarity Score



Notes: These figures illustrate the impact of blockages on the similarity scores between the update logs of foreign-domestic app pairs. The red dots and lines represent the point estimates of β_l from Equation 5, with the shaded areas indicating the corresponding 95% confidence intervals. As a placebo test, the gray dots and lines show the results when blockages of substitutes in Equation 5 are replaced with neutral blockages. These gray markers provide the point estimates of β_l, allowing us to assess the robustness of our findings by comparing them to scenarios where no substantive blockage effect is expected.

versus $(Instagram, TikTok)$). ψ_{Ct} denotes domestic app-month fixed effects, which control for heterogeneous time trends among domestic apps (such as differential changes in update frequency). u_{pt} is the error term.[43]

I present the results in Figure 9. Panel (a) demonstrates the impact of foreign substitute blockages on the average similarity score, $Similarity_{pt}$, with the changes depicted by red lines and dots based on the coefficients β_l from Equation 5. Following the blockage, there is a notable decline in $Similarity_{pt}$, suggesting that domestic apps rapidly diverged from the existing features of their foreign counterparts, developing distinct and innovative features. This shift towards uniqueness aligns with findings from König et al. (2022), which indicate that high-TFP firms are more likely to pursue innovation rather than im-

[43]For apps that update less frequently, the similarity scores will be missing for most months. To ensure that the results are not driven by the frequency of app updates, which could bias the estimation due to the correlation between treatment, update frequency, and update content, I only include apps that have been updated for more than 30 months within a 3-year period to ensure the balance of this sample. Figure A.17 Panel (b) shows the distribution of the number of observations across event time. Additionally, to address carry-over effects (Liu et al., 2024), given that all domestic apps might experience multiple treatments in my setting, I only retain app pairs where the previous treatment occurred more than 12 months prior. Figure A.17 Panels (b) and (c) show average maximum and minimum similarity scores.

itation.[44] The evidence of developing unique features over time not only suggests that domestic apps are innovating more but also indicates an dynamic improvement in TFP.

Gray lines and dots in Figure 9 (a) shows the estimates using Equation 5 but replace $Substitute_p$ with $Neutral_p$, the indicator for whether F and C in pair p are neutrals. Since there is no significant effect from blockages of foreign neutrals, this suggests that the observed results are not driven by spurious correlations between foreign and domestic apps. Additionally, this allows us to distinguish the effects of blockages on substitute pairs that operate through direct competitive dynamics from effects that might arise in neutral pairs due to general market trends or external shocks.

Another validation exercise to assess whether the similarity score captures imitation or innovation behavior involves comparing Chinese update logs not with existing versions of foreign apps but with their *future versions*, which Chinese apps could not have observed prior to their monthly updates.[45] Panel (a) in Figure A.17 presents the results, showing no significant effects for either blockages of substitutes or neutrals. This indicates that the observed decrease in similarity scores in Figure 9 are not due to Chinese apps imitating future foreign app features – which they could not have known – but rather suggest that the initial reductions in similarity are attributable to more genuine innovation following the blockages.

5 Mechanism: The Economy of Data

Infant industry theory supports the growth of new industries by providing temporary protection, which helps them achieve economies of scale (Krugman and Maurice, 2003). In the modern tech industry, data is now acknowledged as a critical input in the production function of firms. By protecting infant industries and allowing them to amass and utilize data, governments can enable these firms to achieve competitive scale, thereby driving innovation and competitiveness in the global market.

Data's role as a novel yet pivotal input in the digital economy is significant. Data can

[44] As outlined in Proposition 1 of König et al. (2022), the model explains how firms, based on a productivity threshold and varying levels of productivity distributed randomly, decide between imitation and innovation. This decision-making process fosters a progression in firm productivity, whereby firms above a certain level increasingly opt for innovation to further their growth, whereas less productive firms may resort to imitation as a catch-up strategy.

[45] Specifically, I define the similarity between a single domestic app's update log l^C and a set of foreign update logs as the maximum cosine similarity across all pairs in $\{l^C\} \times \mathcal{L}^F_{>t}$, where $\mathcal{L}^F_{>t} = \bigcup_{t+1 \leq k \leq t+12} \mathcal{L}^F(k)$:

$$S(\mathcal{L}^F_{>t}, l^C_t) = \max\{d(l^F, l^C) | l^F \in \mathcal{L}^F_{>t}\},$$

directly contribute to productivity, where its accumulation and utilization reduce forecast error (Farboodi, Mihet, Philippon and Veldkamp, 2019; Bajari, Chernozhukov, Hortaçsu and Suzuki, 2019) or enhance a firm's quality of ideas (Jones and Tonetti, 2020), ultimately leading to internal economies. Moreover, since data is nonrival—meaning it can be used by multiple entities simultaneously without being depleted—it facilitates external economies through inter-firm data sharing, amplifying collective technological advancement (Jones and Tonetti, 2020).

I provide three pieces of evidence to show that the expansion of data scale as one of the mechanisms that drives the effect of blockage on innovation: *(i)* data collection efforts significantly expand following the blockage; *(ii)* the resulting increase in user data fosters more internal technological development; and *(iii)* the blockage generates positive spillover effects on other apps via data-sharing networks.

5.1 Expansion in Data Collection and Sharing

In the first step, I show that Chinese apps collect more data from users after the blockage of foreign substitutes. Data collection means that apps transmit data off the device in a manner that allows the app and/or its third-party partners to access it for longer than necessary to fulfill the immediate request, which is also the official definition from Apple.

To track the historical data collection practices of Chinese apps, I extract keys from the `Info.plist` (information property list) file of each app version. As mandated by Apple's privacy policy, developers must disclose their data collection practices during the app submission process by declaring all necessary keys—specific items that define various properties and behaviors of an app—in the `Info.plist` file. For instance, if an app requests access to the camera and location, it must include keys such as `NSCameraUsageDescription` and `NSLocationWhenInUseUsageDescription` in its `Info.plist` file. These practices are further verified by Apple during the app review process.

I then classify[46] the extracted `Info.plist` keys into four privacy groups based on the sensitivity of the data and the scope of data sharing, in accordance with Apple's Privacy

[46] I use Lasso to identify the keys that most significantly contribute to apps' privacy labels, as detailed in Appendix C. I did not rely on (1) the Apple privacy labels published on the iOS Store for each app version, or (2) the official documentation for key usage (`https://developer.apple.com/documentation/bundleresources/information_property_list`). Regarding (1), since Apple's privacy label policy was introduced in December 2020, privacy information is unavailable for app versions published before this time, when many blockages in my sample occurred. As for (2), while the official documentation specifies the types of user data a particular key can access, there isn't always a direct one-to-one mapping from keys to privacy labels. As shown in Figure A.9, some keys do not fully align with a single privacy label. Additionally, some keys originate from third-party companies, not Apple, which may also access sensitive user information.

Labels introduced in December 2020:

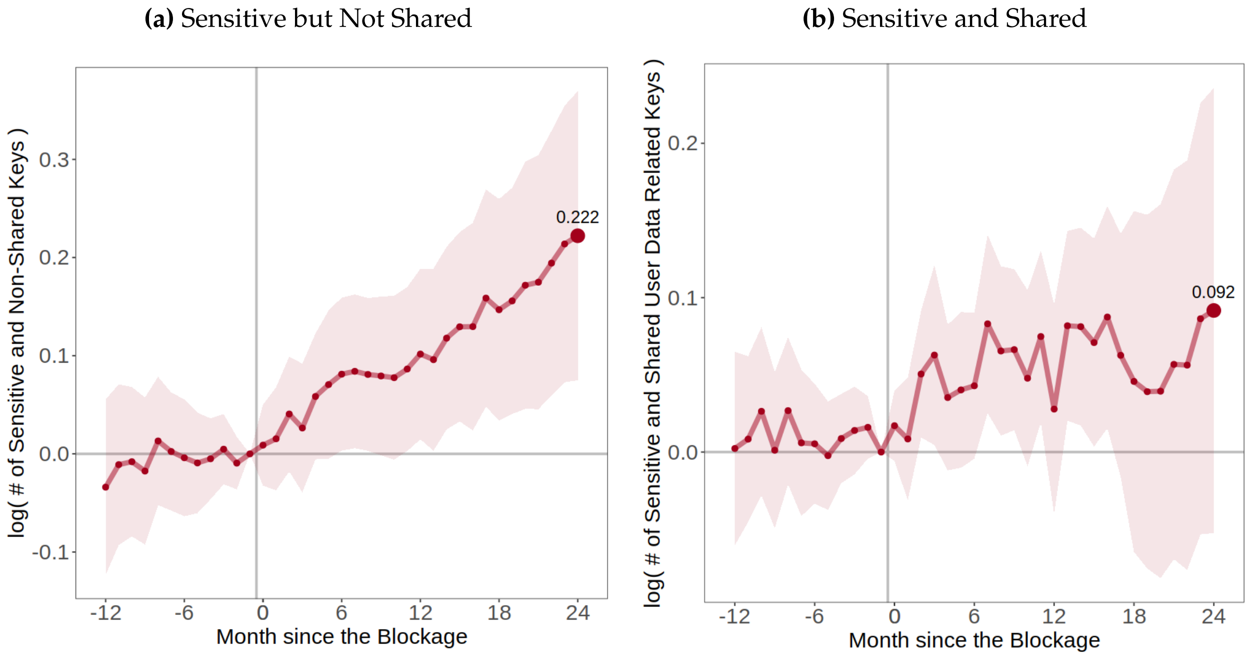
- **Sensitive and Shared**: This corresponds to Apple's privacy label *"Data Used to Track You"*. It includes data that is sensitive and used to track users across different apps and websites. This can include both data shared within a company (like between Instagram and Facebook, both owned by Meta) and data shared with third parties outside the company.
- **Sensitive but Not Shared**: This corresponds to Apple's privacy label *"Data Linked to You"*. It includes sensitive data that is linked to the user's identity but is not shared with third parties.
- **Non-sensitive**: This corresponds to Apple's privacy label *"Data Not Linked to You"*. It includes data that is collected in an anonymized or aggregated form and is not linked to the user's identity.
- **Non User Data Related**: This group includes keys that handle app functionalities, such as technical configurations or interface settings, without accessing or processing user data.

After aggregating the key-privacy label data to the app-month level, I analyze the number of keys under different privacy groups for each app-month. Figure A.18 presents the average number of keys per app by privacy label over time. Panel (a) displays the average trend for Chinese apps, while Panels (b) and (c) show the trends for U.S. and European apps, respectively. There are two key observations: (1) Over the ten years, the average number of sensitive `Info.plist` keys related to user data within an app has increased significantly from 5 to 25. (2) Unlike in the U.S. and European markets, the majority of sensitive user data in Chinese apps is shared with third parties, as indicated by the higher proportion of sensitive and shared keys.

Using the number of keys under different privacy groups as the outcome, I re-estimate Equation 2 to investigate how the blockage of foreign substitutes influences Chinese apps' data collection practices. To address concerns that changes in data collection practices might be mechanical due to apps introducing more functionalities, I control for the total number of updates that apps make monthly.

In Figure 10, I present two types of results: Figure 10 Panel (a) shows that, given the same number of app updates, the number of sensitive but not shared keys gradually began to increase following the blockage of foreign substitutes, reaching a 21.9% increase after two years on average (corresponding to about 0.4 `Info.plist` key). This strongly suggests that the blockage leads to Chinese apps collecting more sensitive user

Figure 10: The Effect of Blockage on Data Collection



Notes: These figures show the effect of blockage on the quantity of data collected from users. The red dots and lines shows the point estimates of β_l in Equation 2 with the total number of updates that apps make monthly controlled. The bandwidth in grey shows the corresponding 95% confidence intervals of β_l. In Panel (a), we show the result of the number of `Info.plist` keys that are classified as sensitive but not shared. In Panel (b), we show the estimates using the number of sensitive and shared keys as the outcome variable.

data within the app for each user. Given Apple's stringent privacy policies compared to Android markets, this increase represents a conservative estimate of the intrusion into user privacy. If I further consider the 23% increase in monthly active users (Figure 5, Panel (a)), the effect of the blockage on the total amount of data that an app could access is likely even more significant, potentially resulting in a 50% increase in overall data collection.

Similarly, in Figure 10 Panel (b), the number of sensitive and shared keys also increased by about 9% (about 0.63 key) after the blockage. Combining this with the fact that Chinese apps are installing more domestic third-party libraries instead of foreign ones (Figure 6), it implies a data spillover effect—Chinese apps are not only relying more on domestic technologies but are also enhancing their data collection capabilities through these local integrations.

The Scale and Scope of Data – The total data volume collected by an app depends on two dimensions: the number of users (the data scale) and the variety of data accessed per user (data scope). Previous evidence indicates that the blockage of foreign substitutes has influenced both scale (Section 4.1) and scope (Section 5.1).

What is the correlation between an app's scale and its data scope (i.e. types of user data collected by the app)? This relationship can vary – it might be negative if users are particularly privacy-conscious, or positive if a larger user base enables more extensive data collection. To investigate this dynamic, I analyzed how apps whose foreign substitutes have never been blocked have adjusted their `Info.plist` keys with growth. Figure A.20 demonstrates the correlation between an app's monthly active users and the number of sensitive `Info.plist` keys implemented each month, showing a minimal correlation. After controlling for app and time heterogeneity, the correlation between monthly active users (per one hundred million users) and the number of sensitive `Info.plist` keys is estimated to be -3.853, which is not statistically significant (see Table A.3).

This minimal correlation implies that the blockage of foreign substitutes affects data collection practices through factors other than the scale of apps. Economies of scale may not significantly improve the capacity to collect more data; as apps grow, they might not necessarily expand the variety of data they collect, possibly due to fixed costs or operational constraints. More plausibly, the absence of foreign competition bolsters the market power of domestic apps, enabling them to adopt more aggressive data collection strategies regardless of user growth. Alternatively, users might become more captive to the remaining apps, thereby giving these apps greater leeway to enhance data collection efforts. Chen and Yang (2019)

5.2 Return of Data on Innovation

The preceding analysis provides direct evidence that the scale of data owned by each Chinese app increases after the blockage of its foreign substitutes, both vertically—through an increase in the total number of users and, consequently, the sample size—and horizontally, with a notable expansion in the information observed per user. The next question is how the increase in data contribute to innovation.

Estimating the return of data on innovation has been challenging due to the endogeneity concerns that arise when regressing measures of innovation on the amount of data collected by an app. First, the relationship between data collection and innovation is likely endogenous. For example, lower-quality apps that focus primarily on harvesting user data may exhibit less innovation, distorting the perceived effectiveness of data collection in driving innovation. Second, the causality between data collection and innovation may be reversed. An app's decision to request more user data may be a response to innovations already implemented.

The endogeneity concerns can affect the estimated coefficient when regressing innovation on data amount. I construct instrumental variable to address the two concerns.

IV: The Introduction of New Access to User Data – The instrumental variable explores the quasi-random variation in the user data that apps can access, which arises from the introduction of new data access capabilities through iOS upgrades.

iOS developer system updates periodically introduce new developer tools and features, which are often reflected in new `Info.plist` keys. These keys can significantly enhance[47] or reduce[48] an app's ability to collect user data across various data types and privacy levels. When an app already possesses the necessary functionalities to handle data of a similar type and has the infrastructure in place for data processing and storage, the introduction of new `Info.plist` keys thus function as natural experiments that exogenously nudge apps to acquire more user data without requiring the addition of new functionalities.[49]

To formalize the IV, consider an app i at time $t-1$. The set of `Info.plist` keys it posses at time $t-1$ is denoted as $\{k_{i,t-1}\}$. We denote the data type of $k_{i,t-1}$ as $d(k_{i,t-1})$ and $s(k_{i,t-1})$ as its privacy level defined in Section 5.1. At time t, a new collection of `Info.plist` keys $\{k_t^{new}\}$ are introduced.

Given a k_t^{new}, if the app already had the necessary functionalities to handle data of a similar type, i.e. $n(\{k_{i,t-1} : d(k_{i,t-1}) = d(k_t^{new})\}) > 0$, then the app is more likely to adopt this key in the following month.

According to whether the introduction of new `Info.plist` keys is enhancing or reducing an app's ability to collect user data, I construct two instrumental variables. The first IV, $DataEnhancingKeys_{it}$, totals the new keys an app is likely to adopt that could potentially enhance its data collection capabilities. It is defined as follows:

[47] For example, the introduction of `UIBackgroundModes` in 2010 marked a significant shift in data collection capabilities. Prior to this update, apps had very limited ability to operate in the background, as they were generally suspended when not in active use. The `UIBackgroundModes` key allowed developers to request permission for their apps to run specific tasks in the background, enabling continuous data collection, such as tracking location updates or gathering sensor data, even when the app was not actively being used.

[48] Before September 2020, user consent for apps to access the photo library was binary: users could either grant full access or deny it completely. If granted, the app could access all photos, providing developers extensive data about the user's photo content. In September 2020, Apple enhanced user privacy by introducing the option for users to grant "Limited" access through a new feature (`PHPhotoLibraryPreventAutomaticLimitedAccessAlert`). This update allows users to select specific photos or albums they wish to share with the app, significantly reducing the amount of personal data shared.

[49] In December 2016, Apple introduced the `NSLocationAlwaysAndWhenInUseUsageDescription` key, allowing developers to request both "When In Use" and "Always" location access in a single prompt. This streamlined the process, making it easier to obtain continuous background location data, thereby increasing the amount of user data that apps could collect.

$$DataEnhancingKeys_{it} = \sum_{k_t^{new}} \Bigg(\underbrace{\mathbb{1}\left\{n(\{k_{i,t-1} : d(k_{i,t-1}) = d(k_t^{new})\}) > 0\right\}}_{(1)} \times \\ \underbrace{\mathbb{1}\left\{s(k_t^{new}) \geq \min\{s(k_{i,t-1}) : d(k_{i,t-1}) = d(k_t^{new})\}\right\}}_{(2)} \Bigg) \tag{6}$$

In this equation, part (1) confirms that app i must already manage data of the same type as the newly introduced key k_t^{new}. Part (2) establishes that k_t^{new} is considered to enhance data collection capabilities for app i at time t if its privacy level is at least as restrictive as the least sensitive data currently managed by the app under the same data type $d(k_t^{new})$.

The second IV, $DataReducingKeys_{it}$, calculates the total number of new keys that could potentially reduce the app's data collection capabilities, introduced at the same time t . It is defined as follows:

$$DataReducingKeys_{it} = \sum_{k_t^{new}} \Bigg(\mathbb{1}\left\{n(\{k_{i,t-1} : d(k_{i,t-1}) = d(k_t^{new})\}) > 0\right\} \times \\ \underbrace{\mathbb{1}\left\{s(k_t^{new}) < \min\{s(k_{i,t-1}) : d(k_{i,t-1}) = d(k_t^{new})\}\right\}}_{(3)} \Bigg) \tag{7}$$

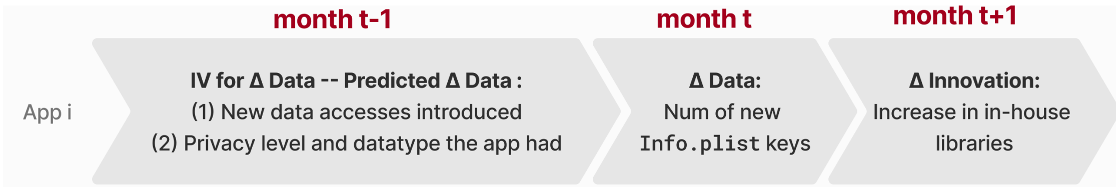
where (3) suggests that k_t^{new} is considered as data-reducing for app i if the privacy level of k_t^{new} is less sensitive than any of the data the app i already manages under the same data type $d(k_t^{new})$.

For both $DataEnhancingKeys_{it}$ and $DataReducingKeys_{it}$, the core variation of the instrumental variables derives from the quasi-random introduction of `Info.plist` keys, differing across time, data type, and privacy level. Figure A.21 visually represents the timeline and diversity in the data type and privacy level of these `Info.plist` keys, illustrating the underlying variability used for the IV estimation.

Results: Return of Data on Innovation – To estimate the return of data on innovation, we ask whether an app that gains one more access to data (i.e., one additional `Info.plist` key) in month t will see a change in how many numbers of in-house libraries it uses in the following period (month $t+1$), compared to apps that do not have the extra data access. The number of new `Info.plist` keys adopted by app i in month t is instrumented by the number of `Info.plist` keys introduced by the iOS developer system in the previous month $t-1$.

Figure 11 illustrates the timeline for variables in the regression, which clarify the temporal relationship between the introduction of new keys, the adoption of the new keys in relevant apps, and the subsequent impact on the number of in-house libraries.

Figure 11: Timeline of Instrumental Variable



Specifically, I estimate the following first-difference equation:

$$\Delta Innovation_{i,t+1} = \eta \Delta Data_{i,t} + \gamma_{f(i),t+1} + \gamma_{d(i,t-1)} + X_{i,t}^{T}\xi + u_{i,t+1} \tag{8}$$

where the independent variable is the measurement for innovation – the number of in-house libraries. The dependent variable is the number of new `Info.plist` keys that are introduced in app i in month t, in other words, the number of keys that do not appear in the `Info.plists` of app i before t. The instrumental variable is the predicted increase in the number of `Info.plist` keys, as constructed above. I include (1) firm and month fixed effects to adjust for any firm-level temporal shocks that might influence apps' data access choices, including internal spillover effects, and (2) data type fixed effects to account for differential impacts various types of data may have on an app's data adoption patterns.

Furthermore, since both the independent variable, $\Delta Data_{it}$, and the instruments, *Data-EnhancingKeys*$_{it}$ and *DataReducingKeys*$_{it}$, rely on apps' existing distribution , I follow the intuition from Borusyak and Hull (2023) and construct average number of data-enhancing and data-reducing keys based on data introduction counterfactuals. This method involves permutating the introduction dates of `Info.plist` keys to simulate various scenarios. Given each random introduction of `Info.plist` keys, I calculate a counterfactual number of data-enhancing and data-reducing keys for each app. These simulations yield an average number of data-enhancing keys, denoted as $AvgDataEnhancingKeys_{it}$, and data-reducing keys, denoted as $AvgDataReducingKeys_{it}$, by averaging the outcomes across all permutations. Controlling for average number of data-enhancing and data-reducing keys helps recenter the independent variable and extract the exogenous variation from the introduction of new access to user data.[50]

[50]Since we are using shift-share style, BH controls help with the share part (i.e. Part (1) in Equation 6). For the shift part, although it also depends on the pre-existing conditions of apps, since it's unlikely that the upgrades in iOS system will be set for specific apps, $s(k_t^{new})$ is exogenous to $s(k_{i,t-1})$.

Table 1: Effects of Data on the Quantity of Technologies

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
	OLS:		First Stage:			IV:	
	Δ Num of Libraries		New `Info.plist` Keys:		Δ Num of	Δ Num of Libraries	
	In-House	Third-Party	User Data	Non-User Data	iOS Libraries	In-House	Third-Party
Instruments:							
Num. of Data-Enhancing Keys			0.016***	-0.0005***	0.147		
			(0.004)	(0.0002)	(0.193)		
Num. of Data-Reducing Keys			-0.043***	0.0002	-0.653		
			(0.008)	(0.0004)	(0.637)		
Endogenous Var $\Delta Data$:							
Num. of New User-Data Keys	2.044***	0.779***				4.713***	0.892
	(0.049)	(0.035)				(1.166)	(0.575)
Firm $\times$ Month FE	Y	Y	Y	Y	Y	Y	Y
Data Type FE	Y	Y	Y	Y	Y	Y	Y
BH Control			Y	Y	Y	Y	Y
Ave. Dep. Var.	17.499	10.208				17.499	10.208
R^2	0.415	0.383	0.092	0.047	0.260		
Observations	22,449	22,449	22,449	22,449	13,249	22,449	22,449
F-stats			28.969	14.454			

Notes: This Table shows regressions of the change in the quantity of technologies on the change in data. I report the OLS results in columns (1)-(2), where the independent variable is the number of new `Info.plist` keys that are related to user data. I report first-stage in column (3)-(5) and IV results in (6)-(7). The instrumental variable is the number of data-enhancing `Info.plist` keys defined by Equation 6 and the number of data-reducing keys defined by Equation 7. Dependent variables are (1) the number of in-house libraries, as measured by the number of libraries that are developed by the parent company of the app, (2) the number of third-party libraries that libraries developed by neither parent company nor Apple. Standard errors are clustered at the app level.

Table 1 presents the IV results. First, Column 3-5 reports the first-stage results. Column 3 shows that the instrumental variables $DataEnhancingKeys_{it}$ and $DataReducingKeys_{it}$ are significantly relevant to the endogenous variable $\Delta Data_{it}$. A new `Info.plist` key enhancing data collection correlates with a 0.016 increase in new user data-related keys, whereas each data-reducing key correlates with a 0.043 decrease in these keys.

Second, a key concern with this method is the exclusion restriction: the introduction of new `Info.plist` keys may coincide with other upgrades in the iOS developer system, potentially influencing the increase in in-house libraries. In Columns 4 and 5, I examine the correlation between the two instrumental variables and the number of new non-user data related keys (Column 4) and the changes in the number of iOS libraries (Column 5). I find little impact of the two IVs on new non-user data related keys and changes in iOS libraries, which suggests that these IVs are unlikely to be confounded by other concurrent system upgrades.

Column 6-7 presents the IV results. They indicate that apps with one additional access to user data (represented by one more `Info.plist` key that is related to user data) tends to develop more in-house libraries. Conversely, the additional user data does not significantly affect third-party libraries. In terms of magnitude, compared to the average number of in-house libraries, the presence of one additional access to user data corresponds to a 40% increase in self-developed libraries.

5.3 Spillover through Data-Sharing Network

The observed increase in data collection following the blockage, combined with data's positive impact on innovation, offers strong evidence of a direct effect of the blockage on within-app innovation through enhanced data access. Additionally, Section 5.1 shows a 9% increase in user data shared with third parties post-blockage, indicating the expansion of data-sharing networks. Since data is nonrivalrous—meaning it can be used by multiple entities simultaneously without depletion (Jones and Tonetti, 2020)—spillover effects are likely to manifest in other apps connected through these data-sharing linkages.

To test this hypothesis, I construct a data-sharing network by analyzing data transfers between apps and third parties over insecure networks. With the established link between data providers and data receivers, I then examine how blockages of foreign substitutes for data providers affect the innovation performance of data receivers, thereby assessing the spillover effects through the data-sharing network.

Detect Data-Sharing Network – Data sharing between firms can be challenging to systematically detect. However, in the setting of apps, a specific form of data sharing practices can be inferred by examining the network security settings found in the `Info.plist` file of an app. Since September 2015, Apple's App Transport Security (ATS) framework has mandated that all app network connections utilize HTTPS, ensuring user data is encrypted during transmission, as opposed to the less secure HTTP. This security requirement is enforced through the `NSAppTransportSecurity` key. Apps can, however, specify a list of domains permitted to bypass these security protocols and allow for the insecure transfer of data[51]. By further linking the specified domains to their corresponding companies, one could establish a data-sharing network through HTTP between an app p – the data provider – and a third-party company r – the data receiver.

However, this method may bias sample selection as companies that continue to use HTTP may be less technologically advanced or productive.[52] This concern is particularly mitigated in China before 2017 due to the low adoption rate of HTTPS in China, largely because the GFW more effectively monitors and controls unencrypted traffic. Firms using HTTPS faced the risk that their domains could be completely blocked, presenting a significant barrier to HTTPS adoption by Chinese firms until the GFW became more sophisticated post-2017.[53]

Spillover Effect of Blockage through Data-Sharing Network – Between 2015 and 2017, I identified 2,989 Chinese apps that used HTTP for data communication, corresponding to interactions with 536 domestic companies. The network helps me to know whether app p is sharing user data with firm r at time t.

For all apps belonging to data receiver firms, I investigate how the blockage of a data provider app p's foreign substitutes influence data receiver firms' app i through the data-sharing between app p and firm $r(i)$. I compare outcomes of app i before and after the blockage of foreign substitutes of app p who share data with app i's firm $r(i)$. The corresponding regression function is as follows:

$$Y_{it} = \exp\left(\sum_{l \neq -1} \beta_l DataShare_{p \rightarrow r(i)} \times SubstituteBlocked^l_{pt} + \psi_{g(i)t} + \delta_i + \gamma_p + X^T_{it}\xi\right) U_{it} \tag{9}$$

where $SubstituteBlocked^l_{pt}$ is an indicator for domestic app p being l months away from

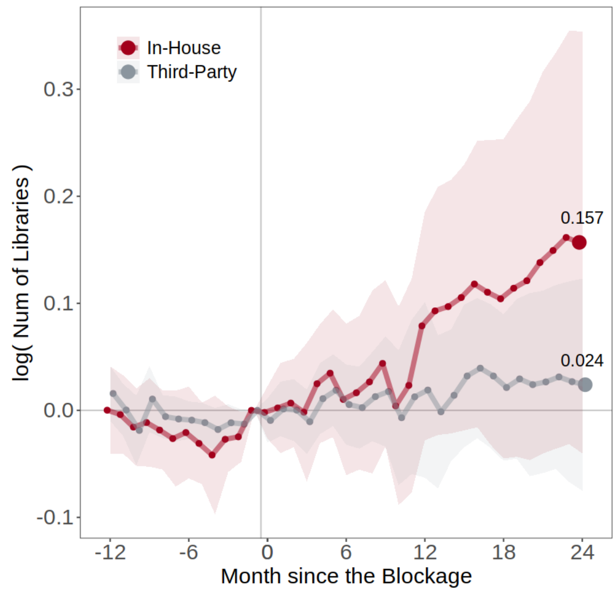
[51] Specifically, apps will include `NSExceptionDomains` and `NSThirdPartyExceptionAllowsInsecureHTTPLoads`. Figure A.22 provides a screenshot of the `Info.plist` of an app which specify a list of domains permitted to bypass these security protocols.

[52] There are several general reasons for apps to use HTTP for data transmission: (1) HTTP is less resource-intensive than HTTPS, making it attractive for apps prioritizing speed and bandwidth efficiency over security; (2) developers may opt for HTTP to maintain compatibility with legacy systems or older devices that do not fully support HTTPS; and (3) smaller or less advanced companies may lack the technical resources or expertise to implement HTTPS protocols effectively.

[53] Source: https://www.thesslstore.com/blog/https-google-china/

Figure 12: Spillover Effect of Blockage through Data-Sharing Network

(a) App Level: Number of Libraries



Notes: These figures show the effect of blockages of foreign substitutes for data providers on the performance of data receivers. In Panel (a), I show the result of number of in-house and third-party libraries. The red dots and lines shows the point estimates of β_l in Equation 9 . The bandwidth in grey shows the corresponding 95% confidence intervals of β_l.

the blockage of its foreign substitute at time t, and $DataShare_{p \to r(i)}$ indicates whether app p is sharing data with the app i's firm $r(i)$ before the blockage. I also include data provider app and data receiver app fixed effects, γ_p and δ_i, to control for inherent app characteristics. The category-month fixed effects, $\psi_{g(i)t}$, take care of app category and time heterogeneity.

Figure 12 presents the results. Following the blockage of data providers' foreign substitutes, there is an upward trend, although not significant due to the limited sample size, in the number of in-house libraries developed by data receiver firms.

Two placebo tests further validate the spillover effects on in-house libraries. First, instead of using the blockage of foreign substitutes of app p as the treatment, I use the blockage of foreign neutrals for the same app. I examine whether apps from companies sharing data with app p exhibit similar patterns in their in-house technology development before and after the blockage, as the blockage of substitutes do. Second, the impact on app i resulting from the blockage of foreign substitutes of app p is assessed *prior to* app p initiating data sharing with firm $r(i)$, the developer of app i. The results of these two placebo tests are presented in Figure A.23, showing negligible impacts, contrasting with the findings in Figure 12.

6 Assessing the Rationale for Industrial Policy

Providing a definitive justification for industrial policy, especially for protectionist policies, requires pinpointing the precise sources and magnitudes of agglomeration economies. This empirical challenge, widely recognized in the literature (Juhász, Lane and Rodrik, 2023), stems from the difficulty of determining whether the benefits are external or internal to firms. For example, in this paper, assessing whether data sharing between firms is contractual and at what price requires detailed data on firm interactions. Despite these challenges, the findings presented in this paper offer suggestive evidence for several mechanisms that justify protectionist policies aimed at correcting static externalities or fostering infant industries.

First, externalities suggested by the growth of an domestic ecosystem (Katz and Shapiro, 1985; Jacobides, Cennamo and Gawer, 2024) and data-sharing networks and the likelihood of knowledge spillovers support the optimality of protectionist policies (Krugman, 1987; Lucas, 1988; Matsuyama, 1992; Young, 1991). Evidence includes: (1) technology spillovers across geographical distances, as evidenced by higher adoption of Chinese libraries in domestic apps compared to other Asian countries and the rest of the world (Figures A.24 and 7), and (2) the expansion of a larger domestic technological ecosystem (Figure 6, Panel (b)) and domestic data-sharing networks (Figure 10, Panel (b)) following the blockage, where the network's value increases as more participants join.

Second, the expansion of data-sharing networks (Section 5.1) and the positive spillovers they generate (Section 5.3) indicate the presence of positive externalities. While the extent to which these externalities are internalized by firms remains uncertain in this paper, Jones and Tonetti (2020) highlights that assigning property rights over data to consumers could yield nearly optimal allocations when data is extensively utilized across firms, even considering privacy concerns; if data exchanges are governed by firms, the rationale for government intervention through protectionist measures may be weakened.

And lastly, dynamic learning from data (Section 5.2) – even if internalized by firms – may justify policy intervention if firms are credit-constrained or myopic. The high fixed costs associated with data learning, such as significant development expenses for internal A/B testing platforms[54] and substantial initial user acquisition costs[55] due to pronounced

[54]For a mid-sized company, initial development costs for an internal A/B testing platform can be substantial, ranging from 3 months for a basic tool to up to a year for more complex systems. Source: `http://alturl.com/m2czx`.

[55]Customer Acquisition Costs (CAC) have increased by 222% over the last decade, rising from \$9 to \$29 per user, and U.S. spending on user acquisition campaigns for shopping apps reached \$6.6 billion in 2023. Source: `http://alturl.com/mpbhf`.

network effects, pose barriers that protectionist policies could help mitigate. Pronounced network effects may strengthen the case for protectionist policies, as acquiring substantial market share from established incumbents can be prohibitively expensive without such interventions. Previous sections have demonstrated clear impacts on domestic firm growth, both within and outside of China, from the blockage of foreign apps. While not definitive, the arguments presented in this section suggest that such protectionist policies may be justified from an efficiency standpoint in this context.

References

Acemoglu, Daron, Philippe Aghion, and Fabrizio Zilibotti, "Distance to frontier, selection, and economic growth," *Journal of the European Economic association*, 2006, *4* (1), 37–74.

Akcigit, Ufuk and Marc Melitz, "International trade and innovation," in "Handbook of International Economics," Vol. 5, Elsevier, 2022, pp. 377–404.

_, Sina T Ates, and Giammario Impullitti, "Innovation and trade policy in a globalized world," Technical Report, National Bureau of Economic Research 2018.

Atkin, David, Amit K Khandelwal, and Adam Osman, "Exporting and firm performance: Evidence from a randomized experiment," *The quarterly journal of economics*, 2017, *132* (2), 551–615.

Autor, David, David Dorn, Gordon H Hanson, Gary Pisano, and Pian Shu, "Foreign competition and domestic innovation: Evidence from US patents," *American Economic Review: Insights*, 2020, *2* (3), 357–374.

Bai, Jie, Panle Jia Barwick, Shengmao Cao, and Shanjun Li, "Quid pro quo, knowledge spillover, and industrial quality upgrading: Evidence from the Chinese auto industry," Technical Report, National Bureau of Economic Research 2020.

Bajari, Patrick, Victor Chernozhukov, Ali Hortaçsu, and Junichi Suzuki, "The impact of big data on firm performance: An empirical investigation," in "AEA papers and proceedings," Vol. 109 American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203 2019, pp. 33–37.

Baldwin, Robert E, "The case against infant-industry tariff protection," *Journal of political economy*, 1969, *77* (3), 295–305.

Bartelme, Dominick G, Arnaud Costinot, Dave Donaldson, and Andres Rodriguez-Clare, "The textbook case for industrial policy: Theory meets data," Technical Report, National Bureau of Economic Research 2019.

Barwick, Panle Jia, Hyuk-Soo Kwon, Shanjun Li, Yucheng Wang, and Nahim B Zahur, "Industrial Policies and Innovation: Evidence from the Global Automobile Industry," Technical Report, National Bureau of Economic Research 2024.

_, Myrto Kalouptsidi, and Nahim Bin Zahur, "Industrial Policy Implementation: Empirical Evidence from China's Shipbuilding Industry," *Review of Economic Studies*, 2024.

Begenau, Juliane, Maryam Farboodi, and Laura Veldkamp, "Big data in finance and the growth of large firms," *Journal of Monetary Economics*, 2018, *97*, 71–87.

Benhabib, Jess, Jesse Perla, and Christopher Tonetti, "Reconciling models of diffusion and innovation: A theory of the productivity distribution and technology frontier," *Econometrica*, 2021, *89* (5), 2261–2301.

Beraja, Martin, David Y Yang, and Noam Yuchtman, "Data-intensive innovation and the State: evidence from AI firms in China," *The Review of Economic Studies*, 2023, *90* (4), 1701–1723.

Bergeaud, Antonin, Philippe Aghion, TimothÃ Gigout-Magiorani, Matthieu Lequien, Marc J Melitz et al., "Exporting Ideas: Knowledge Flows from Expanding Trade in Goods," Technical Report 2024.

Bloom, Nicholas, John Van Reenen, and Heidi Williams, "A toolkit of policies to promote innovation," *Journal of economic perspectives*, 2019, *33* (3), 163–184.

Borusyak, Kirill and Peter Hull, "Nonrandom exposure to exogenous shocks," *Econometrica*, 2023, *91* (6), 2155–2185.

Brand, James, Ayelet Israeli, and Donald Ngwe, "Using GPT for market research," *Harvard Business School Marketing Unit Working Paper*, 2023, (23-062).

Chen, Jiafeng and Jonathan Roth, "Logs with zeros? Some problems and solutions," *The Quarterly Journal of Economics*, 2024, *139* (2), 891–936.

Chen, Yuyu and David Y Yang, "The impact of media censorship: 1984 or brave new world?," *American Economic Review*, 2019, *109* (6), 2294–2332.

Choi, Jaedo and Andrei A Levchenko, "The long-term effects of industrial policy," Technical Report, National Bureau of Economic Research 2021.

Coelli, Federica, Andreas Moxnes, and Karen Helene Ulltveit-Moe, "Better, faster, stronger: Global innovation and trade liberalization," *Review of Economics and Statistics*, 2022, *104* (2), 205–216.

Cory, Nigel and Luke Dascoli, "How barriers to cross-border data flows are spreading globally, what they cost, and how to address them," Technical Report, Information Technology and Innovation Foundation 2021.

Dell, Melissa, "Deep Learning for Economists," Technical Report, National Bureau of Economic Research 2024.

Demirer, Mert, Diego J Jiménez Hernández, Dean Li, and Sida Peng, "Data, privacy laws and firm production: Evidence from the GDPR," Technical Report, National Bureau of Economic Research 2024.

Farboodi, Maryam, Roxana Mihet, Thomas Philippon, and Laura Veldkamp, "Big data and firm dynamics," in "AEA papers and proceedings," Vol. 109 American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203 2019, pp. 38–42.

Garg, Tishara, "Can Industrial Policy Overcome Coordination Failures? Theory and Evidence," *Working Paper*, November 2024.

Goldfarb, Avi and Catherine Tucker, "Digital economics," *Journal of economic literature*, 2019, *57* (1), 3–43.

Gorodnichenko, Yuriy, Jan Svejnar, and Katherine Terrell, "Globalization and innovation in emerging markets," *American Economic Journal: Macroeconomics*, 2010, *2* (2), 194–226.

Hausmann, Ricardo and Dani Rodrik, "Economic development as self-discovery," *Journal of development Economics*, 2003, *72* (2), 603–633.

Hirschman, Albert O, "The strategy of economic development," *Yale University*, 1958.

Hoang, Nguyen Phong, Arian Akhavan Niaki, Jakub Dalek, Jeffrey Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis, "How Great is the Great Firewall? Measuring China's {DNS} Censorship," in "30th USENIX Security Symposium (USENIX Security 21)" 2021, pp. 3381–3398.

Huang, Raffaele, "American Cloud Companies Face Challenge From China in Southeast Asia," *The Wall Street Journal*, 2023.

Jacobides, Michael G, Carmelo Cennamo, and Annabelle Gawer, "Externalities and complementarities in platforms and ecosystems: From structural solutions to endogenous failures," *Research Policy*, 2024, *53* (1), 104906.

Jin, Ginger Zhe, Ziqiao Liu, and Liad Wagman, "The GDPR and SDK Usage In Android Mobile Apps," *Law & Economics Center at George Mason University Scalia Law School Research Paper Series Forthcoming*, 2024.

Jones, Charles I and Christopher Tonetti, "Nonrivalry and the Economics of Data," *American Economic Review*, 2020, *110* (9), 2819–2858.

Juhász, Réka, "Temporary protection and technology adoption: Evidence from the napoleonic blockade," *American Economic Review*, 2018, *108* (11), 3339–3376.

_ , Nathan Lane, and Dani Rodrik, "The new economics of industrial policy," *Annual Review of Economics*, 2023, *16.*

Katz, Michael L and Carl Shapiro, "Network externalities, competition, and compatibility," *The American economic review*, 1985, *75* (3), 424–440.

Klosin, Sylvia, "Dynamic Biases of Static Panel Data Estimators," 2024. Job Market Paper, Massachusetts Institute of Technology.

König, Michael, Kjetil Storesletten, Zheng Song, and Fabrizio Zilibotti, "From imitation to innovation: Where is all that Chinese R&D going?," *Econometrica*, 2022, *90* (4), 1615–1654.

Krueger, Anne O, "Government failures in development," *Journal of Economic perspectives*, 1990, *4* (3), 9–23.

Krugman, Paul, "The narrow moving band, the Dutch disease, and the competitive consequences of Mrs. Thatcher: Notes on trade in the presence of dynamic scale economies," *Journal of development Economics*, 1987, *27* (1-2), 41–55.

— **and Obstfeld Maurice**, "International economics-theory and policy," 2003.

Lane, Nathan, "Manufacturing revolutions: Industrial policy and industrialization in South Korea," *Available at SSRN 3890311*, 2022.

Li, Peiyao, Noah Castelo, Zsolt Katona, and Miklos Sarvary, "Frontiers: Determining the validity of large language models for automated perceptual analysis," *Marketing Science*, 2024, *43* (2), 254–266.

Lileeva, Alla and Daniel Trefler, "Improved access to foreign markets raises plant-level productivity… for some plants," *The Quarterly journal of economics*, 2010, *125* (3), 1051–1099.

Liu, Ernest, "Industrial policies in production networks," *The Quarterly Journal of Economics*, 2019, *134* (4), 1883–1948.

Liu, Licheng, Ye Wang, and Yiqing Xu, "A practical guide to counterfactual estimators for causal inference with time-series cross-sectional data," *American Journal of Political Science*, 2024, *68* (1), 160–176.

Lucas, Robert E, "On the mechanics of economic development," *Journal of monetary economics*, 1988, *22* (1), 3–42.

Manelici, Isabela and Smaranda Pantea, "Industrial policy at work: Evidence from Romania's income tax break for workers in IT," *European Economic Review*, 2021, *133*, 103674.

Matsuyama, Kiminori, "Agricultural productivity, comparative advantage, and economic growth," *Journal of economic theory*, 1992, *58* (2), 317–334.

Melitz, Marc J, "The impact of trade on intra-industry reallocations and aggregate industry productivity," *econometrica*, 2003, *71* (6), 1695–1725.

— , "When and how should infant industries be protected?," *Journal of International Economics*, 2005, *66* (1), 177–196.

Rotemberg, Martin, "Equilibrium effects of firm subsidies," *American Economic Review*, 2019, *109* (10), 3475–3513.

Shangda, Sarah Wang Xu, "16 Changes to the Way Enterprises Are Building and Buying Generative AI," March 2024.

Stiglitz, Joseph E, Justin Yifu Lin, and Ebrahim Patel, *The industrial policy revolution I: The role of government beyond ideology*, Springer, 2013.

Sun, Ruiqi (Rachel), "Blocking the Giants: Theory and Evidence from the Great Firewall," November 2024. Job Market Paper, Rotman School of Management, University of Toronto.

Veldkamp, Laura and Cindy Chung, "Data and the aggregate economy," *Journal of Economic Literature*, 2024, *62* (2), 458–484.

Wooldridge, Jeffrey M, "Simple approaches to nonlinear difference-in-differences with panel data," *The Econometrics Journal*, 2023, *26* (3), C31–C66.

World Bank, "Digital Progress and Trends Report 2023," 2024.

Young, Alwyn, "Learning by doing and the dynamic effects of international trade," *The quarterly journal of economics*, 1991, *106* (2), 369–405.

Appendix

Table of Contents

A Data Construction

A.1 App-level Data

Sample Selection – Our app-level data consists two parts: domestic apps and foreign apps. For domestic apps, we compiled a dataset of 6,000 Chinese companies and obtained comprehensive listings of apps available on the iOS store developed by these companies through qimai.com. Using qimai.com, we scraped firms based on their IDs, which range from 1 to 10,000. These IDs are assigned by the website itself according to the release date of each company's first app. Due to some IDs being empty, we ultimately included 6,000 companies.

This effort resulted in a total of 230,312 apps, representing approximately 17% of the current offerings on the Chinese iOS store. We focused on the iOS store because of the highly fragmented nature of the Android market in China, where each major phone developer maintains its own dedicated app store, resulting in eight separate markets. Of the 230,312 apps we collected, 65% are also available on the Android platform.

For foreign apps, besides the 114 foreign apps that were blocked, we further collect information for approximately 450,000 additional apps, selected by randomly choosing 1% of apps per country (according to the location of developers) per year.

App Basic Information – For each app in our sample, we scrapped historical basic information from qimai.com, including app name, developer name, app description, screenshots, price, version history including version id and update logs, category, compatibility, languages, age rating, developer website, privacy policy and details.

App Performance – We further enhance our data analysis by sourcing monthly active user metrics and user engagement times from qianfan.tech, operated by Analysys. This platform distinguishes itself through its unique data collection methodology, utilizing both proprietary tools and strategic collaborations. qianfan.tech has developed a specialized SDK that is embedded in over 30,000 mobile apps from its partners. This SDK methodically captures detailed user activity data, enabling precise tracking and analysis of user behaviors across these applications. Additionally, they amplifies its data coverage and accuracy by integrating information provided by major wireless carriers in China.

The primary advantage of qianfan.tech is their foundational reliance on actual user behavior data rather than on predictive algorithms commonly used by other data providers in the market. However, one notable limitation with qianfan.tech's approach is the scope of data coverage. While covering the major apps in Chinese app market, our data set currently encompasses app performance metrics for only 11,453 apps. This limitation, primarily due to the direct data collection method.

App Packages and Libraries – To track the evolution of libraries in app over time, we collect IPA file for each app version for 186,220 apps from apkmirror.com and therefore 789,076 IPA files. IPA file is the standard format for distributing and installing applications on Android operating systems, which encapsulates all elements of an app, including its code, resources, assets, and manifest file, which collectively define the app's structure and behavior.

For iOS apps, we utilize `class-dump` to reverse engineering the IPA files (the packages of iOS apps). `class-dump` works by extracting the Objective-C runtime information embedded within an app's binary. This information includes details about the classes, categories, and protocols used in the application, along with their associated instance variables and methods. The tool can generate header files representing the data structures and interfaces that the application uses, effectively providing a snapshot of the app's internal APIs.

To extract libraries from APK files, a systematic and comprehensive approach was employed. First, each application file was unpacked to access its contents by being decompiled with *apktool*. This step accesses the underlying structure and codebase of the application, where SDK references are typically embedded. Subsequently, we parse through the decompiled files to recognize SDK-specific signatures, package names, and initialization routines, thereby enabling the accurate identification of SDK versions and their respective instances across different app versions.

A.2 Library-level Data

For each libraries extracted from apps, recognizing that libraries often include unique identifiers that hint at their origins, we leveraged this information to form hypotheses about the libraries' identities. We utilized CocoaPods' search functionality, accessible both through their website and via the command line (`pod search`), to locate libraries that matched our initial hypotheses.

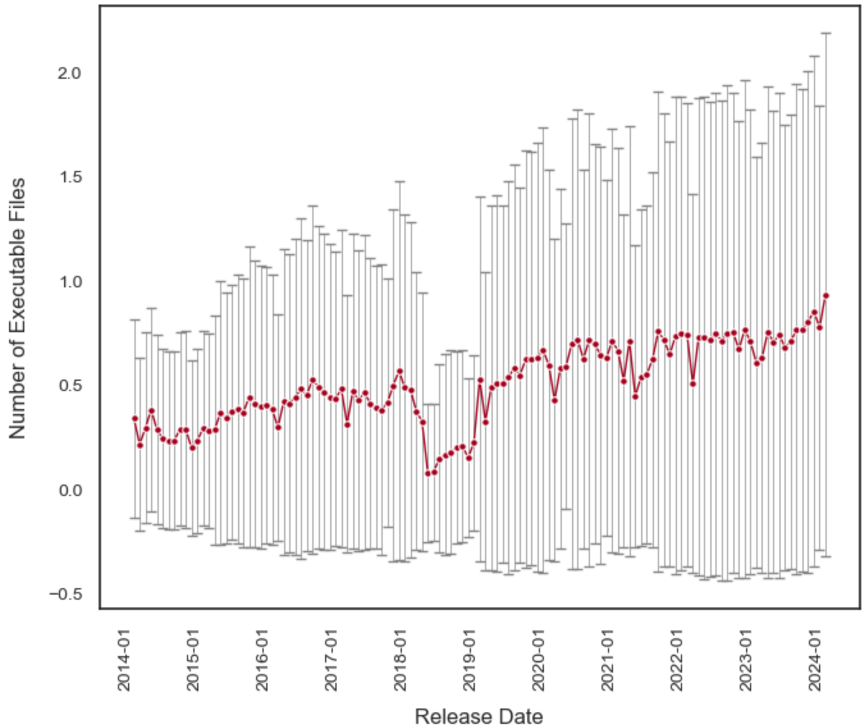
Upon finding potential matches on CocoaPods, we conducted a detailed comparison of the libraries' metadata—such as classes, methods, and other identifiers extracted during our analysis—with the documentation available on CocoaPods and the libraries' public repositories, such as `GitHub`. This meticulous comparison was crucial to ascertain if the public APIs of the libraries corresponded with the headers we generated. We also thoroughly examined the podspec files on CocoaPods, which provided additional valuable metadata about the libraries, including their source files and version history. This information was essential for confirming whether a library used in the app was indeed the one listed on CocoaPods.

In-house libraries are integrated by including them directly in the project or as separate modules. This approach ensures that custom functionalities developed internally are seamlessly incorporated into the app. The decision to include in-house libraries directly in the project or as separate modules depends on the library's complexity and scope. Smaller, utility-focused libraries are often integrated directly into the project for ease of use, while larger, feature-rich libraries are included as separate modules to enhance modularity and maintainability.

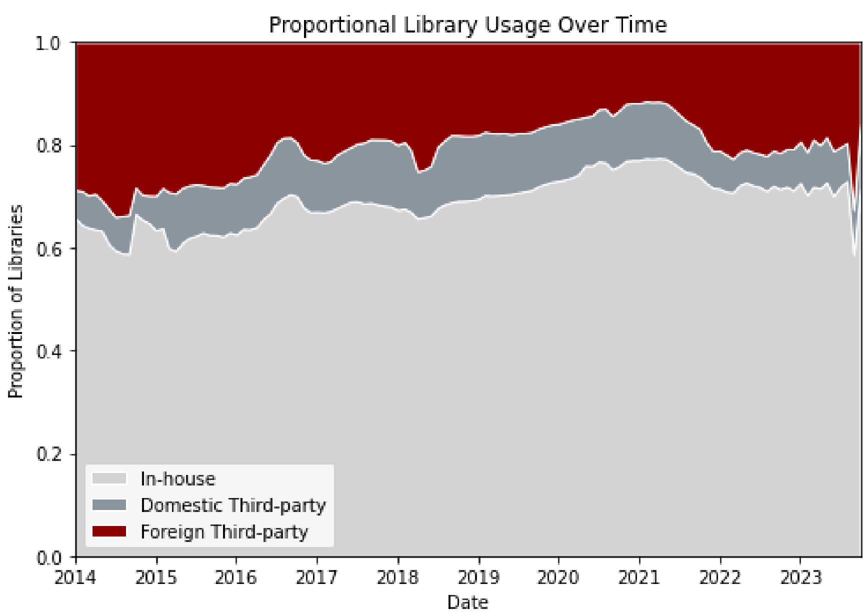
External libraries are added using dependency managers, which streamline the process by automatically downloading and including these libraries as part of the application's executable code. During the build process (Step 4 in Figure 1), these libraries are packaged into the application bundle along with the rest of the app's code. When users download the application from any store or distribution point, they receive the complete application from the app store, inclusive of all integrated libraries. The installation process does not differentiate between the app's original code and the code from the

Figure A.1: Descriptive Statistics for Libararies

(a) Avgerage Number of Executable Files



(b) Proportional Change in Library Usage



libraries—both are seamlessly installed as part of a unified package.

A.3 Firm-level Data

To match company names extracted from libraries, we utilize the API provided by Crunchbase which allows one to effectively search English names and find the corresponding firms at scale.

A.4 Blockage of Foreign Apps

To compile a list of the most notable blocked domains, I first gathered the top 5,000 domains from the Alexa traffic rank, a key metric for assessing website popularity. I then checked these domains against GFWatch (Hoang et al., 2021) to determine if they were blocked and when the blockage happened, as illustrated in Figure A.2.

Figure A.2: Example of GFWatch.


Search for censored domains and their latest blocking rules (may take upto a minute):

www.google.com

SEARCH

Tranco_rank	censored_domain	base_censored_domain	blocking_rules	first_seen	last_seen	base_domain_category
1	www.google.com	www.google.com	www.google.com*	2020/03/20	2022/03/18	search engines and portals

Next, I linked the blocked domains to their corresponding applications by examining the apps' support websites. To further validate the timing of each blockage, I manually coded the dates based on information from news reports and internet searches. Additionally, I utilized data from `greatFire.org`, which has monitored blocked websites and keywords since 2011, confirming most of the necessary historical blockage information.

B Substitutability and Complementarity

B.1 Training Set Construction

To evaluate the relationship between software applications–whether they are substitutes or complements–we utilized a novel approach that leverages the capabilities of the OpenAI language model. This method involves a detailed two-step process aimed at analyzing the descriptions of applications to understand their interactions.

Step 1: Prompt Formulation - We began by crafting a detailed prompt that encapsulates the main functionalities and unique attributes of the applications being analyzed. This prompt aims to succinctly set the stage for the AI to conduct a comparative analysis. As illustrated in Figure A.3, the prompt asks: "Based on the descriptions provided for App A and App B, are these applications more likely to be substitutes, serving similar purposes, or complements, enhancing functionality when used together?"

The setup for this prompt includes specific model parameters to optimize the analysis: we utilized a model configuration with a `temperature` of 0 to maximize response determinism, a `max_tokens` limit of 20 to focus responses, `top_p` set to 1 ensuring only the most likely completions are considered, and zero `frequency_penalty` and `presence_penalty` to allow unbiased consideration of all relevant aspects. These parameters are critical in guiding the AI's response pattern and are illustrated in the figure below.

Figure A.3: Prompt and parameters used for creating training set with OpenAI API.

```python
# Craft the prompt
prompt = f'''Here I describe two apps:

App 1 Description: {app1_description}
App 2 Description: {app2_description}

*Purely* based on their functionalities in the descriptions,
are these apps more likely to be substitutes (serving similar purposes)
or complements (used together for enhanced functionality) or neutral?

Please give the output in a json format:
'relationship': 'substitutes/complements/neutral'
'''

messages = [{"role": "user", "content": prompt}]

# Query the model
response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=messages,
    temperature=0,
    max_tokens=20,
    top_p=1,
    frequency_penalty=0,
    presence_penalty=0
    )
```

Figure A.4: Example of an application analysis using the AI model.

Substitute

```
{   "app1": "WhatsApp",
    "app2": "Messenger",
    "relationship": "substitutes",
    "reason": "Both apps provide similar functionalities such as private messaging, group chats, voice and \
        video calls, sharing of files and media, and cross-platform compatibility. Therefore, they can be \
        used interchangeably for the same purposes."
}
```

Complement

```
{   "app1": "Instagram",
    "app2": "Messenger",
    "relationship": "complements",
    "reason": "While both apps offer social networking features, they serve different purposes. \
        Instagram is primarily for sharing photos and videos, discovering new content, and shopping products. \
        Messenger, on the other hand, is more focused on communication through text, voice, and video calls. \
        They can be used together for enhanced functionality, for instance, sharing Instagram posts through \
        Messenger or communicating with friends found on Instagram through Messenger."
}
```

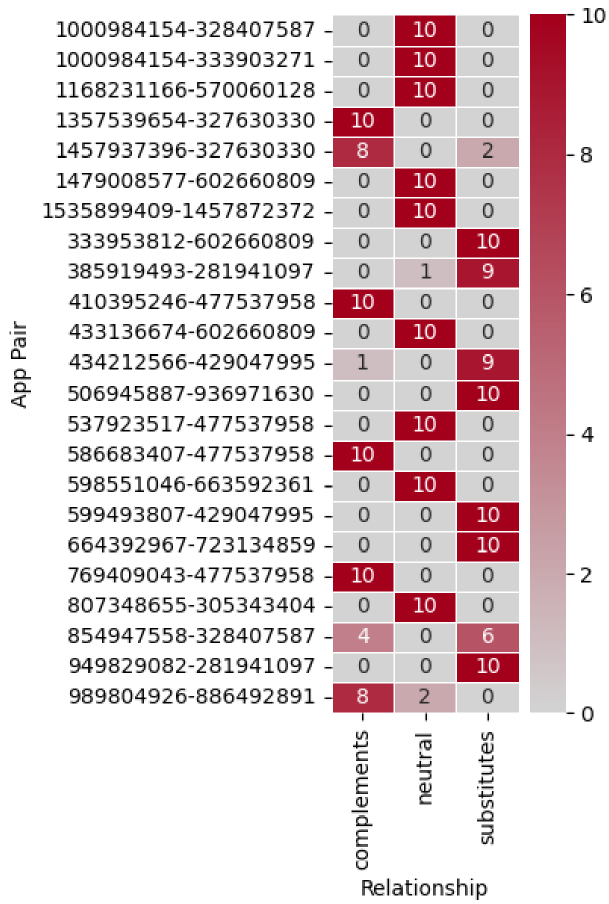
Neutral

```
{   "app1": "Doulingo",
    "app2": "Messenger",
    "relationship": "neutral",
    "reason": "The first app is primarily focused on language, math, and music learning, while \
        the second app is a communication tool with features like text, voice, video calling, \
        and group video chat. They serve different purposes and do not necessarily enhance \
        each other's functionality or serve as substitutes."
}
```

B.2 Validation for Consistency

To demonstrate the consistency of `gpt-4`'s outputs and ensure reproducibility, the `temperature` parameter was set to 0 and `top_k` to 1, with a fixed sampling seed to maintain determinism. Despite these measures, complete determinism is not guaranteed, especially with longer outputs. To evaluate the consistency of labeling, we randomly selected 20 app pairs and ran the model 10 times for each pair using the same inputs. We then calculated the average proportion of times the model produced the most frequent label across these multiple runs, resulting in a consistency rate of 0.94. Figure A.5 illustrates the classification results from these tests, highlighting the robustness of the model under controlled settings.

Figure A.5: Consistency of `gpt-4` Output



Notes: This figure displays the results of ten runs of the `gpt-4` model for each of 20 randomly selected app pairs. The matrix illustrates how often each app pair was classified under each relationship type across these runs. The x-axis represents the relationship types—complement, neutral, and substitute—while each cell shows the number of times a specific app pair was classified into one of these categories across multiple model executions. The intensity of the color in each cell corresponds to the frequency of classifications, ranging from 0 (no occurrences) to 10 (classified as such in all runs). The rightmost column provides a visual summary of the distribution of results, underscoring the model's consistency or variability in classifying each app pair.

B.3 Alignment with Human Evaluation

To confirm the accuracy of `gpt-4`'s classifications, I tasked 11 Chinese annotators with labeling the relationships of 2,667 app pairs, distributing the workload so each annotator evaluated around 700 pairs. These pairs includes:

(1) 1,645 app pairs that are all substitutes classified by `gpt-4`;

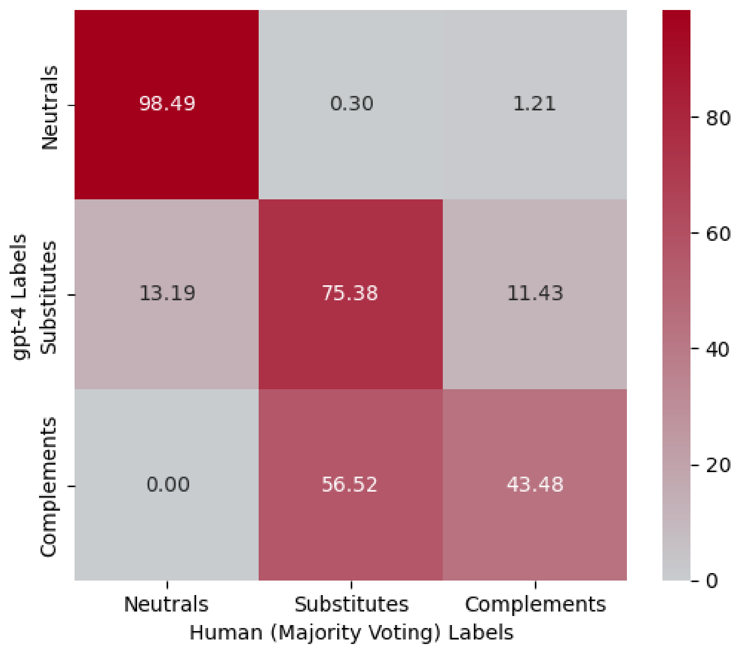
(2) 22 app pairs which are all app pairs that are classified as complements by `gpt-4`;

(3) 1,000 app pairs that are randomly selected from the 16,392 app pairs that are classified as neutral by `gpt-4`;

Each app pair is randomly assigned to annotators. To ensure robust classification results, each app pair was reviewed by three annotators. The final label for each pair was determined through majority voting.

Figure A.6 presents a confusion matrix comparing `gpt-4`'s relationship labels to the human majority voting labels.

Figure A.6: Confusion Matrix: LLM versus Human Labeling



Notes: This figure is a heatmap visualization of the confusion matrix that compares the relationship labels generated by GPT to the human majority voting labels. The matrix is presented in percentages, showing the proportion of each predicted label class relative to the total count of that class.

B.4 Training with `gpt-4` Labeled Data

- Upon completing the prompt formulation, we utilized `GPT-4` to analyze the descriptions of pairs of applications. Figure A.4 showcases examples of app pairs classified as substitutes, complements, or neutral based on their relationship.

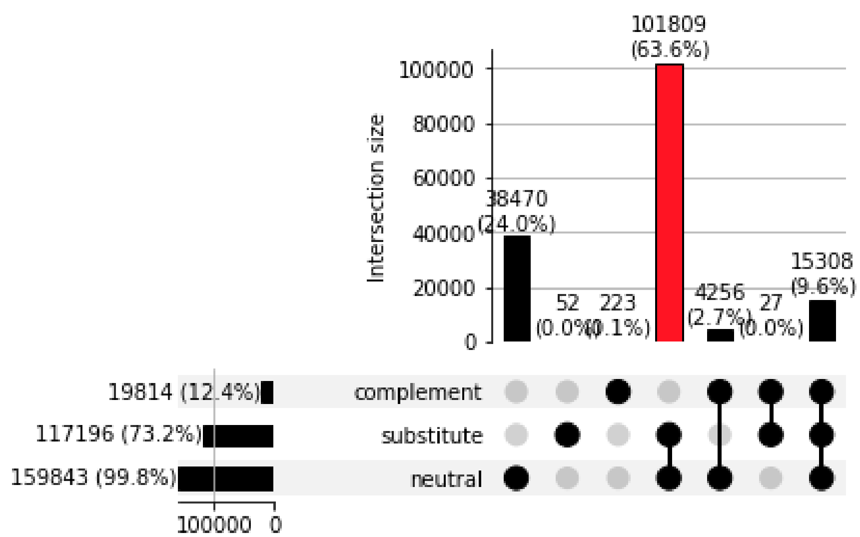
We randomly selected 18,066 pairs of applications and labeled their relationships using `GPT-4`, creating a substantial training dataset for a natural language processing (NLP) model. For each application within these pairs, we encoded the descriptions into word embeddings of size 768 using `bert_multilingual_cased`. We chose this model due to its robust performance across multiple languages and its effectiveness in capturing semantic nuances essential for our analysis.

Subsequently, for each app pair, we combined their corresponding word embeddings and employed various machine learning models to predict their relationships as classified by `GPT-4`. We utilized models such as Logistic Regression, Random Forest Classifier, and Linear SVC for this task. The accuracy rates of these models are summarized in the table below:

Table A.1: Accuracy rate of different models

Model	Accuracy Rate
Logistic Regression	0.86
Decision Tree	0.844
Random Forest	0.8131
Multilayer Perceptron (small)	0.8990
Multilayer Perceptron (large)	0.878

Figure A.7: The Intersection of Treatment Types across Domestic Apps



Notes: This figure visualizes the distribution of interactions between apps, categorized by treatment types. The x-axis represents the interaction size, while the y-axis lists various categories. The largest interaction category, highlighted in red (101,809), accounts for 63.6% of the total interactions, indicating a substantial portion. Smaller groups represent other interaction sizes, with percentages noted for each. Below, the black bars represent different categories (e.g., "substitute") alongside their corresponding sizes, with the bar lengths proportional to the interaction size.

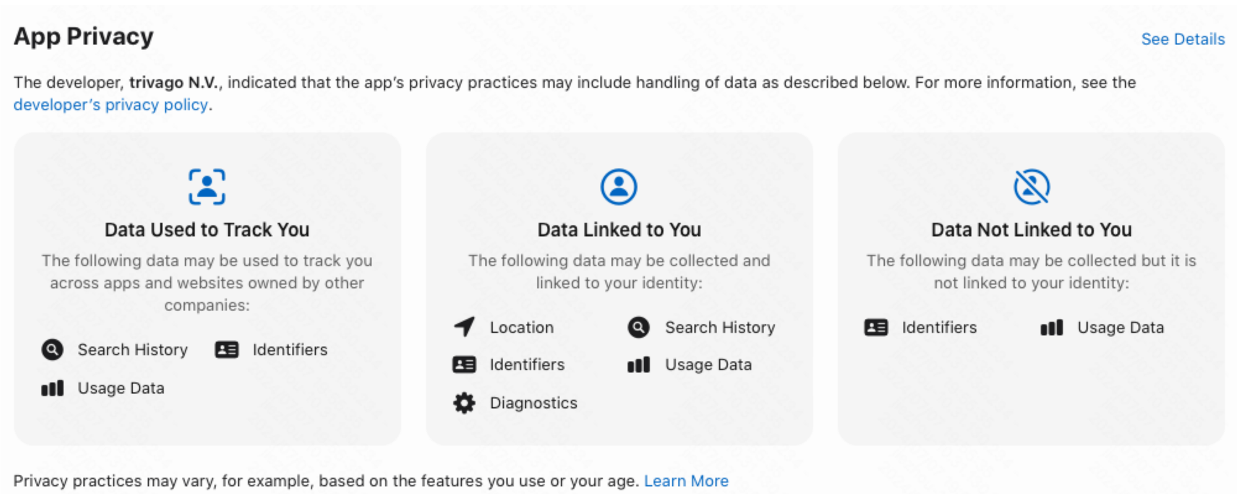
This integrated process establishes a robust NLP algorithm that we utilize to classify the relationships between app pairs. Applying the classifier to the full sample of 15,373,920 foreign-domestic app pairs, we find that 16.5% are classified as substitutes, 2.1% as complements, and the remaining 81.3% as neutral. To ensure that our treatment is not overly concentrated among a few domestic apps, which would indicate a highly selective treatment effect, we further analyze the distribution of treatment types across domestic apps. As illustrated in Figure A.7, 63.6% of the domestic apps have at least one foreign substitute that is blocked, indicating that the treatment is sufficiently widespread. Additionally, 24% of Chinese apps have neither foreign substitutes nor complements blocked, forming the never-treated group in our empirical design.

C User Privacy

C.1 Background

Apple's Privacy Labels – According to Apple's definition, data collection involves transmitting data off the device in a manner that allows you and/or your third-party partners to access it for longer than is necessary to fulfill the immediate request. Starting December 2020, users should be informed of such data collection by categorizing it into three groups: *"Data Used to Track You"*, *"Data Linked to You"*, and *"Data Not Linked to You"*. Developers must disclose their data collection practices during the app submission process, ensuring they comply with Apple's stringent privacy guidelines, which are verified during the app review process.

Figure A.8: An Example of Apple Privacy Labels



The category *"Data Used to Track You"* includes data collected from a user's device that is linked to their identity and used to track them across different apps and websites. Examples of such data include advertising identifiers (like IDFA), email addresses shared with third-party advertising networks, and location data if shared with third parties for tracking purposes.

"Data Linked to You" refers to data collected in a way that is directly associated with your identity, such as your account information, device details, or personal details like your phone number. To classify data as collected but not linked to you, developers must employ privacy measures like removing direct identifiers, such as user IDs, before collecting the data and ensuring they do not link the data back to your identity after collection.

In contrast, *"Data Not Linked to You"* encompasses data collected in an anonymized or aggregated form, not linked to the user's identity. Examples include general app usage statistics without personal identifiers, anonymized crash data, and performance metrics related to network and app performance that are not tied to individual users.

An example of this categorization is shown in Figure A.8.

Property List – In practice, developers must specify the types of data their app collects by including key-value pairs in a file called the `info.plist` (Information Property List) file when they submit the app to Apple Store for review. The `info.plist` file is a structured text file used in iOS development that contains essential metadata about the app. This metadata includes information such as the app's version number, supported device capabilities, and more importantly privacy-related data collection practices.

Each key-value pair in the `Info.plist` file represents a specific piece of information about the app. For things related to data,

C.2 Identify Collected Data and Privacy Labels

Data and Preprocessing – We collect historical privacy labels for 23,643 app versions (3991 apps) and matched them with keys in the `info.plist` of each app version. For the privacy labels, they are dummies to indicate whether the app version are obtaining (1) data used to track users, (2) data linked to users, (3) data not linked to users, or (4) no data are collected by the app at all. Based on the sample, we calculate the probabilities for a key appearing in an app with a specific privacy label and present them in Figure A.9. From the raw data, we already can see strong correlations between keys and privacy labels.

With this, we establish a correspondence between `info.plist` keys, which we could consider as the inputs, and privacy labels which indicate whether the app version contains data that track, link to, not link to users, or no data collected at all.

Model Training – We then use a logistic regression model configured for multinomial outcomes, appropriate for scenarios with more than two class labels. The model is defined mathematically by the softmax function, used to predict the probabilities of the multiple classes:

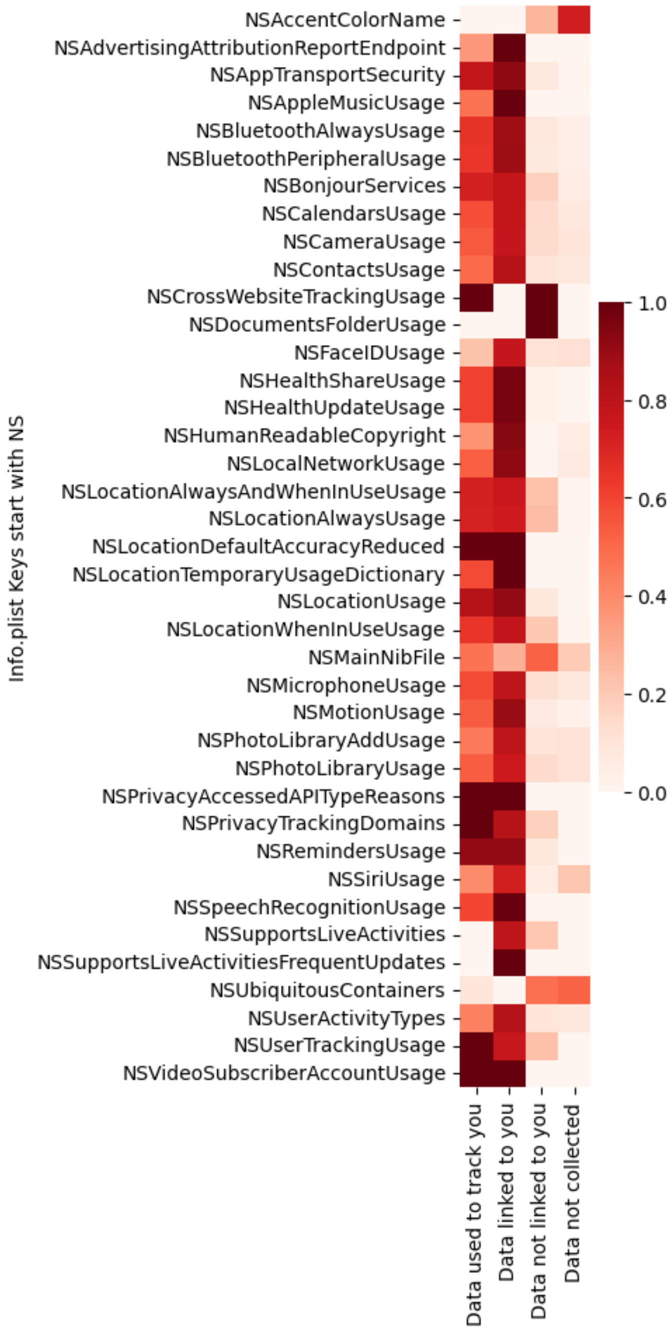
$$P(y = k \mid x) = \frac{\exp\left(x^T \beta_k\right)}{\sum_{j=1}^{K} \exp\left(x^T \beta_j\right)}$$

where x represents the feature vector, β_k is the coefficient vector for class k, and K is the total number of classes. L1 regularization was applied to enhance feature selection, implemented through the penalty=$'\,11'$ parameter in the logistic regression. This regularization adds a penalty equal to the absolute value of the magnitude of the coefficients to the loss function, in order to (1) promote sparsity in the model parameters for feature extraction, and (2) .

The dataset was split into a training set (70%) and a testing set (30%) to validate the model's performance.

Model Visualization and Validation – Initially, we assess the classification results by analyzing the log odds ratios associated with various keys in our model, particularly focusing on the contributions of keys listed within the Info.plist file. This file is vital as it encapsulates configuration details for iOS apps. Our analysis specifically targets keys that start with "NS" and "UI", as these prefixes generally indicate crucial system and user interface configurations, respectively. Keys prefixed with "NS" are often linked to app settings that involve user privacy, functionality, and system integration. These typically demonstrate higher log odds in the model due to their direct impact on how the app interacts with iOS features and handles sensitive user data. Conversely, keys

Figure A.9: Correlation between `Info.plist` Keys and Privacy Labels



Notes: This figure shows the probabilities for a key appearing in an app with a specific privacy label. The x-axis presents privacy labels where are at the app level. The y-axis shows the name of keys. And the color of the heat map indicates the the probabilities for a key appearing in an app (x-axis) with a specific privacy label (y-axis).

Table A.2: Classification Report

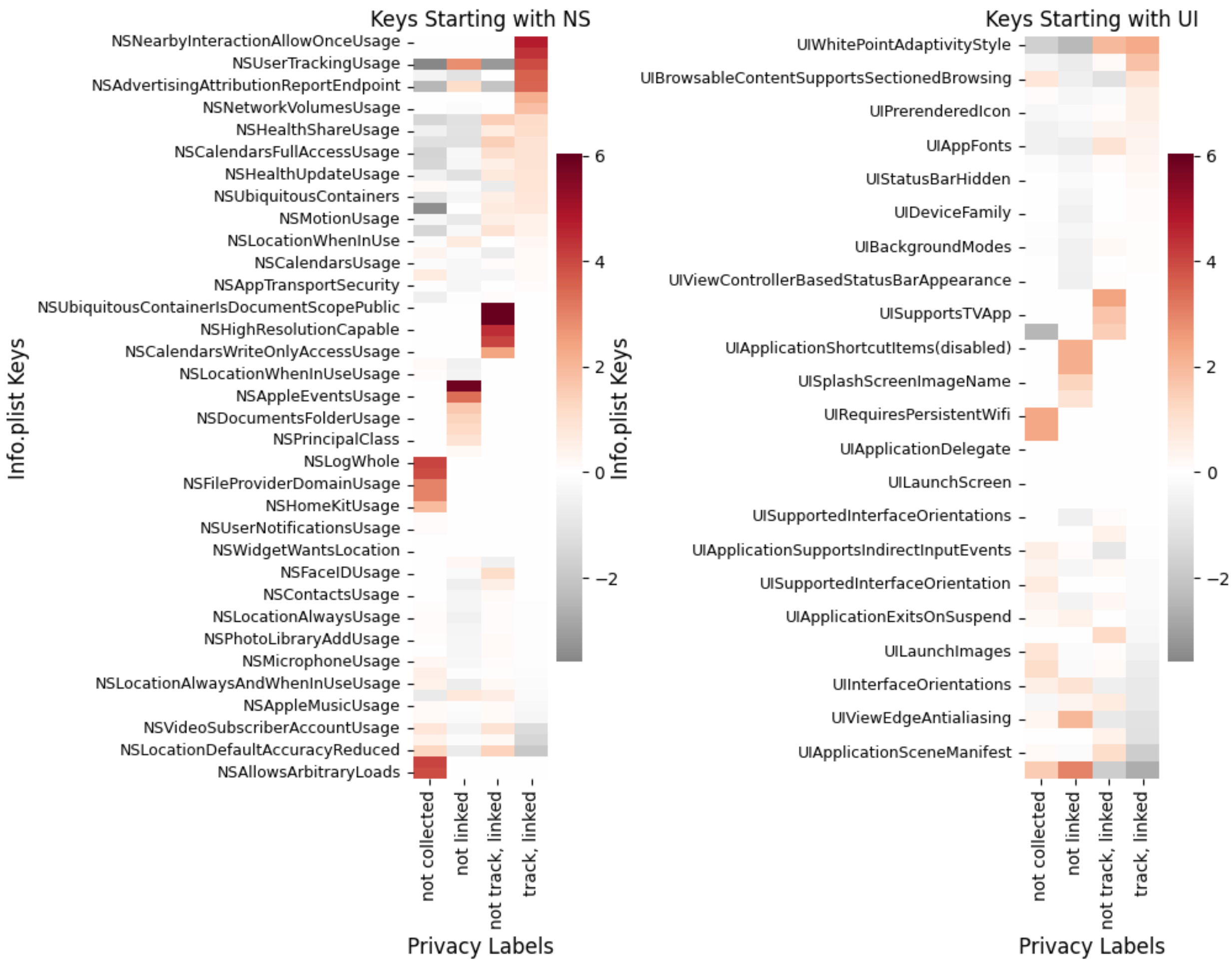
Class	Precision	Recall	F1-score	Support
No data collected	0.97	0.96	0.97	57880
Data that not linked to you	0.99	0.99	0.99	91690
Data that linked to you	0.97	0.97	0.97	84809
Data that used to track you	0.97	0.96	0.96	66208
Macro avg	0.97	0.97	0.97	454931
Weighted avg	0.97	0.97	0.97	454931
Accuracy on test set	0.97 (454931 samples)			

beginning with "UI" are primarily concerned with interface elements, aiming to enhance the intuitiveness and effectiveness of the user experience, which are generally less directly related to privacy concerns but crucial for user interaction. Figure A.10 illustrates these coefficients, aligning with our expectations regarding the relevance of "NS" and "UI" keys to privacy labels and user interface design, respectively.

Second, we conducted a case study on a Chinese iOS app (id 977946724), which has been previously noted for potential user data leakage[1]. Despite the absence of explicit warnings in the privacy labels on the Apple App Store, our investigation revealed that the app contains 44 Info.plist keys that are classified as sensitive and shared.

[1]Source: `https://shorturl.at/ExIYS`

Figure A.10: Log Odds in Classifier

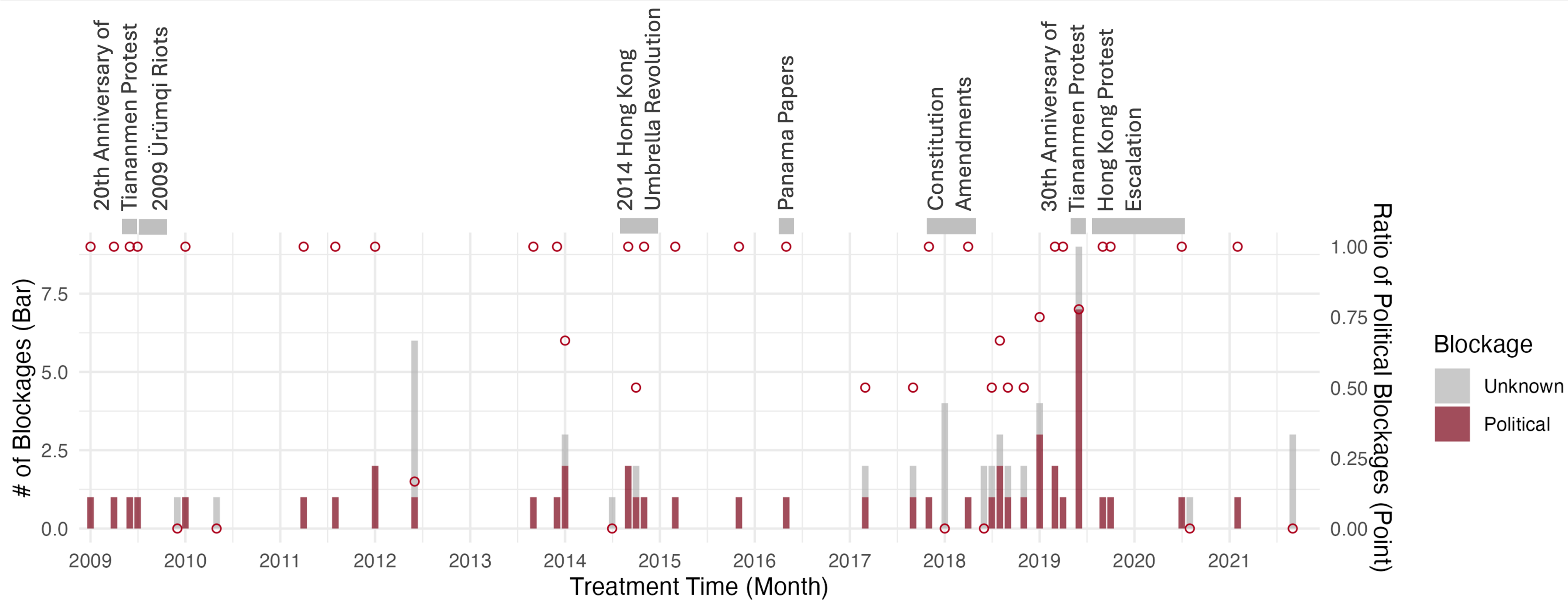


Notes: This figure visualizes the coefficients of a subset of `Info.plist` keys.

D Additional figures and tables

D.1 Temporal Distribution of Foreign App Blockages

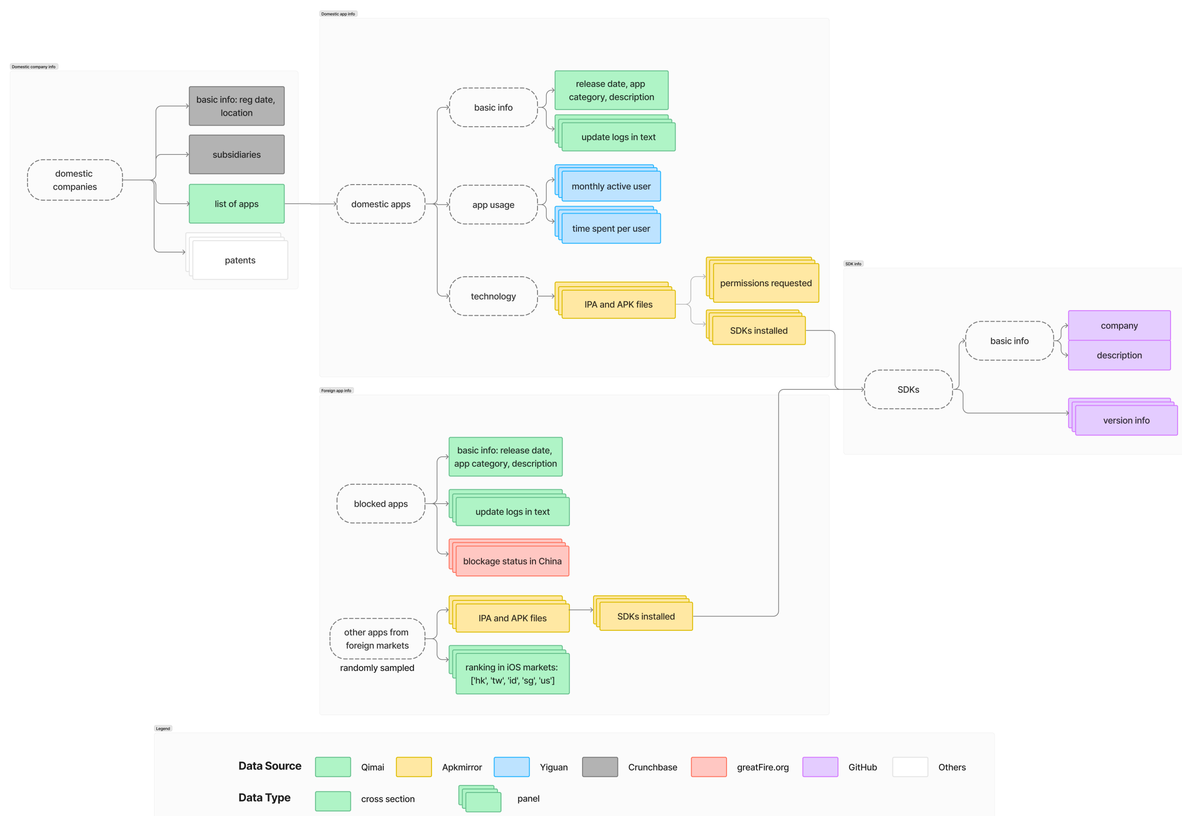
Figure A.11: Political Motivations of Blockages



Notes: This figure visualizes the temporal distribution of foreign app blockages across years, categorized by the reasons for the blockages. The blockages are classified into two categories: "political," representing blockages linked to political motivations based on manual internet research, and "unknown," for blockages where no identifiable reason could be found. The primary y-axis (on the left) represents the total number of blockages as stacked bars, with political blockages highlighted in dark red and unknown blockages in gray. The secondary y-axis (on the right) depicts the ratio of political blockages to the total blockages as red circular points, providing a proportional view of politically motivated blockages over time. Key events are annotated along the x-axis to provide historical context. These annotations align with periods where the frequency or nature of app blockages may have changed. By combining both absolute numbers (bars) and proportions (points), this figure offers insights into the dynamics of app blockages and their underlying causes over the observed period.

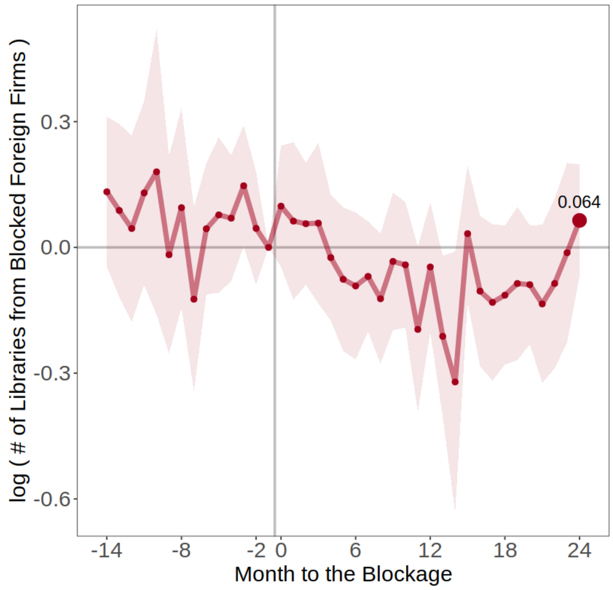
D.2 Overview of Data Structure

Figure A.12: Overview of data structure



D.3 The Effect of Blockage on Libraries from Blocked Firms

Figure A.13: The Effect of Blockage on Libraries from Blocked Firms

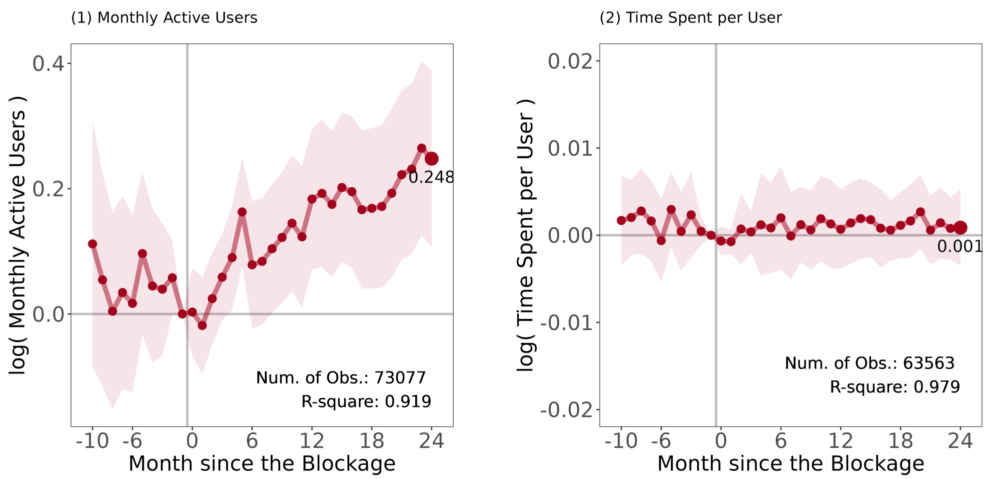


Notes: These figures show the effect of blockage on the adoption of domestic and foreign libraries. The red dots and lines shows the point estimates of β_l, and the bandwidth in grey shows the corresponding 95% confidence intervals of β_l. The effect of blockage on the number of domestic libraries are presented with solid lines, while the estimates for foreign libraries are presented with the dashed line.

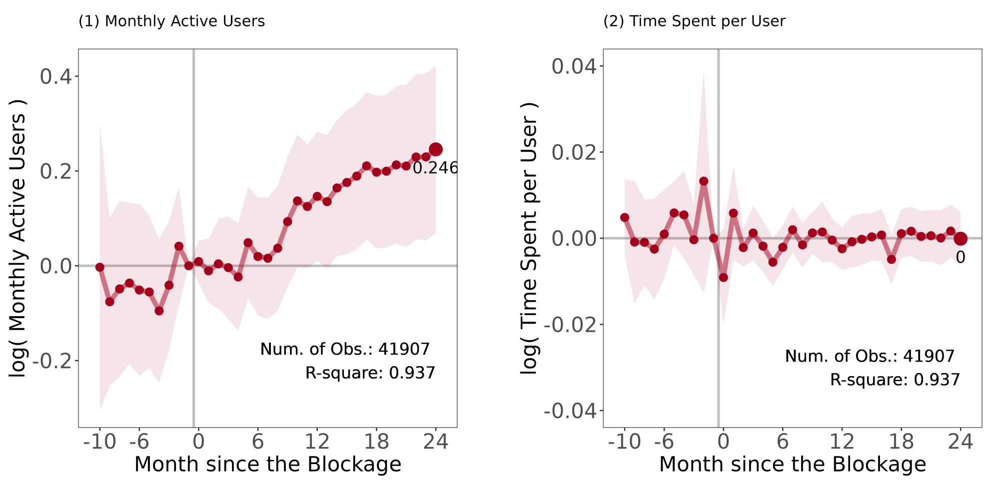
D.4 The Effects of Blockage on Demand: Political versus Non-Political

Figure A.14: The Effects of Blockage on Demand: Political versus Non-Political

(a) Blockages Due to Political Reasons



(b) Blockages Due to Non-Political and Unspecified Reasons

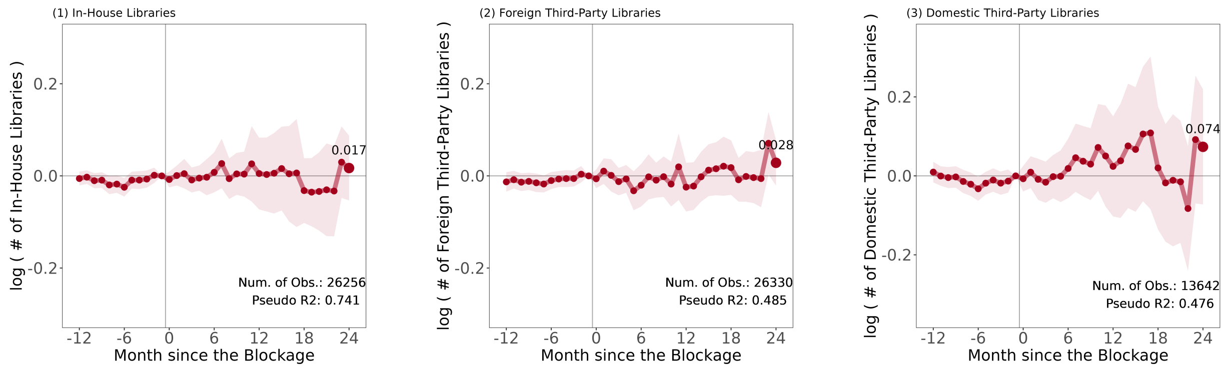


Notes: These figures present the placebo tests for the effects of blockages on demand, divided by political and non-political/unspecified reasons. The red dots and lines represent the point estimates of β_l in Equation 2, while the red bands show the corresponding 95% confidence intervals of β_l.

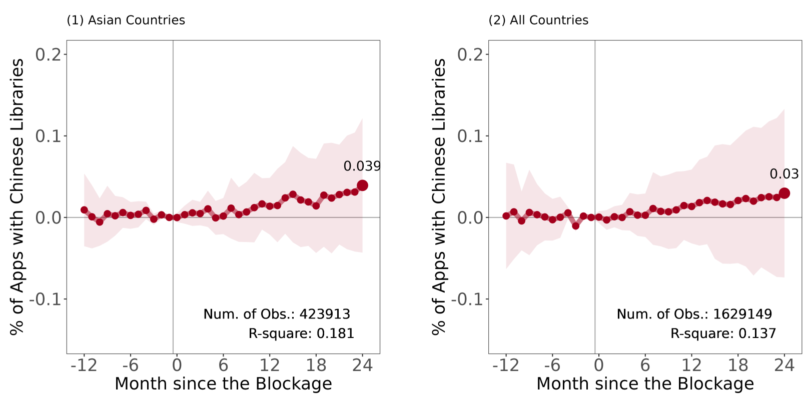
D.5 Placebo Exercises with Foreign Neutrals

Figure A.15: Placebo Exercises with Foreign Neutrals

(a) Quantity: Number of Libraries



(b) Quality: Foreign Installs



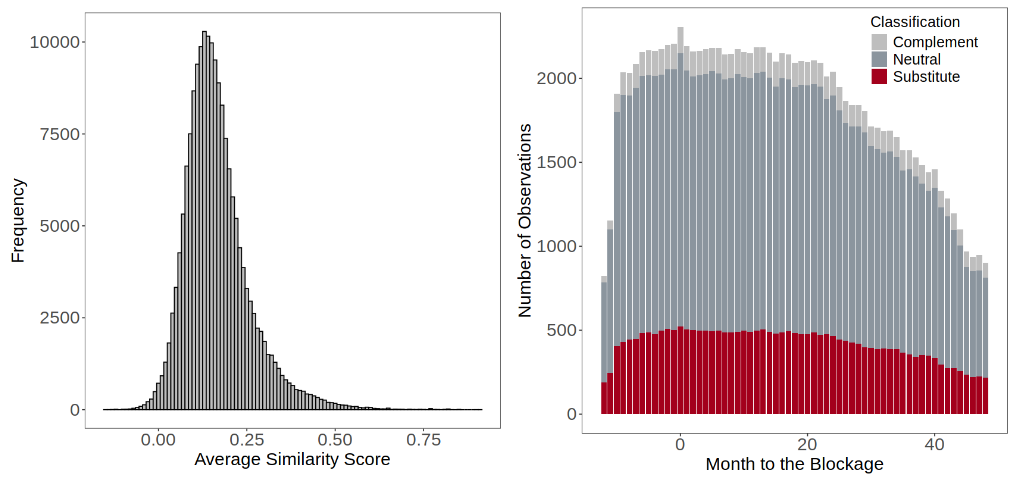
Notes: These figures show the placebo tests for the effects of blockage on innovation. The red dots and lines shows the point estimates of β_I, and the bandwidth in grey shows the corresponding 95% confidence intervals of β_I. The effect of blockage on the number of domestic libraries are presented with solid lines, while the estimates for foreign libraries are presented with the dashed line.

D.6 Descriptives and Robustness of Similarity Score Results

Figure A.16: Descriptives for Similarity Scores

(a) Distribution of Similarity Score

(b) Observations by Event Time and Classification



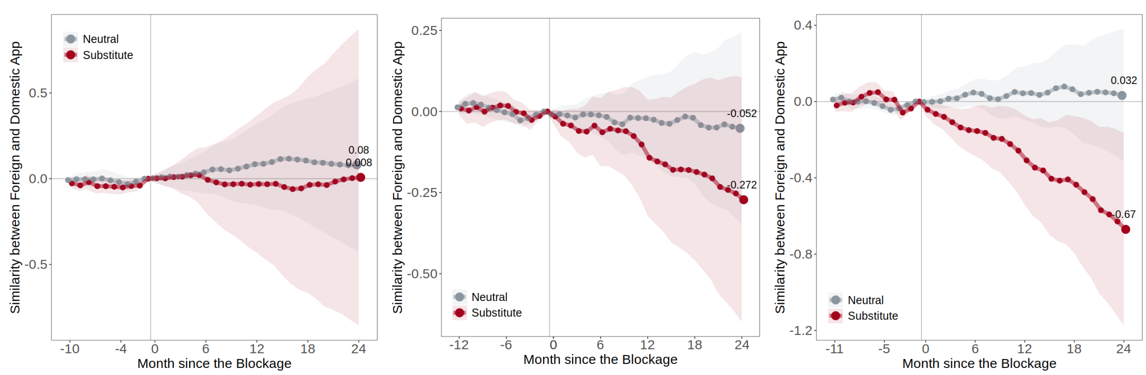
Notes: The figures present distributions of similarity scores between update logs at app pair-month level.

Figure A.17: Other Robustness Exercises

(a) Similarity Score Based on Future Versions

(b) Similarity Score Based on Cutoff=24 Month

(c) Similarity Score Based on Dot Product

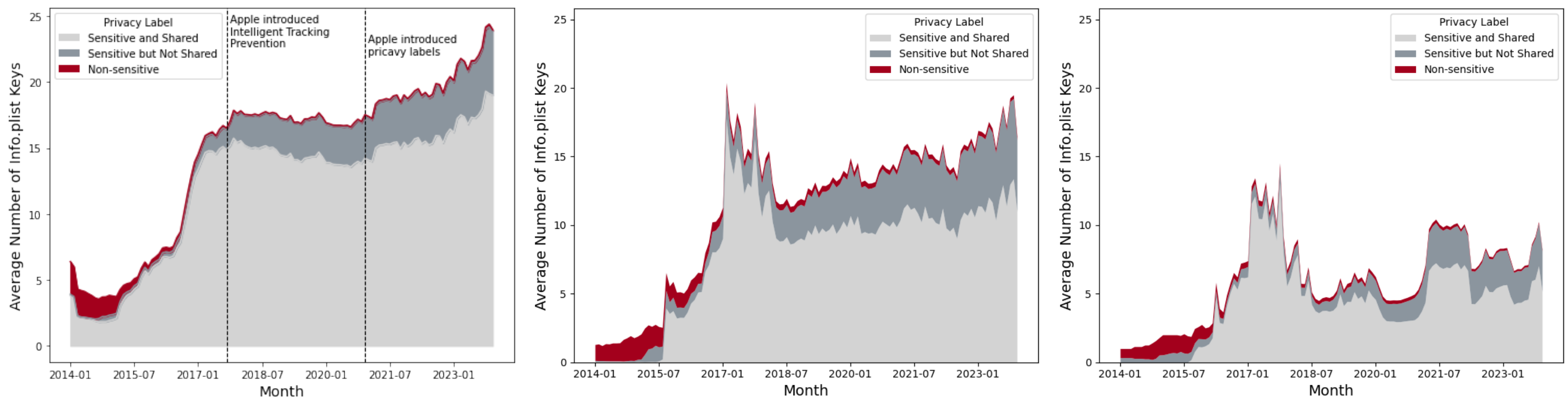


Notes: The figures present results based on different versions of similarity scores to test the robustness of the findings.

D.7 Average Number of `Info.plist` Keys by Privacy Label

Figure A.18: Average Number of `Info.plist` Keys by Privacy Label

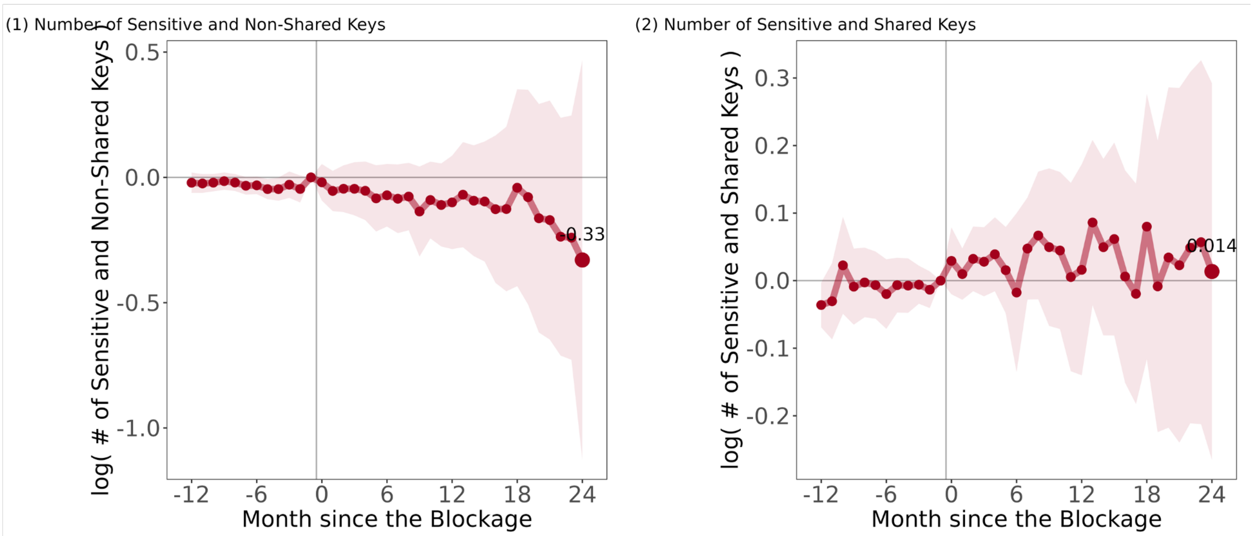
(a) Chinese Apps **(b)** US Apps **(c)** European Apps



Notes: This figure presents the average number of `Info.plist` keys categorized by privacy labels for Chinese (Panel (a)), US (Panel (b)), and European (Panel (c)) apps. The x-axis spans months from January 2014 to December 2023. The y-axis displays the average number of `Info.plist` keys related to user data per app, standardized across the three plots for comparability. In each panel, the light grey bars represent sensitive and shared keys, the dark grey bars represent sensitive but not shared keys, and the red bars represent non-sensitive keys.

D.8 Placebo Exercises for Data Collection & Sharing

Figure A.19: Placebo Exercises with Foreign Neutrals: Data Collection & Sharing



Notes: This figure visualizes the temporal introduction of various data types associated with `Info.plist` keys across different months from 2009 to 2022. Each row represents a specific data type (e.g., advertising data, contacts, customer support data, etc.), and each column represents a month of release. The color gradient, ranging from blue to red, signifies the average privacy level of the respective keys. A value of 0 indicates no user data is involved, 1 represents non-sensitive user data, 2 denotes sensitive data that is not shared, and 3 reflects sensitive data that is shared.

D.9 Correlation Between Data Scale and Scope in Never-Treated Apps

Figure A.20: Correlation Between MAU and Sensitive Keys in Never-Treated Apps

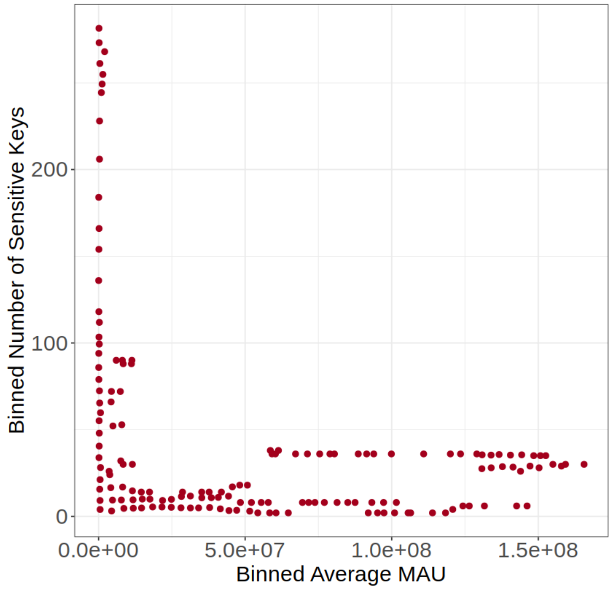


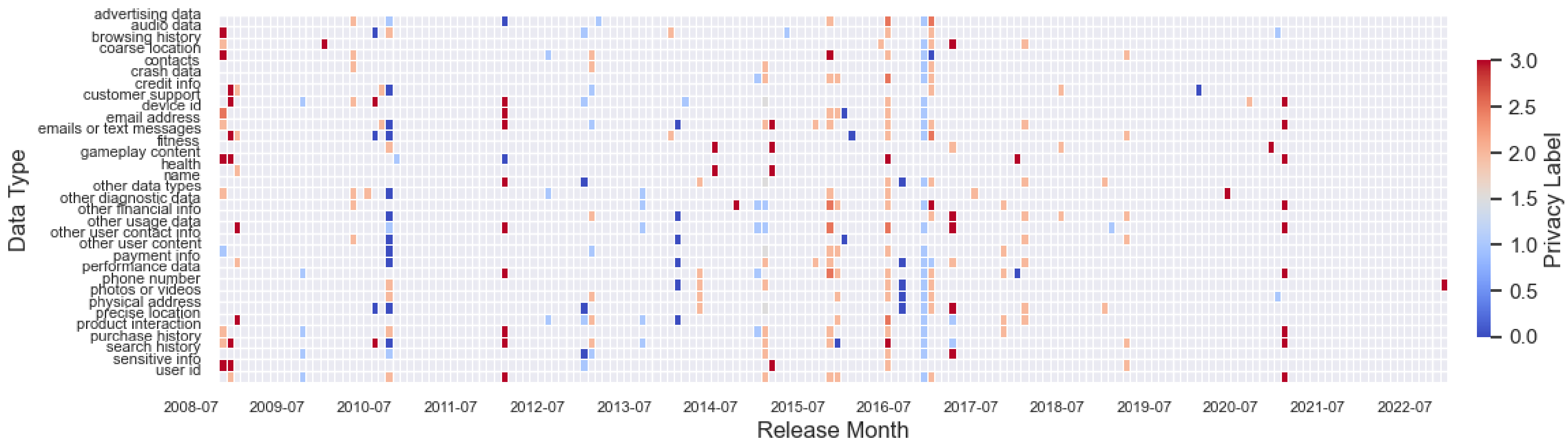
Table A.3: Regression Table

	# of Sensitive `Info.plist` Keys	
MAU	-3.853e-08	2.128e-08
	(3.223e-08)	(3.314e-08)
App FE	Y	Y
Month FE	Y	
R-square	0.8683	0.8579
Obs	12,781	12,781

Notes: This figure and table together show the correlation between monthly active users and number of sensitive `Info.plist` keys.

D.10 Variations in the Introduction of `Info.plist` Keys

Figure A.21: Variations in the Introduction of `Info.plist` Keys



Notes: These figures show the placebo tests for the effects of blockage on innovation. The red dots and lines shows the point estimates of β_l, and the bandwidth in grey shows the corresponding 95% confidence intervals of β_l. The effect of blockage on the number of domestic libraries are presented with solid lines, while the estimates for foreign libraries are presented with the dashed line.

D.11 Example of Domains Permitted to Bypass ATS in an App

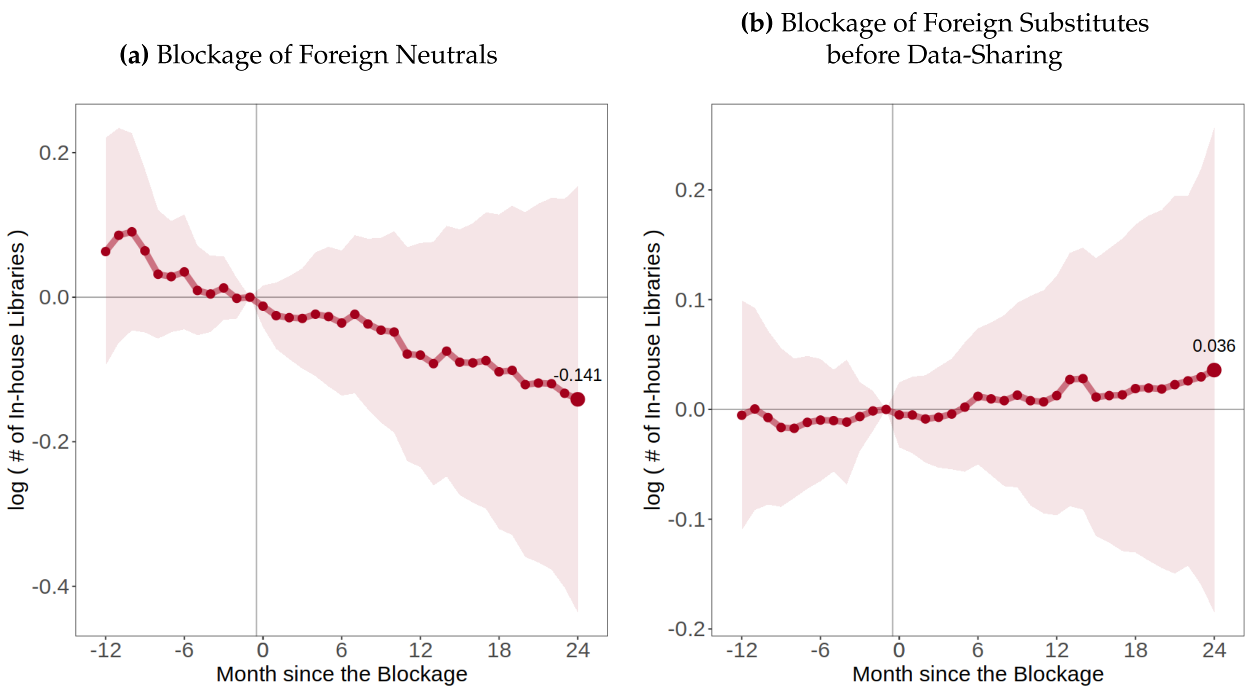
Figure A.22: Example of Domains Permitted to Bypass ATS in an App

▼ App Transport Security Settings	Dictionary	(2 items)
Allow Arbitrary Loads	Boolean	YES
▼ NSAppTransportSecurity	Dictionary	(1 item)
▼ NSExceptionDomains	Dictionary	(7 items)
▶ jpush.cn	Dictionary	(2 items)
▶ sina.cn	Dictionary	(3 items)
▶ sina.com.cn	Dictionary	(3 items)
▼ sinaimg.cn	Dictionary	(3 items)
NSIncludesSubdomains	Boolean	YES
NSThirdPartyExceptionAllowsInsecureHTTPLoads	Boolean	YES
NSThirdPartyExceptionRequiresForwardSecrecy	Boolean	NO
▶ sinajs.cn	Dictionary	(3 items)
▼ weibo.cn	Dictionary	(3 items)
NSExceptionMinimumTLSVersion	String	TLSv1.0
NSIncludesSubdomains	Boolean	YES
NSThirdPartyExceptionRequiresForwardSecrecy	Boolean	NO
▶ weibo.com	Dictionary	(4 items)

Notes: This figure displays a segment of an app's `Info.plist`, highlighting configurations that allow specific domains to bypass standard security protocols. Specifically, it shows the domains `sinaimg.cn` and `weibo.cn` authorized to transfer data via HTTP, as permitted under `NSThirdPartyExceptionAllowsInsecureHTTPLoads`. This example illustrates how exceptions to data security requirements are specified within app settings.

D.12 Placebo Tests for Spillover Effects of Blockage through Data-Sharing Network

Figure A.23: Placebo Tests for Spillover Effects of Blockage through Data-Sharing Network



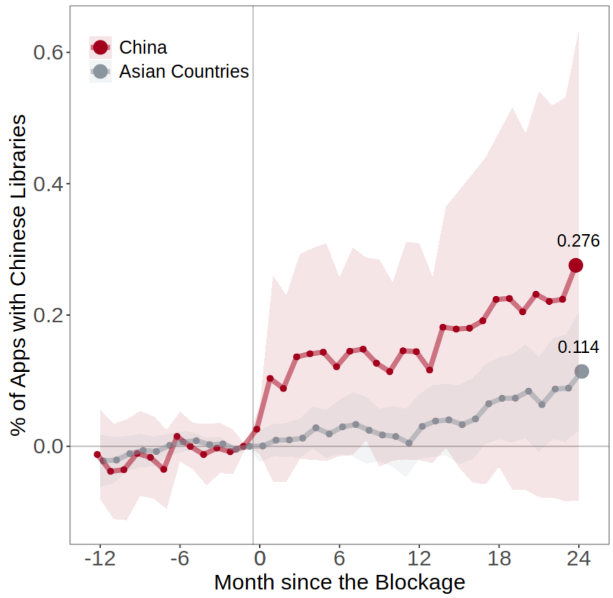
Notes: These figures show the placebo tests for the spillover effects of blockage through data-sharing. In Panel (a), we show the effects of blockages of foreign neutrals on apps that are indirectly affected through the data-sharing network. The red dots and lines shows the point estimates of β_l in Equation

$$Y_{ijt} = \sum_{l \neq -1} \beta_l NeutralBlocked^l_{jt} \times DataShare_{j \to c(i)} + \psi_{g(i)t} + \delta_{ij} + X^T_{it}\xi + u_{ijt}$$

. The bandwidth in light red shows the corresponding 95% confidence intervals of β_l. In Panel (b), for the same set of apps, we show the effects of blockages of foreign substitutes before the data-sharing connection establishes.

D.13 The Effect of Blockage on Library Installs in Domestic Market

Figure A.24: The Effect of Blockage on Library Installs in Domestic Market



Notes: These figures show the effect of blockage on the adoption of libraries from treated companies in Chinese domestic market. The red dots and lines shows the point estimates of β_l, and the bandwidth in grey shows the corresponding 95% confidence intervals of β_l. The effect of blockage on the adoption of libraries from treated companies in Chinese domestic market are presented by red lines/dots, while the estimates for installs in Asian markets are provided as a benchmark in grey lines/dots.