Received: 13 April 2023 | Revised: 17 February 2024 | Accepted: 29 February 2024

DOI: 10.1002/smj.3601

RESEARCH ARTICLE

SMS | Strategic Management Journal

WILEY

Porting learning from interdependencies back home: Performance implications of multihoming for complementors in platform ecosystems

Francisco Polidoro Jr.[1] | **Wei Yang**[2]

[1]The University of Texas at Austin, McCombs School of Business, Austin, Texas, USA

[2]China Europe International Business School, Shanghai, China

Correspondence
Francisco Polidoro, The University of Texas at Austin, McCombs School of Business, 2110 Speedway Stop B6300, Austin, TX 78712-1282, USA.
Email: francisco.polidoro@mccombs.utexas.edu

Abstract

Research Summary: Recognizing the role of complementors in creating value in interdependent platform ecosystems, strategy research has recently started to examine performance heterogeneity across complementors. However, research has thus far focused on the performance implications of dynamics unfolding within a particular ecosystem. We take a step toward exploring influences that arise beyond the focal ecosystem by focusing conceptually on multihoming. We argue that multihoming to another platform produces learning benefits that enhance a complementor's performance on the home platform, especially when dealing with a high level of interdependencies and having greater similarity to other complements. We find supportive evidence in our analysis of open-source software platforms between 2012 and 2018 and discuss implications for research on platform ecosystems, multihoming, and open-source software.

Managerial Summary: Prior studies viewed multihoming as an important strategy for complementors in platform ecosystems. However, little is known about the extent to which such expansion affects the performance of complementors on their home platforms. This study investigates this issue using data on software package complementors in a variety of platforms housed in GitHub, the world's largest repository of open-source software. The findings show that following multihoming, a complementor experiences a performance improvement in the home platform even when compared to the performance change observed during the same period for another complementor with similar attributes but that remains in single-homing. These findings underscore the strategic implications of multihoming as a significant driver of performance heterogeneity across complementors in platform ecosystems.

KEYWORDS

complementors, interdependencies, multihoming, open source, platform ecosystems

1 | INTRODUCTION

Interdependencies in value creation are an essential feature of platform ecosystems (Adner, 2017; Helfat & Raubitschek, 2018; Kapoor, 2018; Kretschmer et al., 2022). In examining the performance implications of interdependencies, research has emphasized the perspective of the firm leading a platform (e.g., Agarwal et al., 2022; Boudreau, 2010; Jacobides et al., 2018). More recently, scholars have shifted their focus toward complementors, showing that although interdependencies within an ecosystem are important sources of value creation, they engender adaptation challenges that adversely affect the performance of complementors (Agarwal & Kapoor, 2023; Burford et al., 2022; Chen, Yi, et al., 2022; Chen, Zhang, et al., 2022; Kapoor & Agarwal, 2017). However, research has overlooked the possibility that important influences on a complementor's performance might arise beyond a focal ecosystem.

Exploring this possibility is relevant because complementors frequently provide products or services on multiple platforms. This practice, often referred to as “multihoming,” is observed in several contexts, such as video games released for competing consoles (Cennamo et al., 2018), merchants with offerings across daily deals online platforms (Li & Zhu, 2021), and applications available both in the Apple App Store and on Google Play (Chen, Yi, et al., 2022; Chen, Zhang, et al., 2022). As a complementor's value proposition is interdependent with other elements in an ecosystem, multihoming to a new platform requires a complementor to adapt its offerings to the structure of the interdependencies of the ported platform (Cennamo et al., 2018; Chen, Yi, et al., 2022; Chen, Zhang, et al., 2022). Just as adaptation to a particular environment creates learning effects that benefit the firm in another environment (Chen et al., 2019), adaptation to a new platform ecosystem may create similar effects across ecosystems. In this study, we explore this possibility by asking how multihoming to a new platform affects a complementor's performance on the home platform.

To investigate this question, we focus conceptually on the performance implications of multihoming, while empirically accounting for the possibility that factors shaping its incidence may also relate to a complementor's performance on the home platform, as we detail later. Previewing the arguments that we develop in the theory section, we note that as a complementor adapts its offering to a new ecosystem, it also learns about upstream interdependent complements and downstream uses of its offering (Adner & Levinthal, 2002; Levinthal, 1998). These insights gained on the new platform can be valuable to the complementor on the home platform, alleviating rigidities that hinder the discovery of superior combinations of interdependent elements (Levinthal, 1997; Rivkin & Siggelkow, 2003). Following these arguments, we posit that multihoming has a positive effect on the complementor's performance on the home platform. Furthermore, we explore contingencies that help us probe the logic underlying our arguments while also highlighting heterogeneities across complementors in benefiting from multihoming. If, as we argue, multihoming produces learning benefits that alleviate rigidities related to interdependencies, then we would expect this effect to be stronger when the complementor deals with a higher level of interdependencies on the home platform (Ganco et al., 2020; Levinthal, 1997). Similarly, we would expect these learning benefits to be more pronounced when the complementor has greater similarity to other complements on the home platform, thus having incentives to differentiate its complement to offset competitive pressures from similar offerings (Boudreau, 2010; Kretschmer et al., 2022).

We tested these arguments using GitHub data on open-source software development platforms. With tens of millions of programs across more than 30 open-source platforms, GitHub is the world's largest repository of open-source software (He et al., 2020). Indeed, GitHub plays a vital role in open-source software development (Lin & Maruping, 2022) and functions as a form of meta-ecosystem, as the open-source development systems that it houses support software technologies that are relevant to other digital ecosystems, such as the Apple App Store and Google Play. Furthermore, although software developers can build on software modules developed independently from each other (Bonaccorsi & Rossi, 2003), interdependencies are a salient feature of the development of software applications, even in an environment that is intentionally designed to be modular (Kapoor & Agarwal, 2017). Using GitHub data enabled us to observe the upstream complements on which a focal complement builds as well as its downstream usage in the ecosystem, an important performance outcome for unpaid complements (Eckhardt, 2016). Also, unlike other ecosystems in which platform owners and price mechanisms play a vital role in coordinating interdependencies, by focusing on a setting where the use of unpaid complements is standard practice (Boudreau & Jeppesen, 2015), we can avoid spurious effects that could otherwise result from a platform owner's strategies and pricing mechanisms.

As we detail in the method section, we used a difference-in-differences (DiD) research design to address the possibility that the performance increase that a complementor experiences on the home platform following multihoming may be confounded by factors related to its decision to multihome in the first place. The analysis captured the causal effect of multihoming on a complementor's performance on the home platform relative to the performance change observed for a complementor on the same platform that, although similar to the focal complementor in several observable attributes expected to shape the incidence of multihoming, did not engage in multihoming. We found that multihoming complementors achieved higher performance on the home platform relative to analogous performance change observed among single-home complementors.

This study advances understanding of heterogeneity across complementors by shifting focus from influences arising within a focal ecosystem (Cennamo et al., 2018; Kapoor & Agarwal, 2017) to factors arising in another ecosystem. By doing so, this study underscores that a complementor's strategy in an ecosystem, in addition to being shaped by network effects that arise on the same side or across sides of a platform (e.g., Agarwal et al., 2022; Helfat & Raubitschek, 2018; Miller & Toh, 2022), is also subject to effects that arise across platforms. Furthermore, by showing that multihoming complementors experience performance gains on the home platform, this study reveals an intertemporal tradeoff overlooked in prior research, namely that multihoming engenders positive effects, as a complementor eventually ports learning back home, which partly counterbalances the immediate adaptation challenges that arise in the new platform. Finally, this study contributes to research on open-source platforms. As managing interdependencies is critical in this context, prior research has explored the role of platform firms in selectively granting access to their platforms in ways that foster the development of complements (Boudreau, 2010; Fosfuri et al., 2008). This study, in turn, underlines the role that a complementor's strategy to multihome plays in helping it deal with interdependencies on the home platform.

2 | THEORY AND HYPOTHESES

2.1 | Duality of interdependencies in platform ecosystems

Interdependencies in value creation are an essential feature of ecosystems (Adner, 2017; Adner & Kapoor, 2010; Kapoor, 2018; Kretschmer et al., 2022), and they are particularly relevant in digital platform ecosystems (Bonaccorsi & Rossi, 2003; Helfat & Raubitschek, 2018). Interdependencies fuel value creation, as participating firms connect their offers with interdependent elements in the ecosystem (Agarwal & Kapoor, 2023; Baldwin, 2020). The existence of complementary technologies increases opportunities for combining interdependent features such that the resulting offering creates more value for users (Lee & Kapoor, 2017), thereby positively affecting the performance of complementors. When examining the performance of apps in Apple's iOS and Google's Android smartphone ecosystems between 2012 and 2014, Kapoor and Agarwal (2017) found that the number of technological interdependencies to which an app was subject in an ecosystem increased the likelihood that it remained among the best performing apps. Similarly, when investigating the performance of apps launched in Apple's App Store between 2008 and 2015, Agarwal and Kapoor (2023) found that the number of complementary technologies in the iPhone ecosystem that a new complement is connected to positively affects its performance.

However, although interdependencies support value creation in a platform ecosystem, they also create adaptation challenges for complementors. As the number of elements in an ecosystem with which a complementor needs to interact to deliver its value proposition increases, the complementor faces higher levels of ecosystem complexity (Chen, Yi, et al., 2022; Chen, Zhang, et al., 2022; Kapoor & Agarwal, 2017). When adapting to an environment marked by high levels of interdependencies, organizations face increasing rigidity, as they tend to get stuck in local optima, making it increasingly challenging for them to find superior combinations of interdependent components (Levinthal, 1997; Rivkin & Siggelkow, 2003). Accordingly, research on platform ecosystems has demonstrated that interdependencies make adaptation more difficult for complementors as the ecosystem evolves. For example, Adner and Kapoor (2010) showed that changes in the technology underpinning the lithography tool, a core feature of the lithography ecosystem, affected the interaction between complementary technologies of the resist and mask. Likewise, Kapoor and Agarwal (2017) found that complexity made it more difficult for apps to maintain their performance in smartphone ecosystems following the introduction of new versions of the operating system underlying their respective platforms. Similarly, Agarwal and Kapoor (2023) showed that interdependencies can also expose complementors to performance bottlenecks as the underlying technology of the platform evolves. Burford et al. (2022), in turn, showed that changes in the structure of interdependencies in e-commerce websites adversely affect performance, making the search for new configurations in the ecosystem more difficult.

While showing that managing interdependencies is important for the success of complementors in a platform ecosystem, research has focused on the perspective of the platform owner or leading firm that orchestrates the ecosystem (Helfat & Raubitschek, 2018; Jacobides et al., 2018). For example, research has shown the role of a firm leading a personal digital assistant handset platform in granting access to independent developers of complements while ensuring interoperability with the platform and other elements in the ecosystem (Boudreau, 2010). Focusing on the Android ecosystem, Wen and Zhu (2019) showed that a leading firm, by entering a domain of complementary technologies (e.g., Google launching a flashlight app), induces complementors to allocate their innovation effort to other domains. By defining the core features of the technological infrastructure of a platform and having the prerogative to choose how and when to change that core technology, the firm leading a platform shapes the nature of interdependencies facing complementors (Agarwal & Kapoor, 2023; Kapoor & Agarwal, 2017).

However, even when a leading firm orchestrates a platform, a complementor can still decide which specific connections to establish with other complements in an ecosystem. For instance, "the Uber app in the smartphone ecosystem did not build its own mapping, payment, and communication technology." (Agarwal & Kapoor, 2023, p. 1219). Similarly, when creating a solution to a complex problem, a software developer can combine other modules of programs in a sequence of interdependent solutions (Bonaccorsi & Rossi, 2003). Therefore, complementors also influence the level of interdependencies with which they operate in a platform.

When examining the performance implications of interdependencies from a complementor's perspective, however, studies have explored factors arising within a particular ecosystem, overlooking influences that might arise beyond the focal ecosystem. This conceptual void is intriguing, given that a critical decision that a complementor can make is whether to multihome, that is, to provide its products or services on multiple platforms. The literature has highlighted multihoming as an important strategy for ecosystem complementors to gain access to more users and obtain cross-platform scale economies (Chen, Yi, et al., 2022; Chen, Zhang, et al., 2022; Chung et al., 2024; Li & Zhu, 2021).

Despite these incentives for complementors to engage in multihoming, doing so requires complementors to adjust their offerings to the specific nature of interdependencies in the ported platform to take full advantage of complementarities in that ecosystem (Schilling, 2000). As the costs of adapting to a new platform reduce the net benefit that a complementor can obtain from multihoming, higher levels of interdependencies in an ecosystem, by making adaptation more difficult, can dissuade a complementor from multihoming (Chen, Yi, et al., 2022; Chen, Zhang, et al., 2022). However, as Cennamo et al. (2018) showed, even when a complementor expects the potential benefits of multihoming to outweigh the adaptation costs, such practice can still result in complementors exhibiting lower-quality performance on the new platform. Although research has demonstrated that adaptation to the ported platform is critical to a complementor's success in multihoming, it has stopped short of examining the effects that such adaptation might have on the performance of a complementor on the home platform, a possibility we explore in the next section.

2.2 | Multihoming and complementor performance on the home platform

We start by noting that, although we focus on complementor performance on the home platform and not on the factors shaping the incidence of multihoming, we acknowledge that multihoming complementors may be inherently superior, such that they might be more likely than single-home complementors to experience subsequent performance increases on the home platform. Therefore, as we detail in the methods section, we empirically accounted for the potential confounding effect of factors associated with the incidence of multihoming.

As we discussed earlier, when multihoming, a complementor faces pressure to adapt its offering to the specific interdependencies of the ported ecosystem to deliver its value proposition in that new environment. For example, a video game developer releasing a game on a new console platform must adapt its game to specific features of the new platform (Cennamo et al., 2018), just as a developer of an app in the smartphone ecosystem launching its app in an alternative ecosystem must adjust its app to account for interdependencies in the new environment (Chen, Yi, et al., 2022; Chen, Zhang, et al., 2022). As adaptation to a particular environment creates learning effects that benefit the firm in another environment (Chen et al., 2019), we argue that in adjusting its offering to the specific interdependencies that exist in the new platform, a complementor gains insights that alleviate the rigidity that hinders its efforts to find superior combinations of interdependent components on the home platform (Levinthal, 1997; Rivkin & Siggelkow, 2003). Multihoming can create such a learning effect in two interrelated ways: exposure to *upstream* interdependent complements on the new platform can help a complementor spot opportunities to refine its complements on the home platform, and observation of *downstream* uses of complements on the new platform can help it identify functionalities that may be useful to users on the home platform.

Starting with the identification of *upstream* opportunities to refine complements, we note that, as the literature has demonstrated, alternative platforms in a given context, such as competing platforms for smartphone apps or for video games, differ in terms of the elements with which a complementor must interact to deliver value in those environments (Cennamo et al., 2018; Kapoor & Agarwal, 2017). Once a complementor launches its offering on a new platform, it can learn from its own efforts to adapt to a distinct set of interdependent complements available in that new environment and spot opportunities to build on new combinations of interdependent elements to deliver value (Ganco et al., 2020). Multihoming also amplifies opportunities for vicarious learning, exposing the complementor to various combinations of interdependent complements underlying the offerings in that environment that point to opportunities to create new configurations on the home platform mirroring those observed on the new platform. With more possibilities to experiment with, a multihoming complementor is in a better position than a single-homing complementor, whose experience is limited to a single ecosystem, to overcome the rigidity that hinders the finding of superior combinations of interdependent complements on the home platform.

Turning to learning about *downstream* uses of complements on a new platform, multihoming exposes a complementor to different uses of complements in the new environment, thus encouraging learning-by-using (Levinthal, 1998). This form of learning is relevant because, as the literature on technological change and adaptation has shown, the downstream uses of a technology in a particular environment reinforce the rigidity that constrains adaptation, as firms tend to focus on known uses and on improvements in the functionalities relevant to those uses (Adner & Levinthal, 2002; Cattani, 2006). However, as this same literature has also demonstrated, once a given technology is exposed to a new environment, where it is used by customers who may value other functionalities, that technology has the potential to significantly improve along those new functionalities, spurring new uses when these improvements are ported back to the original environment (Adner & Levinthal, 2002; Cattani, 2006; Levinthal, 1998). Multihoming complementors can learn about different uses of complements in the new platform, as well as functionalities that users in that platform value, thereby enjoying a greater scope of learning-by-using than single-home complementors. Insights about different uses and different functionalities can fuel a multihoming complementor's search for new configurations on the home platform that increase differentiation relative to other offerings, thereby helping the complementor alleviate the negative effect that similarity to other complements can otherwise have on its performance (Kapoor & Agarwal, 2017).

Whereas both the upstream and the downstream learning opportunities discussed above can benefit multihoming complementors gain an advantage over other complementors on the home platform, this advantage is drastically reduced if the complementors also experience systematically high adaptation costs when porting such learning back to the home platform. However, in contrast with adaptation costs that arise on a new platform, these costs should be lower when a complementor adapts its offering to the home platform, given its greater familiarity with that environment. Moreover, although both multihoming and single-home complementors face costs to refine their offerings on the home platform, the economies of scope associated with multihoming benefit the development of complementor-specific resources and capabilities (Chung et al., 2024). As a result, multihoming complementors have lower adaptation costs on the home platform than do single-homing complementors. Therefore, multihoming increases the potential benefits of adaptation on the home platform by helping complementors identify more opportunities for refining their offerings, while also contributing to reducing the associated costs relative to a single-home complementor, whose experiences are limited to a single ecosystem.

Illustrating the arguments above in the context of open-source software development platforms, we note that multihoming from PyPI to alternative open-source software development platforms such as Rust and Go exposes software developers to upstream complements that enable more flexible use of stack memory and heap memory, two different memory allocation techniques. Once software package complementors become more fully acquainted with these distinctions, they can develop complements for object referencing on the PyPI platform that incorporate those features.[1] In terms of learning about downstream uses of complements on a new platform, an example is the launch of plotnine, a software package in the PyPI platform that incorporates functionalities similar to those of ggplot2, a widely used software package offering graphing functionalities available on the R platform.[2] Another illustration is provided by rvest, a web-scraping software package on the CRAN platform that builds on features of Beautiful Soup, a software package for parsing HTML and XML documents available on the PyPI platform.[3] Furthermore, yhat, a complementor that originally offered software packages

[1]https://medium.com/@rajasekar3eg/rust-for-python-developers-ownership-and-borrowing-cd85fc10cae4.

[2]https://towardsdatascience.com/python-vs-r-the-basics-d754c45c1596.

[3]https://www.kdnuggets.com/2015/12/using-python-r-together.html.

on the PyPI platform and later multihomed to the CRAN platform, eventually launched on the home platform a software package offering functionalities for database management similar to those provided by a complement available on the ported CRAN platform.[4] The context of smartphone ecosystems also offers similar examples: as WeChat, a complementor providing instant messaging apps on both Android and iOS, reported in its blog, it started to provide a float window function in iOS during video streaming that allows for multitasking, which was available first in the Android ecosystem.[5] Additionally, Cydia developed an app for the iOS ecosystem, offering functionalities similar to those available in Android apps that allow users to modify wallpapers and app icons (Zhang et al., 2022).

In summary, multihoming engenders learning effects that help a complementor increase its performance on the home platform, as the complementor identifies opportunities to build on new *upstream* configurations of interdependent complements and to emphasize new *downstream* uses of complements to further differentiate its complement on the home platform. Therefore, we hypothesize the following:

> **Hypothesis (H1).** A complementor's multihoming to a new platform has a positive impact on its performance on the home platform.

Having explained above how multihoming can lead to performance differences between multihoming and single-homing complementors on the home platform, in the following sections we explore two contingencies that help us probe the logic underlying our arguments, namely a complementor's interdependencies and its similarity to other complements on the home platform. The arguments below also help us discuss heterogeneities across multihoming complementors in terms of the extent to which they can benefit from porting learning back to the home platform.

2.3 | Contingent effect of a complementor's interdependencies on the home platform

The level of interdependencies that a complementor experiences on the home platform can play a significant role in the extent to which it can benefit from the learning effects stemming from multihoming. As discussed above, interdependencies entail a tradeoff: just as they can help a complementor enhance the value of its offerings to users, which enhances its performance, they can also result in rigidities that can hamper adaptation, as a complementor must ensure that its offerings are compatible with interdependent elements in the ecosystem.

As the level of interdependencies on the home platform increases, the more that in managing this tradeoff the balance tilts toward ensuring appropriate coordination between interdependencies at the expense of diminished abilities to make further changes that could otherwise enhance the complementor's performance on that platform. However, the learning effects stemming from multihoming that we propose in H1 contribute to attenuating these rigidities that could otherwise hinder subsequent performance. In this case, the complementor's enhanced experience in dealing with interdependencies in another ecosystem and exposure to other combinations of interdependent elements in that ecosystem are particularly meaningful in helping alleviate these rigidities, in turn

[4] https://www.kdnuggets.com/2015/12/using-python-r-together.html.

[5] https://blog.wechat.com/2018/05/28/whats-new-in-wechat-6-6-7-for-ios-float-window-and-in-camera-translation/.

accentuating the positive effects of multihoming on the complementor's performance on the home platform. Conversely, when the complementor faces a low level of interdependencies on the home platform and, accordingly, faces fewer rigidities that could hamper its subsequent performance on that platform, the learning effects stemming from multihoming become less critical to further performance improvements. Thus, we hypothesize as follows:

> **Hypothesis (H2).** The more interdependencies a complementor faces on the home platform, the more multihoming to a new platform positively affects its performance on the home platform.

2.4 | Contingent effect of similarity to other complements on the home platform

The literature on platform ecosystems has underscored the tension surrounding similarity among complements within an ecosystem, which on the one hand helps advance the corresponding type of offerings but on the other hand increases competitive tensions[6] (Boudreau, 2010; Kretschmer et al., 2022). Although similarity to other complements can be beneficial, it also increases the pressure on a complementor to differentiate its offerings in an ecosystem to counter the potential adverse effects of competition. In settings with low barriers to entry, such as the context of digital technologies, similarity to other complements can also make a complementor more vulnerable to competitive threats from entrants offering differentiated products (Helfat & Raubitschek, 2018). In line with the insight that a firm faces incentives to engage in innovative search to offset competitive pressures from similar offerings (Toh & Polidoro, 2013), with higher similarity to other complements in an ecosystem, a complementor has incentives to differentiate its offering to partly offset the negative effect of competition on the usage of its complement (Eckhardt, 2016).

Therefore, when a complementor has a high level of similarity to other complements in a platform ecosystem, differentiating its complement is critical to its performance on that platform. In that situation, the insights it gains into different uses in another environment and distinctive features that are relevant to users in that environment play a key role in helping the complementor meet the greater need for differentiation. Through exposure to another platform, a complementor can gain insights about features that users in that platform value and that complementors on the home platform have overlooked. In contrast, when the complementor shows little similarity to other complements on the home platform, those learning benefits are less critical to furthering its performance on the home platform. While multihoming in this case can still enable the complementor to gain insights into changes in its offering that are meaningful to users, those insights are less critical, as the need to act on them is less pronounced than in the prior situation. Accordingly, we hypothesize:

> **Hypothesis (H3).** The greater a complementor's similarity to other complements on the home platform, the more multihoming to a new platform positively affects its performance on the home platform.

[6]This insight mirrors the argument that the presence of similar organizations in a market produces legitimating benefits that increase the survival prospects of those organizations, but as more similar organizations exist in a market, the resulting competitive tensions supersede those benefits, negatively affecting survival prospects (Baum, 1996).

3 | DATA AND METHODS

3.1 | Empirical setting

To examine the effect of multihoming on a complementor's performance on the home platform, we focused empirically on open-source software development ecosystems housed in GitHub, the world's largest repository of open-source software (He et al., 2020). The activity of complementors in this setting is consequential for other contexts as well, as the software packages available on open-source software platforms support activities that are essential to a variety of other digital ecosystems, such as cryptography, data analytics infrastructure, and the development of applications for smartphones. For example, software packages available on SwiftPM, one of the software platforms in GitHub, are used by third-party software developers to create applications for Apple's iOS platform ecosystem (Rahkema & Pfahl, 2022).

Also, given the nonproprietary nature of open-source software, multihoming in this context is not driven by price considerations or competitive tensions between platform owners and complementors. In some platform ecosystems examined in prior strategy research, such as studies of games created for videogame consoles (Cennamo et al., 2018) or applications launched in Apple's App Store (Agarwal & Kapoor, 2023), the firm leading the platform controls complementors' access to the ecosystem and sometimes directly competes with them (Wen & Zhu, 2019; Zhu & Liu, 2018). In contrast, open-source software package developers can freely select the platform(s) on which they make their technologies available. This allows for a sharper focus on the perspective of complementors when examining the implications of multihoming on their performance, as the influences associated with platform-leading firms shown in other contexts do not apply in this setting.

Furthermore, software package complementors in this setting connect their offerings with other complements available on the respective software development platform, thus instantiating the notion of interdependencies among complementors in an ecosystem. Software development benefits from modularization in that each software module can be developed independently (Bonaccorsi & Rossi, 2003). However, as Kapoor and Agarwal (2017) showed in the context of smartphone ecosystems, interdependencies are a salient feature of the development of software applications even in an environment that is intentionally designed to be modular. Managing interdependencies is critical to open-source software package complementors, given the large variety of modules they can choose from to create offerings that deliver value to users. Moreover, changes to these upstream packages affect the complementor's innovation efficiency in downstream uses, as well as the scope of functions that can be applied downstream. For example, when the core upstream functions on the PyPI platform underwent a major version update from 2.7 to 3 in 2020, all corresponding libraries, including famous AI libraries (e.g., Tensorflow) and data processing libraries (e.g., pandas) also had to update their software packages.[7] Moreover, modules used in open-source development experience changes with considerable frequency, sometimes within days or even hours,[8] in contrast with other settings such as the hardware components of mobile devices, which usually remain stable during several months, sometimes years.

Further, multihoming is a relevant strategy for complementors offering open-source software packages. As Lerner and Tirole (2002) have demonstrated when examining the economics of open-source environments, these settings do not imply the absence of financial incentives

[7] https://www.python.org/doc/sunset-python-2.

[8] On average, the complementors in our dataset make 12 changes in their software packages each month.

but rather offer “delayed rewards” such as “shares in commercial open-source-based companies or future access to the venture capital market,” as well as incentives to send signals of underlying talent to relevant audiences such as peers, the labor market, and the venture capital community (pp. 213–214). By making their complements available through open-source public repositories, software developers can display their technological capabilities to other members of the ecosystem and gain increased visibility. Also, mirroring the notion that an important reason for complementors to engage in multihoming is to gain access to more users and to obtain cross-platform scale economies (Chen, Yi, et al., 2022; Chen, Zhang, et al., 2022; Chung et al., 2024; Li & Zhu, 2021), multihoming enables an open-source software package complementor to increase the number of users of its open-source software, thereby enhancing the pool of potential users that can adopt any closed-source software it may decide to eventually launch. Similarly, such a complementor also has incentives to port learning back to the home platform, as doing so can increase the value of its offering on the home platform and reinforce its visibility on that platform. With more than 30 distinct open-source software development platforms, this setting offers complementors a greater variety of platform ecosystems relative to contexts examined in prior studies, such as app developers operating in the iOS or Android smartphone ecosystems (Kapoor & Agarwal, 2017) or game developers introducing games in the Sega or Nintendo console platforms (Shankar & Bayus, 2003). GitHub reports data on all open-source programs that use a software package, allowing us to empirically observe the extent to which a complement is used on a platform, an important performance outcome for complementors that provide unpaid complements (Eckhardt, 2016).

Finally, the very large number of software packages available across the variety of software development platforms housed in GitHub provides substantial empirical traction that enabled us to pair a complementor that engaged in multihoming with another complementor that is similar in a variety of observable attributes but that remained on a single platform, a feature of the data that we use in our DiD research design, as we explain in Section 3.6.

3.2 | Data sources

GitHub documents all the open-source software development activities in Git-based development process and compiles the metadata that can be publicly retrieved by the GitHub Application Programming Interface (API) service (Dabbish et al., 2012). We collected data on open-source software packages from 2012 to 2018 from GitHub Archive, which contains JavaScript Object Notation (JSON) files on development activities from GitHub API and compiles those metadata files into data dumps on an hourly basis since 2012 (Grigorik, 2012). These JSON files contain the original information of each GitHub activity, including updates and changes, discussions, and pull requests recorded by the GitHub API. We supplemented these data with data from GH Torrent, which compiles the real-time data available in the JSON files into a relational database at the activity, project, individual, and organizational levels (Gousios, 2013). Finally, we obtained data on software package platforms from libraries.io, which contains data on 34 major software management platforms.[9] This source monitors the

[9]Specifically, we obtained data on the following platforms: Alcatraz, Atom, Bower, CPAN, CRAN, Cargo, Carthage, Clojars, CocoaPods, Dub, Elm, Emacs, Go, Hackage, Haxelib, Hex, Homebrew, Inqlude, Jam, Julia, Maven, Meteor, NPM, Nimble, NuGet, Packagist, PlatformIO, Pub, Pypi, Rubygems, Shards, Sublime, SwiftPM, and Wordpress (data retrieved from the official data site of libriaries.io on September 28, 2020).

package complements on those platforms and compiles their usage information by analyzing the entire universe of open-source program code repositories in GitHub. It provides data on upstream dependencies required for implementing a focal code repository (Nesbitt & Pompilio, 2016) as well as on its downstream usage. These detailed data enable us to run analysis at the complementor-month level, as reflected in the description of the study variables and research design.

3.3 | Dependent variable

Exposure to users is an important measure in settings marked by complementarities (Burford et al., 2022). Specifically, usage by other complements is an important performance outcome for complements available in ecosystems in the absence of a price mechanism (Eckhardt, 2016). We thus operationalized a complementor's performance as the extent to which its software package was used on a software development platform, based on data from libraries.io, which compiles a comprehensive set of usage data for all software packages on software platforms in GitHub. Once a package is called as a function to implement a new software program, the event is documented, specifying that the installation of the focal package becomes a prerequisite for the functioning of that software program. Therefore, instances in which a complementor's packages are documented as installation prerequisites capture the extent to which its offerings are used on the software platform. To account for skewness, the variable *Installed repositories* contains the logarithm of the number of programs on GitHub that called a complementor's packages in a software platform ecosystem by the focal month.[10]

3.4 | Independent variable and contingent variables

3.4.1 | Multihoming

As we discuss later, we use a DiD research design to capture the performance change observed for a multihoming complementor relative to analogous change observed for a single-home complementor. In line with recent studies (e.g., Miller et al., 2021; Wen & Zhu, 2019), we used a time-invariant variable to distinguish between observations subject to treatment (i.e., multihoming complementors) and their counterfactuals (i.e., single-home complementors). Specifically, we created the dummy variable *Multihoming*, set to one if a focal complementor provided packages on more than one software platform, and zero otherwise.

3.4.2 | Post

As we discuss later, for a complementor that engaged in multihoming, our analysis includes observations in the months immediately before and immediately after multihoming. To indicate whether a focal observation refers to a month after multihoming, we used the dummy variable *Post*, set to one if, by the focal month, a complementor had published packages on an additional

[10]All of the variables entered as logarithms in this study captured the natural logarithm of the original count added to one.

platform other than its home platform and to zero otherwise. As we explain later, the analysis paired a complementor that engaged in multihoming with a complementor that remained exclusively on the home platform. The dummy variable *Post* in these cases was coded according to the month of multihoming of the complementor for which it functioned as a counterfactual. To capture the effect that H1 hypothesized, the analysis adds the interaction term *Multihoming* × *Post*.

3.4.3 | Interdependencies

As we discussed earlier, in the context of open-source software, interdependencies are reflected in dependency relationships, as a program often depends on other programs, whose functionalities the focal program builds on to deliver its own functionalities (Bonaccorsi & Rossi, 2003). To capture the level of interdependencies that a complementor exhibits on its home platform, we use data on dependency relationships in GitHub to observe all the software programs that the complementor's software package builds on, irrespective of the extent to which these upstream programs are used in other packages. Specifically, the time-varying variable *Interdependencies* contains the number of upstream dependencies that a complementor's packages have built on, by the focal month, on the home platform. To account for skewness, the variable contains the logarithm of this count.

3.4.4 | Similarity

To operationalize a complementor's similarity to other complements on the home platform, we first mapped all other complements on the same platform that target the domains of use of the focal complement by the focal month, as indicated in the keywords for the application domains of complements in GitHub. To ensure that our measure focused on the focal complement's keywords representing the most relevant domains of use on the platform, we considered those that appeared among the top five most frequently used by all complements on the platform, irrespective of the use of those complements. We measured similarity based on the extent to which the upstream packages used by a focal complement by a focal month overlapped with the upstream packages most frequently used by all the other complements in those relevant domains of use.[11] Specifically, we created the time-varying variable *Similarity*, calculated as follows:

$$Similarity_{i,t} = \left(\sum_{1}^{K} \left(\frac{\sum_{1}^{J} c_{kj,t}}{J} \right) \right) / K$$

where K is the number of the most relevant domains of use among the keywords assigned to the focal complement i. Within each of these domains of application, as reflected in the

[11]Restricting the comparison to the set of upstream packages most frequently used by the other complements in a given domain is relevant to avoid incidental overlaps with upstream packages of the focal complement, as this ensured that the measure captured similarity to the upstream packages that were most critical in that domain.

respective keyword k, J is the number of packages that the focal complement, i, builds on, and $c_{kj,t}$ is a dummy variable that takes the value of one if a package j used by the focal complement i is among the top 20 upstream packages that are most frequently used by other complements available in the same domain by month t, and zero otherwise. In other words, this measure captures the extent to which other complements available in the same domain of the focal complement by the observation month draw on functions similar to those of the focal complement. It ranges from zero to one, with higher values denoting higher levels of similarity. For complementors offering multiple complements on the focal platform, this variable contains the average similarity score across its complements on the home platform by the focal month. As we report later, results are robust to alternative ways to measure this variable.

3.5 | Control variables

The analysis accounted for several sources of heterogeneity across complementors and across platforms that may relate to both the independent variables and the dependent variable, as summarized in Table A1. To begin with, the analysis accounted for the main effect stemming from *Interdependencies* and *Similarity* when testing the contingency effects in H2 and H3. Regarding heterogeneity across complementors, a potential concern is that those that more frequently updated their programs on a platform were more likely to experience performance improvements, while the experience gained in the process may also have increased their inclination to multihome. Models account for this potential influence by adding the variable *Pushes* with the total number of times the complementor modified the source files of its packages by the focal month (Saadat et al., 2020). A similar conjecture is that a complementor's ability in spotting opportunities for improving its complements might positively affect its performance on the home platform, while also encouraging multihoming. The analysis addressed this conjecture by adding to models the variable *Improvement issues* containing the number of suggestions for improvements and potential errors posted for the focal complementor's packages, as reported in the "issue" section of their respective GitHub repositories by the focal month (Zhang & Yang, 2022). Likewise, a complementor that garnered a greater number of contributions in the open-source environment may be better able to improve its performance on the home platform, while the greater visibility associated with these contributions and the attendant knowledge sourced through them may render that complementor more likely to multihome. To control for this possibility, models add the variable *Pull requests* with the logarithm of the number of external contributions submitted through the "pull" function to the complementor's package source code repositories in GitHub up to the focal month (Kononenko et al., 2018). Similarly, models also add the variable *Forks* with the logarithm of the number of instances in which other repositories share the focal package and visibility settings to iterate on ideas or propose changes (Haefliger et al., 2008). Further, the extent to which a complementor attracts the attention of other actors in a platform ecosystem may be related to its intrinsic quality and accordingly result in superior performance, and such salience may also be systematically related to a greater propensity to multihome. To control for this conjecture, models add the number of people subscribed to receive updates on the latest development of a complementor's source code repositories, entered in the variable *Watchers* as the logarithm of that count to account for skewness. To account for the potential influence of a complementor's efforts to use documentation as a way of promoting its offerings on the home platform, models enter the dummy variable *Documentation*, set to one if at least one of the focal complementor's source code repositories contained a Readme document, and zero otherwise.[12] Further, although prior research shows that complementors in digital platform ecosystems typically multihome by adapting existing products to the new platform (Cennamo et al., 2018), it is possible that a complementor might launch a new product on both the new platform and the home platform. To account for this conjecture, we controlled for the *Number of packages* that a complementor published on the home platform by the focal month. Finally, the analysis also accounted for any remaining time-invariant heterogeneity across complementors with complementor fixed effects, as well as temporal influences with month fixed effects.

3.6 | DiD research design

Endogeneity concerns surrounding examination of the effect of multihoming on complementor performance on the home platform create challenges for interpretation of such relationship as a causal effect. Although the analysis, as discussed above, controlled for several sources of heterogeneity, the possibility remains that multihoming complementors and single-home complementors exhibit systematically distinct levels in these control variables, posing the risk of biased estimates (Rose & Van der Laan, 2009). Additionally, a complementor's expectation of improving its performance on the home platform may induce its decision to multihome, leading to reverse causality concerns. To address these concerns, we adopted a DiD research design with a matched sample: for each complementor that entered another platform ecosystem, we identified a counterfactual complementor on the same platform that, although similar to the original complementor in several attributes, remained exclusively on that platform. With this design, we can examine the change in the performance of a complementor on the home platform following multihoming relative to the performance change experienced in the same period by a counterfactual complementor that was not subject to the treatment condition, that is, multihoming.

To construct the matched sample, we first identified all complementors that launched software packages on additional software development platforms from 2012 to 2018. To avoid the confounding effect that can arise when a complementor sequentially expands to multiple platforms, we restricted the sample to observations of complementors that multihomed to one additional platform only. This procedure resulted in 22,456 multihoming complementors. Next, we identified all complementors on the respective home platforms that did not engage in multihoming, totaling 26,221 single-home complementors on the same platform as those that experienced multihoming. Following this, for each complementor in the first group of observations (i.e., multihoming complementors) we searched for a matching complementor in the second group of observations (i.e., single-home complementors) that existed on the same platform within the same year (i.e., 12-month window) in which we observe the multihoming complementor and that exhibited similar levels of the contingency and control variables discussed earlier, using a Coarsened Exact Match (Blackwell et al., 2009).[13,14] This procedure resulted in a

[12]As reported later, a sensitivity analysis did not show evidence of influences stemming from other promotion activities.

[13]As we explain in a later section, sensitivity analysis using PSM (Rosenbaum & Rubin, 1983) as an alternative approach to build the sample showed robust results.

[14]By considering as treated observations only complementors that multihomed to one additional platform, the analysis mitigates concerns with bias in staggered DiD estimates stemming from treatment effect heterogeneity (Baker et al., 2021; Callaway & Sant'Anna, 2021). Additionally, pairing them with counterfactual complementors based on all those covariates at time of multihoming and the attendant randomness in residual influence on the incidence of multihoming further allay such concerns.

sample containing 7821 complementors subject to the treatment condition expected to affect performance (i.e., multihoming) and the corresponding 7821 counterfactual complementors that remained in the control condition (i.e., single-home).

As we explained earlier, the setting of open-source software development is a fast-paced environment, with changes in software packages occurring in short periods of time, sometimes within a few days or even hours. Also, changes in upstream interdependent complements and in downstream uses of open-source software packages are immediately observable by other actors in the platform ecosystem. Users in this setting, themselves software programmers, are equipped with skills to observe those changes and make decisions about whether to use certain software packages. These considerations suggest that the performance implications of multihoming can be observed within a few months post-multihoming.[15] Therefore, to capture the effects of multihoming, we compared the performance of complementors in the 6 months post-multihoming with their performance in the previous 6 months, resulting in a final sample with 173,853 complementor-month observations.[16] As we report later, the results were robust to alternative time windows.

3.7 | Estimation

To test how multihoming affects the performance of a complementor on its home platform, we estimated the following equation:

$$Installed\ repositories_{it} = \beta_m Multihoming_i + \beta_p Post_{it} + \beta_{mp} Multihoming_i \times Post_{it} + BX_{it} + \beta_i + \varepsilon_{it} \quad (1)$$

where β_m captures the extent to which the performance on the home platform of complementors that engaged in multihoming differs from the performance of their respective counterfactuals, while β_p refers to changes in performance in months post-multihoming. In line with the implementation of DiD in economics and strategy research (e.g., Miller et al., 2021; Wen & Zhu, 2019), the DiD effect was captured through the inclusion of an interaction between the time-invariant variable denoting observations subject to the treatment effect (i.e., *Multihoming*) and the variable distinguishing between observations referring to the time periods before and after treatment (i.e., *Post*). Specifically, the coefficient β_{mp} captures the DiD effect that is, how the difference in performance in months post-multihoming relative to performance in pre-multihoming months differs between complementors subject to treatment condition (i.e., multihoming) and their respective counterfactuals (i.e., single-home). B is a vector of coefficients on control variables $X_{i,t}$, β_i denotes intercepts, and ε_{it} denotes the error term. With the addition of complementor fixed effects (δC) and month fixed effects (ϑT), $Multihoming_i$ and $Post_{it}$ no longer apply, leading to the following equation:

$$Installed\ repositories_{it} = \beta_{mp} Multihoming_i \times Post_{it} + BX_{it} + \delta C + \vartheta T + \beta_i + \varepsilon_{it} \quad (2)$$

To test the contingency effects in H2 and H3, we follow the standard practice in recent implementations of DiD design with contingency effects of adding the two-way interactions to

[15]This time also aligns with the evidence that developers of apps on the Android platform reacted within the first few months following the threat of Google's launch of similar complements (Wen & Zhu, 2019).

[16]Some complementors start multihoming within 6 months after their initial package release on the home platform. The results were fully robust in models dropping those cases.

control for the possibility that a contingency affects different groups of observations differently (e.g., Chen, Zhang, et al., 2022; Wu & Zhu, 2022). That is, we interact the term capturing the DiD effect (i.e., $Multihoming_i \times Post_{it}$) with the contingency variable, while controlling for the two-way interactions. Accordingly, we expanded Equation 2 into the equation below, in which the variable $Contingency_{it}$ refers to the contingent variables in H2 and H3, respectively *Interdependencies* and *Similarity*:

$$\begin{aligned} Installed\ repositories_{it} = {} & \beta_{mpc} Multihoming_i \times Post_{it} \times Contingency_{it} \\ & + \beta_{mp} Multihoming_i \times Post_{it} + \beta_{mc} Multihoming_i \\ & \times Contigency_{it} + \beta_{pc} Post_{it} \times Contigency_{it} + BX_{it} + \delta \mathrm{C} + \vartheta \mathrm{T} + \beta_i \\ & + \varepsilon_{it} \end{aligned} \tag{3}$$

We estimated the equations above using ordinary least square (OLS) regressions, which facilitate the interpretation of results. As we report later, analyses using Poisson and negative binomial estimation showed robust results.

4 | RESULTS

4.1 | Descriptive statistics

Table 1 reports the *t*-statistics for the difference in the average values of the study variables, revealing that the multihoming complementors in the full sample exhibited systematic differences in those variables relative to the single-homing complementors. This table also indicates that the matching procedure corrected those imbalances, as the multihoming complementors exhibited study variable values similar to those of the corresponding counterfactual single-home complementors, thus assuaging endogeneity concerns.

Before estimating the effect of multihoming on a complementor's performance, we examined whether the data showed patterns in line with the effects we hypothesized. Figure 1 plots the differences in average values of the dependent variable between the 7821 multihoming complementors and the 7821 single-home complementors for the months pre- and post-multihoming, with 95% confidence intervals. It shows that these confidence intervals include zero in the pre-multihoming months and that after the second month post-multihoming, these intervals lie above zero. This pattern aligns with our proposition that multihoming complementors experience an increase in performance on the home platform beyond the performance change observed among single-home complementors.

Further, Table 2 presents the descriptive statistics for the matched sample. Table 3 reports the summary statistics of complementors' performance for the two groups of observations, one subject to the treatment condition (i.e., $Multihoming = 1$) and another with their corresponding counterfactuals (i.e., $Multihoming = 0$). For each group, this table shows the average value of *Installed repositories* for complementors in observation months before multihoming ($Post = 0$) and after multihoming ($Post = 1$). It reveals a positive DiDs (0.066, $p < .01$), indicating that the change in performance for complementors post-multihoming (0.143) was greater than the analogous performance change observed among their corresponding counterfactuals (0.077). This pattern also aligns with H1.

TABLE 1 Characteristics of multihoming and single-homing complementors in full sample and in matched sample.

	Full, unmatched sample			**Matched sample**		
	Treated (*Multihoming* = 1)	**Control (*Multihoming* = 0)**	***t*-Statistic**	**Treated (*Multihoming* = 1)**	**Control (*Multihoming* = 0)**	***t*-statistic**
Interdependencies	0.990 (0.009)	0.928 (0.001)	7.641 [.000]	0.386 (0.009)	0.384 (0.009)	0.138 [.890]
Similarity	0.084 (0.001)	0.074 (0.000)	6.757 [.000]	0.049 (0.002)	0.045 (0.002)	1.240 [.215]
Pushes	4.607 (0.009)	3.911 (0.001)	64.11 [.000]	4.018 (0.015)	4.004 (0.015)	0.646 [.519]
Improvement issues	0.175 (0.005)	0.159 (0.000)	3.749 [.000]	0.018 (0.003)	0.015 (0.002)	1.149 [.251]
Pull requests	0.013 (0.001)	0.009 (0.000)	3.359 [.001]	0.003 (0.001)	0.003 (0.001)	0.625 [.532]
Forks	0.554 (0.007)	0.642 (0.000)	−11.43 [.000]	0.194 (0.006)	0.193 (0.007)	0.066 [.947]
Watchers	0.403 (0.007)	0.385 (0.000)	2.452 [.014]	0.076 (0.005)	0.071 (0.004)	0.830 [.407]
Documentation	0.835 (0.064)	0.854 (0.004)	4.9651 [.000]	0.835 (0.004)	0.835 (0.004)	0.084 [.933]
Number of packages	3.788 (0.064)	3.635 (0.004)	2.113 [.035]	2.397 (0.077)	2.384 (0.095)	0.107 [.915]

Note: Standard errors in parentheses. *p*-values for two-sided tests in brackets.

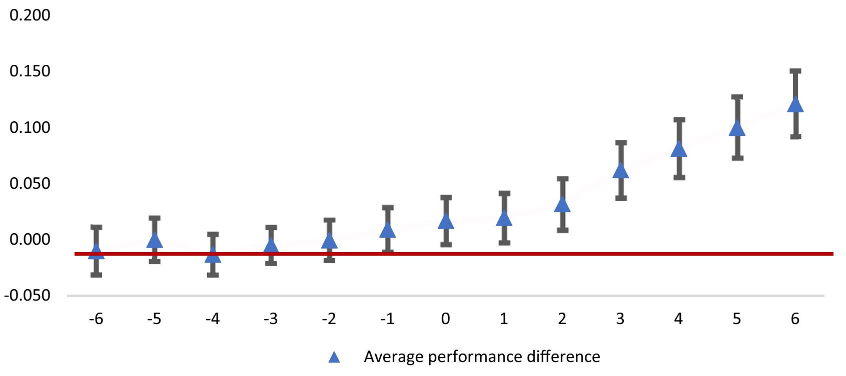

FIGURE 1 Average differences in *Installed repositories* (logged) for complementors with and without multihoming. This figure illustrates the average performance difference (*Installed repositories*) of treated (i.e., *Multihoming* = 1) and control (i.e., *Multihoming* = 0) complementors in months pre- and post-multihoming.

4.2 | Assessment of parallel trend assumption and placebo test

We adopted the procedure used in recent research in economics (Babina et al., 2023; Miller et al., 2021) and strategy (Chen, Zhang, et al., 2022; Zhang & Yang, 2022) to assess whether the patterns in our data aligned with the assumption in the DiD analysis of the parallel trend pre-treatment between the treated and non-treated observations. Specifically, we estimated the dynamic treatment effect by interacting *Multihoming* × *Month* for each month in the pre and post treatment periods. The results, shown in Model 1 in Table A2, do not support the inference that the coefficients on these interactions for the *pre*-treatment months are different from zero, thus validating the parallel trend assumption. Those results also reveal positive coefficients on the interaction terms for months *post multihoming*, which aligns with our argument that multihoming has a positive effect on the performance of complementors on the home platform. Figure 2a plots the average differences in the estimated effects between the treatment and control groups, as well as the corresponding 95% confidence intervals, for each month in the analysis period. Visual inspection of this figure reveals that the confidence intervals of the performance difference between observations in the treated and in the control groups include zero in all pre-multihoming months, which is congruent with the parallel trend pre-treatment assumption. Additionally, this figure shows that for each observation month post-multihoming, the 95% confidence interval with the performance difference for both groups of observation lie above zero, indicating that multihoming resulted in complementors' experiencing a greater positive change in their performance on the home platform relative to the performance change observed among single-home complementors. This pattern is also consistent with H1.

Additionally, in line with recent research (e.g., Babina et al., 2023; Miller et al., 2021; Wen & Zhu, 2019; Zhang & Yang, 2022), we ran a placebo test[17] to further probe the validity of our empirical approach. The results in Model 2 of Table A2 and the corresponding visualization in Figure 2b, analogous to Figure 2a, show that, as expected, no treatment effect was observed in this case.

[17]In this test, we artificially shifted treatment time back by 6 months such that posttreatment months captured the period between 6 months and 1 month prior to actual treatment, when a treatment effect should not be observed.

TABLE 2 Descriptive statistics (N = 173,853).

		Mean	s.d.	1	2	3	4	5	6	7	8	9	10	11
1	Installed repositories	0.28	1.11											
2	Multihoming	0.51	0.50	0.01										
3	Post	0.45	0.50	0.05	0.00									
4	Interdependencies	0.39	0.77	0.13	0.00	0.04								
5	Similarity	0.05	0.19	0.10	0.00	0.03	0.33							
6	Pushes	4.01	1.41	0.12	−0.03	0.17	0.08	0.03						
7	Improvement issues	0.02	0.23	0.07	0.01	0.02	0.01	0.05	0.03					
8	Pull requests	0.00	0.09	0.02	0.01	0.01	0.01	0.00	0.02	0.17				
9	Forks	0.21	0.61	0.14	0.00	0.07	0.01	0.08	0.22	0.05	−0.01			
10	Watchers	0.08	0.43	0.06	0.01	0.04	0.00	0.06	0.07	0.62	0.15	0.04		
11	Documentation	0.83	0.36	0.07	0.00	0.01	0.04	0.02	0.06	0.03	0.01	0.09	0.05	
12	Number of packages	2.45	7.88	0.03	0.00	0.02	−0.03	−0.01	0.14	0.04	0.00	0.21	0.09	0.05

TABLE 3 Difference-in-differences in *Installed repositories*.

	Pre	Post	Difference
Treated (*Multihoming* = 1)	0.230 (0.004)	0.372 (0.006)	0.143 (0.007)
Control (*Multihoming* = 0)	0.234 (0.004)	0.311 (0.006)	0.077 (0.007)
Difference-in-differences			0.066 (0.012)
t-Statistic	0.634	6.966	5.640
p-Value	.526	.000	.000

Note: Standard errors in parentheses.

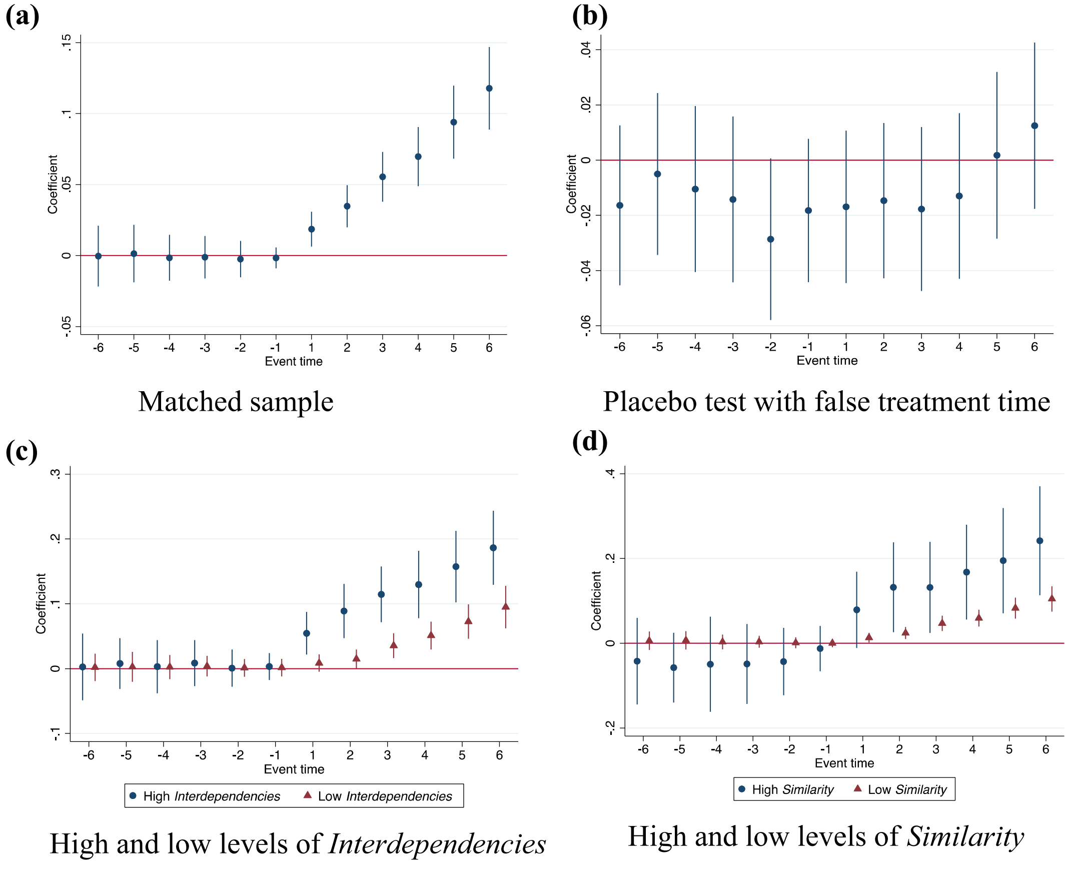

FIGURE 2 Multihoming and complementors' performance in home platform. Figure 2 plots the coefficients of dynamic treatment effects of multihoming with 95% confidence intervals for months pre and post multihoming. (a) The coefficients for the entire matched sample based on Model 1 of Table A2. (b) The placebo test with a false treatment time (6 months before actual multihoming). (c, d) The effect of multihoming over time by high and low levels of contingency effects of *Interdependencies* and *Similarity* (see Sections 4.1, 4.2, and 4.3 for additional information).

4.3 | DiD regression analysis

Table 4 presents OLS estimates of influences on complementors' *Installed repositories*. Model 1 estimates the influences on the performance of complementors according to Equation 1, without fixed effects for complementors and months. Model 2 includes these fixed effects, in line with Equation 2. Model 3 additionally includes the three-way interaction terms used to test the contingent effects predicted in H2 and H3, as well as the corresponding two-way interactions, as per Equation 3. The VIF scores were below the threshold of 10 as well as the more stringent threshold of 5 (Kleinbaum et al., 1988) across all models, thereby mitigating concerns about collinearity. Table A3 shows comparable results in partial models with the gradual inclusion of these variables. Table A3 also shows equivalent results in models entering orthogonalized values for interactions terms, further allaying potential concerns regarding collinearity. Consistent with H1, the coefficient on *Multihoming* × *Post* is positive ($\beta = .028$, $p = .004$ in Model 3 in Table 4), revealing a 2.8% increase in the performance of a multihoming complementor in a post-multihoming month, or a compounded increase of 18% within 6 months following multihoming, after accounting for other influences, including complementor and month fixed effects.

In line with H2, Model 3 shows a positive coefficient on the interaction term *Multihoming* × *Post* × *Interdependencies* ($\beta = .046$, $p = .009$). The findings reveal that the performance of a multihoming complementor increases by 6.3% in a post-multihoming month when *Interdependencies* increase by one standard deviation. In line with H3, Model 3 reveals that the coefficient on the interaction term *Multihoming* × *Post* × *Similarity* term is positive ($\beta = .129$, $p = .032$). The results indicate a performance increase of 5.3% in a post-multihoming month when *Similarity* increases by one standard deviation. These findings strongly support H2 and H3.

Figure 2c,d illustrates these contingency effects by plotting the average differences in the estimated effects between the treatment and control groups, as well as the corresponding 95% confidence intervals, for each month in the analysis period, for observations with high (i.e., at or above the mean) or low (i.e., below the mean) values of the contingency variables, *Interdependencies* and *Similarity*, respectively. Consistent with H2 and H3, both figures show that the complementors that engaged in multihoming experienced performance increases following the multihoming and that this effect was greater at higher levels of these contingency variables.

4.4 | Sensitivity analyses

4.4.1 | Alternative time windows around treatment

Although, as we discussed earlier, considering 6 months pre and postmultihoming is appropriate to our analysis, a potential concern is that the results showing the effect of multihoming on the performance of complementors on the home platform might be sensitive to this specific window around the treatment. To investigate this possibility, we reran the analysis, dropping the first two post-multihoming months to allow more time to elapse before observing the performance implications. Model 4 in Table 4 shows robust results. Additionally, we reran the models with a narrower window (i.e., 3 months) and broader windows (i.e., 9 and 12 months) around the treatment. Models 5–7 in Table 4 show similar results.

4.4.2 | Robustness to count model estimation

We ran additional models using Poisson and negative binominal to estimate influences on the number of complementors' installed repositories. Table A4 shows robust results, thereby

TABLE 4 Difference-in-differences estimates of influences on *Installed repositories*.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
				Sensitivity to alternative time windows around multihoming			
	Partial model without fixed effects	**Partial model with fixed effects**	**Main model**	**Excluding first 2 months post-multihoming**	**3-Month window**	**9-Month window**	**12-Month window**
Multihoming × Post (DD, H1: $\beta > 0$)	0.024 (0.037)	0.034 (0.001)	0.028 (0.004)	0.031 (0.009)	0.023 (0.003)	0.030 (0.011)	0.030 (0.022)
Multihoming	0.018 (0.289)						
Post	0.034 (0.000)						
Multihoming × Post × Interdependencies (H2: $\beta > 0$)			0.046 (0.009)	0.050 (0.017)	0.053 (0.000)	0.048 (0.016)	0.051 (0.021)
Multihoming × Interdependencies			0.030 (0.522)	0.030 (0.564)	0.052 (0.244)	0.084 (0.085)	0.110 (0.032)
Post × Interdependencies			0.039 (0.000)	0.048 (0.000)	0.018 (0.014)	0.046 (0.000)	0.052 (0.000)
Multihoming × Post × Similarity (H3: $\beta > 0$)			0.129 (0.032)	0.143 (0.044)	0.094 (0.053)	0.125 (0.069)	0.113 (0.128)
Multihoming × Similarity			0.174 (0.139)	0.174 (0.167)	0.236 (0.055)	0.175 (0.153)	0.136 (0.275)
Post × Similarity			−0.082 (0.014)	−0.096 (0.019)	−0.047 (0.041)	−0.079 (0.048)	−0.078 (0.082)
Interdependencies	0.157 (0.000)	0.220 (0.000)	0.152 (0.000)	0.163 (0.000)	0.095 (0.001)	0.161 (0.000)	0.173 (0.000)
Similarity	0.252 (0.000)	0.131 (0.040)	0.010 (0.863)	0.000 (0.994)	−0.067 (0.077)	0.036 (0.553)	0.077 (0.252)
Improvement issues	0.239 (0.012)	0.092 (0.129)	0.091 (0.125)	0.104 (0.135)	0.055 (0.230)	0.153 (0.018)	0.173 (0.009)
Pull requests	0.045 (0.803)	−0.075 (0.001)	−0.083 (0.000)	−0.089 (0.000)	−0.076 (0.008)	−0.043 (0.298)	0.097 (0.310)
Watchers	0.046 (0.178)	0.044 (0.051)	0.045 (0.041)	0.055 (0.037)	0.023 (0.263)	0.077 (0.001)	0.100 (0.000)
Forks	0.201 (0.000)	0.151 (0.000)	0.154 (0.000)	0.167 (0.000)	0.132 (0.000)	0.192 (0.000)	0.230 (0.000)
Pushes	0.062 (0.000)	0.033 (0.000)	0.036 (0.000)	0.035 (0.000)	0.025 (0.000)	0.042 (0.000)	0.048 (0.000)
Documentation	0.147 (0.000)	0.032 (0.496)	0.012 (0.800)	0.026 (0.611)	−0.017 (0.777)	−0.034 (0.460)	−0.025 (0.593)
Number of packages	−0.001 (0.507)	0.002 (0.046)	0.002 (0.029)	0.002 (0.058)	0.005 (0.004)	0.002 (0.132)	0.001 (0.438)
Constant	−0.167 (0.000)	0.075 (0.086)	0.080 (0.064)	0.069 (0.132)	0.129 (0.011)	0.096 (0.025)	0.074 (0.103)
Observations	173,853	173,853	173,853	142,613	99,319	240,582	288,871
R-squared	.053	.858	.859	.853	.908	.821	.803
Complementor and month fixed effects	No	YES	YES	YES	YES	YES	YES

Note: *p*-values based on robust standard errors in parentheses.

allaying the potential concern that the results might have been driven by using OLS regressions.

4.4.3 | Alternative measure of similarity

As Model 1 in Table A5 shows, the results were robust when we measured similarity based on the cosine measure between the upstream software packages used by that complementor and the upstream software packages used by other complementors in the same domains of use. We also probed the concern that considering the top five most frequently used keywords to identify the most relevant domains for comparison may have deflated the measure of similarity, as with fewer domains, there is less chance of a potential overlap in upstream packages. Specifically, we created an alternative measure of similarity that expanded the domains used in the comparison to those indicated by the top 10 keywords most frequently used by complementors in the focal platform. Model 2 in Table A5 shows robust results, thereby allaying this concern.

4.4.4 | Investigating concerns with promotion as alternative explanation

As discussed in Section 3.5, we included in our analysis controls to account for the potential influence stemming from a complementor's efforts to promote its offering on the home platform. A potential concern is that promotion efforts may have comprised other activities not fully captured by Readme files, such as email campaigns or trending repositories in GitHub. If a complementor increased these promotion efforts on the home platform following multihoming, then this could partly explain the performance increase following the multihoming reported above. To investigate this possibility, we ran models predicting promotion activities as measured in a variety of ways, as detailed in Table A6. The results showed that the multihoming complementors did not increase any such promotion activities following multihoming, thereby further allaying this concern.

4.4.5 | Investigating concerns with new product launch as alternative explanation

As discussed in Section 3.5, the analysis accounted for the potential influence stemming from the number of packages that a complementor published on the home platform as of the focal month (*Number of packages*), thereby accounting for the effects stemming from the launch of new packages. Further, as explained in Section 3.6, the values of this variable of multihoming complementors were similar to those of their single-homing counterfactuals in the matched sample. Notwithstanding these considerations, we further investigated the possibility that the launch of new packages might have driven the findings. We started by running a model using in the dependent variable the average installed repositories across packages. Even if a new launch resulted in an increase in a complementor's installed repositories, it is unlikely that the new launch would perform so much better than preexisting packages as to move the overall average up. The results, presented in Model 1 of Table A7, were robust, thus mitigating this concern. Additionally, we ran an analysis dropping observations in the top 10th percentile of

Number of packages. Model 2 of Table A7 shows consistent results, further allaying this concern.

4.4.6 | Alternative approach to build matched sample

We also conducted sensitivity analysis using propensity score matching (PSM) (Rosenbaum & Rubin, 1983) as an alternative approach to construct a matched sample. Following this approach, for each multihoming complementor, we identified a counterfactual complementor with the smallest difference in probability scores of multihoming estimated in a probit model with the control variables used in the main analysis. Table A8 shows that the results based on the PSM sample were consistent with those in our main analysis, further increasing confidence that the baseline results were not an artifact of the specific sample in the main analysis.

4.5 | Post hoc analyses

In addition to the analyses above showing evidence in support of the causal effect of multihoming on a complementor's performance on the home platform, this section provides additional evidence that aligns with the logic underlying our argumentation.

We started by considering that if the performance gains accruing to multihoming complementors indeed relate to learning that occurs on the new platform, as we argue, then we would expect such gains to be stronger when the platforms shared traits facilitating the porting of learning back to the home platform. We investigated this conjecture by considering commonality between the new platform and the home platform in the programming language used for open-source software development, as such commonality can facilitate learning and reduce adaptation costs while still preserving learning opportunities.[18] The results, shown in Model 1 in Table A9, are consistent with this conjecture, showing that the performance gains on the home platform following multihoming were greater when the platforms shared the same programming language.

Further, if the increase in performance was indeed attributable to the porting of learning back to the home platform, then we would expect this increase to be more pronounced for complementors that have been on the home platform for longer, as the attendant familiarity with that environment should make adaptation easier and, accordingly, facilitate the porting of learning back to the home platform. The findings, presented in Model 2 in Table A9, cohere with this conjecture, showing that the performance increase on the home platform following multihoming was greater among the complementors with longer tenure on the home platform.

Finally, we also investigated the extent to which the performance increase on the home platform following multihoming was linked to activities that occurred on the new platform, which would be indicative that such effect was related to the learning that occurred in that platform. We gathered data on two activities on the new platform that could create learning opportunities for the multihoming complementors, namely code changes made to the corresponding repositories, captured by the variable with the number of *Pushes* in the new platform, and

[18]For example, different platforms building on the same programming language can still exhibit remarkable differences in terms of the upstream interdependent complements a focal complementor builds on, as well as in terms of the downstream use of the complements.

communication with downstream users in the new ecosystem, as captured by *Issue comments* with the number of messages about bugs and improvement suggestions for the complementor's repositories in the new platform. Models 3 and 4 in Table A9 report, respectively, that the multihoming complementors that exhibited higher levels of activity on the new platform, as reflected in these two variables, experienced greater performance improvements on the home platform than those with lower levels in those variables. These results provide additional evidence in line with the logic underlying our arguments. Additionally, these results show that, although multihoming provides learning opportunities to multihoming complementors, there is also important variation across complementors in the extent to which they capture these opportunities.

5 | DISCUSSION

This study takes a key step toward exploring the possibility that complementor performance in a platform ecosystem might also be shaped by influences arising beyond the focal ecosystem. Specifically, it investigates how multihoming to a new platform affects complementor performance on the home platform. We propose that complementor performance on a home platform is positively affected by multihoming. Underlying our proposition is the argument that multihoming engenders learning effects that benefit a complementor on the home platform, as adaptation to another platform contributes to making it better able to manage interdependencies and identify opportunities to further differentiate its complement on the home platform. We further probe this logic by proposing that this positive performance implication of multihoming is stronger when the complementor faces higher levels of interdependencies and greater similarity to other complements on the home platform. Our findings show that a software package complementor that multihomes to a new software development platform attains greater improvement in performance post multihoming, relative to the performance change observed during the same period for another complementor on the same platform that shares with the focal complementor similar levels in a variety of attributes expected to relate to multihoming, but that remained in single-homing. Our findings also reveal that this effect is stronger for software package complementors with higher levels of interdependencies and with greater similarity to other complements on the home platform.

5.1 | Limitations

We note that our findings are subject to potential caveats. With our DiD design we were able to account for a variety of observable attributes that could also affect the incidence of multihoming. The extensive supplemental analyses, including the use of an alternative approach to build a matched sample, further mitigate endogeneity concerns. Additionally, post hoc analyses showed evidence consistent with the logic underlying our arguments about the performance implications of multihoming. Although these analyses generated a preponderance of evidence aligned with our arguments, opportunities remain for future research to further probe the consequences of multihoming for complementors' adaptation and learning across ecosystems.

Further, future research can examine the generalizability of our findings to other platform ecosystems. As discussed above, the open-source nature of the software development platforms examined in this study helps us focus on the perspective of complementors and capture more sharply the causal effect of multihoming on their performance on the home platforms. This effect stemming from multihoming might be even more important in helping complementors overcome the rigidities associated with interdependencies in ecosystems, such as smartphone application platforms, in which certain firms play a leading role in setting the technological infrastructure that shapes the interdependencies facing complementors. Similarly, the learning benefits stemming from multihoming might be even stronger in ecosystems wherein IP protection and pricing are prevalent, as complementors may benefit from insights gained on another platform to offset competitive pressures they face on a focal platform. The opportunity remains for future inquiry to investigate these conjectures.

5.2 | Theoretical contributions and opportunities for future research

This study advances strategy research on platform ecosystems by demonstrating that the performance that a complementor attains in an ecosystem is also shaped by its presence in another ecosystem, especially when higher levels of interdependencies and greater similarity to other complements are involved. This study also uncovers a new set of influences shaping heterogeneity across firms in ecosystems. Specifically, advancing prior strategy discussing effects that arise on the same side or across sides of a platform (e.g., Agarwal et al., 2022; Helfat & Raubitschek, 2018), this study underscores effects that arise *across* platforms. Thus, in addition to variation in performance across complementors, sources of variation also exist that can explain within-complementor heterogeneity across ecosystems. A fruitful opportunity for future research would be to unpack the extent to which performance heterogeneity in platform ecosystems is driven by characteristics of a focal ecosystem, by variation across firms, and by other effects associated with firms' presence in other ecosystems.

This study also underlines the role of complementors in managing interdependencies. Although much value creation in ecosystems is driven by complementors, research has emphasized the role of the firm leading a platform in managing interdependencies by controlling access or by defining the technological infrastructure of an ecosystem that sets the nature of interdependencies affecting complementors' value creation activities. This study redresses this imbalance by showing that a complementor's decision to launch complements on an additional platform helps it become better able to deal with interdependencies on the home platform.

Additionally, this study also contributes to the literature on multihoming. Existing literature on multihoming has explored the factors driving a complementor's expansion across platforms; the implications of such expansion remain under-explored. Research on multihoming has related its incidence to the benefits that complementors expect to obtain from such expansion, such as gaining access to the user base in another platform or mitigating dependency on any one platform (Li & Zhu, 2021; Wang & Miller, 2020). Research directly examining the implications of multihoming from the perspective of complementors remains scant, with the investigation by Cennamo et al. (2018), showing that video games launched on a more complex console platform were of lower quality, being a notable exception. This study advances this recent literature by showing the implications of multihoming for the performance of complementors on the home platform.

By showing that the adaptation challenges following multihoming create a beneficial byproduct, this study reveals an important intertemporal tradeoff that prior research has overlooked: the positive effects that multihoming engenders, as a complementor eventually ports learning back home, partly counterbalance the immediate adaptation challenges that arise in the new platform. Similar effects may arise when another platform on which a complementor operates undergoes generational changes; although adapting to such changes can hinder performance on this other platform (Burford et al., 2022), what the complementor learns in the process may be beneficial on the focal platform. This conjecture warrants further inquiry. More broadly, by indicating that a complementor's adaptation to an ecosystem creates performance benefits in another, this study suggests that multihoming across platform ecosystems is a promising context for future research to advance understanding of the evolution of firm capabilities and technological trajectories (Helfat, 2000; Polidoro Jr & Jacobs, 2024; Polidoro Jr & Yang, 2021). Examining such evolution in the context of open-source software platforms is particularly fruitful, as it enables researchers to map the accumulation of technological capabilities that precede the creation of startups exploring the market potential of closed-source software complements in digital ecosystems.

Finally, this study advances the literature on open-source software. As we discussed earlier, interdependencies are especially salient in the context of open-source software, which can make the management of interdependencies particularly challenging for firms in the ecosystem. Echoing this insight, research on open-source platforms has explored the role of leading firms in managing interdependencies across complements by selectively granting access to their platforms in ways that foster the development of complements (Boudreau, 2010; Fosfuri et al., 2008). This study, in turn, underscores the role of complementors in managing interdependencies, as their decision to multihome can make them better able to deal with interdependencies. Moreover, while the use of unpaid competing complements is standard practice in the software development platform ecosystems that we investigated in this study, the availability of open-source software can also facilitate innovative competition, providing the building blocks that complementors can use when developing priced complements (Helfat & Raubitschek, 2018). Indeed, as we discussed earlier, the software packages available on the open-source software platforms we examined support activities that are essential to a variety of other digital platform ecosystems, such as cryptography, data analytics infrastructure, and the development of applications for smartphones. A fruitful avenue for future research would be to explore the performance effects of multihoming for complementors that use software packages available in open-source environments to launch paid-for complements in another environment. Examining complementors' dual presence in both open-source platforms and ecosystems driven by proprietary knowledge can result in a more comprehensive understanding of complementor strategy and performance effects that arise across ecosystems.

ACKNOWLEDGMENTS

The authors gratefully acknowledge valuable comments on earlier versions of this manuscript provided by seminar participants at the Strategic Management Society Conference, the Academy of Management Annual Conference, George Mason University, INSEAD, Insper, Louisiana State University, Michigan State University, Nova University Lisbon, Santa Clara University, Syracuse University, Texas A&M University, University of Massachusetts Amherst, and University of Wisconsin Milwaukee. The authors thank the editor and two anonymous reviewers for constructive feedback.

DATA AVAILABILITY STATEMENT

The authors elect to not share data.

ORCID

Francisco Polidoro Jr. https://orcid.org/0000-0002-8131-3740
Wei Yang https://orcid.org/0000-0002-0142-0707

REFERENCES

Adner, R. (2017). Ecosystem as structure: An actionable construct for strategy. *Journal of Management*, *43*(1), 39–58.

Adner, R., & Kapoor, R. (2010). Value creation in innovation ecosystems: How the structure of technological interdependence affects firm performance in new technology generations. *Strategic Management Journal*, *31*(3), 306–333.

Adner, R., & Levinthal, D. A. (2002). The emergence of emerging technologies. *California Management Review*, *45*(1), 50–66.

Agarwal, S., & Kapoor, R. (2023). Value creation tradeoff in business ecosystems: Leveraging complementarities while managing interdependencies. *Organization Science*, *34*(3), 1216–1242.

Agarwal, S., Miller, C. D., & Ganco, M. (2022). Growing platforms within platforms: How platforms manage the adoption of complementor products in the presence of network effects? *Strategic Management Journal*, *44*(8), 1879–1910.

Babina, T., He, A. X., Howell, S. T., Perlman, E. R., & Staudt, J. (2023). Cutting the innovation engine: How federal funding shocks affect university patenting, entrepreneurship, and publications. *The Quarterly Journal of Economics*, *138*(2), 895–954.

Baker, A., Larcker, D. F., & Wang, C. C. (2021). How much should we trust staggered difference-in-differences estimates? European Corporate Governance Institute finance working paper, No. 736/2021.

Baldwin, C. Y. (2020). Design rules: How technology shapes organizations (Vol. 2). Working paper, Harvard Business School, Cambridge, MA.

Baum, J. A. C. (1996). Organizational ecology. In C. H. Clegg & W. R. Nord (Eds.), *Handbook organization studies* (pp. 77–114). Sage.

Blackwell, M., Iacus, S., King, G., & Porro, G. (2009). cem: Coarsened exact matching in Stata. *The Stata Journal*, *9*(4), 524–546.

Bonaccorsi, A., & Rossi, C. (2003). Why open-source software can succeed. *Research Policy*, *32*(7), 1243–1258.

Boudreau, K. (2010). Open platform strategies and innovation: Granting access vs. devolving control. *Management Science*, *56*(10), 1849–1872.

Boudreau, K. J., & Jeppesen, L. B. (2015). Unpaid crowd complementors: The platform network effect mirage. *Strategic Management Journal*, *36*(12), 1761–1777.

Burford, N., Shipilov, A. V., & Furr, N. R. (2022). How ecosystem structure affects firm performance in response to a negative shock to interdependencies. *Strategic Management Journal*, *43*(1), 30–57.

Callaway, B., & Sant'Anna, P. H. C. (2021). Difference-in-differences with multiple time periods. *Journal of Econometrics*, *225*, 200–230.

Cattani, G. (2006). Technological pre-adaptation, speciation, and emergence of new technologies: How Corning invented and developed fiber optics. *Industrial and Corporate Change*, *15*(2), 285–318.

Cennamo, C., Ozalp, H., & Kretschmer, T. (2018). Platform architecture and quality tradeoffs of multihoming complements. *Information Systems Research*, *29*(2), 461–478.

Chen, L., Yi, J., Li, S., & Tong, T. W. (2022). Platform governance design in platform ecosystems: Implications for complementors' multihoming decision. *Journal of Management*, *48*(3), 630–656.

Chen, L., Zhang, P., Li, S., & Turner, S. F. (2022). Growing pains: The effect of generational product innovation on mobile games performance. *Strategic Management Journal*, *43*(4), 792–821.

Chen, M., Kaul, A., & Wu, B. (2019). Adaptation across multiple landscapes: Relatedness, complexity, and the long run effects of coordination in diversified firms. *Strategic Management Journal*, *40*(11), 1791–1821.

Chung, H. D., Zhou, Y. M., & Ethiraj, S. (2024). Platform governance in the presence of within-complementor interdependencies: Evidence from the rideshare industry. *Management Science*, *70*(2), 799–814.

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub: Transparency and collaboration in an open software repository. *Paper presented at the Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*.

Eckhardt, J. T. (2016). Welcome contributor or no price competitor? The competitive interaction of free and priced technologies. *Strategic Management Journal*, *37*(4), 742–762.

Fosfuri, A., Giarratana, M. S., & Luzzi, A. (2008). The penguin has entered the building: The commercialization of open-source software products. *Organization Science*, *19*(2), 292–305.

Ganco, M., Kapoor, R., & Lee, G. K. (2020). From rugged landscapes to rugged ecosystems: Structure of interdependencies and firms' innovative search. *Academy of Management Review*, *45*(3), 646–674.

Gousios, G. (2013). The GHTorent dataset and tool suite. In *Proceedings of 2013 10th Working Conference on Mining Software Repositories* (*MSR*). (pp. 233–236). IEEE.

Grigorik, I. (2012). The github archive. Retrieved from https://githubarchive.org

Haefliger, S., Von Krogh, G., & Spaeth, S. (2008). Code reuse in open-source software. *Management Science*, *54*(1), 180–193.

He, V. F., Puranam, P., Shrestha, Y. R., & von Krogh, G. (2020). Resolving governance disputes in communities: A study of software license decisions. *Strategic Management Journal*, *41*(10), 1837–1868.

Helfat, C. E. (2000). Guest editor's introduction to the special issue: The evolution of firm capabilities. *Strategic Management Journal*, *21*(20–11), 955–959.

Helfat, C. E., & Raubitschek, R. S. (2018). Dynamic and integrative capabilities for profiting from innovation in digital platform-based ecosystems. *Research Policy*, *47*(8), 1391–1399.

Jacobides, M. G., Cennamo, C., & Gawer, A. (2018). Towards a theory of ecosystems. *Strategic Management Journal*, *39*(8), 2255–2276.

Kapoor, R. (2018). Ecosystems: Broadening the locus of value creation. *Journal of Organization Design*, *7*(1), 1–16.

Kapoor, R., & Agarwal, S. (2017). Sustaining superior performance in business ecosystems: Evidence from application software developers in the iOS and android smartphone ecosystems. *Organization Science*, *28*(3), 531–551.

Kleinbaum, D., Kupper, L., Muller, K., & Nizam, A. (1988). The correlation coefficient and straight-line regression analysis. In *Applied regression analysis and other multivariable methods* (pp. 80–95). Wadsworth Publishing Company.

Kononenko, O., Rose, T., Baysal, O., Godfrey, M., Theisen, D., & De Water, B. (2018). Studying pull request merges: A case study of Shopify's active merchant. In *Proceedings of the 40th International Conference on Software Engineering*: *Software Engineering in Practice* (pp. 124–133).

Kretschmer, T., Leiponen, A., Schilling, M., & Vasudeva, G. (2022). Platform ecosystems as meta-organizations: Implications for platform strategies. *Strategic Management Journal*, *43*(3), 405–424.

Lee, J. M., & Kapoor, R. (2017). Complementarities and coordination: Implications for governance mode and performance of multiproduct firms. *Organization Science*, *28*(5), 931–946.

Lerner, J., & Tirole, J. (2002). Some simple economics of open-source. *The Journal of Industrial Economics*, *50*(2), 197–234.

Levinthal, D. A. (1997). Adaptation on rugged landscapes. *Management Science*, *43*(7), 934–950.

Levinthal, D. A. (1998). The slow pace of rapid technological change: Gradualism and punctuation in technological change. *Industrial and Corporate Change*, *7*(2), 217–247.

Li, H., & Zhu, F. (2021). Information transparency, multihoming, and platform competition: A natural experiment in the daily deals market. *Management Science*, *67*(7), 4384–4407.

Lin, Y. K., & Maruping, L. M. (2022). Open-source collaboration in digital entrepreneurship. *Organization Science*, *33*(1), 212–230.

Miller, C. D., & Toh, P. K. (2022). Complementary components and returns from coordination within ecosystems via standard setting. *Strategic Management Journal*, *43*(3), 627–662.

Miller, S., Johnson, N., & Wherry, L. R. (2021). Medicaid and mortality: New evidence from linked survey and administrative data. *The Quarterly Journal of Economics*, *136*(3), 1783–1829.

Nesbitt, A., & Pompilio, M. (2016). Libraries.io.

Polidoro, F., Jr., & Jacobs, C. (2024). Knowledge diffusion in nascent industries: Asymmetries between startups and established firms in spurring inventions by other firms. *Strategic Management Journal*, *45*, 807–845.

Polidoro, F., Jr., & Yang, W. (2021). Corporate investment relationships and the search for innovations: An examination of startups' search shift toward incumbents. *Organization Science*, *32*(4), 909–939.

Rahkema, K., & Pfahl, D. (2022). SwiftDependencyChecker: Detecting vulnerable dependencies declared through CocoaPods, carthage and swift PM. In *Proceedings of the 9th IEEE/ACM International Conference on Mobile Software Engineering and Systems* (pp. 107–111).

Rivkin, J. W., & Siggelkow, N. (2003). Balancing search and stability: Interdependencies among elements of organizational design. *Management Science*, *49*(3), 290–311.

Rose, S., & Van der Laan, M. J. (2009). Why match? Investigating matched case-control study designs with causal effect estimation. *International Journal of Biostatistics*, *5*(1), 1.

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*(1), 41–55.

Saadat, S., Newton, O. B., Sukthankar, G., & Fiore, S. M. (2020). Analyzing the productivity of GitHub teams based on formation phase activity. In *2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology* (*WI-IAT*). (pp. 169–176). IEEE.

Schilling, M. A. (2000). Toward a general modular systems theory and its application to interfirm product modularity. *Academy of Management Review*, *25*(2), 312–334.

Shankar, V., & Bayus, B. L. (2003). Network effects and competition: An empirical analysis of the home video game industry. *Strategic Management Journal*, *24*(4), 375–384.

Toh, P. K., & Polidoro, F. (2013). A competition-based explanation of collaborative invention within the firm. *Strategic Management Journal*, *34*(10), 1186–1208.

Wang, R. D., & Miller, C. D. (2020). Complementors' engagement in an ecosystem: A study of publishers' e-book offerings on Amazon kindle. *Strategic Management Journal*, *41*(1), 3–26.

Wen, W., & Zhu, F. (2019). Threat of platform-owner entry and complementor responses: Evidence from the mobile app market. *Strategic Management Journal*, *40*(9), 1336–1367.

Wu, Y., & Zhu, F. (2022). Competition, contracts, and creativity: Evidence from novel writing in a platform market. *Management Science*, *68*(12), 8613–8634.

Zhang, Y., Li, J., & Tong, T. W. (2022). Platform governance matters: How platform gatekeeping affects knowledge sharing among complementors. *Strategic Management Journal*, *43*(3), 599–626.

Zhang, Y., & Yang, W. (2022). Breakthrough invention and problem complexity: Evidence from a quasi-experiment. *Strategic Management Journal*, *43*(12), 2510–2544.

Zhu, F., & Liu, Q. (2018). Competing with complementors: An empirical look at Amazon.com. *Strategic Management Journal*, *39*(10), 2618–2642.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Polidoro, F. Jr., & Yang, W. (2024). Porting learning from interdependencies back home: Performance implications of multihoming for complementors in platform ecosystems. *Strategic Management Journal*, *45*(9), 1791–1821. https://doi.org/10.1002/smj.3601