Co(de)nnection:
Developer Networks and Digital Asset Fragility *

Zhiheng He[†] Huiyi Litan[‡]

Abstract

Digital assets are interconnected on the technical implementation side. We uncover a dynamic network linking 1,510 tokens through shared developers using 47 million GitHub commits by nearly 280,000 developers from 2013 to 2025. We document a risk-shielding effect: tokens connected to riskier peers subsequently exhibit lower idiosyncratic volatility, tail risk and asymmetry in semi-side systematic exposures, in contrast with conventional spillovers observed in a parallel social-media co-mention network built from 41 million Reddit posts. The effect is strongest for infrastructure-critical and ecosystem-linked tokens, and becomes more pronounced alongside the network segmentation. An instrument based on historical intraday commit-timing similarity is used to address measurement error and endogenous network formation. We discuss several explanations and test a labor mobilization mechanism: peer distress triggers intensified code maintenance, diffusing development-flow information that reduces implementation uncertainty. Our findings highlight developers as a novel form of financial infrastructure shaping risk transmission of digital assets.

JEL: *G12, G14, D85, O36, J24*

Keywords*: Digital Assets, Developer Networks, Fragility, Idiosyncratic Risk, Open-Source Development*

*We thank Alfred Qi Fan, Weichen Zhang and seminar and conference participants at Shanghai Jiao Tong University Antai College, for helpful comments and suggestions. All errors are our own. First draft: January 7, 2026. This draft: May 31, 2026.

[†]Renmin University of China. hez@ruc.edu.cn.

[‡]Renmin University of China. huiyi.litan@ruc.edu.cn.

Financial research has long studied how links between assets shape interdependence of their risk, typically through fundamental channels (e.g., Anton & Polk, 2014; Cohen & Frazzini, 2008; C. M. Lee et al., 2019) and investor-side channels (e.g., Ali & Hirshleifer, 2020; Peng et al., 2023). Yet digital assets, including utility tokens, stablecoins, and tokenized real-world assets,[1] are also deeply determined by their technical implementation: their economic functions are implemented in code, and their reliability depends on continuous developer effort in maintaining, upgrading, and securing the underlying protocols. This gives rise to an underexplored implementation layer in the valuation process between issuer fundamentals and market pricing, through which technical uncertainty and developer activities may shape asset-level risk. This implementation-side impact can be increasingly important as an expanding range of assets is becoming digitized.

In this paper, we uncover the extensive and dynamic linkages among digital assets formed through shared developers, and examine how it shapes digital asset idiosyncratic risk and fragility. We study this question in the crypto token market, where open-source development leaves rich traces of granular, real-time technical activity. Specifically, we first construct the networked linkages via tracking 26,866 GitHub repositories associated with 1,510 tokens, covering the vast majority of the total cryptocurrency market capitalization, and identify 47,384,938 code commits made by 279,422 developers between January 2013 and March 2025. Figure 1 provides an overview of the resulting developer network, in which each token is a node, and the weighted edges are constructed based on overlapping developers who contribute to both tokens. The network landscape shows that digital assets are not implemented in isolation; in contrast, they interact widely and frequently with each other through shared developers, typically reflecting shared standards and overlapping technical labor communities. We term this phenomenon as "co(de)nnection." We further show that the developer network experienced a remarkably dynamic evolutionary process over the decade. The rich structural characteristic of developer networks potentially leads to common exposures to shocks and knowledge spillovers for digital assets.

We then study the interdependence of risk across tokens linked in developer networks. We document an uncommon *risk-shielding* relation: focal tokens whose peers exhibit higher idiosyncratic and tail risk will themselves exhibit lower corresponding risk measures. To

[1]Notwithstanding the ongoing evolution and expansion of this field, Treasury/IRS defines digital assets as digital representations of value that use cryptography to secure transactions that are digitally recorded using blockchain or other similar technologies. See https://www.federalregister.gov/d/2023-17565. Furthermore, it is becoming increasingly recognized that digital assets should not be viewed exclusively through the lens of traditional assets. For instance, the SEC Release in March 2026 indicates that digital commodities, tools, collectibles and stablecoins are not to be regarded as securities. See https://www.sec.gov/files/rules/interp/2026/33-11412.pdf.

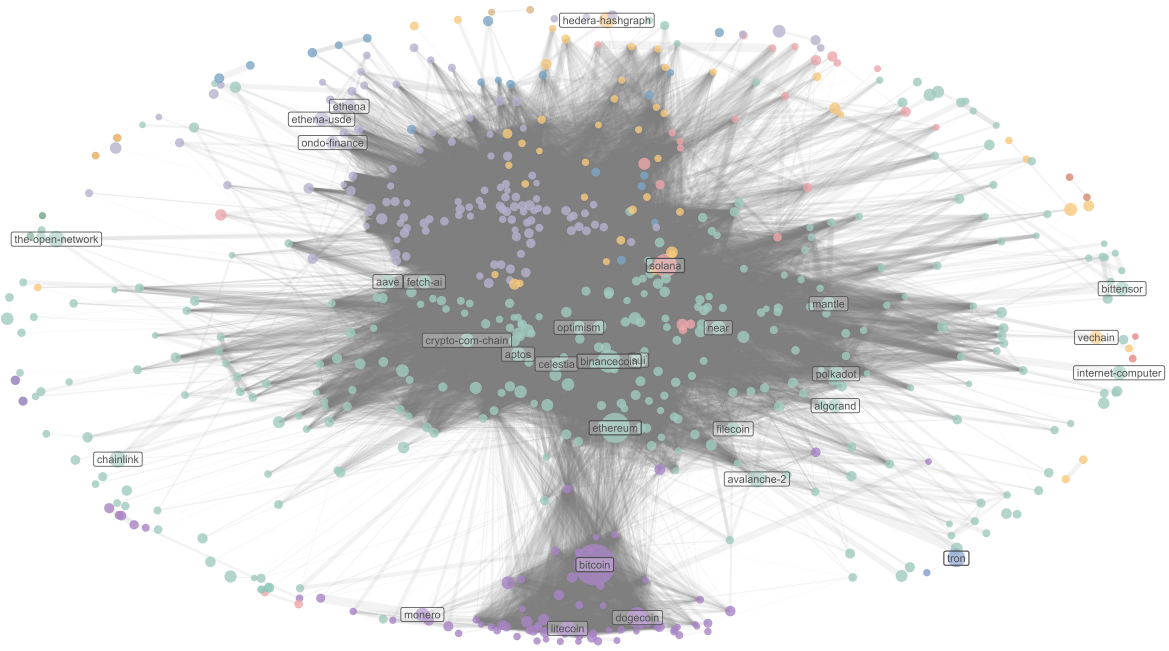


Figure 1: The Developer Network

This figure visualizes the network of digital assets (tokens) connected by shared coding developers. Each node represents a token project, with the node size representing its average daily market capitalization in January 2025. An edge connects two nodes if the same developers actively submitted commits to both projects during the period from January 2013 to March 2025. The edge width reflects the number of such shared developers. For visualization clarity, only edges corresponding to at least 10 co-developers are retained, and isolated nodes are excluded. The top 30 tokens based on average daily market cap in January 2025 are labeled (note that some large-cap tokens, such as Ripple, are absent due to insufficient shared developer activity). The network layout and colored node clustering are determined by the Fruchterman-Reingold algorithm.

empirically analyze this relation, we evaluate monthly developer networks constructed within a weekly rolling window, and merge them with the token-week panel of market variables. Specifically, we define *Developer Network Peer Risk*$_{i,t} \equiv \left(\sum_{j\neq i} w_{i,j,t} \cdot x_{j,t}\right) / \left(\sum_{j\neq i} w_{i,j,t}\right)$, i.e., the weighted average connected peer j's risk measure $x_{j,t}$ for the focal token i, as our main independent variables. The weights $w_{i,j,t}$ are determined by the cosine similarity between the sets of developers associated with tokens i and j, where association is defined by having submitted coding commits for the respective token within the month preceding week t. We consider a broad set of measures $x_{i,t}$, including idiosyncratic volatility (IVOL), downside volatility (DSVOL), negative conditional skewness (NCSKEW), expected shortfall at the 5% level (ES5), and semi Betas.

The risk-shielding effect is both statistically and economically significant: a one–standard-deviation rise in peers' IVOL corresponds to an 11.31%–of–standard-deviation drop in the focal token's IVOL. This interdependence is even stronger for downside and tail-risk metrics. A one–standard-deviation increase in peers' ES5 lowers the focal token's ES5 by 14.02%,

which translates into avoiding an additional 167-basis-point loss on a crash day. Furthermore, developer connections dampen lottery-like characteristics of tokens and lessen the imbalance between systematic downside and upside risk exposures. Although average Downside and Upside Betas are usually asymmetric (1.0 vs. 0.7), tokens connected to peers with higher Upside Betas experience a convergence of their Semi Betas toward 1. This indicates that the developer network boosts an asset's participation in market upswings without amplifying its vulnerability to downside crashes.

To understand the risk-shielding effect in relation to developer networks, we break down the economic intuition behind *Developer Network Peer Risk* by decomposing the token-specific risk $x_{i,t}$ and connectedness $w_{i,j,t}$. From the viewpoint of tokenomics theories (e.g., Cong et al., 2021b; Sockin & Xiong, 2023b), unlike traditional securities whose valuation is tied to comparatively concrete fundamentals, token values are driven by the functional usefulness and productivity of their underlying platforms — features that are typically complex and opaque.[2] Accordingly, reductions in idiosyncratic volatility and tail risk can be interpreted as diminished uncertainty regarding implementation quality and platform fragility.[3] From the token issuer's perspective, lower return volatility also serves as a proxy for improved operating performance (Lyandres et al., 2022), supporting expectations of credible future development. Turning to connectedness $w_{i,j,t}$, developers who work on multiple tokens provide a direct channel for technological spillovers and for reacting to common implementation-side shocks (e.g., resolving analogous bugs, addressing similar functional needs). Given that explicit cash-flow news is largely limited in crypto markets (e.g., Kogan et al., 2024), developer activity can instead generate information about the development status that propagates through the developer network. Taken together, we argue that the developer network encodes implementation-side inter-connectedness and transmits information, thereby shaping risk transmission across assets, particularly in downside states when technical fragility is most pronounced.

In our empirical framework, a key difficulty is isolating implementation-side developer linkages from the better-known investor-side connections. Since developers may trade and hold assets, overlap in developer activity might, in principle, reflect joint attention or sentiment rather than genuine technical interdependence. To disentangle these mechanisms, we build parallel social-media co-mention networks from over 41 million Reddit posts, defining weighted edges via co-mentions in a way conceptually similar to shared analyst

[2] In this sense, digital assets more or less resemble energy or agricultural commodities, especially with highly uncertain quality and settlement standards.

[3] Idiosyncratic volatility also captures speculative trading. Yet this channel is less informative for the risk-shielding pattern, as speculation would more naturally propagate rather than dampen risk.

coverage (Ali & Hirshleifer, 2020). The comparison is striking: social links are associated with conventional risk spillovers, whereas developer links are associated with risk insulation. In direct horserace regressions, social ties raise focal IVOL by 16.4%, while developer connections reduce it by 13.5%. Controlling for social connectedness further strengthens the shielding estimate, implying that the developer network reflects a distinct economic force.

Another empirical concern is measurement error. Coding logs include casual contributions, peripheral work, and even "code-washing" behavior (Gefen et al., 2024), all of which tend to attenuate OLS coefficients. To isolate the structural component of developer coordination, we use *Historical Commit Timing Similarity* as an instrument for developer links. Using synchronization in intraday commit timestamps and imposing a one-month lag, we extract a predetermined portion of developer overlap that more likely reflects systematic collaboration rather than incidental noise. The resulting 2SLS estimates corroborate the risk-shielding effect and indicate that attenuation bias is quantitatively meaningful in the baseline OLS results. The historical timing-based instrument also alleviates concerns on endogenous network formation, since weekly market outcomes are unlikely to drive historical, within-day, high-frequency collaborations. Notably, commit-timing similarity may partly capture persistent coordination environments, such as shared geography or time zones, which naturally fall under the broader notion of structured collaboration we aim to measure.[4]

We further demonstrate that our result reflects the actual architecture of the developer network rather than arbitrary connectivity. Through permutation and perturbation tests, we randomly delete real edges and add artificial ones, and observe that the estimated shielding effect reaches its maximum exactly at the true network. In particular, the estimated effect declines smoothly as true signal is removed, but breaks down abruptly when noise is introduced. This asymmetry exactly matches theoretical predictions on how network disruption responds to varying degrees of signal loss and noise addition. Overall, the topology-based evidence rules out the concern that the shielding effect is shaped by incidental overlap or generic properties of dense networks.

Regarding the mechanisms, we show that the estimated shielding effect is not explained by mechanical mean reversion, factor absorption, investor inattention, or liquidity-driven channels, and is less likely to be driven by developers' wealth reallocation. Alternatively, we propose and test the labor-mobilization channel grounded in developer activity: distress

[4]To the extent that these factors also load on regional shocks, they would mechanically bias estimates toward positive spillovers. The fact that we obtain a strongly negative coefficient therefore implies that our measured shielding effect is, if anything, a conservative lower bound.

among connected peers functions as an early warning, motivating developers to accelerate maintenance, implement pending updates, and reinforce the focal token's technical infrastructure. This behavior may directly resolve vulnerabilities (e.g., Ransbotham & Mitra, 2009), or indirectly generate observable development activity that signals technical robustness to the market. In this sense, developer networks constitute conduits for implementation-side information diffusion. Aligned with this view, we explore effect heterogeneity across network positions, using the centrality measures following Hirshleifer et al. (2025). We find a competitive interaction between developer and social connections: when an asset is centrally positioned in the social network, noisy social sentiment overwhelms developer signals, weakening the shield; in contrast, high centrality within the developer network strengthens the development signal and offsets social spillover.

We also document several extended findings. In the cross section, the shielding effect is most pronounced for infrastructure-critical and ecosystem-linked assets (e.g., Ethereum-based utility tokens and DePIN tokens), where coding efforts are essential for operation. In contrast, it is weaker or nonexistent in more marketing-oriented segments (e.g., Meme and Gaming tokens), where technical feature plays a minimal role in valuation. In the time series, the effect intensifies after 2020, consistent with a maturing developer community. To further trace this evolution, we document pronounced topological dynamics in the developer network, which is clearly divided into three stages: incubation, expansion, and specialization. Thus, these topological metrics exhibit potential as an underlying support for the crypto market segmentation documented in (Cong et al., 2025).

Literature. Our paper contributes first to the literature on blockchain economics. Existing literature has extensively documented the critical technological underpinnings that bring new functionalities and economic features in blockchain economics.[5] Meanwhile, tokenomics theories highlight the central role of technical capacity and functional utility in determining token valuation (e.g., Cong et al., 2021b; Sockin & Xiong, 2023b), and emphasize the importance of developers in token-based platform finance (Cong et al., 2021a). We build on these insights by tracking the micro-level technical implementation of digital assets and examining developers' impact on the market. To our knowledge, we are likely the first to uncover the dynamic digital asset linkages based on sharing developers, and formally analyzes how the implementation stage serves as a fundamental determinant of risk interconnections

[5]Key focuses span, for example, consensus algorithms (e.g., Biais et al., 2019; Saleh, 2021), smart contracts (e.g., Brzustowski et al., 2023; Cong & He, 2019; John et al., 2023), scalability solutions (e.g., Cong et al., 2023; Harvey et al., 2025), DeFi (e.g., Capponi et al., 2025; Lehar & Parlour, 2025; Makarov & Schoar, 2022), as well as programmable tokens and tokenized assets embedded within these platforms (e.g., Borri et al., 2022; Chod & Lyandres, 2023; Goldstein et al., 2024; He et al., 2024; Sockin & Xiong, 2023a).

across digital assets.

A closely related strand of literature empirically analyzes examines risk and return structures of crypto assets. Liu and Tsyvinski (2021) study the market's exposure to a wide range of risk factors in the time series. In the cross-section, Liu et al. (2022) examine the crypto-CAPM and three-factor model comprising market, size, and momentum; Cong et al. (2025) introduce the value factor based on token adoption; Fieberg et al. (2025) extract trend factor from technical indicators. A substantial component of crypto risks, however, remains unexplained by common risk factors. Meanwhile, the crypto market exhibits extreme tail risks, and is susceptible to crashes, pump-and-dump schemes and potential nefarious activities regarding specific assets (e.g., Borri, 2019; Griffin & Shams, 2020; S. S. Lee & Wang, 2025; Li et al., 2018). Our paper adds to this literature by studying how token-specific idiosyncratic and tail risks are transmitted across developer-connected peers. The documented risk shielding effect in a sense reminds Schwenkler and Zheng (2025), who document opposite-signed return relation among crypto peers connected through news co-mention networks and attribute to investor-side overreaction and substitution in information processing. Our findings are based on a distinct developer network structure, pointing to mechanism implications on labor mobilization from the implementation side.[6] Thus, we add to the literature exploring unique features of crypto markets. In particular, Kogan et al. (2024) suggests that the lack of cash-flow news leads to different investor trading models in the crypto market. Our findings imply that development flows may constitute alternative information. Although they do not necessarily carry information about expected returns like fundamental news, they complement the information environment about implementation-side uncertainties and technical fragility.

We also add to the literature connecting open-source innovation and financial outcomes. Open-source implementation is characterized by transparency, distributed collaboration, and code-based observability (Hahn et al., 2008; Lerner & Tirole, 2002; Nagle, 2018), and these features are especially relevant in digital asset markets, where programmability and code implementation are core economic primitives (Sharma et al., 2025). Prior work uses GitHub activity to proxy for project quality, resource provision, and fundraising prospects in token markets (e.g., Howell et al., 2020; Lyandres et al., 2022). Standing on the intersection of code production and market performance, Gefen et al. (2024) uncovers the novel pattern of "code washing," documenting that projects inflating coding records can misleadingly boost fundraising success, yet exhibit weak long-term market performance compared to those with

[6]Because developer links are unique to digital asset markets, our findings may help explain why Schwenkler and Zheng (2020) finds that, in stock markets, the news network observes positive-signed relation among linked stocks, in contrast with implications in the crypto market (Schwenkler & Zheng, 2025).

substantive coding efforts. Whereas existing literature mainly focus on levels of commits, we exploit the network structure based on developers in open-source communities. Our approach thus extends this literature from project-level coding measures to the cross-project collaboration structure.

A broader literature investigates networked linkages in finance. While an exhaustive enumeration is beyond our scope, firms are interconnected by a multitude of networks, related to fundamental elements, e.g., technology (e.g., Acemoglu et al., 2016; Bloom et al., 2013; C. M. Lee et al., 2019), production (e.g., Barrot & Sauvagnat, 2016; Cohen & Frazzini, 2008; Herskovic, 2018), ownership (e.g., Anton & Polk, 2014), as well as non-fundamental and behavioral factors, such as analyst coverage (e.g., Ali & Hirshleifer, 2020) and social ties (e.g., Hirshleifer et al., 2025; Kuchler et al., 2022; Peng et al., 2023). We add to this literature by introducing a new networked relation formed through shared developers. Perhaps interestingly, developers have hybrid economic features as they are micro-carriers of technologies while simultaneously acting as mobile labor entities with behavioral traits. The developer network might therefore provide new observation on the intersection of labor mobility and finance (e.g., Baghai et al., 2021; Donangelo, 2014). Within the literature on labor linkages, Liu and Wu (2023) pioneered to infer firms' human capital connections from online job postings, which primarily capture labor demand. In contrast, our developer network—though limited in its industry coverage—offers a supply-side measure of labor links, allowing for relatively high-frequency tracking of realized labor inputs.

In the rest of the paper, Section 1 describes the data, stylized facts, and our identification strategy; Section 2 presents the main empirical results; Section 3 discusses the mechanisms; Section 4 documents extended findings and robustness tests; Section 5 concludes. Additional robustness tests, covering alternative edge definitions, coding-activity metrics, and inference methods, are provided in the online appendix OA-C1–OA-C9.

1. Data and Empirical Strategy

1.1. Data Sources and Variable Construction

Our analysis primarily relies on a comprehensive dataset collected from GitHub. We process over 47 million unstructured granular development records to construct a dynamic, weekly-rolling developer network, where nodes represent token projects and edges denote shared developer relationships. To provide a comparative perspective regarding social ties, we further collect the complete history of Reddit posts from the subreddit `r/CryptoCurrency`, processing over 41 million posts into a parallel social network based on shared post mentions.

We merge these network data with crypto market data and obtain a token-week panel with 250,157 observations, covering the majority of the cryptocurrency market capitalization.

1.1.1. Processing GitHub Data

We select GitHub to analyze coding developers of token projects for several reasons. First, most token projects use GitHub to host their open-source code repositories, enabling developers worldwide to transparently collaborate, track, and manage contributions.[7] Second, GitHub provides comprehensive records of contribution histories, user interactions, and project evolution, offering valuable insights into the network structures and collaborative dynamics. Third, given GitHub's dominant role in the open-source ecosystem, our findings can offer generalizable insights into open-source innovation.[8]

On GitHub, project development and management are organized around the repository (*repo*) as the fundamental unit. A repo functions as a specific storage location for a project's source code, associated files, and each file's revision history, typically maintained collaboratively by multiple developers.[9] Within a repo, the development process follows a standard distributed version-control workflow: developers implement changes (typically locally) and submit them as *commits*, which document every instance where changes or updates are submitted to the codebase by an author (e.g., Zheng & Wang, 2025).[10] The commits are then pushed to the remote repo and incorporated into the main codebase (typically with peer review process in many projects). Technically, a commit is therefore an atomic unit of coding work that leaves a permanent trace in the repo history. For this reason, commit-level records provide a transparent proxy for realized project work and ongoing technical maintenance in open-source projects. Importantly, this infrastructure supports a decentralized ecosystem where contributions are permissionless: it allows external developers, not just repo managers, to participate in the coding collaboration.

[7] Open-source transparency is a defining feature of the decentralized ecosystem; and public code contributions are vast and important in the context of blockchain economies and token project development. Therefore, although some token projects may not or just partially open source, they do not undermine the economic significance of our findings.

[8] Furthermore, the increasing institutionalization of GitHub-based developer output measures facilitates future cross-setting and cross-industry comparisons. For example, the World Intellectual Property Organization (WIPO)'s *Global Innovation Index* incorporates a GitHub-based indicator under "developer creative outputs," defined using GitHub commit pushes received (scaled by working-age population); see https://github.blog/news-insights/policy-news-and-insights/developers-are-now-included-in-the-wipo-global-innovation-index/ (accessed December 22, 2025).

[9] See GitHub Docs, "About repositories,"https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories (accessed December 22, 2025).

[10] See GitHub Docs, "About commits," https://docs.github.com/en/pull-requests/committing-changes-to-your-project/creating-and-editing-commits/about-commits (accessed December 22, 2025).

We collect tokens with relevant open source development data on GitHub. For each token, we first identify its repos via its publicly listed GitHub links. One token project could have multiple repos, e.g., each managing different functions. We aggregate all the mapped repos as the token's corresponding project development dataset. Then, for each identified repo, we track detailed records of its commits with information about submission timestamp, associated GitHub login (the author's unique identifier), etc. We term these authors as (coding) *developers*. We identify and remove official bot logins based on user identifiers (e.g., *dependabot*, *github-actions*) and further manually screen for generic automation patterns of suspected logins with extremely frequent and extensive commit records.

By compiling this data, we construct a comprehensive dataset of commit-level records from January 2013 to March 2025. It collects 26,866 repositories that belong to 1,510 tokens, comprising 47,384,938 commits submitted by 279,422 developers.

1.1.2. Developer Network Construction

This section constructs the developer networks using the commit-level records described above. Our approach exploits the fact that developers frequently contribute to multiple projects simultaneously, thereby creating linkages between tokens. Conceptually, we project these bipartite developer-token relationships into a token-token network: tokens serve as nodes, and the dynamic extent of overlapping developers constitutes the time-varying edges connecting tokens.

We construct the network at weekly frequency with a monthly estimation window. Formally, for each week t, define a rolling window $W(t)$ as the preceding month (30 days) ending at t. This rolling-window design yields a "fish-scale" series of networks without any record looking ahead over t. Define a developer of token i evaluated at time t as a GitHub login that submits at least one commit to any repo mapped to token i during $W(t)$, and let $D_{i,t}$ denote the corresponding set of token i's developers. Tokens i and j are linked in the developer network at time t if they share at least one developer within $W(t)$, i.e., if $D_{i,t} \cap D_{j,t} \neq \emptyset$.[11] Appendix Figures A4 and A5 illustrate the above process from commit activities to token-token links: Figure A4 provides a schematic from commit records to shared-developer linkages, and Figure A5 presents a real example, in which a GitHub login contributes to repos associated with two different tokens within the same rolling window.

To quantify the strength of these linkages, we use *developer (cosine) similarity* as our

[11] Meanwhile, although uncommon, one repo might be associated with more than one token, which implies the corresponding tokens are inherently related. Our network construction captures this relationship well, where any developers of this repo are identified as developers for both tokens.

baseline edge weight. Specifically, for each token pair (i, j) at time t, their developer similarity $w_{i,j,t}$ solves

$$w_{i,j,t} = \frac{|D_{i,t} \cap D_{j,t}|}{\sqrt{|D_{i,t}| \times |D_{j,t}|}} \tag{1}$$

where $|D_{i,t}|$ denotes the number of unique developers contributing to token i during the window $W(t)$. This measure captures the *relative overlap* in technical human capital. The denominator serves as a normalization factor that mitigates mechanical scale effects, ensuring that large projects do not exhibit artificially high connectivity simply due to the sheer size of their developer pools.

Therefore, for each week t, we obtain an undirected weighted developer network represented by an adjacency matrix $G_t = \{w_{i,j,t}\}_{n_t \times n_t}$, where n_t is the number of tokens. By construction, $w_{i,j,t} = w_{j,i,t}$ for $i \neq j$ and and self-loops are excluded, $w_{i,i,t} = 0$. Intuitively, a greater $w_{i,j,t}$ implies that the developers of token i and j are largely overlapped.

To ensure that our results are not driven by the specific functional form of edge weights or data artifacts, we construct a suite of alternative measures for robustness checks, as detailed in Online Appendix OA-C3, including (i) the number of overlapping developers, and (ii) weighted similarity (weighting each developer by their number of commits within $W(t)$) to account for contribution extensity and intensity, respectively, and (iii) three corresponding measures that restrict commits *not* marked as forks by GitHub, and the corresponding authors are refered as active developers.[12] These active-developer measures mitigate mechanically inherited histories through forking.

1.1.3. Social Network Construction

While our primary focus is the developer network, the market is intrinsically socially connected. Crypto assets often lack fundamental cash flow news, making them highly sensitive to community sentiment and synchronization of attention (e.g., Liu et al., 2022; Maitre et al., 2025). Furthermore, since developers are themselves human participants embedded in the broader community, it is crucial to distinguish whether the observed network effects stem from specialized technical coordination or merely reflect general social interaction. To address this, we construct a parallel *social network* to serve as a comparative benchmark.

[12]Forked commits inherit upstream commit histories, potentially introducing mechanical developer overlap. See GitHub Docs, "About forks," https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/about-forks (accessed December 22, 2025). In our sample, active commits account for approximately 30% of the total commits, providing a stringent subsample for robustness testing.

We utilize Reddit (specifically the subreddit `r/CryptoCurrency`) as the representative platform for constructing this network. Reddit serves as the central "town square" for the crypto community, where retail investors and crypto enthusiasts exchange information and form consensus.[13] Following Wang and Zhang (2024), we interpret co-mentions in social media discussions as a proxy for the "collective perception" of asset relatedness: when two tokens are frequently discussed together, they may share a common investor base or narrative theme. This co-mention network approach conceptually parallels the literature on shared analyst coverage in equity markets (Ali & Hirshleifer, 2020).

Specifically, we parse the textual content of the full sample of submissions and comments in `r/CryptoCurrency` since its creation in March 2013, aggregating 41,432,920 posts, and identify mentioned token tickers or names in each post. A "social link" is formed between token i and token j at time t if both tokens are explicitly mentioned in the same posts. To align with the developer network's frequency, we aggregate these co-mentions over the same rolling window $W(t)$ (the past 30 days). The social network at t is then represented by an adjacency matrix $S_t = \{s_{i,j,t}\}_{n_t \times n_t}$, where $s_{i,j,t}$ denotes the count of Reddit posts within $W(t)$ that mention both token i and token j.

1.1.4. Crypto Market and Token Characteristics

We source daily price, trading volume, and market capitalization data of tokens from CoinGecko, one of the leading crypto market aggregators. We apply standard data cleaning filters following Liu et al. (2022). After merging the market data with the developer and social networks, our main sample consists of 1,510 unique tokens with available GitHub project repositories. These tokens account for the vast majority of the total cryptocurrency market capitalization as Figure A1 shows, ensuring the aggregate representativeness of our analysis.

Risk measures. Our primary measure of risk is *Idiosyncratic Volatility (IVOL)*, which captures the token-specific uncertainty orthogonal to common market risks. Following Liu et al. (2022), we calculate IVOL as the annualized standard deviation of the residuals from a CAPM regression of daily excess returns on the crypto market factor. To comprehensively characterize the risk profile, we further consider a set of measures: (1) Tail risks. To capture asymmetry and extreme downside potential, we compute *Downside Volatility (DSVOL)*, *Downside-to-Upside Volatility (DUVOL)*, *Negative Conditional Skewness (NCSKEW)* (Chen

[13] A compelling example is the close tie between the birth and rise of Dogecoin and the Reddit community. See Wikipedia: "Dogecoin quickly gained traction, particularly on social platforms like Reddit." https://en.wikipedia.org/wiki/Dogecoin

et al., 2001) and *Expected Shortfall at 5% level (ES5)*. Greater values of these measures indicate that the asset is subject to greater downside or crash risk; (2) Lottery-like features. Given that cryptocurrencies are often viewed as assets with lottery-like payoffs, we construct *MAX* (maximum daily return) (Bali et al., 2011) and *SMAX* (MAX scaled by volatility) to capture these properties; (3) Systematic risks. We also examine *Downside Beta* and *Upside Beta* (Ang, Chen, & Xing, 2006) to test for asymmetric market exposure. Detailed definitions are provided in Appendix Table A1.

A critical methodological choice concerns the estimation window of these measures. Given our focus on risk transmission across tokens, a relatively short window allows us to identify the effects that might be smoothed out under longer evaluation horizons. Furthermore, compared to equity markets, the crypto market is characterized by significantly higher velocity and structural breaks, further justifying the adoption of a shorter time window. Therefore, in the main text, we employ a 30-day rolling window for volatility and beta measures, and a 90-day rolling window for higher-moment measures (NCSKEW and ES5). Our main results remain robust under longer windows (e.g., 45-day, 60-day, and 90-day for volatility measures).

Other characteristics. Beyond risk measures, we collect a comprehensive set of token-level control variables, including standard characteristics such as size, volume, past returns, and token age (in weeks). We also construct network-specific controls such as *eigenvector centrality* to account for the token's position within the networks, which captures how important a token is in the network information transmission (Hirshleifer et al., 2025). That is, a token is more central if it is connected to other well-connected tokens. Details are provided in the Online Appendix OA-B2. We also use the number of GitHub stars as an alternative proxy. To proxy for general public attention, we obtain *Google Search Volume (SVI)* (Da et al., 2011) for each token, as detailed in Online Appendix OA-B3.

1.2. Summary Statistics and Stylized Facts

Table 1 reports summary statistics of the main network variables and risk measures.

Network characteristics. As the gross developer networks over a decade (Figure 1) shows, tokens tend to form distinct clusters based on shared-developer connectedness. Prominent public-chain ecosystems (e.g., Bitcoin, Ethereum, and Solana) appear in separate clusters, each linked to a set of tokens. This clustering pattern is consistent with the notion that projects built within the same ecosystem often share libraries, standards, and compatibility requirements, which may encourage developers to contribute across multiple token projects

within that ecosystem. *Panel A* summarizes the edge characteristics of weekly developer networks. Notably, the low mean of raw developer similarity (0.74%) versus the higher mean for non-zero edges (12.07%) indicates that the developer network is sparse in general,[14] yet characterized by pockets of strong connectivity. In addition, network positions are highly heterogeneous: the distribution of eigenvector centrality is strongly right-skewed with high kurtosis, consistent with a structure in which a small set of tokens occupy hub-like positions while many tokens remain peripheral or isolated. Interestingly, some large-cap tokens (e.g., Ripple and Tron) appear peripheral in the developer network.

Market characteristics. Table 1 Panel B reports time-series statistics for aggregate market returns. The mean weekly value-weighted return of 1.20%, along with the higher-moment statistics, correspond closely to the magnitudes reported in Liu et al. (2022) (hereafter LTW), confirming the representativeness of our sample. Abnormal returns are computed from the CAPM and the LTW three-factor model, resulting in a mean weekly value-weighted return of -0.86% and -0.81%, respectively. The declines in mean (in magnitude) and standard deviation validate the factors' explanatory power. The higher-order moments, however, imply remarkable risks that lie outside the scope of factor models.[15] Panel C documents several token characteristics, with all suggesting significant heterogeneity.

Table 1 Panel D summarizes the distribution of the main risk measures. Despite controlling for common risk factors, tokens exhibit immense idiosyncratic volatility and extreme tail risk. The mean annualized IVOL remains high at 139.30%, and the Expected Shortfall (ES 5%) averages 14.32%, implying that investors face severe losses during tail events that cannot be diversified away by standard factors. Furthermore, tokens feature lottery-like properties, and crucially, feature asymmetric systemic exposure: the average Downside Beta (1.00) is noticeably higher than the Upside Beta (0.72). Such asymmetry suggests that tokens are structurally more prone to crashing with the market than participating in rallies. These facts highlight that the crypto market is not only volatile but structurally fragile. The inability of traditional asset pricing factors to fully account for these extreme risks underscores the importance of investigating fundamental non-financial mechanisms that may deepen or govern risk propagation and stability.

[14]The overall network is shown in Online Appendix Figure OA-1, where isolated tokens (without developer links) are almost as numerous as connected tokens.

[15]To intuitively interpret the substantial higher kurtosis of abnormal returns (AR) relative to raw returns, one may simplify the market as consisting of Bitcoin and various alt tokens. In this case, value-weighted AR roughly reflects the portion of alt token returns not explained by Bitcoin correlation. The high kurtosis and negative skew reveal significant tail risks in this segment, particularly the risk of crashes.

Table 1: Summary Statistics

This table reports descriptive statistics for the key variables used in the analysis. The main sample consists of a token-week unbalanced panel of 1,510 tokens over 581 weeks, totaling 250,157 observations. *Panel A* summarizes network characteristics, including developer networks (DevNet), social networks (SocNet), and commit timing networks (TimingNet). Developer networks are constructed based on six different edge measures, where *Developer (Cosine) Similarity* is used in subsequent main-text analysis. AD refers to active developers that submit non-fork commits, as detailed in Online Appendix OA-C3. Edge-level statistics of weights are reported across all potential token pairs ("Raw Edges", representing the full density) and within existing links only ("Non-zero Edges", representing the sparse structure). Network Eigenvector Centrality is calculated at the token-week level. *Panel B* reports time-series statistics for aggregate value-weighted returns and abnormal returns. *Panel C* presents the main token characteristics. *Panel D* summarizes the risk measures used in the paper. For expositional convenience, return and volatility measures are reported in percentages in this table; however, the main regression analyses employ raw values (with volatility annualized) unless otherwise specified.

Panel A: Network Characteristics	Raw Edges					Non-zero Edges		Network Eigenvector Centrality				
	N	Mean	Std.Dev.	Min	Max	Mean	Std.Dev	N	Mean	Std.Dev.	Skewness	Kurtosis
DevNet — *Developer Network*												
Developer Similarity (%)	197,317,431	0.74	6.13	0.00	100.00	12.07	21.76	250,157	0.06	0.21	3.69	15.05
# Overlapping Developer	197,317,431	1.01	12.19	0.00	2,573.00	16.41	46.47	250,157	0.07	0.22	3.57	14.21
Weighed Similarity (%)	197,317,431	0.53	5.47	0.00	100.00	8.54	20.41	250,157	0.06	0.21	3.81	16.09
AD Similarity (%)	197,317,431	0.21	3.15	0.00	100.00	11.30	19.96	250,157	0.04	0.18	4.92	25.45
# Overlapping AD	197,317,431	0.14	4.17	0.00	2,450.00	7.42	29.40	250,157	0.04	0.18	4.58	22.57
Weighed AD Similarity (%)	197,317,431	0.14	2.82	0.00	100.00	7.30	19.22	250,157	0.04	0.18	4.79	24.57
SocNet — *Social Network*												
# Reddit Co-mention	197,317,215	0.47	9.33	0.00	11,365.00	6.54	34.05	250,157	0.04	0.12	4.49	26.20
TimingNet — *Timing Network*												
Commit Timing Similarity (%)	197,317,431	18.40	32.48	0.00	100.00	63.25	28.10	250,157	0.44	0.43	0.07	1.14

Panel B: Statistics of Aggregate Market Returns	N	Mean	Std.Dev.	10%	Median	90%	Skewness	Kurtosis
Value-Weighted Return, %	581	1.20	9.51	-9.39	0.74	13.24	0.19	5.16
Value-Weighted Abnormal Return (CAPM), %	570	-0.86	6.25	-2.37	-0.32	0.58	-1.37	53.92
Value-Weighted Abnormal Return (3-Factor), %	570	-0.81	6.16	-2.41	-0.30	0.65	-0.62	53.02

Panel C: Token Characteristics	N	Mean	Std.Dev.	10%	Median	90%	Skewness	Kurtosis
Market Cap *($M)*	250,157	1,525.84	30,325.68	0.27	9.62	332.69	38.20	1,795.27
Volume *($M)*	250,157	110.57	1,777.91	0.00	0.22	24.18	48.37	4,853.14
Attention *(Search Volume Index)*	250,157	31.83	130.24	0.00	0.00	64.83	6.46	46.24
# GitHub Stars	233,673	701.47	3,986.04	3.00	62.00	1,109.00	14.99	263.54

Panel D: Risk Measures	N	Mean	Std.Dev.	10%	25%	Median	75%	90%
IVOL *(CAPM, Annual %)*	219,717	139.30	357.58	35.57	55.39	88.72	144.68	237.42
IVOL *(3-Factor, Annual %)*	219,717	131.91	342.19	32.81	51.32	83.14	137.00	226.35
DSVOL *(Downside Volatility, Annual %)*	219,717	118.07	82.76	47.92	70.48	101.01	142.38	200.81
DUVOL *(Down/Up Volatility Log Ratio)*	219,646	-0.55	0.90	-1.57	-0.93	-0.42	-0.02	0.32
NCSKEW *(Negative Coefficient of Skewness)*	219,717	-1.31	1.71	-3.46	-1.97	-0.97	-0.31	0.23
ES5 *(Expected Shortfall at 5%, %)*	219,667	14.32	8.81	6.86	9.31	12.55	16.99	22.86
MAX *(Daily Return, %)*	219,717	25.56	79.57	5.00	8.15	13.69	24.04	43.97
SMAX *(Daily Return Scaled by Volatility)*	219,717	2.59	0.87	1.58	1.96	2.46	3.10	3.82
DSBETA *(Downside β)*	219,717	1.00	3.15	-0.17	0.43	1.03	1.58	2.26
USBETA *(Upside β)*	219,717	0.72	3.35	-0.56	0.09	0.67	1.25	2.01

1.3. Empirical Strategy

1.3.1. Baseline Specification

To examine the effect of the developer network peers' risk on the focal token's risk, we employ the following panel regression model (taking IVOL as the baseline example):

$$IVOL_{i,t+1} = \eta\ DevNet\ Peer\ IVOL_{i,t} + \gamma' X_{i,t} + \alpha_i + \delta_t + \epsilon_{i,t}. \tag{2}$$

The key independent variable, *DevNet Peer IVOL*, is the weighted average of idiosyncratic volatility across connected peers,

$$DevNet\ Peer\ IVOL_{i,t} \equiv \begin{cases} \dfrac{\sum_{j\neq i} w_{i,j,t} IVOL_{j,t}}{\sum_{j\neq i} w_{i,j,t}}, & \text{if} \quad \sum_{j\neq i} w_{i,j,t} > 0; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where weights $w_{i,j,t}$ is from the adjacency matrix G_t, i.e., the developer similarity between token i and j. In our panel dataset, 41.98% of the observations have positive $\sum_{j\neq i} w_{i,j,t}$, i.e., have at least one linked peer. [16] We adopt a predictive setup where the dependent variable is measured at week $t+1$. This lag structure helps isolate the directional transmission of risk from the network to the focal asset. We also process several tests to rule out contemporaneous price pressure and mechanical return reversals in Section 3.

We include a comprehensive set of controls $X_{i,t}$ to account for confounding factors, such as size, momentum, volume, public attention (proxied by search volume), and token age effects evaluated at time t. Crucially, to control for the persistent nature of token risk characteristics, we include the focal token's *Past IVOL* as control. However, since IVOL is constructed using a 30-day (approximately 4-week) rolling window, using the first lag (e.g., $t-1$) would result in substantial overlap in the underlying estimation windows, thereby generating mechanical autocorrelation. Instead, we control for $IVOL_{i,t-4}$, which captures the persistent feature as much as possible without any mechanical autocorrelation. Appendix Figure A2 shows the correlation plot of IVOL, DevNet Peer IVOL and control variables. The high correlation (0.52) between IVOL and Past IVOL ($IVOL_{i,t-4}$) confirms the persistent nature is captured.

Statistical inference with this network setting requires addressing complex residual structures. Given that the regression involves linkages between focal token i at time t and peer tokens j at time t', the potential correlation between the error terms $\epsilon_{i,t}$ and $\epsilon_{j,t'}$ should

[16] Unlike dense networks where peer averages essentially proxy for the market average, the developer network is (relatively) sparse and clustered; thus, this measure captures distinct local exposure rather than global market trends.

be taken into account, which is not considered in a standard two-way clustering scheme. We therefore employ Driscoll-Kraay standard errors (Driscoll & Kraay, 1998) with a bandwidth set to cover the full temporal span of the regression specification (from $t+1$ to $t-4$). This approach yields test statistics that are robust to general forms of spatial and temporal dependence. We also verify robustness using standard two-way clustering by token and week in Table OA-2, where all the main results remain statistically significant.

1.3.2. Instrumental Variable

Identifying the causal effect of the developer network on asset risk transmission faces two fundamental challenges: measurement error and endogenous network formation. First, although we implement filters to remove bot accounts, the commit records may still be noisy, containing casual contributions, "tourist" contributors, and even "code-washing" activities that inflate contribution metrics without substantive coding efforts (Gefen et al., 2024). This may introduce considerable attenuation bias in the OLS estimates. Second, the formation of developer ties may be subject to endogenous sorting.[17]

To isolate the genuine, structural developer linkages from both casual noise and reverse causality, we propose an instrumental variable (IV) approach based on *Commit Timing Similarity*, which conceptually echoes the shift-share framework (Goldsmith-Pinkham et al., 2020). Specifically, we utilize the persistent and synchronized working habits of developers, reflected in timing similarities, to construct pre-determined structural exposure shares, while treating the peer asset's risk fluctuations as the shifts. Consequently, as suggested in Goldsmith-Pinkham et al. (2020), the identification here relies on the exogeneity of the exposure structure rather than the randomness of peer outcomes.

Formally, for each token i at time t, we map all second-level timestamps of commits within the time window $W(t)$ onto a daily cycle $[0, T]$, where $T = 86,400$ seconds. We estimate the probability density function of commit timing, $f_{i,W(t)}(\tau)$, using Kernel Density Estimation with a Gaussian kernel.The edge weight between token i and j in the Timing Network W^{time} is defined as the uncentered correlation (cosine similarity) of their temporal density profiles:

$$w_{i,j,t}^{time} = \frac{\int_0^T f_{i,W(t)}(\tau) f_{j,W(t)}(\tau) d\tau}{\sqrt{\int_0^T f_{i,W(t)}^2(\tau) d\tau} \sqrt{\int_0^T f_{j,W(t)}^2(\tau) d\tau}}. \tag{4}$$

[17] As noted by Goldsmith-Pinkham and Imbens (2019), peers may actively selected based on latent characteristics. If these determinants of network formation are correlated with the outcome, standard OLS estimates will be biased.

We then calculate the *TimingNet Peer IVOL* and use its *one-month lag*, $\textit{TimingNet Peer IVOL}_{i,t-4} = \sum_{j\neq 1} w^{time}_{i,j,t-4} IVOL_{j,t-4} / (\sum_{j\neq i} w^{time}_{i,j,t-4})$, as the instrument for the endogenous $\textit{DevNet Peer IVOL}_{i,t}$.

The validity of this instrument rests on several pillars. For *relevance*, genuine technical collaboration typically requires temporal synchronization. Developers working on interrelated projects tend to coordinate schedules, participate in synchronized sprints, or share highly consistent working hours. Also, synchronization nature is relatively persistent as collaboration remains. Therefore, this instrument acts as a structural filter: it naturally purifies the network by stripping away casual, unsynchronized noise linkages, and amplifies the connections characterized by high-intensity, structured labor input, especially when processing urgent attacks and pushes. As a result, the IV approach should increase the coefficient magnitude relative to the baseline OLS estimates, thereby mitigating the attenuation bias.

Regarding *exclusion restriction*, the specific intraday distribution of commit time is plausibly exogenous to the daily and longer-level market indicators. Even for market-wide news that might trigger an immediate shock, it is implausible that token-specific price fluctuations would systematically alter the intraday work flows of developers (e.g., shifting a commit from 2:00 PM to 2:30 PM) in a way that generates spurious correlation across projects. Also, the randomness inherent in specific second-level commit timing, driven by personal schedules and sleep cycles, is orthogonal to fundamental asset risks, and is relatively hard to manipulate.

A standard concern of network identification is the reflection problem (Manski, 1993) or omitted correlated effects. In our context, one might intuitively challenge that Commit Timing Similarity heavily reflects shared geographic locations or timezones. Consequently, tokens linked by timing similarity might be exposed to shared regional macro shocks (e.g., local regulatory changes), which directly affect token volatility. However, the direction of this potential omitted variable bias, either interpreted as reflection problems or geographic/timezone homophily, strictly works in our favor: These unobserved confounders would mechanically drive the risks of the connected tokens to co-move, thereby inducing a positive bias (i.e., spurious contagion or risk spillover), while we find a highly robust negative association. That is, the developer network obtains a powerful counterforce that completely overcomes this natural positive spillover (if any). This suggests that our estimates of the risk shielding effect should represent a conservative lower bound of the true effect.

Furthermore, by using the *TimingNet* constructed from the *previous month* to instrument for the current developer network, we rule out any potential feedback from immediate market shocks to developer behavior. Thus, the lagged Timing Network affects current asset

risk transmission merely through the relatively persistent (or slow-moving) structure of real collaboration efforts.

2. The Risk Shielding Effect of Developer Networks

This section documents the market implications of the developer network. We find that a token's idiosyncratic volatility, tail risk, and lottery-like features are *negatively* associated with those of its linked peers in the developer network. This pattern stands in stark contrast to risk propagation and spillover commonly observed in networks (e.g., Barrot & Sauvagnat, 2016; Cohen & Frazzini, 2008), suggesting a surprising *risk shielding* effect. We then confirm the topological validity of our findings through network permutation and perturbation tests. Finally, we show that developer network links extend to mitigate asymmetric systematic exposures.

2.1. The Shielding Effect on Idiosyncratic Volatility

We begin by examining the effect of developer network (DevNet) peer IVOL on focal token's IVOL. Table 2 reports the OLS and 2SLS results. Under a standard risk spillover hypothesis, one would expect the coefficient η on *DevNet Peer IVOL* to be positive. However, across all OLS specifications in Columns (1) to (3), ranging from the baseline fixed-effects model to the full model controlling for past IVOL and token characteristics, the estimated η is negative and statistically significant at the 1% level. This indicates that tokens connected to peers with higher IVOL tend to exhibit *lower* subsequent idiosyncratic risk.

Columns (4)-(5) report the 2SLS specification instrumented using the 1-month lagged Commit Timing Network peer IVOL, as detailed in Section 1.3.2. We first verify the validity of our instrument. The instrument exhibits high relevance, with a t-statistic of 21.69 and a Kleibergen-Paap F-statistic of 470.6, well above standard critical values. Additionally, the results alleviate concerns that the instrument merely proxies for the token's own volatility momentum. As shown in Column (4), the first-stage coefficient on *Past IVOL* is statistically indistinguishable from zero (−0.001), and the correlation between the instrument and *Past IVOL* is negligible (−0.05, see Appendix Figure A2). Moreover, the coefficient on *Past IVOL* in the second stage (0.409) remains remarkably stable compared to the OLS baseline (0.410). These findings indicate that the developer network exerts an independent structural influence on risk, distinct from the token's own volatility momentum.

Regarding the estimated magnitude in Column (5), the coefficient increases substantially in the 2SLS specification ($\hat{\eta}_{2SLS} = -0.555$ versus $\hat{\eta}_{OLS} = -0.042$). This increase is consistent with the correction of attenuation bias caused by noise commit records, as discussed in

Table 2: Impact of Developer Networks on Idiosyncratic Volatility

This table reports OLS and 2SLS estimates examining the impact of developer network peer idiosyncratic volatility (*DevNet Peer IVOL*) on focal token idiosyncratic volatility $IVOL_{i,t+1}$ evaluated at the end of week $t+1$. Columns (1)–(3) present OLS estimates with token progressively added controls. Column (4) reports the first-stage regression results, where *DevNet Peer IVOL* is instrumented by *Past TimingNet Peer IVOL* (peer volatility weighted by the 1-month lagged commit timing network). Column (5) reports the second-stage 2SLS estimate. All specifications include token and time fixed effects. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with a lag of 5 weeks. $^{***}p<0.01$, $^{**}p<0.05$, $^{*}p<0.10$.

	Dependent: $IVOL_{i,t+1}$				
	OLS			2SLS Stage-1	2SLS Stage-2
	(1)	(2)	(3)	(4)	(5)
DevNet Peer IVOL	-0.100***	-0.058***	-0.042***		-0.555***
	(-4.806)	(-3.734)	(-3.259)		(-3.949)
Past IVOL		0.437***	0.410***	-0.001	0.409***
		(7.928)	(8.119)	(-1.512)	(8.103)
Beta			0.233	-0.005**	0.230
			(1.158)	(-2.349)	(1.146)
ln(Cap)			-0.184***	0.021***	-0.168***
			(-6.601)	(7.524)	(-6.126)
ln(Vol)			-0.036***	0.006***	-0.032***
			(-4.224)	(2.645)	(-3.869)
Ret 4,1			1.801***	0.008*	1.803***
			(10.987)	(1.755)	(11.031)
ln(SVI)			0.023	0.016***	0.032*
			(1.168)	(5.263)	(1.673)
ln(Age)			-0.005	-0.001	-0.011
			(-0.158)	(-0.104)	(-0.317)
Past TimingNet Peer IVOL				0.206***	
				(21.692)	
Token FE	✓	✓	✓	✓	✓
Time FE	✓	✓	✓	✓	✓
Observations	219,598	209,752	209,752	209,752	209,752
Adjusted R^2	0.164	0.324	0.376	0.454	0.370
Kleibergen-Paap F-stat				470.6	
Wu-Hausman p-val					<0.001

Section 1.3.2. The 2SLS estimate suggests that the coefficient is significant both statistically and economically: for an average token, a one-standard-deviation increase in peer IVOL is associated with an 11.31% standard-deviation reduction in future IVOL,[18] implying a strong risk shielding effect.

In terms of economic intuition, we interpret our findings through the lens of tokenomics theories (e.g., Cong et al., 2021b). A token's volatility typically proxies for the uncertainty of its underlying platform productivity (e.g., security, scalability), which is strictly bound to its technical implementation. Since code dictates the actual execution of embedded functional

[18] Calculated as $\hat{\eta}_{2SLS} \times \sigma(\textit{DevNet Peer IVOL})/\sigma(\textit{IVOL}) = -0.555 \times 72.840\%/357.576\%$.

rights, ranging from transaction clearing to smart contract governance, elevated IVOL often reflects technological uncertainty or operational fragility. The observed risk shielding effect thus implies that developer linkages actively reduce this uncertainty. Furthermore, to the extent that high IVOL also captures speculative trading activities, our findings suggest that developer linkages help insulate the asset from the spillover of speculative sentiment. Section 3 explores the specific economic mechanisms.

2.2. Developer Shielding vs. Social Contagion

One potential concern in interpreting the risk shielding effect is that developer linkages might simply proxy for social attention or investor sentiment, as developers are also social media users and are also susceptible to behavioral biases. In particular, a social network typically drives risk spillover, and if developers merely flock to (or leave) "socially hyped" projects, the developer network should structurally resemble (inversely resemble) the social network, exhibiting a risk contagion (shield) pattern. To distinguish our mechanism from the social network effect, we perform a horserace analysis by explicitly controlling for the IVOL of social network peers,

$$\textit{SocNet Peer IVOL}_{i,t} \equiv \begin{cases} \dfrac{\sum_{j\neq i} s_{i,j,t} IVOL_{j,t}}{\sum_{j\neq i} s_{i,j,t}}, & \text{if} \quad \sum_{j\neq i} s_{i,j,t} > 0; \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where the construction of the social network adjacency matrix $S_t = \{s_{i,j,t}\}_{n_t \times n_t}$ is detailed in Section 1.1.3. In our dataset, 57.44% of observations have positive $\sum_{j\neq i} s_{i,j,t}$, i.e., have at least one linked peer in the social network. As a preliminary check, and as shown in Appendix Figure A2, the correlation between *DevNet Peer IVOL* and *SocNet Peer IVOL* is only 0.12, suggesting that these two networks capture distinct links.

Table 3 presents the results. Column (1) first examines the effect of social peers alone. The coefficient on *SocNet Peer IVOL* is positive and highly significant (1.049, $t = 9.13$), confirming that risk spills over through social networks. Column (2) introduces *DevNet Peer IVOL* alongside social peers. The results reveal a striking divergence in direction. The coefficient on *SocNet Peer IVOL* remains significantly positive and even slightly increases, while the coefficient on *DevNet Peer IVOL* remains significantly *negative*. Columns (3)-(4) report the full OLS and 2SLS regressions comparable to Columns (3) and (5) in Table 2, respectively. The shielding effect of the developer network becomes even more pronounced with $\hat{\eta}_{2SLS} = -0.660$ ($t = -4.65$). Economically, a one-standard-deviation higher developer peer IVOL is associated with a reduction in future IVOL of 13.45% of a standard deviation,

whereas a one-standard-deviation higher social peer IVOL is associated with an increase in future IVOL of 16.35% of a standard deviation. This sharp contrast indicates that the two networks capture fundamentally distinct channels: while social linkages facilitate the propagation of idiosyncratic risks, developer linkages function as a distinct firewall that dampens volatility, exhibiting a fundamental dimension of asset resilience.

Table 3: Developer Network Shield vs. Social Network Spillover

This table reports OLS and 2SLS estimates from a "horserace" specification comparing the impact of the Developer Network (*DevNet*) peers and the Social Network (*SocNet*) peers on focal token idiosyncratic volatility $IVOL_{i,t+1}$. Columns (1)–(3) present OLS estimates. Column (4) reports the second-stage 2SLS estimate, where *DevNet Peer IVOL* is instrumented by *Past TimingNet Peer IVOL*. All specifications include token and time fixed effects and full controls (omitted for brevity) as in Table 2 Column (3). T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with a lag of 5 weeks. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

	Dependent: $IVOL_{i,t+1}$			
	OLS			2SLS
	(1)	(2)	(3)	(4)
DevNet Peer IVOL		-0.113***	-0.049***	-0.660***
		(-5.287)	(-3.730)	(-4.652)
SocNet Peer IVOL	1.049***	1.050***	0.842***	0.846***
	(9.127)	(9.123)	(6.934)	(6.958)
Controls			✓	✓
Token FE	✓	✓	✓	✓
Time FE	✓	✓	✓	✓
Observations	219,598	219,598	209,752	209,752
Adjusted R^2	0.195	0.195	0.393	0.385
Kleibergen-Paap F-stat				472.3
Wu-Hausman p-val				<0.001

2.3. Tail Risks and Systematic Exposures

We next consider a wider set of risk measures to examine whether the shielding effect is pronounced beyond decreasing IVOL. In particular, crypto tokens are characterized by high tail risks and negative skewness (e.g., Borri, 2019). Table 4 reports the 2SLS estimates for four distinct measures (Y) of tail risk: *Downside Volatility* (*DSVOL*), *Downside-to-Upside Volatility Ratio* (*DUVOL*), *Negative Coefficient of Skewness* (*NCSKEW*), and *Expected Shortfall at 5%* (*ES*5). Across all specifications, the coefficient on *DevNet Peer Y* is negative and statistically significant at the 1% level.

First, the developer network mitigates the magnitude of extreme losses with lower downside volatility and expected shortfall. A one-standard-deviation increase in peer ES5 is associated with a 14.02% standard-deviation reduction in the focal token's ES5, roughly equivalent to preventing an additional 167-bp loss during the extreme market crash day. Second, and perhaps more intriguingly, the network improves the *shape* of the return

Table 4: Developer Networks, Tail Risks and Systematic Exposures

This table reports 2SLS estimates examining the impact of developer network peer risk using various measures of downside and tail risk. The dependent variables are: Downside Volatility (*DSVOL*), Downside-Upside Volatility Ratio (*DUVOL*), Negative Coefficient of Skewness (*NCSKEW*), and Expected Shortfall at the 5% level (*ES*5), Downside Beta (*DSBETA*) and Upside Beta (*USBETA*), respectively. In each specification, the explanatory variable *DevNet Peer Y* corresponds to the DevNet-weighted peer average of the respective dependent variable. *SocNet Peer Y* refers to the corresponding social network peer risk and lagged focal risk measures for that column's dependent variable. All specifications include token and time fixed effects and full controls, except that Beta is excluded in Columns (5)-(6) to avoid mechanical correlation. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with lags of 5 weeks for Columns (1)-(2)and (5)-(6), and 15 weeks for Columns (3)-(4) to cover the 90-day calculation window of the skewness and expected shortfall measures. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Dependent (Y):	$DSVOL_{i,t+1}$ (1)	$DUVOL_{i,t+1}$ (2)	$NCSKEW_{i,t+1}$ (3)	$ES5_{i,t+1}$ (4)	$DSBETA_{i,t+1}$ (5)	$USBETA_{i,t+1}$ (6)
DevNet Peer Y	-0.200***	-0.291***	-0.693***	-0.190***	0.028	0.482**
	(-4.901)	(-4.689)	(-3.330)	(-3.153)	(0.172)	(2.351)
ScoNet Peer Y	0.146***	0.373***	0.526***	0.121***	0.460***	0.931***
	(10.793)	(13.719)	(14.834)	(8.133)	(4.473)	(6.441)
Controls	✓	✓	✓	✓	✓	✓
Token FE	✓	✓	✓	✓	✓	✓
Time FE	✓	✓	✓	✓	✓	✓
Observations	209,752	209,651	195,083	195,040	209,752	209,752
Adjusted R^2	0.519	0.510	0.160	0.568	0.055	0.072
Kleibergen-Paap F-stat	308.5	184.5	60.2	126.2	269.5	224.9
Wu-Hausman p-val	<0.001	<0.001	<0.001	<0.001	0.994	0.001

distribution. The robust negative effect on *NCSKEW* (-0.693, $t = -3.33$) suggests that the developer shield actively dampens the propensity for returns to exhibit left-skewed, crash-prone behavior, thereby rendering tokens less fragile to catastrophic outliers. In sharp contrast, the coefficients on *SocNet Peer Y* remain consistently positive across all specifications, confirming that social hype exacerbates crash risks and tail exposures through a contagion channel.

We then investigate whether developer linkages alter the tokens' exposures to systematic risks. As reported in Table 1, tokens are structurally more prone to crashing with the market (high Downside Beta) than participating in rallies (low Upside Beta). Table 4 Columns (5) and (6) turn to systematic risk exposures, suggesting a striking asymmetry: while the developer network has no significant impact on Downside Beta, it significantly enhances Upside Beta. For comparison, social network connections simultaneously increase systematic exposure on both sides.

To further investigate these asymmetric effects, Figure 2 plots the binned scatter (following the methodology of Cattaneo et al., 2024) of the focal token's semi-betas against its peers'. Panel A demonstrates that the developer network acts as a normalizing force.

As the token is linked to peers with higher upside beta in the developer network, its own upside beta rises from 0.5 to 1, whereas its downside beta remains largely invariant around 1. Given that the sample average upside and downside betas are 0.72 and 1.00, respectively, this indicates that the developer network effectively "recovers" the token's responsiveness to market rallies, thereby restoring the symmetry of its systematic risk exposure. In sharp contrast, Panel B shows that the social network drives a synchronized increase of semi betas, i.e., effectively merely translating into changes in speculative leverage.

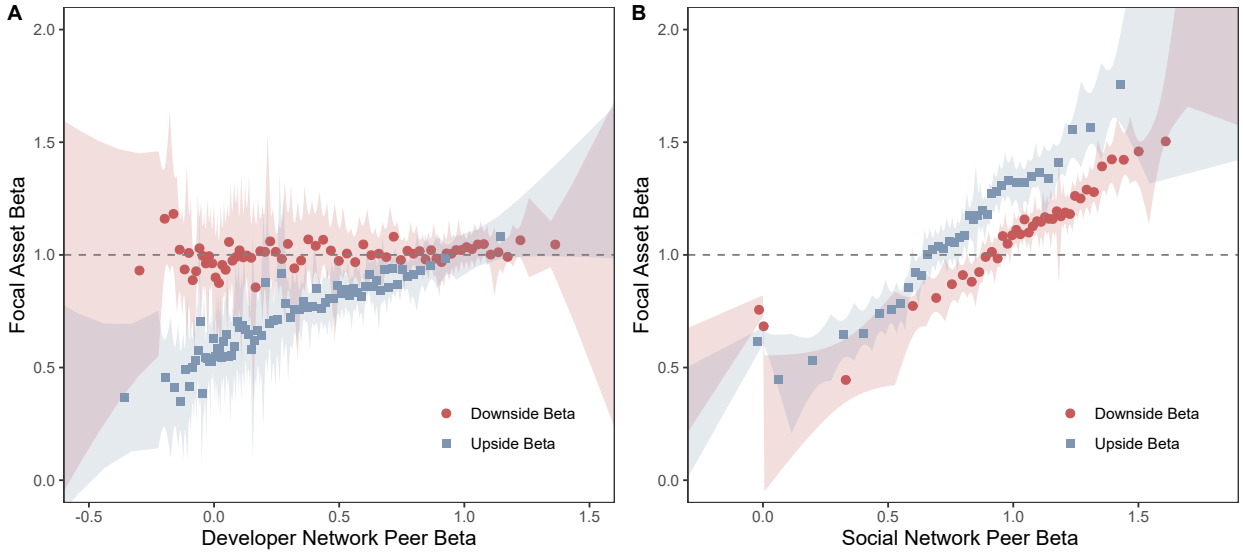


Figure 2: Network Spillovers of Tail Risks and Systemic Exposures

This figure presents covariate-adjusted binned scatter plots, using the methodology of Cattaneo et al. (2024), to illustrate the relationship between network peer and focal systematic risk. The analysis distinguishes between Downside Beta (red circles) and Upside Beta (blue squares). *Panel A* focuses on the Developer Network. The flat slope of the red series versus the positive slope of the blue series illustrates the network's role in facilitating upside participation without increasing downside exposure (the "shield" effect). *Panel B* focuses on the Social Network, where both series exhibit positive slopes, indicating symmetric risk transmission. Shaded areas represent 95% confidence intervals. Estimations include token and time fixed effects plus full controls. Axis ranges are zoomed to dense regions for visualization clarity, though calculations utilize the full sample with optimally chosen bins suggested by Cattaneo et al. (2024).

Collectively, these findings suggest that the developer network promotes economic maturity: it reshapes token properties by suppressing downside tail risks and repairing asymmetric exposure to systematic risks.

2.4. Falsification Tests on Topological Validity

We demonstrate that the documented risk shielding effect is a unique consequence of actual developer linkages, rather than an artifact of natural market fluctuations or chance. Essentially, the social network analyzed in Section 2.2 already offers a counterfactual that a

network does not necessarily create a shield effect. We next conduct a permutation test to rigorously rule out spurious correlations. Specifically, we randomly permute the mapping between tokens and network nodes while preserving the network structure and the tokens' time-series characteristics. We use the randomized mappings to re-estimate the full OLS model (Table 3, Column 3),[19] and draw the distribution of estimated coefficient η from 1,000 simulations as Figure A3 shows. The true estimate lies far in the left tail of the counterfactual distribution, indicating that the probability of observing such a shielding effect by chance is negligible ($p = 0.01$).

More rigorously, we employ a "Peak Test" to show that the shielding effect stems strictly from the observed developer network topology, rather than accidental connectivity or unobserved aggregate shocks unrelated to network structure. We systematically perturb the network by either randomly removing existing edges (signal loss) or adding spurious edges (noise injection) weighted by the empirical distribution.[20] If the observed developer network represents the precise channel for risk absorption, any topology distortion, whether by losing true information or mixing in false signals, should diminish the estimated effect.

Figure 3 visualizes the trajectory of the estimated coefficients across varying degrees of perturbation. The coefficient magnitude peaks exactly at the unperturbed state (0% perturbation) and decays toward zero as the perturbation ratio increases in magnitude, confirming that the shield effect is driven by the topological structure of the observed developer network.

Moreover, the decay pattern is highly asymmetric: when randomly removing edges, the effect decays slowly at the beginning, suggesting a degree of structural redundancy where the core network maintains functionality even with partial data loss. However, the effect collapses precipitously upon the addition of spurious edges. To better understand this pattern, we further simulate the theoretical network structural change ratio against perturbation ratio, which is independent of any token characteristics and market variables, as plotted in Figure OA-7. Remarkably, the asymmetric decay in Figure 3 closely mirrors the theoretical topological shock curve. These findings yield two critical insights beyond a falsification test. First, the tight alignment between the regression estimates and the theoretical topological curve confirms that our results are consistent with the observed topology carrying economically relevant information — if the findings were driven by omitted aggregate shocks or mere variable values, the coefficient trajectory would not track the

[19] For intuitive interpretation, we use the OLS baseline to perform the test, since the IV estimation involves an additional mapping process between developer network nodes and commit timing network nodes.

[20] To ensure that the added edges structurally mimic real connections, their weights are sampled with replacement from the empirical distribution of the observed edge weights.

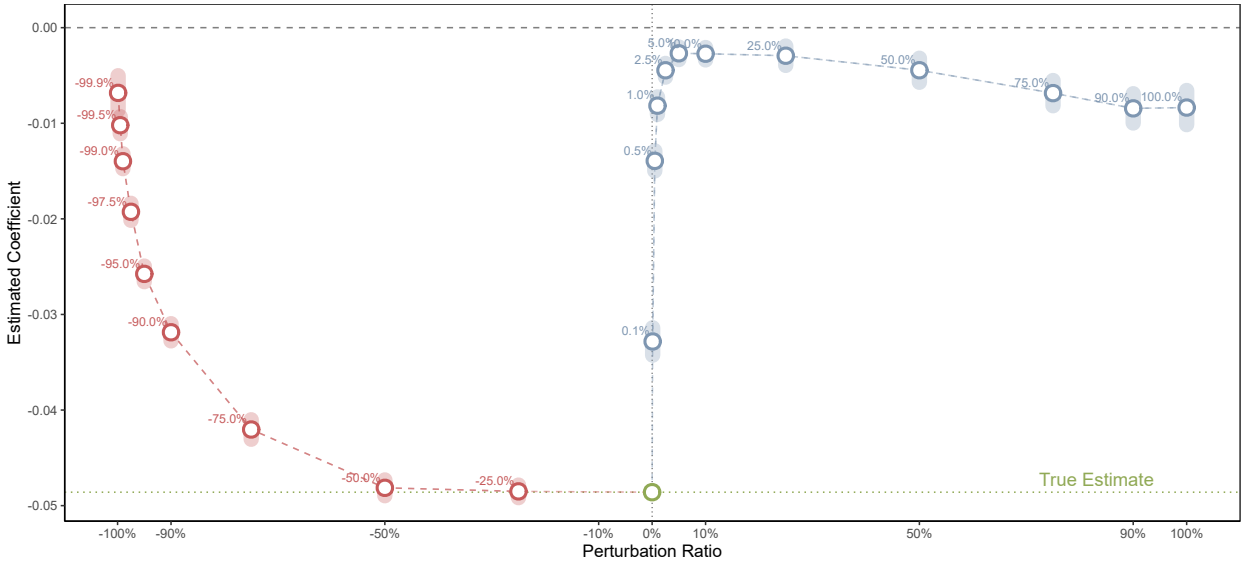


Figure 3: Network Structure Validity: The "Peak" Test

This figure illustrates the sensitivity of the estimated peer risk effect to random perturbations of the developer network structure. We employ the baseline OLS specification with full controls and fixed effects (corresponding to Column (3) of Table 2) to estimate the coefficient of *DevNet Peer IVOL*. The x-axis represents the perturbation ratio relative to the total number of existing edges, 24,309,556, i.e., a 0.1% perturbation roughly refers to removing or adding 24 thousands of edges. Negative perturbation ratios (red markers) indicate random edge removal (signal loss), where a percentage of true edges are randomly deleted. Positive perturbation ratios (blue markers) indicate random edge addition (noise injection), where spurious edges are randomly added between unconnected nodes. To ensure that the results are not driven by edge weight magnitude, the weights for added edges are drawn without replacement from the empirical distribution of existing edge weights. The y-axis reports the estimated coefficient. The green marker at 0% represents the "true estimate" (-0.042) from the actual observed network. Each point represents the mean coefficient from 100 independent simulations, with 95% confidence intervals (often small due to the large number of edge operations in each single simulation). The pattern demonstrates that the estimated effect magnitude peaks (in absolute terms) at the true network structure and attenuates significantly as the network is randomized.

topological deterioration so precisely. Second, the sharp decay under noise injection aligns with small-world network properties (Watts & Strogatz, 1998), where introducing even a few random links creates topological "shortcuts" that drastically shorten the characteristic path length. These spurious shortcuts create influential false pathways, dilute the true signal with noise, thereby severely attenuating the estimated shielding coefficient. This sensitivity to noise injection further confirms the structural specificity of our mechanism.

These observations have two implications for interpreting empirical data. On the one hand, the observed development records are highly authentic: real efforts leave distinct, high-fidelity traces that cannot be simulated by random noise. On the other hand, the presence of any casual links in the real-world data likely causes us to underestimate the true magnitude of the shielding effect. This supports the necessity of our IV approach, suggesting

that the reported estimates likely represent a conservative lower bound of the developer network's actual power.

3. Mechanisms

3.1. The Mobilization Channel: Developers' Response

The counter-intuitive risk shielding effect, distinct from the typically observed spillover effect of many networks, motivates us to suspect non-financial mechanisms, especially those rooted in the network's micro-foundations, i.e., the developers themselves. We propose the *Mobilization Hypothesis*: the developer network functions as an early warning system and enables developers to actively respond to peer risk. Given that connected projects often share underlying functional complementarities or code dependencies, a spike in risk among peers — often signaling technical vulnerabilities or exploits — serves as a wake-up call. This signal triggers a "rush to fix" effect, mobilizing developers to proactively update code, e.g., patch bugs and reinforce security. In this context, coding updates need not necessarily yield fundamental technological breakthroughs, but may be sufficient to influence market as effort signals. As a result, the mobilized constructive activities dampen idiosyncratic risks, either by resolving actual threats [21] or projecting developer competence to the market, generating a developer network shield.

To empirically validate this hypothesis, we examine whether developer network peer IVOL triggers higher intensity of developer efforts $M_{i,t+1}$, proxied as the number of commits within the next week, $ln(Commits)_{i,t+1}$,[22] and then we show that higher $M_{i,t+1}$ is associated with lower focal IVOL.

Table 5 presents the results. Columns (1)-(3) test whether peer risk triggers mobilized response, reflected as greater $M_{i,t+1}$. Column (1) shows that *DevNet Peer IVOL* has a significantly positive coefficient (0.485, t = 8.66), which supports our hypothesis. As our IV approach suggests, commit records are mixed with noise components such as casual commits. Column (2) adopts a reduced-form test that directly regresses M on the instrument, *Past TimingNet Peer IVOL*. The highly significant positive coefficient validates that the actual

[21] For example, Ransbotham and Mitra (2009) demonstrate that in the broader software context, timely remediation effectively disrupts the attack escalation, preventing potential risks from materializing into actual compromises.

[22] To accommodate zero values, we apply the inverse hyperbolic sine (asinh) transformation, $\text{asinh}(x) = \ln(x + \sqrt{x^2 + 1})$, which behaves similarly to a logarithmic transformation, is well defined and approximately linear at zero. To deal with the recent critique regarding the scale-dependence of log-like transformations (Chen & Roth, 2024), we explore alternative specifications in Online Appendix OA-C5, and show that peer IVOL promotes developer activity both extensively and intensively.

collaboration structure has strong impact on the intensity of developer efforts. Column (3) shows the 2SLS estimate. The estimated coefficient of the instrumented *DevNet Peer IVOL* is 4.451 (t = 18.59), indicating that developer network peer IVOL induces a striking rush in commits for the focal token project. Notably, Column (3) also suggests that peer risk from social links has minimal impact on developer efforts (−0.008, t = −0.61).

Table 5: Test The Mobilization Channel

This table reports the regression results for the developer mobilization channel. Columns (1)-(3) examine the mobilization response, where the dependent variable is developer effort M, in week $t+1$, measured as $\ln(Commits)_{i,t+1}$. Column (1) reports the OLS estimate using *DevNet Peer IVOL* as main independent. Column (2) reports the reduced-form OLS estimate using the instrument, *Past TimingNet Peer IVOL*, as main independent. Column (3) reports the 2SLS estimate using instrumented *DevNet Peer IVOL* as main independent. Columns (4)-(6) examine the relationship between mobilization $M_{i,t}$ and the focal token's IVOL $IVOL_{i,t+1}$. Column (4) presents the OLS estimation. Column (5) presents the 2SLS estimation where M is instrumented by *Past TimingNet Peer IVOL*. Column (6) presents a decomposition analysis where developer effort is split into *Structural Effort* $\widehat{M_{i,t+1}}$, the fitted values from Column (2), and *Residual Effort* $\widehat{\varepsilon_{M_{i,t+1}}}$, the residuals from Column (2). All specifications include token and time fixed effects and full controls. The variable *Commits*, which contain zero values, is transformed using the inverse hyperbolic sine function, $\text{asinh}(x) = \ln(x + \sqrt{x^2+1})$ to account for zeros while approximating a logarithmic interpretation, denoted as $\ln(\cdot)$ for interpretation simplicity. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with a lag of 5 weeks. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Dependent:	$M_{i,t+1} \equiv ln(Commits)_{i,t+1}$			$IVOL_{i,t+1}$		
	OLS (1)	OLS (2)	2SLS (3)	OLS (4)	2SLS (5)	OLS (6)
$M_{i,t+1}$				-0.021**	-0.148***	
				(-2.429)	(-4.410)	
DevNet Peer IVOL	0.485***		4.451***			
	(8.686)		(18.592)			
Past TimingNet Peer IVOL		0.917***				
		(19.598)				
$\widehat{M_{i,t+1}}$						-0.148***
						(-4.651)
$\widehat{\varepsilon_{M_{i,t+1}}}$						-0.010
						(-1.176)
SocNet Peer IVOL	0.019**	0.015*	-0.008	0.842***	0.845***	0.845***
	(2.486)	(1.915)	(-0.606)	(6.932)	(6.928)	(6.946)
Controls	✓	✓	✓	✓	✓	✓
Token FE	✓	✓	✓	✓	✓	✓
Time FE	✓	✓	✓	✓	✓	✓
Observations	210,202	210,202	210,202	209,752	209,752	209,752
Adjusted R^2	0.802	0.806	0.193	0.393	0.391	0.393
Kleibergen-Paap F-stat			477.2		382.7	
Wu-Hausman p-val			<0.001		<0.001	

Table 5 Columns (4)–(6) further examine the consequence stage. The OLS estimate in Column (4) shows a negative coefficient (-0.021, t = −2.43), supporting the shielding effect, yet likely attenuated by measurement error from ineffective "noise" commits. Column (5) addresses this by instrumenting effort with *Past TimingNet Peer IVOL*, which isolates

the structural component of mobilization. The coefficient intensifies sevenfold to −0.148 ($t = -4.41$), confirming that structure-induced efforts exert a powerful shielding force. Notably, the product of the first-stage mobilization elasticity (4.451) and this shielding efficacy (−0.148) yields an implied total effect of −0.658, effectively replicating the baseline estimate of −0.660 (Table 3).[23] Column (6) further decomposes effort into structural ($\widehat{M}$) and residual ($\widehat{\varepsilon_M}$) components. The results confirm that only efforts rooted in the collaboration structure shield value; residual efforts are statistically irrelevant, consistent with the "code washing" phenomenon (Gefen et al., 2024).

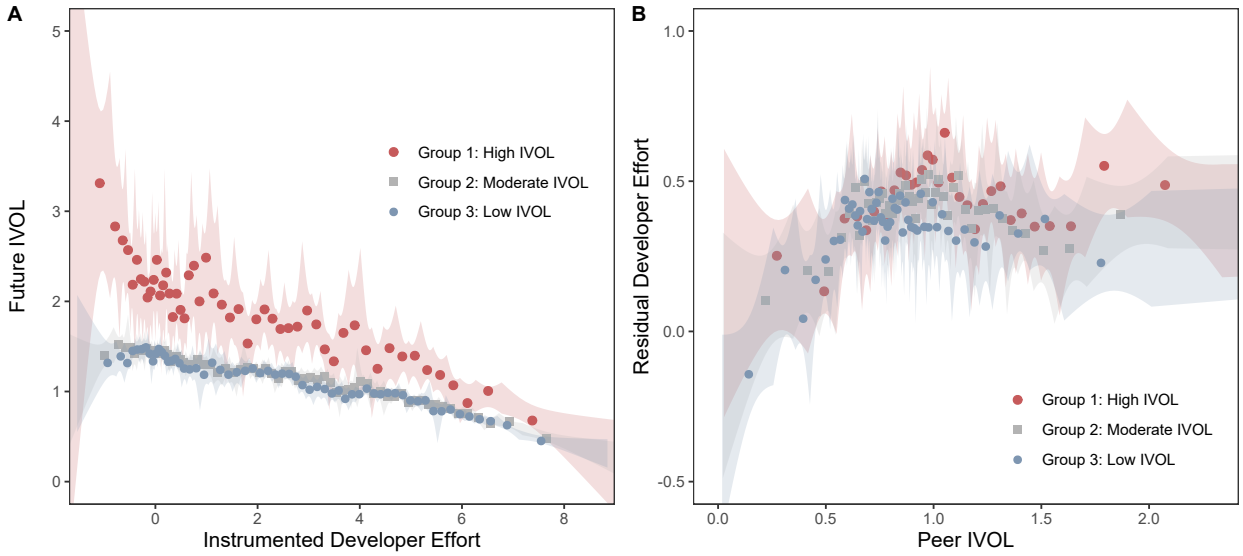


Figure 4: Decomposing Developers' Commit Effort: Mobilization vs. Panic Response

This figure presents covariate-adjusted binscatter plots (Cattaneo et al., 2024) visualizing the dual channels of developer mobilization. The sample is stratified by lagged focal IVOL into High (30%), Moderate (40%), and Low (30%) groups. Shaded areas represent 95% confidence intervals. *Panel A (Mobilization Channel)* plots future focal IVOL against Instrumented Developer Effort (fitted values of regressing $\ln(Commits)$ on lagged TimingNet Peer IVOL). *Panel B (Noise Channel)* plots Residual Developer Effort (residuals of regression used in Panel A, representing uncoordinated activity) against DevNet Peer IVOL. Note that the single data point in the negative region comes from the algorithm's handling of mass points; this bin represents a substantial sample share, statistically balancing the dispersed positive residuals. Estimations include token and time fixed effects plus full controls. Axis ranges are zoomed to dense regions for visualization clarity, though calculations utilize the full sample with optimally chosen bins suggested by Cattaneo et al. (2024).

To further understand how this mobilization effect interacts with one token project's own risk, Figure 4 Panel A plots the binned scatter (Cattaneo et al., 2024) of future IVOL against instrumented effort ($\widehat{M}$) across terciles of current IVOL. The uniformly negative slopes confirm that the structural defense is universal: mobilization reduces future risk

[23] The indirect effect can be roughly calculated as the product of the mobilization elasticity (4.451) and the shielding efficacy (−0.148). Conceptually, $\frac{\Delta \text{FocalRisk}}{\Delta \text{PeerRisk}} = \frac{\Delta \text{LogCommits}}{\Delta \text{PeerRisk}} \times \frac{\Delta \text{FocalRisk}}{\Delta \text{LogCommits}}$.

regardless of whether the asset is currently stable or volatile. Panel B further explores the residual effort. We observe a positive relationship between Peer IVOL and residual effort, suggesting that peer distress does simultaneously trigger unstructured efforts. Notably, such relationship appears to be concave and similar across IVOL terciles. We term this "panic coding": a reactive compulsion to simply signal presence in response to peer shocks, but does not actually related to the level of peer shocks. Since this panic-induced activity generates noise without shielding value, it attenuates OLS estimates, further rationalizing our IV approach in isolating the effective structural developer mobilization.

3.2. Network Information Transmission

Given that individual code commits may have limited immediate impact on asset fundamentals, we propose that the mobilization channel achieves the shielding effect mainly through a signaling mechanism. Drawing on Hirshleifer et al. (2025), we hypothesize that the risk shield effect is heterogeneous with regard to token's network positions, proxied by eigenvector centrality, reflecting information diffusion within the network.

Table 6 examines the interplay between the developer and social networks. In Panel A, the interaction between *DevNet Peer IVOL* and *SocNet EC* is positive and statistically significant(0.755, t = 4.81). This result indicates that high social network prominence significantly attenuates the risk-shielding effect. That is, when a token is central in the social network, the intense transmission of sentiment and noise drowns out developer signals, rendering the shield ineffective.

In Panel B, we interact social peer risk measures with developer network centrality. The interaction terms exhibit significantly negative coefficients, e.g., −0.447 (t = −4.63) for *SocNet Peer IVOL* × *DevNet EC*. This indicates that when the focal asset is central in the developer network, the amplified transmission of its resilience signal effectively counteracts the risk contagion from social channels. Together, Panels A and B suggest that two networks function as competing transmission infrastructures and attention amplifiers (Hirshleifer et al., 2025) for different types of information. When a project occupies a central position in social (developer) networks, its technical activities (sentiment spillovers) become less pronounced in driving risk transmission.

Table 6 Panel C investigates whether the shielding effect varies with the focal token's centrality within the developer network itself. Unlike the competitive effects above, the interaction terms (*DevNet Peer IVOL*×*DevNet EC*) are statistically insignificant, suggesting that the intrinsic benefit of the shield is universal across the developer network. This aligns with the logic of the main channel discussed above: mobilization is not a privilege reserved for

Table 6: Network Centrality and Risk Transmission

This table reports 2SLS estimates examining how the impact of peer risk is moderated by the focal token's network position, proxied by Eigenvector Centrality (EC). The dependent variables (Y) across columns are: Idiosyncratic Volatility ($IVOL$), Downside Volatility ($DSVOL$), Maximum Return (MAX), Expected Shortfall ($ES5$), Downside Beta ($DSBETA$), and Upside Beta ($USBETA$). *Panel A* interacts *DevNet Peer Y* with the token's centrality in the Social Network (*SocNet EC*) to test if social noise amplifies or dampens the developer shield. *Panel B* interacts *SocNet Peer Y* with the token's centrality in the Developer Network (*DevNet EC*) to test if developer coordination mitigates social contagion. *Panel C* interacts *DevNet Peer Y* with the token's centrality in the Developer Network (*DevNet EC*) to test for structural heterogeneous effects within the developer network. All specifications include the full set of controls, token fixed effects, and time fixed effects. In particular, both DevNet Peer Y and SocNet Peer Y are included in each panel. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with lags corresponding to the dependent variable's window (5 or 15 weeks). $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Dependent (Y):	$IVOL_{i,t+1}$ (1)	$DSVOL_{i,t+1}$ (2)	$MAX_{i,t+1}$ (3)	$ES5_{i,t+1}$ (4)	$DSBETA_{i,t+1}$ (5)	$USBETA_{i,t+1}$ (6)
Panel A: Heterogeneous Developer Network Effects across Social Network Positions						
DevNet Peer Y	-0.689***	-0.207***	-0.825***	-0.195***	-0.003	0.425**
	(-4.775)	(-4.989)	(-4.165)	(-3.222)	(-0.019)	(2.030)
DevNet Peer Y × SocNet EC	0.755***	0.220***	0.909***	0.301***	0.644***	0.800**
	(4.814)	(3.170)	(3.890)	(3.665)	(2.857)	(1.968)
SocNet EC	-0.383*	0.011	-0.086*	0.001	-0.877***	-0.436
	(-1.900)	(0.196)	(-1.834)	(0.093)	(-3.612)	(-1.439)
Observations	209,752	209,752	209,752	195,040	209,752	209,752
Adjusted R^2	0.385	0.519	0.287	0.569	0.054	0.071
Panel B: Heterogeneous Social Network Effects across Developer Network Positions						
SocNet Peer Y	0.859***	0.140***	1.012***	0.117***	0.543***	0.948***
	(6.954)	(10.151)	(6.176)	(7.643)	(4.595)	(6.716)
SocNet Peer Y × DevNet EC	-0.447***	0.034*	-0.520***	0.025	-0.289**	-0.445***
	(-4.633)	(1.778)	(-4.565)	(1.130)	(-2.468)	(-2.673)
DevNet EC	0.162	-0.028	0.034	-0.004	0.263	0.071
	(1.494)	(-1.497)	(1.373)	(-1.579)	(1.543)	(0.325)
Observations	209,752	209,752	209,752	195,040	219,598	219,598
Adjusted R^2	0.393	0.526	0.299	0.575	0.045	0.058
Panel C: Heterogeneous Developer Network Effects across Developer Network Positions						
DevNet Peer Y	-0.676***	-0.219***	-0.210***	-0.804***	0.037	0.476**
	(-4.535)	(-3.166)	(-4.852)	(-3.942)	(0.217)	(2.212)
DevNet Peer Y × DevNet EC	-0.185	-1.211	-0.115	-0.595	-0.382	0.395
	(-0.549)	(-1.077)	(-0.584)	(-1.305)	(-0.274)	(1.120)
DevNet EC	0.355	0.159	0.207	0.136	0.315	-0.376*
	(0.907)	(1.133)	(0.960)	(1.484)	(0.236)	(-1.906)
Observations	209,752	195,040	209,752	209,752	209,752	209,752
Adjusted R^2	0.384	0.558	0.518	0.286	0.054	0.072
Controls	✓	✓	✓	✓	✓	✓
Token FE	✓	✓	✓	✓	✓	✓
Time FE	✓	✓	✓	✓	✓	✓

central projects, but rather a micro-founded property of the collaborative structure itself.[24]

[24]In contrast with this universal nature, we show in Online Appendix OA-C9 that the social network is stratified, the transmission of social risk is significantly amplified for central nodes, consistent with Hirshleifer et al. (2025).

3.3. Alternative Explanations

3.3.1. Distinguishing from Mechanical Mean-Reversal

A potential concern is that our findings might be driven by mechanical mean reversion of risk measures: if the developer network merely proxies for a common latent risk factor, a spike in peer risk at time t might coincide with a transitory peak in focal risk, followed by a natural reversal at $t+1$. Table 7 presents a series of tests to rule out this possibility.

First, we examine contemporaneous correlations. If the mechanical reversal hypothesis held, focal and peer risks should be positively correlated within the same period. However, Panel A Columns (1)–(2) show that DevNet Peer $\text{IVOL}_{i,t+1}$ is significantly negatively associated with $\text{IVOL}_{i,t+1}$. Second, we perform a horse-race test between lagged and contemporaneous peer risks.[25] Columns (3)–(4) show that both terms remain significantly negative, confirming that the shielding effect is not an artifact of timing selections. Third, and most importantly, Column (5) uses 2SLS estimation to decompose the effect. We instrument the peer IVOL at t to isolate the mobilization channel, while controlling for contemporaneous peer IVOL ($t+1$) to capture instantaneous market spillovers, yielding two opposing estimates: (i) The contemporaneous coefficient turns positive (1.452, $t = 3.50$), consistent with standard risk spillover, possibly due to common market exposure; (ii) The instrumented lagged coefficient becomes strongly negative (−2.025, $t = 3.99$). Notably, the sum of these two forces (-0.57) aligns with the estimated total effect in our baseline specification (−0.66). That is, despite the existence of contemporaneous co-movement, the developer network generates a powerful counterforce that overcome the natural spillover, resulting in the net shielding effect. Table 7 Panel B adopts the same 2SLS approach to other main risk measures, including *DSVOL*, *DUVOL*, *NCSKEW*, *ES*5 and *MAX*, with all confirming the decomposed effects. Therefore, despite the common spillovers, the developer network effect is the dominant driver of the observed net shield.

3.3.2. Factor Absorption

Another mechanical concern, though does not apply to every risk measure, is that peer uncertainty is merely absorbed into common risk factors, mechanically reducing idiosyncratic volatility. Table 8 Columns (1)-(3) reject this hypothesis. Column (1) orthogonalizes peer risk against factor exposures,[26] the estimated coefficient (−0.660) is negative and persistent. Moreover, if the observed shielding effects were due to factor absorption, then

[25] Note that there is mechanical overlap inherent in rolling volatility windows, which typically biases coefficients toward similarity, while this does not affect our main inference.

[26] By explicitly conditioning on factor exposures within the 2SLS framework, this specification isolates the variation in peer risk that is independent of systematic market co-movements.

Table 7: Simultaneous Effect Test

This table reports OLS and 2SLS estimates designed to examine mechanical mean reversion as an alternative explanation. *Panel A* focuses on idiosyncratic volatility ($IVOL_{i,t+1}$). Columns (1)-(2) present OLS estimates of the aggregate net effect using only the contemporaneous term (*DevNet Peer* $IVOL_{i,t+1}$). Columns (3)-(4) present OLS estimates with both the contemporaneous term and the lagged term (*DevNet Peer* $IVOL_{i,t}$). Column (5) presents 2SLS estimates, which instruments the lagged term (*DevNet Peer* $IVOL_{i,t}$) to separate the latent active mobilization channel from the aggregate net effect, leaving the contagion channel loaded on the contemporaneous coefficient. *Panel B* extends this 2SLS test to other risk measures, including Downside Volatility (*DSVOL*, Column 1), Downside-Upside Volatility Ratio (*DUVOL*, Column 2), Negative Coefficient of Skewness (*NCSKEW*, Column 3), and Expected Shortfall at the 5% level (*ES*5, Column 4), and 30-day Maximum Daily Return (*MAX*, Column 5). All specifications include the full set of controls, token fixed effects, and time fixed effects. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with lags corresponding to the dependent variable's window. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Panel A: Tests on Idiosyncratic Volatility

	Dependent Variable: $IVOL_{i,t+1}$				
	OLS				2SLS
	(1)	(2)	(3)	(4)	(5)
DevNet Peer $IVOL_{i,t+1}$	-0.098***	-0.051***	-0.055***	-0.027**	1.452***
	(-4.815)	(-3.799)	(-3.738)	(-2.346)	(3.504)
DevNet Peer $IVOL_{i,t}$			-0.058***	-0.032***	-2.025***
			(-3.677)	(-2.695)	(-3.992)
SocNet Peer $IVOL_{i,t+1}$		1.185***		1.185***	1.192***
		(12.490)		(12.491)	(12.549)
Controls		✓		✓	✓
Token FE	✓	✓	✓	✓	✓
Time FE	✓	✓	✓	✓	✓
Observations	219,607	209,752	219,598	209,752	209,752
Adjusted R^2	0.164	0.407	0.164	0.407	0.369
Kleibergen-Paap F-stat					60.3
Wu-Hausman p-val					<0.001

Panel B: 2SLS Tests on Other Measures

Dependent (Y):	$DSVOL_{i,t+1}$	$DUVOL_{i,t+1}$	$NCSKEW_{i,t+1}$	$ES5_{i,t+1}$	$MAX_{i,t+1}$
	(1)	(2)	(3)	(4)	(5)
DevNet Peer $Y_{i,t+1}$	0.706***	0.885***	2.041***	0.832***	2.785***
	(5.002)	(5.011)	(3.332)	(3.211)	(2.645)
DevNet Peer $Y_{i,t}$	-0.847***	-1.097***	-2.605***	-0.993***	-4.111***
	(-4.968)	(-5.015)	(-3.370)	(-3.231)	(-3.186)
SocNet Peer $Y_{i,t}$	0.182***	0.490***	0.595***	0.128***	1.475***
	(12.250)	(15.011)	(14.654)	(8.299)	(10.673)
Controls	✓	✓	✓	✓	✓
Token FE	✓	✓	✓	✓	✓
Time FE	✓	✓	✓	✓	✓
Observations	209,752	209,651	195,083	195,040	205,651
Adjusted R^2	0.488	0.493	-0.027	0.516	0.134
Kleibergen-Paap F-stat	173.6	207.5	62.7	122.8	17.0
Wu-Hausman p-val	<0.001	<0.001	<0.001	<0.001	<0.001

we would see insignificant impacts on total volatility and increased factor-implied volatility (FVOL).[27] However, Column (2)-(3) show that the total volatility of the focal token declines significantly (-0.643, $t = -4.45$), while there is no significant FVOL changes (-0.052, $t = -1.36$). Collectively, these results confirms a net reduction of risk, rather than a reallocation from idiosyncratic to systematic exposures.

3.3.3. Investor Neglect

We further address behavioral interpretations related to limited investor attention. Shocks to peer assets might occupy investor attention, leaving the focal asset overlooked. Such neglect could lead to thin trading and stale prices, resulting in low realized volatility. Table 8 Columns (4)-(5) reject this interpretation. If the observed risk reduction were driven by neglect, we would expect a decline in trading activity and a decoupling from market fundamentals. In contrast, Column (4) shows a significant increase in log Turnover (0.313, $t = 2.54$), indicating that the focal asset remains actively traded rather than being overlooked. Column (5) shows an increase in CAPM R^2 (0.028, $t = 2.07$), suggesting that the asset's price efficiency and synchronicity with market factors actually improve. These results confirm that the main findings are not driven by passive neglect or stale pricing.

3.3.4. Liquidity Channel

Interpretations regarding liquidity are twofold. On the one hand, there might be a liquidity contagion, where peer distress drains liquidity from related assets. However, microstructure theory suggests that such liquidity dry-ups typically exacerbate volatility due to increased price impact—contradicting our finding of reduced risk. Table 8 Column (6) further shows that DevNet Peer Risk is associated with a significantly lower Amihud Illiquidity (-0.571, $t = -3.30$), indicating that market depth actually improves rather than deteriorates. On the other hand, a possible channel is that the observed risk shield might merely reflect a relative advantage where the focal asset becomes significantly more liquid than its crashing peers. Column (7) tests this by regressing Relative Illiquidity, i.e., the gap between focal and peer Illiquidity, on peer risk. The coefficient is statistically insignificant (-0.054, $t = -0.54$), suggesting no systematic shift in relative liquidity dominance. Furthermore, Column (8) confirms that even after explicitly controlling for Peer Illiquidity, the shielding effect remains robust (-0.795, $t = -4.48$). Collectively, these results demonstrate that liquidity channels

[27] Mathematically, total volatility is decomposed into systematic and idiosyncratic components: $VOL^2_{i,t} \approx FVOL^2_{i,t} + IVOL^2_{i,t}$, where $FVOL$ captures the risk explained by common factors.

are not the main driver of our findings.[28]

3.3.5. Flight-to-Quality and Pricing Implications

The above tests point to a potential flight-to-quality phenomenon. This inference does not undermine our mechanism, but essentially can be viewed as investors' realized market reaction in response to the signals from the developer network. A key question is whether flight-to-quality here induces speculative price distortions (e.g., from panic buying) and influences market stability. Table 8 Columns (9)-(11) show that DevNet Peer Risk has an insignificant impact on future cumulative abnormal returns (CAR) across multiple daily windows, including [1,3], [4,10], and [1-14], suggesting no significant panic buying. This null result further complements the characterization of the shielding effect: it effectively operates on the second moment (risk/volatility) without distorting the first moment (price drifts), implying that developers' mobilization is less likely to carry cash-flow news but influences the information environment.[29] Moreover, the market identifies the developer shield as a source of fundamental stability, rather than a speculative opportunity, providing a source of resilient quality.

3.3.6. Developers' Wealth Reallocation

Finally, we discuss a potential financial channel where developers themselves are influential investors. It is plausible that co-developers hold tokens as incentives or investment stakes in projects they contribute to (e.g., Cong et al., 2021a). Then a crash in a peer token could lead developers to rebalance allocations, generating a risk transmission through the developer network independent of productive coding efforts. While current data limitations prevent us from directly verifying this channel, it conceptually parallels the "common ownership" channel (e.g., Anton & Polk, 2014), which is more likely to generate a risk spillover rather than a risk-shielding effect. Rigorous investigation might benefit from the comprehensive and transparent on-chain transaction data, leaving for future research.

[28] In contrast, SocNet Peer Risk is associated with a significantly lower CAPM R^2 (−0.014) and higher illiquidity (0.106). This implies that while social contagion also triggers turnover, it generates a speculative anomaly that decouples the asset from fundamentals and deteriorates liquidity.

[29] From an asset pricing perspective, the "lower risk without sacrificing returns" might theoretically suggest profitable strategies, which is beyond the scope of this paper. We view the critical value of the developer network lies in downside protection (avoiding the crash) rather than alpha generation. In addition, in the crypto market, severe limits to arbitrage and high failure risks likely prevent this resilience premium from being easily captured as pure alpha, which probably explains why we observe rational price stability rather than efficient arbitrage corrections.

Table 8: Tests on Alternative Explanations

This table reports two-stage least squares (2SLS) panel regressions examining the effects of developer network peer volatility (*DevNet Peer IVOL*) on risk, liquidity, and return outcomes. The dependent variables include next-period idiosyncratic volatility ($IVOL_{i,t+1}$), total volatility ($VOL_{i,t+1}$), factor-implied volatility ($FVOL_{i,t+1}$), log turnover ($Turnover_{i,t+1}$), $R^2_{i,t+1}$ from a CAPM regression, log Amihud illiquidity ($ILLIQ_{i,t+1}$), relative illiquidity ($Rel.ILLIQ_{i,t+1} = ILLIQ_{i,t+1} - DevNet\ Peer\ ILLIQ_{i,t+1}$), and cumulative abnormal returns (CAR) over various daily horizons. *DevNet Peer IVOL* is instrumented using the 1-month lagged commit timing network peer IVOL (*TimingNet Peer IVOL*). In Column (1), peer IVOL is orthogonalized within the 2SLS framework by conditioning on exposures to factors. Past dependent variable (Y) controls are included where appropriate, depending on the outcome variable. Characteristic controls include full controls used in baseline specifications as in Table 2, except that trading volume ($\ln(Vol)$) is explicitly excluded when Y is turnover or illiquidity-related measures to avoid mechanical correlation. Past monthly cumulative abnormal returns are controlled in the CAR specifications. All regressions include token and time fixed effects. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with a lag of 5 weeks. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Testing Channel:	Factor Absorption			Investor Neglect		Liquidity Channel			Flight-to-Quality		
Dependent (Y):	*IVOL*	*VOL*	*FVOL*	*Turnover*	R^2	*ILLIQ*	*Rel.ILLIQ*	*IVOL*	$CAR_{[1,3]}$	$CAR_{[4,10]}$	$CAR_{[1,14]}$
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
DevNet Peer IVOL	-0.660***	-0.643***	-0.052	0.313**	0.028**	-0.571***	-0.054	-0.795***	0.006	-0.013	-0.002
	(-4.652)	(-4.454)	(-1.360)	(2.543)	(2.070)	(-3.297)	(-0.537)	(-4.481)	(0.668)	(-0.642)	(-0.056)
SocNet Peer IVOL	0.847***	0.855***	0.109***	0.025**	-0.014***	0.106***	0.077***	0.847***	-0.009	-0.006	-0.029
	(7.043)	(6.965)	(5.036)	(2.163)	(-5.977)	(5.293)	(6.760)	(6.966)	(-1.568)	(-0.414)	(-1.066)
DevNet Peer ILLIQ								-0.070***			
								(-2.843)			
Orthogonalized (IV)	✓										
Past Y	✓	✓	✓				✓	✓			
Past CAR									✓	✓	✓
Controls	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Token FE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Time FE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Observations	209,752	209,752	209,752	209,752	209,752	209,752	209,752	209,752	211,642	211,620	211,661
Adjusted R^2	0.385	0.385	0.332	0.660	0.554	0.823	0.843	0.381	0.225	0.244	0.384
Kleibergen-Paap F-stat	472.1	472.3	472.8	482.2	472.3	482.2	512.0	359.9	475.9	475.7	475.8
Wu-Hausman p-val	<0.001	<0.001	0.011	<0.001	<0.001	<0.001	<0.001	<0.001	0.618	0.443	0.850

4. Extended Implications

4.1. Heterogeneous Effects across Token Categories and Narratives

We examine the effect heterogeneity across 55 token categories and narratives. Sourced from CoinMarketCap, one of the leading crypto market aggregators, these categories cover all existing labels with over 100 tokens, supplemented by economically significant ecosystems (e.g., Bitcoin Ecosystem) to ensure comprehensive coverage without selection bias. We employ 55 separate 2SLS regressions with each interacting the developer network variable with one category label, and report the total semi-elasticity (main effect plus interaction) for four risk measures: *IVOL*, *DSVOL*, *ES*5%, and *MAX*.[30]

Figure 5 visualizes the results. First, the shielding effect is systemically pervasive. The vast majority of coefficients lie to the left of zero (negative), confirming that developer-induced stability is a fundamental feature rather than a niche phenomenon restricted to specific sectors. Second, the magnitude of the shield aligns with technical complexity. Ecosystem-based categories (e.g., Ethereum Ecosystem) and infrastructure-critical categories (e.g., DeFi, DePIN) exhibit strong shielding effects, validating our mechanism that professional maintenance is most valuable for assets where code quality and coordinated problem solving are extremely important.[31] Third, the shield weakens in sentiment-driven sectors. Categories such as Memes and Gaming show negligible effects, in which the necessity for technical maintenance is ordinarily minimal.[32]

4.2. Time-series Heterogeneity

To investigate whether the shielding effect is a transient anomaly or a persistent market feature, Figure 6 plots the time-varying total effects of developer peer risk derived from year-interacted 2SLS regressions. During the nascent phase (2014–2019), estimates exhibit high variance and statistical insignificance, reflecting the noise inherent in an early-stage market with fewer assets and likely unstandardized development practices. However, a structural convergence is evident post-2020: the coefficients stabilize significantly below zero across all tail-risk measures (e.g., *IVOL*, *NCSKEW*, and *ES*5) and tightly track the pooled

[30]Classifications are not mutually exclusive: a single token may be associated with multiple labels simultaneously (e.g., a token can be classified under both Layer 2 and Zero Knowledge Proofs). Running separate regressions helps avoid multicollinearity issues arising from overlapping categories, and reporting total semi-elasticity helps comparison across categories.

[31]In particular, Optimism Ecosystem and Arbitrum Ecosystem exhibit the strongest effects, which are categories that with both ecosystem-based and infrastructure-critical (both are Layer-2 solutions) features.

[32]This is also consistent with our findings in Section 3.2: for assets driven primarily by social attention rather than technological utility, the developer defense is naturally less pronounced.

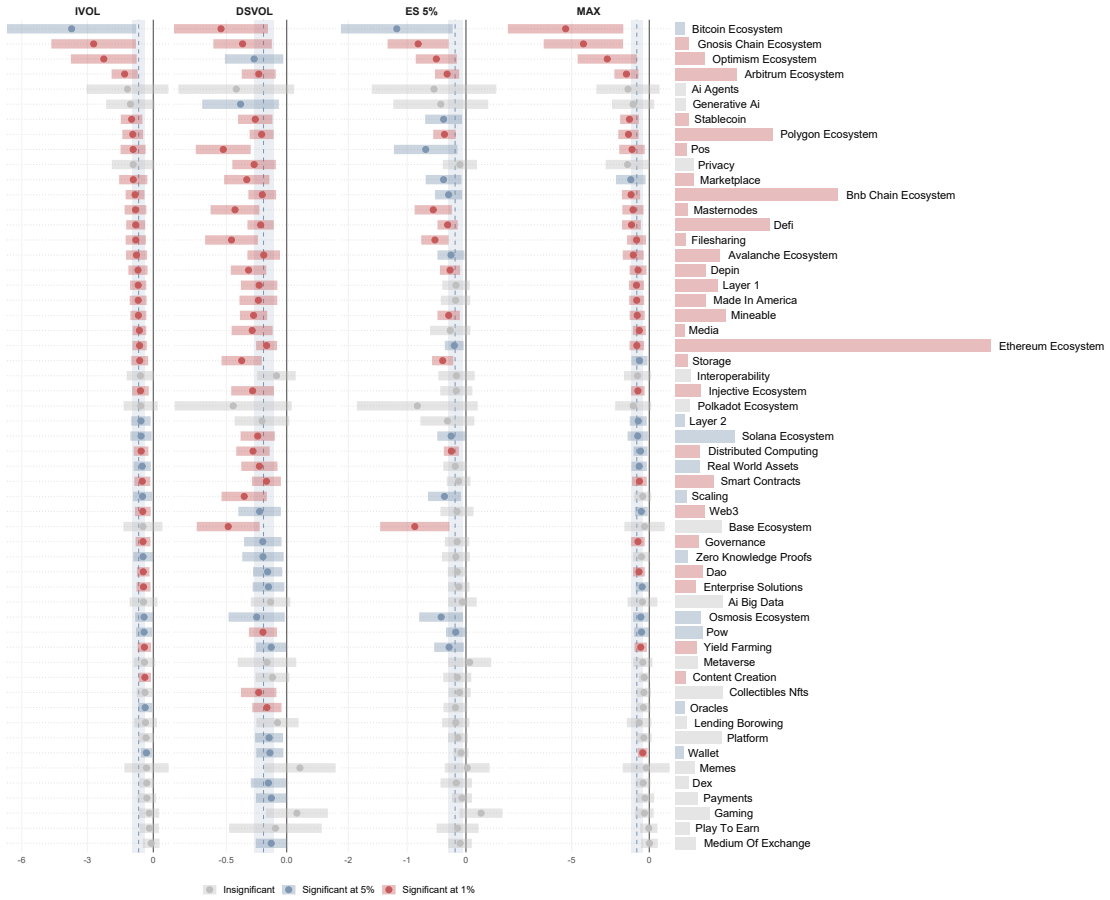


Figure 5: Heterogeneous Effects across Token Categories and Narratives

This figure illustrates the heterogeneous effects of the developer network across 55 token categories / narratives. The category information is collected from *CoinMarketCap*. The examined categories covers all the existing labels with more than 100 tokens as of March 2025. To ensure comprehensive coverage without introducing arbitrary selection bias, we also include specific economically significant categories with 20–100 tokens (e.g., Bitcoin Ecosystem) based on their established market relevance. Note that the categories are not mutually exclusive: a single token may be classified into multiple narratives (e.g., a token can belong to both "Layer 2" and "Zero Knowledge Proofs"). The panels report the estimated total effect coefficients of developer network peer risk on four measures: Idiosyncratic Volatility (*IVOL*), Downside Volatility (*DSVOL*), Expected Shortfall at the 5% level (*ES*5%), and Maximum Return (*MAX*). The coefficients are derived from IV regressions that interact the developer network variable with category indicators; the reported effect is the sum of the main effect and the specific category interaction term, where the standard errors are calculated via delta method. The 95% confidence intervals of θ_y are based on Driscoll–Kraay standard errors with lags corresponding to the measures' evaluation windows. The dashed vertical line indicates the benchmark IV estimates without category interactions. All regressions include token and date fixed effects as well as full controls. The right panel reports the sample size (number of unique tokens) for each category. Categories are sorted by the magnitude of the effect on *IVOL*. The color coding corresponds to the statistical significance of the coefficients in the *IVOL* model (red: 1% level; blue: 5% level; grey: insignificant).

baseline (dashed line). This finding suggests that as the digital asset ecosystem came to a mature stage, technical resilience has became more pronounced, serving as a robust feature across market cycles.

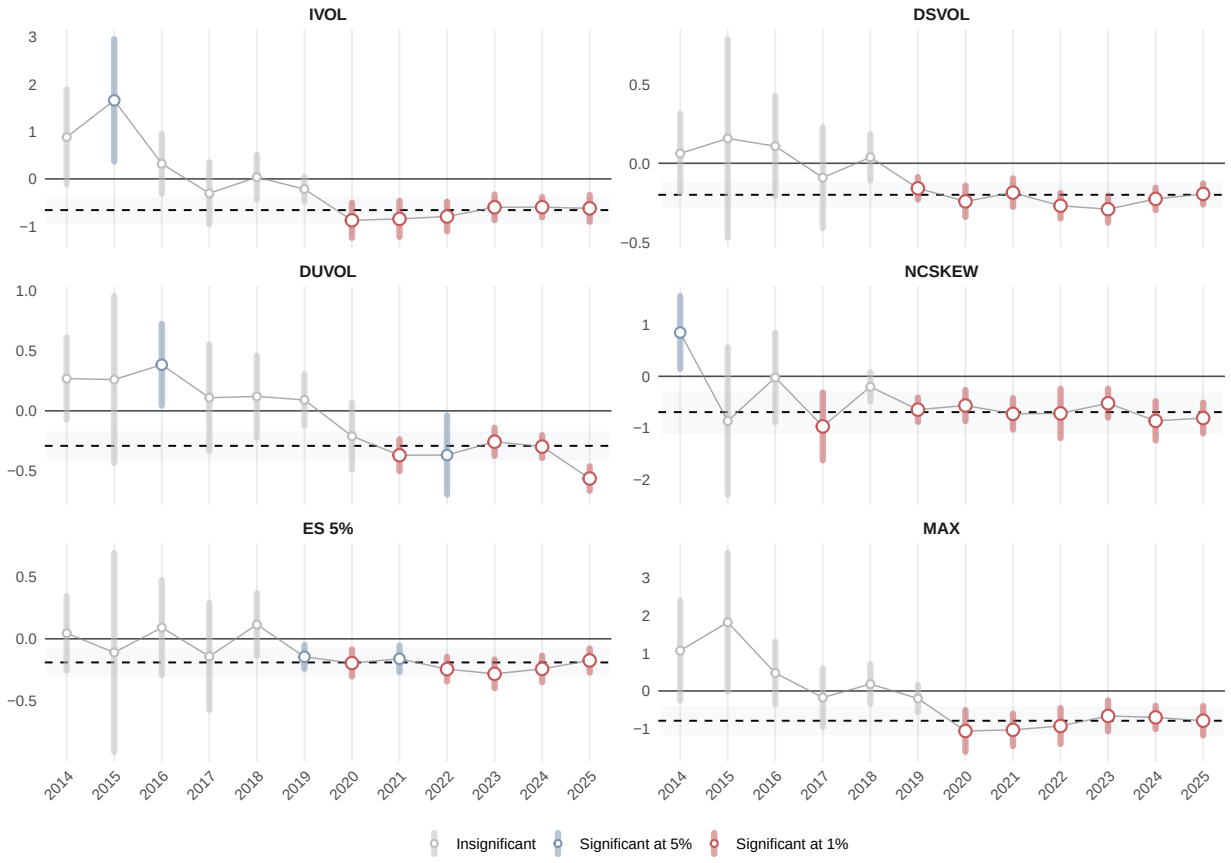


Figure 6: Time-Series Heterogeneity and Maturation of the Developer Network Shield
This figure illustrates the year-by-year evolution of the effects of developer network on six risk measures: Idiosyncratic Volatility (*IVOL*), Downside Volatility (*DSVOL*), Upside Volatility (*DUVOL*), Negative Conditional Skewness (*NCSKEW*), Expected Shortfall at the 5% level (*ES*5), and 30-Day Maximum Return (*MAX*). Each panel reports the estimated annual total effect from IV regressions that interact the endogenous developer-network variable with year dummies. The annual effect is computed as the sum of the main effect and the corresponding year interaction term, where the standard errors are calcuated via delta method. The 95% confidence intervals are based on Driscoll–Kraay standard errors with lags corresponding to the measures' evaluation windows. The dashed horizontal line indicates the benchmark IV estimates without year interactions. Points are colored by the statistical significance of θ_y (red: 1% level; blue: 5% level; grey: insignificant).

4.3. Network Topology Dynamics and Segmentation

The time-series heterogeneity of the risk-shielding effect suggests the potential structural change of developer networks. We therefore look into the dynamic evolution of the network topology. Figure 7 Panel A shows that the ecosystem has reached a remarkable scale. Since 2013, the total number of tokens continues to grow exponentially, and the average effort (commits) per developer is relatively stable, suggesting economic significance. Notably, since 2022, the network dynamics indicate a structural evolution towards modularity. The stabilization of the giant component (largest connected subgraph), despite the increasing

number of tokens, reflects a transition from a monolithic mesh to a more specialized, multi-cluster ecosystem.

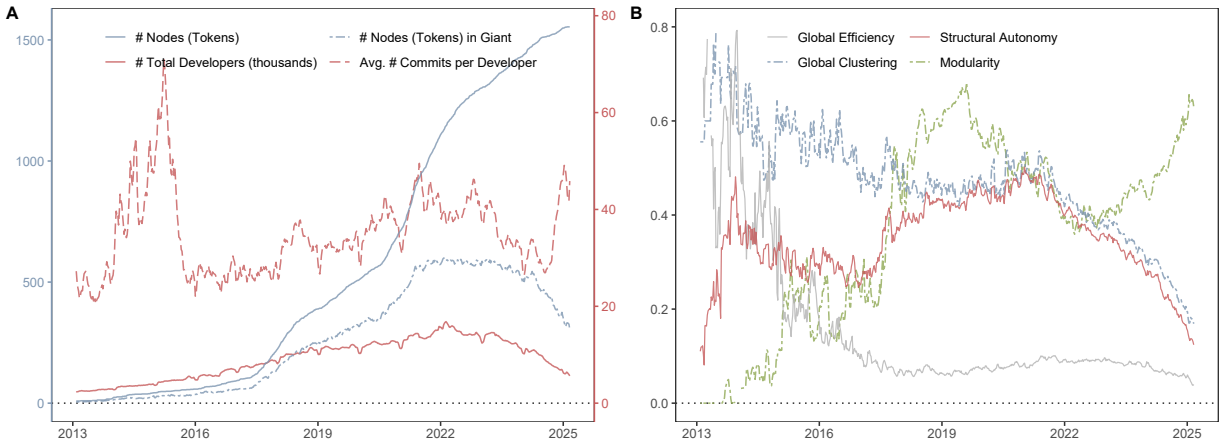


Figure 7: Network Topology Dynamics

This figure presents the dynamic evolution of the developer network's aggregate statistics and topological characteristics. The metrics are evaluated at a weekly frequency. Panel A illustrates the network size and developer activity. The left y-axis tracks the total number of token nodes (solid blue line) and the number of nodes in the *giant component* (dashed blue line), i.e., the largest interconnected subgraph within the network. The right y-axis tracks the total number of developers that have commit submissions within a month (solid red line), and the average number of commits per developer (dashed red line). Panel B shows the evolution of network topology metrics. Solid lines show the integration and brokerage capabilities: *Global Efficiency* (gray line) measures the ease of information transfer across the network, where higher values indicate faster global propagation; *Structural Autonomy* (red line) quantifies the abundance of structural holes, where higher values imply greater potential for brokerage and non-redundant connections between communities. Dashed lines represent cohesion and segregation: *Global Clustering* (blue line) measures the degree of local cohesiveness, where higher values indicate that nodes tend to form tightly knit cliques; *Modularity* (green line) assesses the strength of division into communities, where higher values indicate a network fragmented into distinct, self-contained subgroups with sparse external connections. Details of these topology measures are provided in Online Appendix OA-B1.

Figure 7 Panel B further explores the network's structural evolution through topological measures: *Global Clustering (GC)*, which proxies for local cohesiveness (Newman, 2003); *Modularity (Q)*, which captures the degree of segmentation into distinct communities (Newman, 2006); and *Structural Autonomy (SA)*, which reflects the brokerage potential of tokens to bridge disconnected groups (Burt, 1992). Tracking their joint dynamics suggests three distinct phases: (1) *Incubation* (high GC, low Q, low SA). In the early years, the network resembles a small-world "clique." Developers are tightly interconnected in a single cohesive mesh, leaving few structural holes to bridge (hence low autonomy). (2) *Expansion* (decreasing GC, rising Q, rising SA). As the ecosystem grows, it begins to splinter into distinct clusters (Rising Modularity). This structural divergence creates rich opportunities for cross-community brokerage, driving SA to its peak. (3) *Specialization* (low GC, high Q, low SA). Post-2022, the network enters a mature stage. While Modularity remains high,

SA drops sharply. This combination implies that technical boundaries have hardened: the ecosystem has transitioned from a bridged network into distinct, specialized silos where developers prioritize commit within specific stacks over cross-ecosystem brokerage.[33] This structural timeline offers a novel granular view of market evolution. Notably, the transition to the Specialization Phase precedes the financial market segmentation across token categories documented by Cong et al. (2025) by approximately one year, suggesting the developer network topology as a potential leading indicator.

Furthermore, the developer network is topologically distinct from social networks (for example, a token with more social ties does not necessarily have more peers in the developer network, and vice versa), indicating that code collaboration is not merely a projection of social interactions. Details are provided in Online Appendix OA-A.

4.4. Additional Tests and Robustness

We conduct a series of robustness checks to ensure that our findings are not driven by specific measurement choices or statistical assumptions. First, regarding *network measurement*, we show that the risk-shielding effect is robust across alternative edge definitions, including raw overlap counts, active-only developer subsets, and commit-weighted similarity measures. Second, regarding *statistical inference*, our results remain robust when replacing the baseline Driscoll-Kraay standard errors with standard two-way clustering. Third, we verify the *interplay between networks* using alternative specifications: we replace the baseline cosine similarity with raw overlapping developer counts, and substitute network centrality with Google Search Volume (for social attention) and GitHub Stars (for technical prominence). Finally, we examine *aggregate network dynamics* and find that the shielding mechanism operates independently of macro-level structural evolution, such as changes in global density or modularity. Collectively, these tests validate that the our main results are both statistically and economically robust. We further provide additional results regarding topology dynamics, crash risks and social network implications. Detailed results are reported in Online Appendix OA-C1–OA-C9.

5. Conclusion

As assets become digitized and even programmable, their risks are shaped not only by issuer fundamentals and investor trading, but also by the technical labor that implements and maintains them. This paper studies this implementation layer by constructing a dynamic

[33]The overall decreasing *Global Efficiency* further corroborates the segmentation, signaling that information no longer traverses the network through global shortcuts but remains localized.

network of digital assets linked through shared open-source developers. Using 47.4 million GitHub commits for 1,510 tokens, we document an opposite-signed form of risk transmission: higher idiosyncratic and tail risk among developer-connected peers predicts lower subsequent fragility of the focal asset. This risk-shielding pattern contrasts sharply with the spillover effect observed in a parallel social co-mention network, and is corroborated by an instrument based on lagged commit-timing similarity and topology-based falsification tests. By examining a battery of explanations, the evidence points to developer mobilization as an operative channel. Peer distress appears to function as a warning signal, inducing maintenance activity that reduces uncertainty about technical robustness. Consistent with this interpretation, the shielding effect is strongest for infrastructure-critical and ecosystem-linked assets, where code reliability is central to economic value. These findings suggest developers as a financial infrastructure: they are not merely individual employees, but also highly-mobile carriers of implementation-side information.

More broadly, our findings suggest that digitization creates a new layer of asset valuation and connectedness. This perspective may bring implications for asset pricing, disclosure, and market design in digital finance. As tokenization and software- and AI-intensive financial products diffuse into mainstream financial markets, measuring the organization and responsiveness of technical labor may become central to understanding how financial risks propagate — and when they are contained.

References

Acemoglu, D., Akcigit, U., & Kerr, W. R. (2016). Innovation network. *Proceedings of the National Academy of Sciences*, *113*(41), 11483–11488.

Ali, U., & Hirshleifer, D. (2020). Shared analyst coverage: Unifying momentum spillover effects. *Journal of Financial Economics*, *136*(3), 649–675.

Amihud, Y. (2002). Illiquidity and stock returns: Cross-section and time-series effects. *Journal of Financial Markets*, *5*(1), 31–56.

Ang, A., Chen, J., & Xing, Y. (2006). Downside risk. *Review of Financial Studies*, *19*(4), 1191–1239.

Ang, A., Hodrick, R. J., Xing, Y., & Zhang, X. (2006). The cross-section of volatility and expected returns. *Journal of Finance*, *61*(1), 259–299.

Anton, M., & Polk, C. (2014). Connected stocks. *The Journal of Finance*, *69*(3), 1099–1127.

Baghai, R. P., Silva, R. C., Thell, V., & Vig, V. (2021). Talent in distressed firms: Investigating the labor costs of financial distress. *The Journal of Finance*, *76*(6), 2907–2961.

Bali, T. G., Cakici, N., & Whitelaw, R. F. (2011). Maxing out: Stocks as lotteries and the cross-section of expected returns. *Journal of Financial Economics*, *99*(2), 427–446.

Barrot, J.-N., & Sauvagnat, J. (2016). Input specificity and the propagation of idiosyncratic shocks in production networks. *The Quarterly Journal of Economics*, *131*(3), 1543–1592.

Biais, B., Bisiere, C., Bouvard, M., & Casamatta, C. (2019). The blockchain folk theorem. *Review of Financial Studies*, *32*(5), 1662–1715.

Bloom, N., Schankerman, M., & Van Reenen, J. (2013). Identifying technology spillovers and product market rivalry. *Econometrica*, *81*(4), 1347–1393.

Borri, N. (2019). Conditional tail-risk in cryptocurrency markets. *Journal of Empirical Finance*, *50*, 1–19.

Borri, N., Liu, Y., & Tsyvinski, A. (2022). The economics of non-fungible tokens. *Available at SSRN*, *4052045*.

Brzustowski, T., Georgiadis-Harris, A., & Szentes, B. (2023). Smart contracts and the coase conjecture. *American Economic Review*, *113*(5), 1334–1359.

Burt, R. S. (1992). *Structural holes: The social structure of competition*. Harvard University Press.

Capponi, A., Jia, R., & Yu, S. (2025). Price discovery on decentralized exchanges. *Available at SSRN 4236993*.

Cattaneo, M. D., Crump, R. K., Farrell, M. H., & Feng, Y. (2024). On binscatter. *American Economic Review*, *114*(5), 1488–1514.

Chen, J., & Roth, J. (2024). Logs with zeros? some problems and solutions. *The Quarterly Journal of Economics*, *139*(2), 891–936.

Chen, J., Hong, H., & Stein, J. C. (2001). Forecasting crashes: Trading volume, past returns, and conditional skewness in stock prices. *Journal of Financial Economics*, *61*(3), 345–381.

Chod, J., & Lyandres, E. (2023). Product market competition with crypto tokens and smart contracts. *Journal of Financial Economics*, *149*(1), 73–91.

Cohen, L., & Frazzini, A. (2008). Economic links and predictable returns. *The Journal of Finance*, *63*(4), 1977–2011.

Cong, L. W., & He, Z. (2019). Blockchain disruption and smart contracts. *Review of Financial Studies*, *32*(5), 1754–1797.

Cong, L. W., Hui, X., Tucker, C., & Zhou, L. (2023). Scaling smart contracts via layer-2 technologies: Some empirical evidence. *Management Science*, *69*(12), 7306–7316.

Cong, L. W., Karolyi, G. A., Tang, K., & Zhao, W. (2025). Value premium, network adoption, and factor pricing of crypto assets. *Management Science (forthcoming)*.

Cong, L. W., Li, Y., & Wang, N. (2021a). Token-based platform finance. *Journal of Financial Economics*.

Cong, L. W., Li, Y., & Wang, N. (2021b). Tokenomics: Dynamic adoption and valuation. *Review of Financial Studies*, *34*(3), 1105–1155.

Da, Z., Engelberg, J., & Gao, P. (2011). In search of attention. *The journal of finance*, *66*(5), 1461–1499.

Donangelo, A. (2014). Labor mobility: Implications for asset pricing. *The Journal of Finance*, *69*(3), 1321–1346.

Driscoll, J. C., & Kraay, A. C. (1998). Consistent covariance matrix estimation with spatially dependent panel data. *Review of Economics and Statistics*, *80*(4), 549–560.

Fieberg, C., Liedtke, G., Poddig, T., Walker, T., & Zaremba, A. (2025). A trend factor for the cross section of cryptocurrency returns. *Journal of Financial and Quantitative Analysis*, *60*(7), 3116–3153. https://doi.org/10.1017/S0022109024000747

Gefen, O., Rabetti, D., Sun, Y., & Zhang, C. (2024). Code-washing: Evidence from open-source blockchain startups [Available at SSRN]. *Working Paper*.

Goldsmith-Pinkham, P., & Imbens, G. W. (2019). Social networks with endogenous sortings. *Journal of Business & Economic Statistics*, *37*(3), 447–458.

Goldsmith-Pinkham, P., Sorkin, I., & Swift, H. (2020). Bartik instruments: What, when, why, and how. *American Economic Review*, *110*(8), 2586–2624.

Goldstein, I., Gupta, D., & Sverchkov, R. (2024). Utility tokens as a commitment to competition. *The Journal of Finance*, *79*(6), 4197–4246.

Griffin, J. M., & Shams, A. (2020). Is bitcoin really untethered? *The Journal of Finance*, *75*(4), 1913–1964.

Hahn, J., Moon, J. Y., & Zhang, C. (2008). Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties. *Information Systems Research*, *19*(3), 369–391.

Harvey, C. R., Saleh, F., & Sverchkov, R. (2025). An economic model of the l1-l2 interaction. *Available at SSRN 5163823*.

He, Z., Rogoff, K. S., & You, Y. (2024). *Market power and redeemable loyalty token design* (tech. rep.). National Bureau of Economic Research.

Herskovic, B. (2018). Networks in production: Asset pricing implications. *The Journal of Finance*, *73*(4), 1785–1818.

Hirshleifer, D., Peng, L., & Wang, Q. (2025). News diffusion in social networks and stock market reactions. *The Review of Financial Studies*, *38*(3), 883–937.

Howell, S. T., Niessner, M., & Yermack, D. (2020). Initial coin offerings: Financing growth with cryptocurrency token sales. *The Review of Financial Studies*, *33*(9), 3925–3974.

John, K., Kogan, L., & Saleh, F. (2023). Smart contracts and decentralized finance. *Annual Review of Financial Economics*, *15*(1), 523–542.

Kogan, S., Makarov, I., Niessner, M., & Schoar, A. (2024). Are cryptos different? evidence from retail trading. *Journal of Financial Economics*, *159*, 103897.

Kuchler, T., Li, Y., Peng, L., Stroebel, J., & Zhou, D. (2022). Social proximity to capital: Implications for investors and firms. *The Review of Financial Studies*, *35*(6), 2743–2789.

Lee, C. M., Sun, S. T., Wang, R., & Zhang, R. (2019). Technological links and predictable returns. *Journal of Financial Economics*, *132*(3), 76–96.

Lee, S. S., & Wang, M. (2025). Variance decomposition and cryptocurrency return prediction. *Journal of Financial and Quantitative Analysis*, *60*(4), 1859–1890.

Lehar, A., & Parlour, C. (2025). Decentralized exchange: The uniswap automated market maker. *The Journal of Finance*, *80*(1), 321–374.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. *Journal of Industrial Economics*, *50*(2), 197–234. https://doi.org/10.1111/1467-6451.00174

Li, T., Shin, D., & Wang, B. (2018). Cryptocurrency pump-and-dump schemes. *Journal of Financial and Quantitative Analysis*, 1–59.

Liu, Y., & Tsyvinski, A. (2021). Risks and returns of cryptocurrency. *Review of Financial Studies*, *34*(6), 2689–2727.

Liu, Y., Tsyvinski, A., & Wu, X. (2022). Common risk factors in cryptocurrency. *The Journal of Finance*, *77*(2), 1133–1177.

Liu, Y., & Wu, X. (2023). Labor links, comovement, and predictable returns. *Journal of Financial and Quantitative Analysis*, 1–57.

Lyandres, E., Palazzo, B., & Rabetti, D. (2022). Initial coin offering (ico) success and post-ico performance. *Management Science*, *68*(12), 8658–8679.

Maitre, A. T., Pugachyov, N., & Weigert, F. (2025). Social media-based attention and the cross-section of cryptocurrency returns. *Journal of Banking & Finance*, 107518.

Makarov, I., & Schoar, A. (2022). Cryptocurrencies and decentralized finance (defi). *Brookings Papers on Economic Activity*, *2022*(1), 141–215.

Manski, C. F. (1993). Identification of endogenous social effects: The reflection problem. *Review of Economic Studies*, *60*(3), 531–542.

Nagle, F. (2018). Learning by contributing: Gaining competitive advantage through contribution to crowdsourced public goods. *Organization Science*, *29*(4), 569–587. https://doi.org/10.1287/orsc.2018.1202

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, *45*(2), 167–256.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, *103*(23), 8577–8582.

Peng, L., Titman, S., Yönaç, M., & Zhou, D. (2023). It's a small world: Social ties, comovements, and predictable returns. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4240056

Ransbotham, S., & Mitra, S. (2009). Choice and chance: A conceptual model of paths to information security compromise. *Information Systems Research*, *20*(1), 121–139.

Roll, R. (1988). R-squared. *Journal of Finance*, *43*(3), 541–566.

Saleh, F. (2021). Blockchain without waste: Proof-of-stake. *Review of Financial Studies*, *34*(3), 1156–1190.

Schwenkler, G., & Zheng, H. (2020). *The network of firms implied by the news*. ESRB Working Paper Series.

Schwenkler, G., & Zheng, H. (2025). News-driven peer co-movement in crypto markets. *Journal of Corporate Finance*, *93*, 102772.

Sharma, V., Agarwal, A., & Barua, A. (2025). Demand-side effects of open innovation: The case of cryptocurrency forking. *Management Science*, *71*(8), 7136–7160.

Sockin, M., & Xiong, W. (2023a). Decentralization through tokenization. *The Journal of Finance*, *78*(1), 247–299.

Sockin, M., & Xiong, W. (2023b). A model of cryptocurrencies. *Management Science*, *69*(11), 6684–6707.

Wang, H., & Zhang, W. (2024). Network through social media connections [Available at SSRN: https://ssrn.com/abstract=4536567]. *Working Paper*.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442.

Zheng, Y., & Wang, Q. R. (2025). Blood diamonds? Responses of open-source software developers to the Facebook–Cambridge Analytica scandal. *Strategic Management Journal*, *46*(7). https://doi.org/10.1002/smj.3712

Appendix

A. Additional Figures and Tables

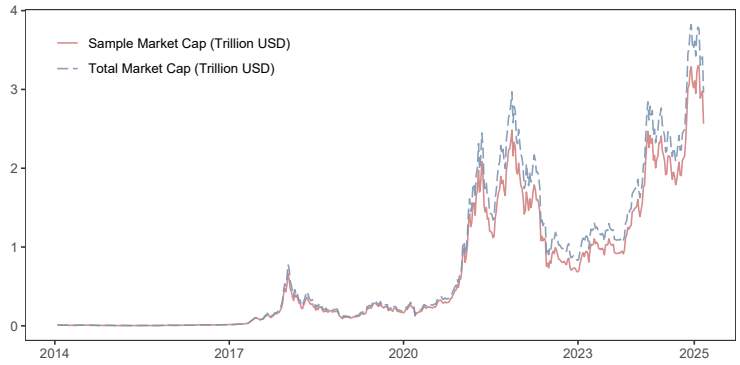


Figure A1: Aggregate Market Cap of the Sample Set and the Whole Market

This figure compares the aggregate market capitalization (in Trillions of US Dollars) of the tokens included in our regression sample (the red solid line) against the total global cryptocurrency market (the blue dashed line) from January 2014 to March 2025.

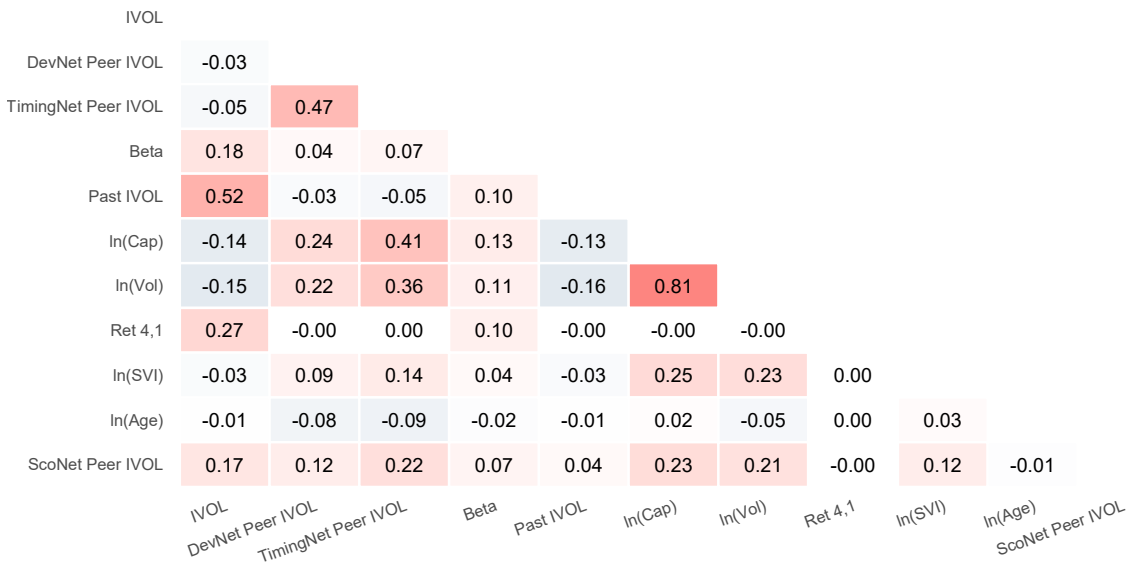


Figure A2: Correlation Matrix

This figure presents the pairwise correlation coefficients for the main variables included in the baseline regression (Table 3 Column (4)).

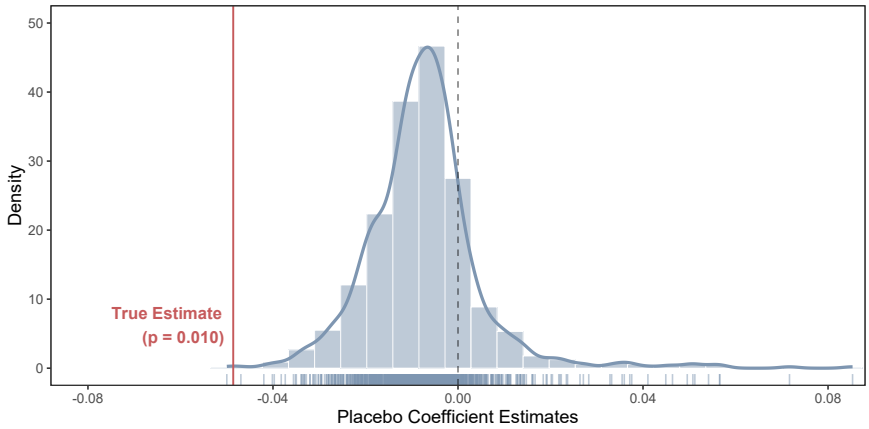


Figure A3: Permutation Test: Randomizing Token-Node Mappings

This figure reports the results of a permutation test designed to verify that the estimated stabilization effect arises from actual developer connections rather than random chance or unobserved heterogeneity. In this test, we randomly shuffle the mapping between tokens and network nodes and estimate the baseline OLS specification with full controls and token and time fixed effects. This procedure preserves the topological structure of the developer network and the distributional properties of the variables but destroys the specific, true links between a token and its peer group. The histogram (blue bars) and kernel density estimate (blue line) display the distribution of the regression coefficients for *DevNet Peer IVOL* obtained from these 1,000 counterfactual simulations. The dashed vertical line at zero indicates the mean of the counterfactual estimates. The solid red vertical line represents the *True Estimate* from the baseline specification using the actual network data. The empirical p-value ($p = 0.010$) indicates that the observed developer shield effect is statistically significant and distinguishable from noise at the 1% level. In addition, the mean value of the estimates is slightly (negatively) away from zero, which aligns the peak test: the random mapping more or less retain (overlap) a number of original edges, preserving some of the original estimated effects despite significant noise is introduced by new (fake) edges.

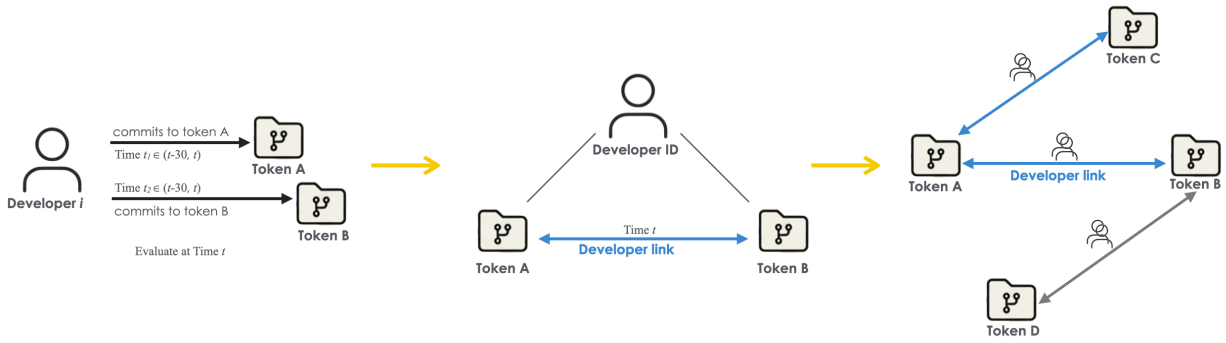
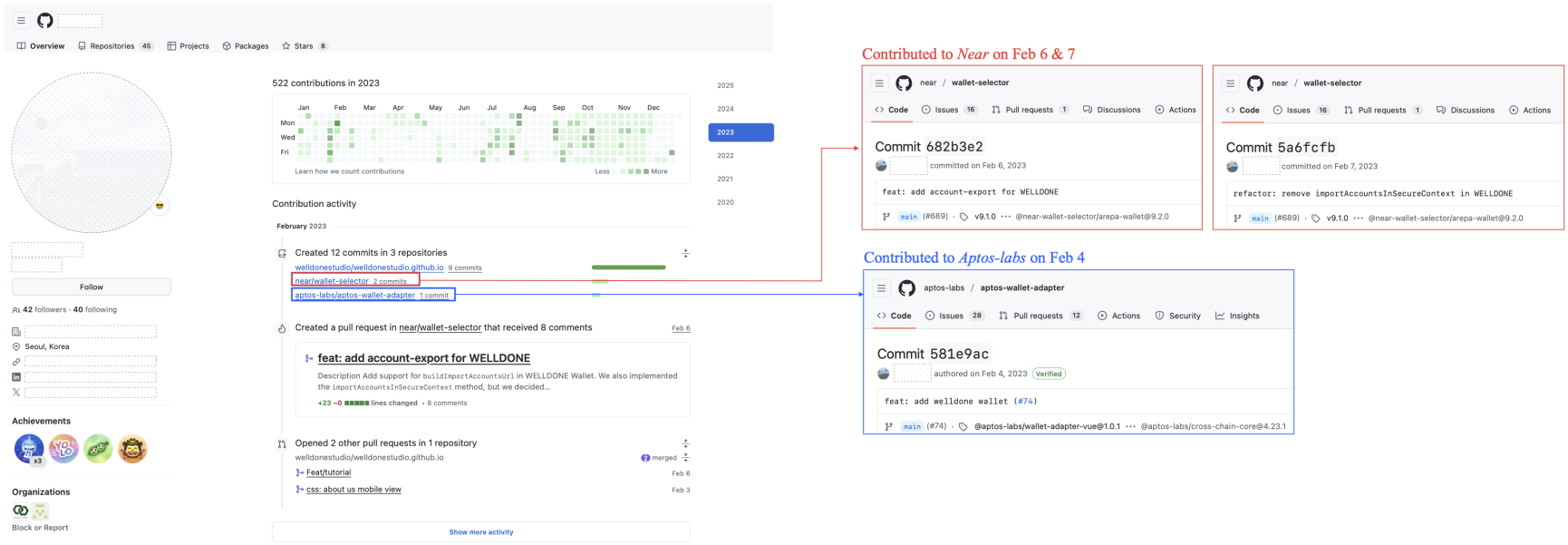


Figure A4: From Commit Activities to Developer Network

This figure illustrates how we construct the developer network from GitHub commit records. *Step 1 (Data Extraction)* identifies any developer i who submits code (commits) to a repository associated with *Token A* at time t_1 and also commits to a repository associated with *Token B* at time t_2. And both t_1 and t_2 are within $(t-30, t)$. *Step 2 (Link Formation)* projects the cross-project activity to a *developer link* between Token A and Token B at time t, using the developer ID as the linking identifier. *Step 3 (Network Aggregation)* demonstrates how multiple developers generate multiple developer links across tokens (e.g., Token A linked to Token B and Token C), which collectively form the token-level developer network used in our empirical analyses. This procedure converts the raw bipartite developer-token graph into the weighted one-mode token projection used in the empirical analysis.



Panel A. Github Contribution Activity of an Example Developer

Panel B. Example Commits

Figure A5: Micro-foundation of Network Edges: An Illustrative Co-Developer

This figure provides a concrete example of a shared developer (anonymized) connecting two distinct tokens within a short time window, validating the network construction methodology. *Panel A* displays the developer's GitHub contribution heatmap and activity feed. It captures concurrent activity in repositories associated with the token project *NEAR* (`near/wallet-selector`) and *Aptos Labs* (e.g., `aptos-labs/aptos-wallet-adapter`) during the same month. *Panel B* verifies the substance of this cross-project activity at the commit level. The commit logs confirm that the developer contributed code to *Aptos Labs* on Feb 4 and to *NEAR* on Feb 6 and 7. Notably, the commit messages (e.g., "add welldone wallet" and "add account-export") indicate that the developer is applying specialized technical expertise (wallet integration) across different token projects, supporting the assumption that these links represent meaningful channels for knowledge and risk transmission.

Table A1: Variable Definitions

Variable	Definition	Interpretation & Reference
Panel A: Main Risk Measures		
IVOL	The annualized standard deviation of daily idiosyncratic returns. It is calculated as the standard deviation of the residuals from a CAPM regression of daily excess returns on the market factor excess returns over the past 30 days, multiplied by $\sqrt{365}$.	Measures idiosyncratic volatility and arbitrage risk. (e.g., Ang, Hodrick, et al., 2006; Liu et al., 2022)
DSVOL	The annualized downside volatility, calculated as the square root of the annualized second lower partial moment of daily excess returns over the past 30 days: $\sqrt{365 \times \frac{1}{T} \sum_{r_{i,t}<0} r_{i,t}^2}$, where $r_{i,t}$ is the daily excess return.	Measures downside tail risk without assuming normality.
DUVOL	The log ratio of downside to upside volatility over the past 30 days. Calculated as $\ln\left(\frac{(n_u-1)\sum_{down}(r_{i,t}-\bar{r}_i)^2}{(n_d-1)\sum_{up}(r_{i,t}-\bar{r}_i)^2}\right)$, where days are split into "down" (return below mean) and "up" (return above mean).	A measure of crash risk; higher values indicate a more left-skewed distribution. (e.g., Chen et al., 2001)
NCSKEW	The negative coefficient of skewness of CAPM residuals over the past 90 days. Calculated as $-\frac{n(n-1)^{3/2}\sum \varepsilon_{i,t}^3}{(n-1)(n-2)(\sum \varepsilon_{i,t}^2)^{3/2}}$.	Measures the asymmetry of the return distribution; higher values imply higher crash risk. (e.g., Chen et al., 2001)
*ES*5	Expected Shortfall at the 5% level, calculated as the negative average of daily returns that fall into the lowest 5% quantile over the past 90 days.	Captures the expected loss conditional on an extreme tail event.
DSBETA	Downside Beta. The covariance between the asset's excess return and the market excess return conditional on the market return being below its mean, divided by the downside variance of the market, estimated over the past 30 days.	Measures systematic risk exposure during market downturns. (e.g., Ang, Chen, & Xing, 2006)
USBETA	Upside Beta. The covariance between the asset's excess return and the market excess return conditional on the market return being above its mean, divided by the upside variance of the market, estimated over the past 30 days.	Measures systematic risk exposure during market upturns. (e.g., Ang, Chen, & Xing, 2006)
Panel B: Other Variables		
Beta	The CAPM beta estimated from a regression of daily excess returns on market excess returns over the window $[t-89, t-7]$.	Measures standard systematic risk exposure.
$\log(ILLIQ)$	The natural logarithm of the average daily Amihud illiquidity ratio over the past 30 days, where daily illiquidity is defined as $\lvert Ret\rvert/(30\times Volume)$. Scaled by 10^6 before taking logs.	Proxies for price impact and illiquidity. (e.g., Amihud, 2002)
$\log(Turnover)$	The natural logarithm of the average daily turnover ratio (Volume / Market Cap) over the past 30 days.	Proxies for trading activity and investor attention.
R^2	The R-squared statistic from the CAPM regression estimated over the past 30 days.	Inverse proxy for price informativeness (synchronicity). (e.g., Roll, 1988)
CAR	Cumulative Daily Abnormal Return calculated over future windows. Abnormal returns are computed as the difference between realized excess returns and expected returns based on the ex-ante CAPM beta.	Measures future performance or return reversal.

Online Appendix

for

"Co(de)nnection: Developer Networks and Digital Asset Fragility"

OA-A Supplementary Stylized Facts 51

OA-B Detailed Remarks on Measures 55

OA-B1 Network Topology Measures 55

OA-B2 Centrality Measures 57

OA-B3 Public Attention Measures 57

OA-C Additional Results and Robustness Tests 59

OA-C1 Additional Results on Network Spillovers of Crash Risk 59

OA-C2 Additional Results on Network Topology 60

OA-C3 Alternative Network Weight Measures 64

OA-C4 Alternative Statistical Inferences 66

OA-C5 Additional Discussion on Mobilization Channel 67

OA-C6 Alternative Proxies for Network Interactions 68

OA-C7 Effect Robustness across Aggregate Network States 68

OA-C8 Max Returns and Lottery-like Features 69

OA-C9 Social Network Centrality and Contagion Amplification 70

June 1, 2026

OA-A. Supplementary Stylized Facts

In this section, we provide a comprehensive visualization of the developer network to complement the stylized facts reported in the main text.

Developer network panorama: network with isolated points and annual networks. Figure OA-1 presents the *full-sample* panorama of the developer network, explicitly retaining isolated tokens that share no developers with other projects. This visualization reveals that the developer network is characterized by a distinct "core-periphery" structure: a dense interconnected core of established ecosystems (e.g., Ethereum and Solana) surrounded by a vast periphery of isolated projects.

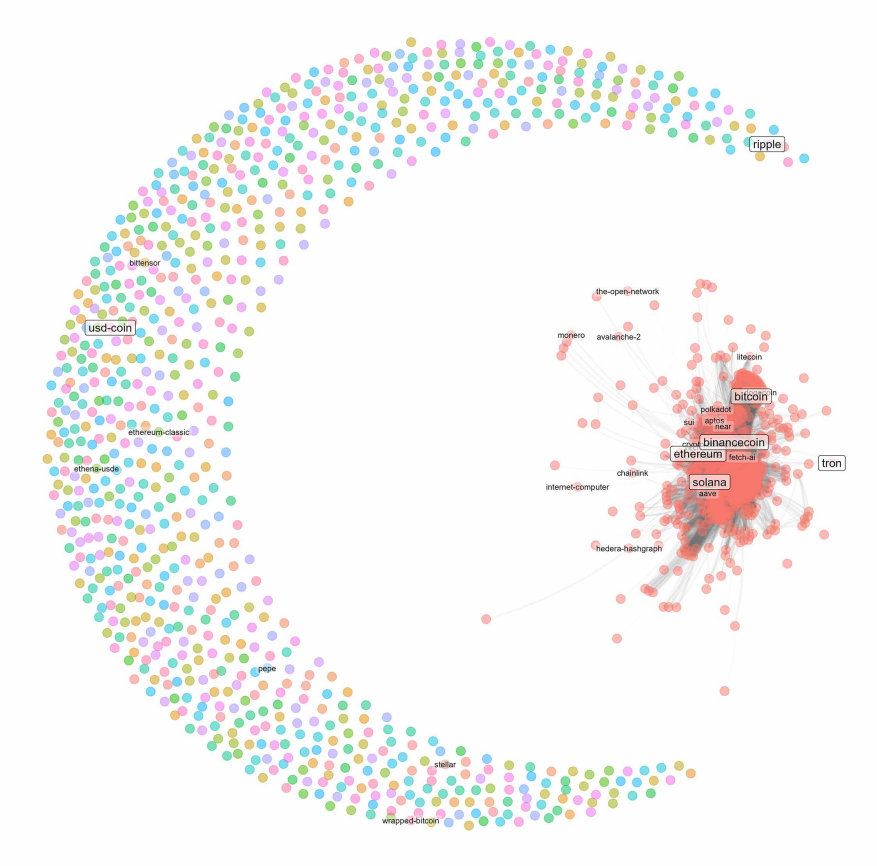


Figure OA-1: Overall Developer Network of Token Projects

This figure visualizes the full-sample developer network of token projects over January 2013 to March 2025. Each in-sample token project is a node. An edge is drawn between two projects if at least 100 developers contributed to both projects, either through actively submitted commits or forked commits. We set this lower limit on the number of overlapping developers for readability, otherwise there would be overly dense. The network layout and node clusters (colored) are determined by the Fruchterman-Reingold algorithm. Therefore, compared to Figure 1, this figure focus on the overall network structure rather than the core subgraphs of the active commit network.

Furthermore, Figure OA-2 decomposes this overall structure into annual snapshots from

2013 to 2024. This temporal decomposition documents the evolution of the crypto industry from a monocentric network dominated by Bitcoin in the early years to a polycentric system with distinct, modular ecosystems in the mature period.

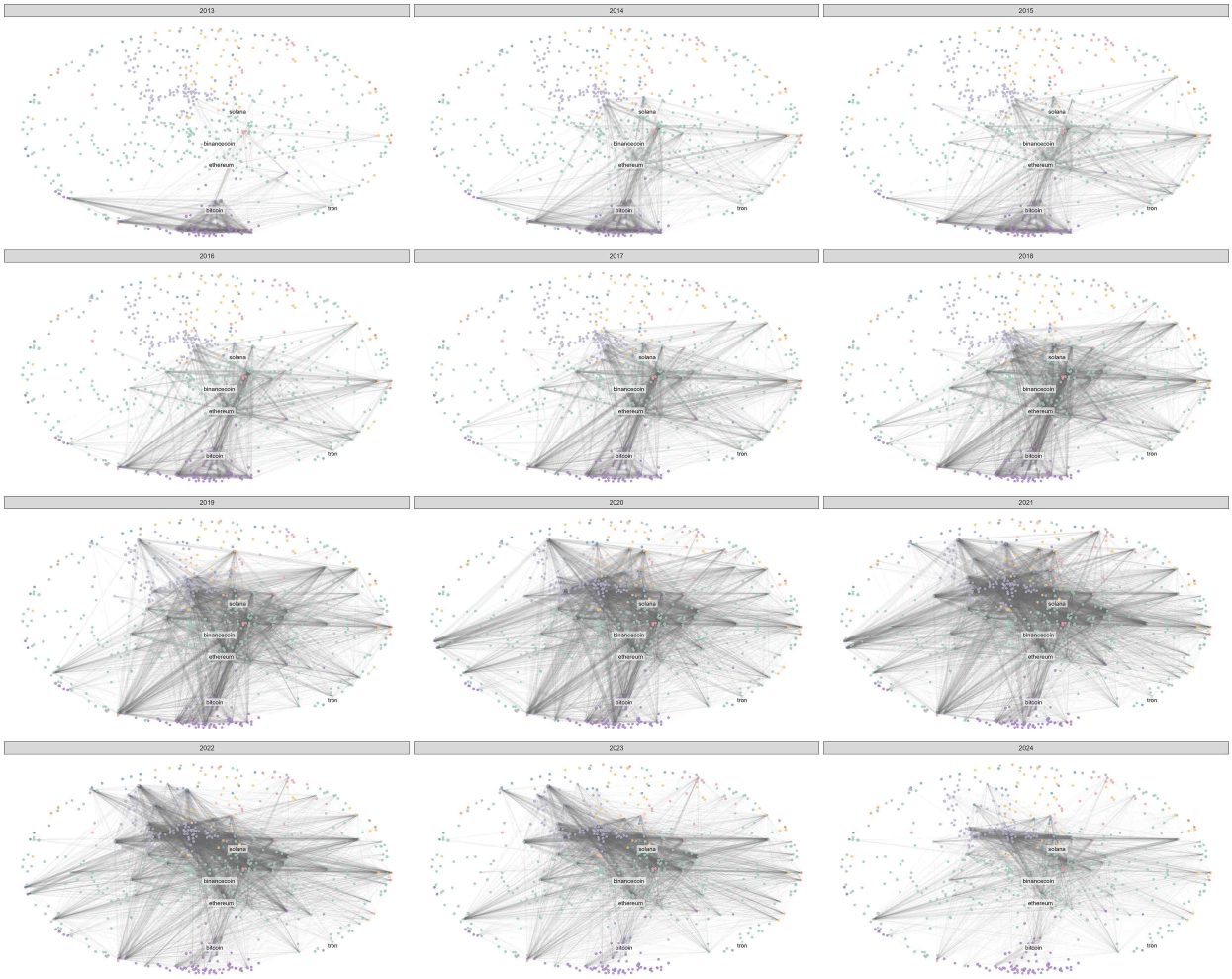


Figure OA-2: Yearly Coding Developer Network: 2013 – 2024

This figure visualizes the network of token projects connected by overlapping coding developers for each year. For comparability across years and with Figure 1, we maintain the nodes, clusters, and positions from Figure 1. In any given year, two projects are connected if at least one developer contributes to both projects — either by actively submitting commits or through passively forked commits — within that year.

Trending on developer networks and social media. Figure OA-3 examines the cross-sectional correlation between Reddit popularirty and developer network centrality. The lack of a systematic relationship—in sharp contrast to the strong correlation between Reddit rank and market capitalization—confirms that technical centrality is effectively orthogonal to social popularity and token size.

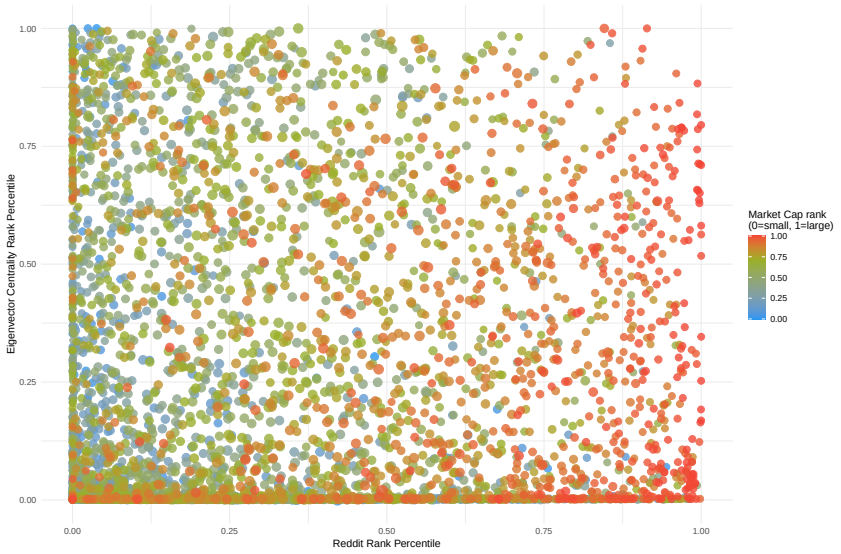


Figure OA-3: Reddit Rank and Developer-network Centrality

This figure plots each token's **Reddit rank percentile** (x-axis) against its **developer-network eigenvector centrality rank percentile** (y-axis), with point colors indicating **market capitalization rank**. Two patterns stand out. First, Reddit prominence is strongly aligned with token size: higher Reddit rank percentiles are predominantly associated with larger market-cap tokens. Second, developer-network centrality is weakly related to both Reddit prominence and market cap. Overall, the figure supports the interpretation that Reddit-based networks primarily capture *attention*, whereas developer networks capture a distinct *technical linkage* dimension.

Figure OA-4 reinforces this distinction through the time-series dynamics of Bitcoin and Ethereum. Notably, while Bitcoin retains top-tier social attention (indicated by high Reddit rank, as colored in red), its centrality within the developer network has first increased and then declined. In contrast, Ethereum displays significant variation in technical centrality, rising and then fluctuating over cycles, even as its social attention has been increasing. These divergent trajectories reinforce that the developer network is orthogonal to social popularity.

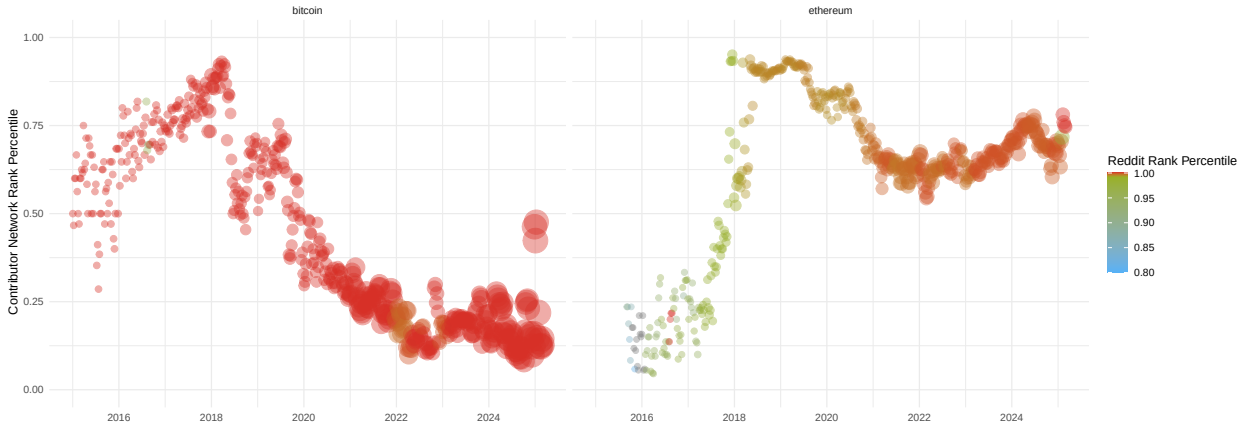


Figure OA-4: Reddit Rank and Developer-network Centrality: Bitcoin and Ethereum
This figure tracks the time-varying **developer-network prominence** of **Bitcoin** and **Ethereum**. The y-axis reports each token's **developer-network rank percentile**, while point colors indicate the corresponding **Reddit rank percentile**. The figure shows that developer-network prominence can rise or fall independently of Reddit attention, underscoring that developer networks capture a dimension that is distinct from *attention/market prominence*.

OA-B. Detailed Remarks on Measures

OA-B1. Network Topology Measures

In this appendix, we provide formal definitions, calculation methods, and interpretations for the network metrics utilized in Figure 7. Let $\mathcal{G} = (V, E)$ denote an undirected, weighted graph, where V is the set of nodes (tokens) with $N = |V|$, and E is the set of edges representing developer overlap. Let A be the adjacency matrix where $A_{ij} > 0$ if nodes i and j are connected, and w_{ij} represents the weight of the edge.

Giant Component Size (N_{giant}). The giant component is the largest connected subgraph within the network, containing the maximum number of connected nodes. Let $C_1, C_2, \ldots, C_k$ be the set of connected components in $\mathcal{G}$. The size of the giant component is defined as:

$$N_{giant} = \max_k |V(C_k)| \tag{OA-1}$$

Interpretation: N_{giant} represents the functional backbone of the network. A high ratio of N_{giant}/N indicates a unified market structure where information and resources can theoretically traverse between the majority of projects, whereas a low ratio suggests a fragmented ecosystem of isolated islands.

Average Commits per Developer. A proxy for the aggregate work intensity or activity level of the developer population. Let D be the set of unique developers and c_k be the number of commits made by developer k. The metric is calculated as $\frac{1}{|D|}\sum_{k\in D} c_k$.
Interpretation: This non-topological metric controls for the volume of activity, ensuring that network structural changes are not merely artifacts of changing data volume.

Global Efficiency (E_{glob}). Global efficiency measures the efficiency of information exchange across the entire network, assuming that information flows along the shortest paths. It is defined as the average inverse shortest path length between all pairs of nodes (Latora & Marchiori, 2001):

$$E_{glob} = \frac{1}{N(N-1)} \sum_{i\neq j\in V} \frac{1}{d_{ij}} \tag{OA-2}$$

where d_{ij} is the shortest path length (geodesic distance) between node i and node j. If nodes are disconnected, $d_{ij} = \infty$ and $1/d_{ij} = 0$.
Interpretation: Higher values indicate a more integrated network where information propagates rapidly. Unlike average path length, global efficiency is robust to disconnected networks.

A decline in E_{glob} typically signals structural disintegration or the loss of critical hubs.

Structural Autonomy (SA). Derived from Burt's seminal theory of structural holes (Burt, 1992), structural autonomy measures the extent to which a network is rich in non-redundant connections. It is the complement of network constraint. For a given node i, Burt's constraint C_i is defined as:

$$C_i = \sum_{j \in \Gamma(i)} \left(p_{ij} + \sum_{q \neq i,j} p_{iq} p_{qj} \right)^2 \tag{OA-3}$$

where p_{ij} is the proportional investment of node i's relations in node j. The network-level Structural Autonomy is calculated as:

$$SA = 1 - \frac{1}{N} \sum_{i=1}^{N} C_i \tag{OA-4}$$

Interpretation: Higher SA implies that the average node possesses high brokerage potential by connecting diverse, disconnected groups.

Global Clustering Coefficient (C_{glob}). Also known as transitivity, this metric quantifies the tendency of nodes to cluster together. It measures the probability that two neighbors of a node are connected to each other.
It is the ratio of closed triplets (triangles) to the total number of connected triplets (open and closed) in the network (Newman, 2003):

$$C_{glob} = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets}} = \frac{3 \sum_i \delta(i)}{\sum_i k_i (k_i - 1)} \tag{OA-5}$$

where $\delta(i)$ is the number of triangles node i participates in, and k_i is the degree of node i.

Interpretation: High clustering indicates a network dominated by "cliques" and local cohesion. In the context of developer networks, it suggests "social reinforcement" where developers working on project A and B imply a high likelihood of direct overlap between A and B.

Modularity (Q). Modularity measures the strength of division of a network into modules (communities). It compares the density of edges within modules to what would be expected in a random network (Newman & Girvan, 2004; Newman, 2006). Using the Louvain method

for community detection (Blondel et al., 2008), modularity is defined as:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \tag{OA-6}$$

where m is the total number of edges, A_{ij} is the adjacency matrix weight, k_i is the degree of node i, c_i is the community assignment of node i, and $\delta(c_i, c_j)$ is the Kronecker delta function (1 if $c_i = c_j$, 0 otherwise).

Interpretation: High modularity ($Q \to 1$) indicates a fragmented network structure with dense internal connections within communities but sparse connections between them. An upward trend in Q signifies the solidification of community boundaries and tribalization.

OA-B2. Centrality Measures

We construct a panel of weekly eigenvector centrality scores for each token to capture its position in the developer network. Using the weekly developer-similarity matrix G_t, we compute eigenvector centrality to measure the relative influence of each token within the developer network on a weekly basis. Specifically, for each week t, we construct the symmetric similarity matrix G_t, where each element $G_{i,j,t}$ represents the developer similarity between tokens i and j as defined in Section 1.1.2. The eigenvector centrality of token i in week t, denoted by $EC_{i,t}$, is obtained by solving the following eigenvalue equation:

$$G_t x_t = \lambda_t x_t \tag{OA-7}$$

where x_t is the eigenvector corresponding to the largest eigenvalue λ_t. In this formulation, a token's centrality depends not only on the number (and strength) of its direct connections but also on the centrality of the tokens to which it is connected. A higher eigenvector centrality score thus indicates a more influential position of the token within the developer network in a given week. Therefore, we obtain a panel dataset of weekly centrality scores $\{EC_{i,t}\}$ for all tokens, capturing their evolving positions within the developer network over time.

OA-B3. Public Attention Measures

This subsection documents the construction of weekly social attention measures based on Google Trends following Da et al. (2011).

We collect weekly Google Trends search interest for each token using the token name as the baseline keyword. To reduce ambiguity for tokens with common names, we manually refine the search term by appending "coin" when necessary (e.g., "Diamond coin"). A key limitation of Google Trends is that the reported index is normalized within each query

(scaled to the maximum search interest of that keyword over the selected time window), which prevents direct cross-sectional comparisons across different tokens. To address this issue, we implement a pairwise benchmarking strategy: we query two token keywords jointly to obtain their relative search interest within the same request, and use the implied ratio to place token-specific indices on a common scale. By iterating this procedure over token pairs, we recover a standardized weekly attention index that is comparable across all tokens in the sample.

OA-C. Additional Results and Robustness Tests

OA-C1. Additional Results on Network Spillovers of Crash Risk

Compared to other measures in our main analysis, the risk measures *DUVOL* and *NCSKEW* are relatively complex to draw implications from regression coefficients. Figure OA-5 further visualizes the structural relationship between the focal asset's tail risk measures (*DUVOL* in Panel A and *NCSKEW* in Panel B) and those of its peers. The uniformly negative slopes in both panels reveal a striking pattern: as peer crash risk intensifies (higher *DUVOL* or *NCSKEW*), the focal asset exhibits a significantly more favorable return distribution (lower crash risk).

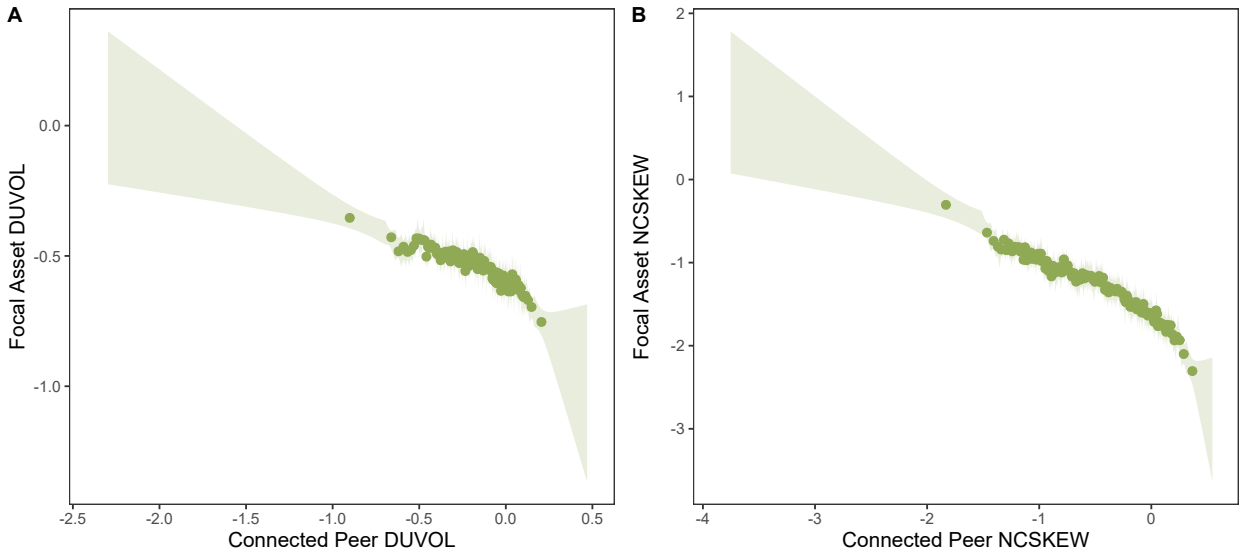


Figure OA-5: Binned Scatter Plots of Focal vs. Peer Tail Risk

This figure displays binned scatter plots of the focal asset's tail risk measures against those of its connected peers. Panel A plots the focal asset's Down-to-Up Volatility (*DUVOL*) against the Connected Peer *DUVOL*. Panel B plots the focal asset's Negative Coefficient of Skewness (*NCSKEW*) against the Connected Peer *NCSKEW*. Both panels utilize the covariate-adjustment method following Cattaneo et al. (2024), controlling for market capitalization, momentum, and trading volume, as well as token and time fixed effects. The grey areas represent 95% confidence intervals.

These plots provide extended insight beyond the baseline regression estimates in Table 4. First, they confirm the monotonicity of the shielding effect. The relationship remains strictly negative throughout the distribution, ruling out the concern where the shielding mechanism might collapse under extreme peer distress. Second, the plots show the integrity of the shield in the right tail. Even when connected peers exhibit heightened crash risk (the far right of the x-axis), the focal asset consistently exhibits lower skewness (safer tail distribution). The even steeper slope suggests that the developer-induced differentiation is

robust and more pronounced when the network faces severe structural shocks, implying the "flight-to-quality" channel documented in our main results.

OA-C2. Additional Results on Network Topology

Additional topology measure: small-world sigma. To rigorously assess the network's topological position between regular lattices and random graphs, we compute the "small-world" metric, σ, following the quantitative framework proposed by Humphries and Gurney (2008). The metric is defined as:

$$\sigma = \frac{\gamma}{\lambda} = \frac{C_{real}/C_{rand}}{L_{real}/L_{rand}} \tag{OA-8}$$

where C and L denote the clustering coefficient and average path length, respectively, and the subscript *rand* refers to the expected values in an equivalent Erdős-Rényi random graph with the same number of nodes and edges. A network is considered to exhibit small-world properties if $\sigma > 1$, which typically arises from a combination of high local clustering ($\gamma \gg 1$) and short global path lengths ($\lambda \approx 1$) (Watts & Strogatz, 1998).

Figure OA-6 presents the evolution of σ on a natural logarithmic scale. We observe that the network exhibits substantial small-world characteristics throughout the sample period, with $\ln(\sigma)$ stabilizing in the range of $4 \sim 6$ (implying $\sigma \approx 50 \sim 400$). This magnitude is primarily driven by the network's high clustering coefficient relative to random graphs ($\gamma \gg 1$), a common feature in social collaboration networks where interactions are locally dense.

The trajectory suggests a rapid establishment of small-world topology during the initial phase (2013–2015), coinciding with the network's expansion. In the later period (2016–2025), despite the rising modularity and declining global efficiency observed in the main text, the giant component retains a σ significantly above the random baseline. This suggests that while the macro-structure has fragmented into distinct communities, the internal structure within the giant component remains highly optimized for local information flow relative to a random configuration.

Theoretical benchmark for topological structural change. To rigorously interpret the asymmetry observed in the empirical Peak Test, we construct a theoretical model quantifying the topological structural change ($\Delta\mathcal{S}$) of the developer network under random data perturbations. We model the aggregate developer network as a weighted graph derived from the sequence of weekly snapshots $\{G_t\}$. For any pair of nodes i and j, the weight ω_{ij} is

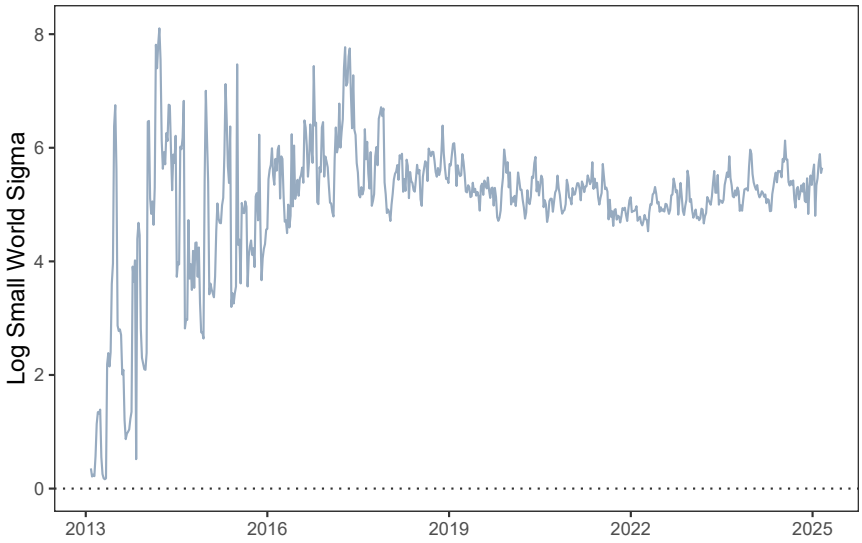


Figure OA-6: Temporal Evolution of Small-Worldness

This figure tracks the trajectory of the small-world coefficient, σ, on a natural logarithmic scale. The metric is calculated as the ratio of the normalized clustering coefficient to the normalized characteristic path length, $\sigma = \frac{C/C_{rand}}{L/L_{rand}}$, relative to an equivalent Erdős-Rényi random graph. The dotted horizontal line at $y = 0$ represents the baseline for a random network ($\ln(\sigma) = 0$ or $\sigma = 1$). Values significantly above this baseline indicate strong small-world topology, characterized by high local clustering combined with short global path lengths within the giant component.

defined as the *temporal persistence* of their collaboration:

$$\omega_{i,j} = \sum_t \mathbb{I}(w_{i,j,t} > 0), \tag{OA-9}$$

where $w_{i,j,t} > 0$ indicates the existence of a link in week t. A greater $\omega_{i,j}$ implies that token i and j are connected by co-developers across a longer time. The total volume of observable links is $E_{total} = \sum \omega_{ij}$, and the total number of unique links is $E_{unique} = \sum_{i,j} \mathbb{I}(\omega_{i,j} > 0)$. To capture the heterogeneity of ω across token pairs, we assume it follows a log-normal distribution with parameters (μ, σ) where we take $\sigma = 1.5$ to capture a large heterogeneity. According to the property of log normal distribution,

$$\frac{E_{total}}{N_{nodes} \times (N_{nodes} - 1)/2} = \bar{\omega} = \exp\left\{\mu + \frac{1}{2}\sigma^2\right\}. \tag{OA-10}$$

Then we obtain the calibrated $\mu \simeq 45$ based on the actual E_{total}, and the number of tokens N, and the assumed σ.

We define the *Topological Structural Change* (ΔS) as the relative deviation in the set of unique edges E_{unique}. The theoretical simulation proceeds in two directions:

1. The Pruning Process (Temporal Redundancy). In the pruning test, we randomly remove a fraction α of the total temporal links E_{total}. For a unique connection (i, j) to be topologically severed (i.e., removed from the aggregate graph), all its ω_{ij} times of temporal connections must be removed. Assuming random sampling, and the perturbation ratio is α (in magnitude), the probability of complete severance is $\alpha^{\omega_{ij}}$. The expected structural loss is:

$$\Delta\mathcal{S}_{prune}(\alpha) = \frac{1}{E_{unique}} \sum_{(i,j)} \alpha^{\omega_{ij}} = \mathbb{E}[\alpha^{\omega}]. \tag{OA-11}$$

Because ω follows a heavy-tailed distribution with a high mean ($\bar{\omega} \approx 45$), the function is strictly convex. This captures *temporal redundancy*: stable developer pairs interact over many weeks ($\omega_{ij} \gg 1$), creating a robust buffer where the topological relationship survives even if a large portion of weekly records are lost.

2. The Noise Injection Process (Sparsity-Driven Sensitivity). In the noise injection test, we add spurious links amounting to a fraction α of the total volume E_{total}. Crucially, since the potential space of all possible node pairs is vast compared to the existing connections (high sparsity), a randomly added link almost certainly connects a pair with $\omega_{ij} = 0$, thereby creating a new unique edge with weight 1. The number of spurious unique edges is thus αE_{total}. The relative structural shock is derived as:

$$\Delta\mathcal{S}_{noise}(\alpha) = \frac{\alpha(E_{total} - E_{unique})}{E_{unique}} = \alpha \times (\bar{\omega} - 1) \simeq 44\alpha. \tag{OA-12}$$

This reveals a powerful *multiplier effect*. Because the existing network is constructed from highly persistent ties (mean weight 45), while random noise consists of ephemeral ties (weight 1), injecting noise equivalent to just 1% of the data volume induces a topological distortion of approximately 44%. This mathematical property explains the sharp collapse observed in the Peak Test: the network's small-world structure is extremely sensitive to the introduction of random shortcuts.

Implications for inference and economic interpretation. The distinct geometric properties of the structural change curve offer three critical insights into the validity and nature of the developer network.

First, the results align with *small-world network theory*, where the introduction of random shortcuts (spurious edges) drastically alters global connectivity properties. The fact that our shielding effect vanishes precisely as the topological noise increases confirms that the risk absorption capacity is an emergent property of the specific, organized collaboration architecture, rather than a generic feature of connectivity.

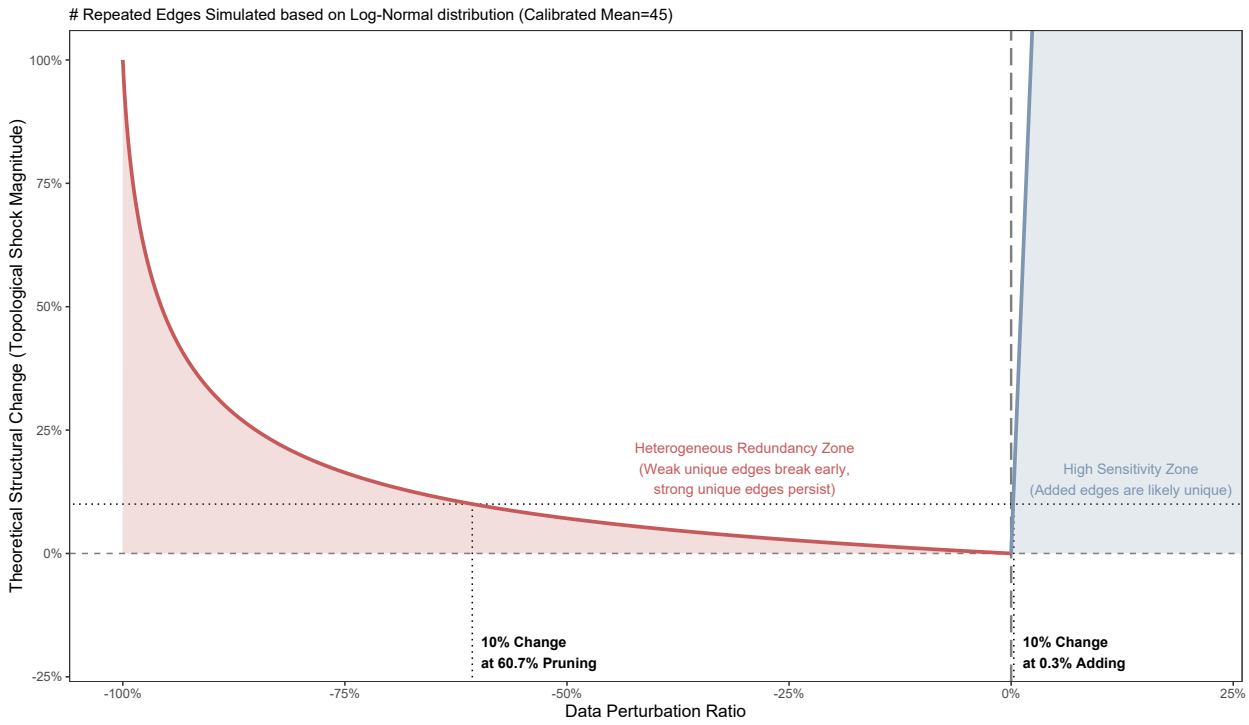


Figure OA-7: Theoretical Structural Change

This figure illustrates the theoretical structural change of the developer network under random perturbations, simulated based on the heterogeneous distribution of developer ties. We simulate edge weights (collaboration intensity) using a Log-Normal distribution calibrated to match the empirical mean connectivity of the observed network ($\mu \approx 45$, $\sigma = 1.5$). The x axis reports the data perturbation ratio, where negative values (red zone) represent edge pruning (removing a fraction of observed edges) and positive values (blue zone) represent edge adding (introducing additional edges). The y axis plots the resulting theoretical structural change (topological shock magnitude). The convex decay red curve indicates *structural redundancy*: due to the heavy-tailed weight distribution, the network structure is resilient to random pruning, pruning primarily removes weak, redundant edges early, so substantial pruning is required to generate a material structural change (e.g., a 10% change occurs at 60.7% pruning). The steep blue linear rise illustrates *high sensitivity*: added edges are likely to be unique and therefore disproportionately alter network topology (e.g., a 10% change occurs at only 0.3% adding). Dashed and dotted reference lines mark the zero-perturbation benchmark and the 10% change threshold, respectively.

Second, the asymmetry provides strong assurances regarding data quality and inference. The redundancy observed in the pruning domain implies that our results are robust to *false negatives* (missing data). Even if we miss-record a significant portion of developer activities, the convex decay suggests that the core topological structure—and thus the estimated shielding effect—remains largely intact. Conversely, the hypersensitivity in the noise domain implies that our results are strictly driven by *authentic* linkages. If the observed network contained substantial noise or casual, non-substantive links, the linear multiplier effect would have likely washed out the results entirely. This suggests that any potential measurement error in our sample likely biases our estimates toward zero, implying that the true economic

magnitude of the developer network's influence is a conservative lower bound.

Finally, it is worth noting that this theoretical benchmark is constructed purely based on *unweighted* topological existence, independent of the weight measure choices, and importantly, peer volatility values. The fact that the empirical regression coefficients track this theoretical topological loss curve so closely indicates that the mechanism is fundamentally structure-driven. It is the existence of the robust collaboration structure themselves — validated by the difficulty of breaking them and the ease of diluting them — that determines the system's resilience.

OA-C3. Alternative Network Weight Measures

Here we provide rigorous mathematical representations for the alternative developer network adjacency matrices, in addition to the baseline cosine similarity of developers documented in Section 1.1.2. Specifically, let M denote the total number of potential GitHub developers over the sample period. For each token i at time t, define the binary developer indicator vector

$$\boldsymbol{c}_{i,t} \;=\; \big(c_{i,t}^{(1)},\, c_{i,t}^{(2)},\, \ldots,\, c_{i,t}^{(M)}\big)',$$

where

$$c_{i,t}^{(k)} \;=\; \begin{cases} 1, & \text{if developer } k \text{ is a developer to token } i \text{ at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

Equivalently, the set of developers to token i at time t is $S_{i,t} \;=\; \big\{\, k : c_{i,t}^{(k)} = 1 \,\big\}$.

Next, let $w_{i,t}^{(k)}$ equals the number of commits made by developer k to token i within the month as of t, and define the weighted vector

$$\boldsymbol{w}_{i,t} = \big(w_{i,t}^{(1)},\, w_{i,t}^{(2)},\, \ldots,\, w_{i,t}^{(M)}\big)'.$$

Note that $c_{i,t}^{(k)} = \mathbb{I}(w_{i,t}^{(k)} \geq 1)$.

Similarly, define the active developer indicator vector

$$\boldsymbol{c}_{i,t}^{\text{act}} = \big(c_{i,t}^{\text{act},(1)},\, c_{i,t}^{\text{act},(2)},\, \ldots,\, c_{i,t}^{\text{act},(M)}\big)', \qquad c_{i,t}^{\text{act},(k)} = \mathbf{1}\{\, w_{i,t}^{(k)} \geq 1 \,\},$$

and the active-weighted vector

$$\boldsymbol{w}_{i,t}^{\text{act}} = \left(w_{i,t}^{\text{act},(1)},\, w_{i,t}^{\text{act},(2)},\, \ldots,\, w_{i,t}^{\text{act},(M)}\right)', \quad w_{i,t}^{\text{act},(k)} = \begin{cases} w_{i,t}^{(k)}, & \text{if } w_{i,t}^{(k)} \geq 1, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, for each pair of tokens (i, j) at time t, we construct five alternative undirected adjacency matrices $G_t^x = \left\{G_{i,j,t}^x\right\}_{n_t \times n_t}$. Each element $G_{i,j,t}^x$ is defined as follows (with $i \neq j$ and $G_{i,i,t}^x = 0$):

1. *Number of Overlapping Developers*:

$$G_{i,j,t}^{\text{NC}} \;=\; \left|S_{i,t} \,\cap\, S_{j,t}\right| \;=\; \sum_{k=1}^{M} c_{i,t}^{(k)}\, c_{j,t}^{(k)}. \tag{OA-13}$$

2. *Weighted (Developer) Similarity*:

$$G_{i,j,t}^{\text{WCS}} \;=\; \frac{\displaystyle\sum_{k=1}^{M} w_{i,t}^{(k)}\, w_{j,t}^{(k)}}{\|\boldsymbol{w}_{i,t}\|\,\|\boldsymbol{w}_{j,t}\|} \;=\; \frac{\boldsymbol{w}_{i,t}'\,\boldsymbol{w}_{j,t}}{\sqrt{\boldsymbol{w}_{i,t}'\,\boldsymbol{w}_{i,t}}\,\sqrt{\boldsymbol{w}_{j,t}'\,\boldsymbol{w}_{j,t}}}. \tag{OA-14}$$

3. *Number of Overlapping Active Developers (AD)*:

$$G_{i,j,t}^{\text{NAC}} \;=\; \sum_{k=1}^{M} c_{i,t}^{\text{act},(k)}\, c_{j,t}^{\text{act},(k)} \;=\; \left|S_{i,t}^{\text{act}} \,\cap\, S_{j,t}^{\text{act}}\right|. \tag{OA-15}$$

4. *AD Similarity*:

$$G_{i,j,t}^{\text{ACS}} \;=\; \frac{\displaystyle\sum_{k=1}^{M} c_{i,t}^{\text{act},(k)}\, c_{j,t}^{\text{act},(k)}}{\|\boldsymbol{c}_{i,t}^{\text{act}}\|\,\|\boldsymbol{c}_{j,t}^{\text{act}}\|} \;=\; \frac{\left(\boldsymbol{c}_{i,t}^{\text{act}}\right)'\boldsymbol{c}_{j,t}^{\text{act}}}{\sqrt{\left(\boldsymbol{c}_{i,t}^{\text{act}}\right)'\boldsymbol{c}_{i,t}^{\text{act}}}\,\sqrt{\left(\boldsymbol{c}_{j,t}^{\text{act}}\right)'\boldsymbol{c}_{j,t}^{\text{act}}}}. \tag{OA-16}$$

5. *Weighted AD Similarity*:

$$G_{i,j,t}^{\text{WACS}} \;=\; \frac{\displaystyle\sum_{k=1}^{M} w_{i,t}^{\text{act},(k)}\, w_{j,t}^{\text{act},(k)}}{\|\boldsymbol{w}_{i,t}^{\text{act}}\|\,\|\boldsymbol{w}_{j,t}^{\text{act}}\|} \;=\; \frac{\left(\boldsymbol{w}_{i,t}^{\text{act}}\right)'\boldsymbol{w}_{j,t}^{\text{act}}}{\sqrt{\left(\boldsymbol{w}_{i,t}^{\text{act}}\right)'\boldsymbol{w}_{i,t}^{\text{act}}}\,\sqrt{\left(\boldsymbol{w}_{j,t}^{\text{act}}\right)'\boldsymbol{w}_{j,t}^{\text{act}}}}. \tag{OA-17}$$

By construction, each adjacency matrix G_t^x is symmetric ($G_{i,j,t}^x = G_{j,i,t}^x$) and has zeros on the diagonal ($G_{i,i,t}^x = 0$). These six networks capture progressively finer aspects of developer linkage among tokens, ranging from simple headcounts of overlapping developers to fully weighted measures that account for both developer overlap and activity intensity.

We employ these alternative adjacency matrices to re-estimate the baseline instrumental variable specifications, ensuring that our main findings are not driven by the specific choice of the cosine similarity metric. Table OA-1 presents the second-stage 2SLS results across five alternative network definitions. The estimates exhibit remarkable stability across all specifications, reinforcing two critical dimensions of our mechanism.

First, the consistency between simple count-based measures (Columns 1-2) and normalized similarity measures (Columns 3-5) indicates that the results are robust to how we measure this network. Second, and perhaps more importantly, the results hold robustly when restricting the network to *active* developers (Columns 2, 3, and 5). This rules out the concern that the effect might stem from passive historical linkages (e.g., forked repositories with no recent activity) and confirms that it is the real-time, active coordination of human capital that generates asset resilience.

In Panel A, the coefficients on *DevNet Peer IVOL* remain negative and statistically significant at the 1% level, with magnitudes ranging from -0.536 to -0.564, closely mirroring the baseline estimates. This pattern persists for downside and tail risk measures in Panel B, as well as lottery-like features in Panel C. Furthermore, Panel D confirms the asymmetric transmission of systematic risk: the developer network facilitates upside participation (positive *USBETA*) while blocking downside comovement (insignificant *DSBETA*). Collectively, these findings confirm that the risk-shielding effect is a fundamental feature of developer connectedness, independent of the functional form of edge weights or the inclusion of passive contributors.

OA-C4. Alternative Statistical Inferences

To ensure that our statistical inference is not an artifact of the specific lag structure employed in the Driscoll-Kraay estimation, Table OA-2 re-estimates the main results using standard two-way clustered standard errors (clustered by token and week). This conservative benchmark yields results consistent with our baseline. Panel A confirms that the risk-shielding effect on idiosyncratic volatility remains statistically significant at the 1% level. Furthermore, the results for downside risks (Panel C) and structural characteristics (Panel D) hold robustly. This consistency verifies that the identified network effects are insensitive to alternative assumptions regarding the error covariance matrix.

OA-C5. Additional Discussion on Mobilization Channel

To address potential concerns regarding the arbitrary unit dependence of log-like transformations (e.g., *arcsinh*) in the presence of zero-valued outcomes (Chen & Roth, 2024), Table OA-3 presents three alternative specifications. Column (1) employs a Poisson regression to estimate the total effect. The coefficient on DevNet Peer IVOL is positive and highly significant (0.086, $t = 6.46$). Since Poisson estimates are scale-invariant, this result confirms that the aggregate volume of developer effort unambiguously increases in response to peer structural risk, yielding an approximate semi-elasticity interpretation.

We further decompose this mobilization into extensive and intensive margins. Column (2) uses a Linear Probability Model (LPM) to test the extensive margin, i.e., the mobilization effect on whether or not commit. The significant positive coefficient (0.100, $t = 8.67$) indicates that peer risk acts as a wake-up call, significantly increasing the probability that a dormant repository becomes active.

Finally, Column (3) examines the effort intensification among active developers. We use a subsample where the token-week observations are restricted to have strictly positive number of commits within the corresponding week. This subsample has 104,164 observations. Even conditional on such active community, peer risk is associated with a significant increase in the volume of commits (0.080, $t = 5.91$). Collectively, these results suggest that the developer's mobilized response is pronounced regarding both the extensive and intensive margin: peer risk not only triggers the activation of "standby" developers, but also spurs active maintainers to intensify their contributions (intensive margin).

In addition, the coefficients on control variables provide additional validity checks for our specification.

Size Effect: $ln(Cap)$ is consistently positive and highly significant across all columns, reflecting that larger projects naturally have more active and consistent development teams.

Life-cycle Effect: $ln(Age)$ shows a negative association with both the probability of activity (-0.060) and the intensity of work (-0.209). This is consistent with a "project life-cycle" view where older projects enter a maintenance phase with lower frequency of code updates compared to younger, fast-iterating projects.

Performance Feedback: Interestingly, Ret 4,1 (past month return) is negatively associated with the probability of committing code (-0.010 in Col 2). This implies a counter-cyclical effort pattern: developers are more likely to be active when prices are dropping (low returns), possibly to fix bugs or restore confidence, rather than when prices are surging (wealth effect inducing laziness). This aligns perfectly with the "Panic Coding" narrative.

Impact of Social Network: The coefficients on SocNet Peer IVOL offer a sharp contrast to those of the developer network. While social network risk shows a slight positive association with the extensive margin (0.006 in Column (2)), it is significantly negatively associated with the intensive margin (−0.014 in Column (3)), and fails to generate a significant aggregate mobilization (insignificant in Column (1)). To some extent, the extensive margin is easier to fake in practice, as the project somehow only needs to submit whatever one commit. This is particularly "useful" for socially hyped token projects: social panic might generate "noise" that attracts superficial attention (slightly higher probability of activity), it does not translate into productive output and may even distract existing developers (negative intensive margin). This disparity highlights the unique nature of the developer channel. In contrast, structural risk in the code base (DevNet) serves as a precise "call to action," driving meaningful increases in both participation and effort intensity.

OA-C6. Alternative Proxies for Network Interactions

To verify that the interplay between developer and social networks is not an artifact of the specific cosine similarity measure used in the baseline, we replicate the interaction analysis using the raw count of overlapping developers as the alternative link definition. As shown in Table OA-4, the results are qualitatively identical to the main findings. Panel A confirms the significant interference effect, where high social centrality dampens the developer shield. Panel B shows no significant heterogeneity within the developer network itself, reinforcing that the shielding benefit is a broad structural feature. Panel C robustly supports the buffering effect, demonstrating that deep integration into the technical network helps assets resist social contagion, regardless of the functional form of network weights.

To further test the robustness of network interactions, we use Google Search Volume and GitHub Stars as observable proxies for social attention and technical prominence, respectively. The results are reported in Table OA-5. Panel A confirms the "interference hypothesis": the interaction with search volume is significantly positive, indicating that retail noise actively disrupts the developer network's shielding capacity. In Panel B, we note that GitHub Stars serve as a "hybrid" proxy—capturing technical endorsement but also reflecting general popularity, making it structurally less clean than eigenvector centrality. Consistent with this dual nature, the interaction with Stars is statistically weaker than the baseline, as the latent social noise component within the metric likely partially offsets the buffering role of technical integration.

OA-C7. Effect Robustness across Aggregate Network States

We further investigate the structural stability of the risk-shielding mechanism. A potential

concern is that the shielding effect might be a regime-dependent phenomenon, operative only when the global network possesses high connectivity or efficiency. To address this, we interact the peer risk measure with a comprehensive set of time-varying global network characteristics, including network density, average betweenness, and modularity. The results in Table OA-6 show that the interaction terms are statistically indistinguishable from zero across all specifications. It implies that the localized "small-world" mechanism absorbs shocks independently of the aggregate network's shift from cohesion to fragmentation. This also aligns the implications from Table 6 Panel C, i.e., the risk shield effect is universal across the developer network. Mobilization is not a privilege reserved for central projects, but rather a micro-founded property of the collaborative structure itself.

OA-C8. Max Returns and Lottery-like Features

The literature documents that crypto tokens display pronounced lottery-like features (Chava et al., 2025). Lottery-like assets are typically characterized by a small probability of extremely large positive payoffs, strong positive skewness, and salient upside realizations that attract speculative demand (Bali et al., 2011). A widely used empirical proxy for such lottery-like payoffs is *MAX*, defined as the maximum daily return over a recent measurement window. The intuition is that recent extreme positive returns are highly salient to investors and summarize the right-tail payoff realization of an asset more directly than average returns or volatility alone.

As documented in Table 1, crypto tokens exhibit substantial right-tail return realizations, making MAX-based measures particularly relevant in this setting. While a comprehensive analysis of lottery demand and its pricing implications is beyond the scope of this paper, this section provides a reduced-form test of whether network linkages shape the transmission of lottery-like payoff realizations. Specifically, we examine whether peers' MAX-based lottery proxies predict the corresponding MAX-based measures of the focal token. This exercise does not directly estimate investors' lottery preferences; instead, it asks whether developer and social connectedness are associated with the propagation or attenuation of extreme upside return spikes.

Table OA-7 presents the results. We examine *MAX*, measured as the maximum daily return within the past month, and *SMAX*, measured as MAX scaled by volatility. The coefficients on *DevNet Peer MAX* and *DevNet Peer SMAX* are significantly negative, equal to −0.791 and −0.050, respectively. These results suggest that developer connectedness dampens extreme right-tail payoff realizations. In other words, when a token is more closely embedded in a shared developer community, extreme upside spikes among its developer-connected peers

are followed by weaker, rather than stronger, lottery-like realizations of the focal token. This pattern is consistent with the broader risk-shielding interpretation: developer networks appear to stabilize not only downside fragility and idiosyncratic volatility, but also excessive right-tail price movements that often characterize speculative digital assets. In contrast, the coefficients associated with social-network peers are positive and statistically significant, with the coefficient on *SocNet Peer MAX* equal to 1.000. That is, social co-mentions tend to transmit salient return realizations and may amplify attention-driven speculative demand, thereby increasing the likelihood that lottery-like payoff realizations propagate across socially connected tokens.

OA-C9. Social Network Centrality and Contagion Amplification

Table OA-8 investigates whether social network centrality moderates the transmission of peer risk, serving as a complementary test for Table 6. We interact SocNet Peer Y with SocNet EC to test if central nodes are more susceptible to sentiment propagation.The results reveal an amplification effect, particularly for downside and lottery-like risks. While the interaction term for overall volatility (*IVOL*) is positive but insignificant, the coefficients for Downside Volatility (0.745, t = 9.08), Maximum Returns (0.585, t = 7.71), and Downside Beta (1.623, t = 5.25) are positive and highly significant. These findings suggest that social centrality acts as a "super-conductor" for sentiment-driven contagion. Consistent with Hirshleifer et al. (2025), who characterize social connectivity as a conduit for opinion divergence and churning, we find that central nodes are disproportionately exposed to the propagation of peer distress.

This result stands in sharp contrast to the developer network findings in Panel C of Table 6. There, the interaction between peer risk and centrality was insignificant, indicating that the developer shield is universal and benefits peripheral and central projects equally. Comparing the two reveals a fundamental structural dichotomy: risk contagion in social networks is structurally stratified (central nodes suffer more), whereas risk shielding in developer networks is topologically universal (all nodes benefit equally). This reinforces our conclusion that the developer network functions as a fundamental infrastructure for stability, unlike the social network which functions as an attention amplifier of noise.

Table OA-1: Robustness Check Using Alternative Measures of Developer Network

DevNet Weight Measure	# DevOverlap	# Act DevOverlap	Act DevSim (%)	Wtd DevSim (%)	Wtd Act DevSim (%)
	(1)	(2)	(3)	(4)	(5)
Panel A: $IVOL_{i,t+1}$					
DevNet Peer *IVOL*	-0.564***	-0.554***	-0.536***	-0.556***	-0.538***
	(-3.926)	(-3.942)	(-3.917)	(-3.902)	(-3.935)
Observations	209,752	209,752	209,752	209,752	209,752
Adjusted R^2	0.370	0.369	0.369	0.368	0.367
Panel B1: $DSVOL_{i,t+1}$					
DevNet Peer *DSVOL*	-0.202***	-0.197***	-0.195***	-0.199***	-0.194***
	(-4.931)	(-5.050)	(-5.048)	(-4.920)	(-5.032)
Observations	209,752	209,752	209,752	209,752	209,752
Adjusted R^2	0.519	0.518	0.518	0.518	0.518
Panel B2: $ES5_{i,t+1}$					
DevNet Peer *ES*5	-0.193***	-0.188***	-0.185***	-0.190***	-0.185***
	(-3.159)	(-3.182)	(-3.183)	(-3.165)	(-3.187)
Observations	195,040	195,040	195,040	195,040	195,040
Adjusted R^2	0.568	0.568	0.568	0.568	0.567
Panel C: $MAX_{i,t+1}$					
DevNet Peer *MAX*	-0.809***	-0.773***	-0.741***	-0.799***	-0.747***
	(-4.144)	(-4.133)	(-4.099)	(-4.098)	(-4.133)
Observations	209,752	209,752	209,752	209,752	209,752
Adjusted R^2	0.287	0.285	0.285	0.281	0.281
Panel D1: $DSBETA_{i,t+1}$					
DevNet Peer *DSBETA*	0.028	0.026	0.026	0.028	0.026
	(0.172)	(0.172)	(0.172)	(0.172)	(0.172)
Observations	209,752	209,752	209,752	209,752	209,752
Adjusted R^2	0.055	0.054	0.054	0.055	0.055
Panel D2: $USBETA_{i,t+1}$					
DevNet Peer *USBETA*	0.486**	0.482**	0.473**	0.494**	0.481**
	(2.343)	(2.381)	(2.400)	(2.365)	(2.410)
Observations	209,752	209,752	209,752	209,752	209,752
Adjusted R^2	0.072	0.072	0.072	0.070	0.071

Notes. This table reports robustness checks for main 2SLS results in Table 3-4 using alternative measures of the developer network. Columns (1)–(5) use #DevOverlap (Number of Overlapping Developers), #Act DevOverlap (Number of Active Overlapping Developers), Act DevSim (%) (Active Developer Similarity, %), Wtd DevSim (%) (Weighted Developer Similarity, %), and Wtd Act DevSim (%) (Weighted Active Developer Similarity, %) as the key network measure, respectively. All specifications include controls and fixed effects as in the main analysis. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with lags corresponding to the dependent variable's window. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Table OA-2: Robustness Checks Using Two-Way Clustered Standard Errors

Panel A: Impact on Idiosyncratic Volatility

Dependent Variable:	$IVOL_{i,t+1}$			
	OLS			2SLS
Model:	(1)	(2)	(3)	(4)
DevNet Peer IVOL	-0.100***	-0.058***	-0.042***	-0.555***
	(-2.663)	(-2.862)	(-2.596)	(-2.666)
Observations	219,598	209,752	209,752	209,752
Adjusted R^2	0.164	0.324	0.376	0.370

Panel B: Developer Networks v.s. Social Networks

Dependent Variable:	$IVOL_{i,t+1}$			
DevNet Peer IVOL		-0.113***	-0.049***	-0.660***
		(-2.969)	(-2.941)	(-3.156)
SocNet Peer IVOL	1.049***	1.050***	0.842***	0.846***
	(7.794)	(7.788)	(6.392)	(6.420)
Observations	219,598	219,598	209,752	209,752
Adjusted R^2	0.195	0.195	0.393	0.385

Panel C: Developer Networks and Downside Risks

Dependent Variables (Y):	$DSVOL_{i,t+1}$	$DUVOL_{i,t+1}$	$NCSKEW_{i,t+1}$	$ES5_{i,t+1}$
DevNet Peer Y	-0.200***	-0.291***	-0.693***	-0.190**
	(-3.426)	(-4.184)	(-3.402)	(-2.161)
Observations	209,752	209,651	195,083	195,040
Adjusted R^2	0.519	0.510	0.160	0.568

Panel D: Developer Networks and Structural Risk Characteristics

Dependent Variables (Y):	$MAX_{i,t+1}$	$SMAX_{i,t+1}$	$DSBETA_{i,t+1}$	$USBETA_{i,t+1}$
DevNet Peer Y	-0.791***	-0.050**	0.028	0.482**
	(-2.711)	(-2.075)	(0.109)	(2.244)
Observations	209,752	209,752	209,752	209,752
Adjusted R^2	0.287	0.164	0.055	0.072

Notes. This table re-estimates the baseline specifications using two-way clustered standard errors at the token and time levels. Panel A reports OLS and 2SLS estimates for $IVOL_{i,t+1}$. Panel B includes both developer-network and social-network peer measures. Panels C–D report results for additional risk measures as indicated. All specifications include the full set of baseline controls and fixed effects as in the main analysis. t-statistics based on two-way (token and time) clustered standard errors are reported in parentheses. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Table OA-3: Alternative Measures of Developer Mobilization

Dependent:	$\text{Commits}_{i,t+1}$	$\mathbb{I}(\text{Commits}_{i,t+1} > 0)$	$\ln(\text{Commits})_{i,t+1}$
	Poisson	OLS	OLS
	(1)	(2)	(3)
DevNet Peer IVOL	0.086***	0.100***	0.080***
	(6.456)	(8.666)	(5.905)
Past IVOL	-0.026***	-0.001***	-0.004*
	(-3.253)	(-2.626)	(-1.755)
SocNet Peer IVOL	-0.007	0.006***	-0.014*
	(-0.367)	(3.324)	(-1.945)
Beta	0.048**	0.002	0.011
	(1.984)	(1.553)	(0.909)
ln(Cap)	0.113***	0.041***	0.076***
	(5.453)	(23.966)	(8.989)
ln(Vol)	-0.023	0.003**	0.016***
	(-1.431)	(2.410)	(4.151)
Ret 4,1	-0.009	-0.010***	-0.017
	(-0.575)	(-5.253)	(-1.602)
ln(SVI)	0.023	0.001	0.011
	(1.295)	(0.333)	(1.372)
ln(Age)	0.058	-0.060***	-0.225***
	(1.221)	(-12.642)	(-8.857)
Token FE	✓	✓	✓
Time FE	✓	✓	✓
Observations	183,711	210,202	104,164
Adjusted R^2		0.696	0.685
Pseudo R^2	0.800		
Mean(Dependent)	87.2	0.534	4.62

Notes. This table presents estimates of the effect of peer risk on developer activity using alternative specifications to address the concerns regarding "log with zeros", and to distinguish the extensive margin and intensive margin. Column (1) reports the Poisson Pseudo-Maximum Likelihood (PPML) estimates where the dependent variable is the raw count of commits within week $t+1$ ($Commits_{i,t+1}$). Column (2) reports OLS estimates of the extensive margin, where the dependent variable is a binary indicator equal to one if the token has at least one commit in week $t+1$ ($\mathbb{I}(\text{Commits}_{i,t+1} > 0)$). Column (3) reports OLS estimates of the intensive margin restricted to observations with positive commits, where the dependent variable is the log commits ($ln(Commits)_{i,t+1}$), which is actually transformed using the inverse hyperbolic sine function, $\text{asinh}(x) = \ln(x + \sqrt{x^2+1})$, to be consistent with the main text in Table 5, denoted as $\ln(\cdot)$ for interpretation simplicity. All specifications include token and time fixed effects. Standard errors are clustered at the token level. t-statistics are reported in parentheses based on Driscoll-Kraay standard errors with a lag of 5 weeks. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Table OA-4: Robustness of Network Interactions using Overlapping Developers

Dependent Variables (Y):	$IVOL_{i,t+1}$	$DSVOL_{i,t+1}$	$MAX_{i,t+1}$	$ES5_{i,t+1}$	$DSBETA_{i,t+1}$	$USBETA_{i,t+1}$
Model:	(1)	(2)	(3)	(4)	(5)	(6)
Panel A: Heterogeneous Developer Network Effects across Social Network Positions						
DevNet Peer Y	-0.704***	-0.209***	-0.845***	-0.198***	-0.003	0.431**
	(-4.838)	(-5.021)	(-4.246)	(-3.229)	(-0.015)	(2.029)
DevNet Peer Y × SocNet EC	0.780***	0.223***	0.940***	0.306***	0.649***	0.764*
	(4.836)	(3.235)	(3.908)	(3.703)	(2.861)	(1.837)
SocNet EC	-0.397*	0.008	-0.088*	0.000	-0.882***	-0.411
	(-1.922)	(0.147)	(-1.817)	(0.038)	(-3.599)	(-1.342)
Observations	209,752	209,752	209,752	195,040	209,752	209,752
Adjusted R^2	0.386	0.519	0.288	0.569	0.054	0.071
Panel B: Heterogeneous Developer Network Effects across Developer Network Positions						
DevNet Peer Y	-0.691***	-0.212***	-0.822***	-0.223***	0.037	0.480**
	(-4.591)	(-4.888)	(-4.010)	(-3.155)	(0.216)	(2.208)
DevNet Peer Y × DevNet EC	-0.189	-0.111	-0.589	-1.262	-0.341	0.393
	(-0.567)	(-0.580)	(-1.320)	(-1.048)	(-0.280)	(1.115)
DevNet EC	0.369	0.204	0.138	0.166	0.279	-0.378*
	(0.955)	(0.973)	(1.536)	(1.104)	(0.237)	(-1.921)
Observations	209,752	209,752	209,752	195,040	209,752	209,752
Adjusted R^2	0.385	0.518	0.287	0.557	0.054	0.072
Panel C: Heterogeneous Social Network Effects across Developer Network Positions						
SocNet Peer Y	0.859***	0.140***	1.012***	0.117***	0.543***	0.948***
	(6.954)	(10.151)	(6.176)	(7.643)	(4.595)	(6.716)
SocNet Peer Y × DevNet EC	-0.447***	0.034*	-0.520***	0.025	-0.289**	-0.445***
	(-4.633)	(1.778)	(-4.565)	(1.130)	(-2.468)	(-2.673)
DevNet EC	0.162	-0.028	0.034	-0.004	0.263	0.071
	(1.494)	(-1.497)	(1.373)	(-1.579)	(1.543)	(0.325)
Observations	209,752	209,752	209,752	195,040	219,598	219,598
Adjusted R^2	0.393	0.526	0.299	0.575	0.045	0.058

*Notes.*This table reports robustness checks on the interaction between the developer network (DevNet) and the social network (SocNet). The alternative developer-network measure is based on the number of overlapping developers. Panel A interacts DevNet peer Y with the focal asset's SocNet centrality (SocNet EC). Panel B interacts DevNet peer Y with the focal asset's DevNet centrality (DevNet EC). Panel C interacts SocNet peer Y with the focal asset's DevNet centrality (DevNet EC). All specifications include the full set of baseline controls and fixed effects as in the main analysis. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with lags corresponding to the dependent variable's window. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Table OA-5: Robustness of Network Interactions Using Stars and Attention

Dependent Variables (Y):	$IVOL_{i,t+1}$	$DSVOL_{i,t+1}$	$MAX_{i,t+1}$	$ES5_{i,t+1}$	$DSBETA_{i,t+1}$	$USBETA_{i,t+1}$
Model:	(1)	(2)	(3)	(4)	(5)	(6)
Panel A:Heterogeneous Developer Network Effects across Social Network Positions						
DevNet Peer Y	-0.784***	-0.203***	-0.972***	-0.190***	0.030	0.565***
	(-4.931)	(-5.022)	(-4.397)	(-3.117)	(0.156)	(2.636)
DevNet Peer Y × Attention	0.092***	0.003	0.132***	0.000	-0.002	-0.067
	(3.138)	(0.259)	(2.964)	(-0.022)	(-0.039)	(-1.371)
Attention	-0.032	0.044***	-0.009	0.004***	0.037	0.014
	(-1.224)	(6.441)	(-1.512)	(4.469)	(1.304)	(0.496)
Observations	209,752	209,752	209,752	195,040	209,752	209,752
Adjusted R^2	0.384	0.519	0.284	0.568	0.055	0.071
Panel B:Heterogeneous Developer Network Effects across Developer Network Positions						
DevNet Peer Y	-0.832**	-0.198**	-0.770*	-0.148	-0.477	0.599
	(-2.321)	(-2.470)	(-1.686)	(-1.238)	(-0.973)	(0.957)
DevNet Peer Y × Stars	0.038	0.000	-0.002	-0.007	0.105	-0.017
	(0.576)	(-0.033)	(-0.029)	(-0.393)	(0.957)	(-0.161)
Stars	-0.044	-0.022*	-0.003	-0.001	-0.095	-0.066
	(-0.850)	(-1.673)	(-0.283)	(-0.418)	(-1.109)	(-0.865)
Observations	196,912	196,912	196,912	183,570	196,912	196,912
Adjusted R^2	0.392	0.527	0.291	0.578	0.053	0.071
Panel C: Heterogeneous Social Network Effects across Developer Network Positions						
SocNet Peer Y	1.349***	0.110***	1.405***	0.093***	0.661***	1.253***
	(4.618)	(5.336)	(6.977)	(3.833)	(2.874)	(4.720)
SocNet Peer Y × Stars	-0.142**	0.008**	-0.116**	0.007	-0.035	-0.090*
	(-2.041)	(2.149)	(-2.252)	(1.151)	(-0.948)	(-1.742)
ln(Stars+1)	0.095	-0.026***	0.015	-0.002	0.006	-0.007
	(1.634)	(-3.430)	(1.530)	(-1.384)	(0.137)	(-0.134)
Observations	196,912	196,912	196,912	183,570	205,812	205,812
Adjusted R^2	0.401	0.535	0.304	0.585	0.044	0.058

Notes. This table reports robustness checks on developer–social network interactions using alternative proxies for network positions: public attention and GitHub stars. Panel A interacts DevNet peer Y with the focal asset's social-network position measured by Attention= $\ln(SVI)$. Panel B interacts DevNet peer Y with the focal asset's developer-network position measured by logged number of stars. Panel C interacts SocNet peer Y with the focal asset's developer-network position measured by logged number of stars. To avoid zero values, we add one to SVI and # stars. All specifications include the full set of baseline controls and fixed effects as in the main analysis. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with lags corresponding to the dependent variable's window. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Table OA-6: Additional Analysis for Heterogeneity of Network Characteristics

Dependent Variables(Y):	$IVOL_{i,t+1}$	$DSVOL_{i,t+1}$	$DUVOL_{i,t+1}$	$NCSKEW_{i,t+1}$	$ES5_{i,t+1}$
Model:	(1)	(2)	(3)	(4)	(5)
Panel A: Network Density					
DevNet Peer Y	-0.547***	-0.241***	-0.336***	-0.782***	-0.204**
	(-3.711)	(-4.113)	(-4.738)	(-2.880)	(-2.498)
DevNet Peer Y × Density	-1.463	0.530	0.745	1.341	0.184
	(-0.968)	(1.140)	(1.011)	(0.682)	(0.331)
Adjusted R^2	0.385	0.519	0.511	0.160	0.568
Panel B: Average Betweenness					
DevNet Peer Y	-0.333*	-0.158***	-0.189	-0.704**	-0.166
	(-1.804)	(-2.686)	(-1.530)	(-2.218)	(-1.523)
DevNet Peer Y × Avg Betweenness	-0.002*	0.000	-0.001	0.000	0.000
	(-1.802)	(-0.801)	(-1.058)	(0.056)	(-0.311)
Adjusted R^2	0.386	0.519	0.510	0.159	0.568
Panel C: Average Strength					
DevNet Peer Y	-0.600***	-0.210***	-0.231***	-0.683**	-0.191**
	(-3.627)	(-4.226)	(-2.778)	(-2.558)	(-2.287)
DevNet Peer Y × Avg Strength	0.000	0.000	0.000	0.000	0.000
	(-0.606)	(0.290)	(-1.377)	(-0.081)	(0.017)
Adjusted R^2	0.385	0.519	0.510	0.160	0.568
Panel D: Global Clustering					
DevNet Peer Y	-0.484**	-0.266***	-0.433***	-0.944**	-0.276**
	(-2.163)	(-3.082)	(-4.067)	(-2.414)	(-2.248)
DevNet Peer Y × Global Clustering	-0.438	0.162	0.405	0.666	0.210
	(-0.744)	(0.959)	(1.371)	(0.943)	(0.930)
Adjusted R^2	0.385	0.519	0.511	0.160	0.568
Panel E: Modularity					
DevNet Peer Y	-0.816***	-0.275***	-0.133	-0.923**	-0.310***
	(-3.385)	(-3.056)	(-0.956)	(-2.426)	(-3.463)
DevNet Peer Y × Modularity	0.478	0.230	-0.499	0.779	0.391
	(0.861)	(1.044)	(-1.219)	(1.043)	(1.589)
Adjusted R^2	0.385	0.518	0.511	0.161	0.569

Notes. This table provides additional analysis on whether the effect of DevNet peer Y varies with the focal asset's developer-network characteristics. Each panel interacts DevNet peer Y with a different DevNet characteristic: density (Panel A), betweenness centrality (Panel B), average tie strength (Panel C), global clustering coefficient (Panel D), and modularity (Panel E). The interaction terms are generally insignificant, suggesting no systematic heterogeneity in the DevNet effect along these network characteristics. All specifications include the full set of baseline controls and fixed effects as in the main analysis. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with lags corresponding to the dependent variable's window. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Table OA-7: Developer Networks and Lottery-like Features

This table reports 2SLS estimates examining the impact of developer network peer risk on lottery-like features. The dependent variables are: 30-day Maximum Daily Return (*MAX*) and Scaled 30-day Maximum Daily Return (*SMAX*). In each specification, the explanatory variable *DevNet Peer Y* corresponds to the DevNet-weighted peer average of the respective dependent variable. *SocNet Peer Y* refers to the corresponding social network peer risk and lagged focal risk measures for that column's dependent variable. All specifications include token and time fixed effects and full controls. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with a lag of 5 weeks. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Dependent (Y):	$MAX_{i,t+1}$ (1)	$SMAX_{i,t+1}$ (2)
DevNet Peer Y	-0.791***	-0.050***
	(-4.080)	(-2.691)
SocNet Peer Y	1.000***	0.047***
	(6.132)	(10.993)
Controls	✓	✓
Token FE	✓	✓
Time FE	✓	✓
Observations	209,752	209,752
Adjusted R^2	0.287	0.164
Kleibergen-Paap F-stat	287.0	721.8
Wu-Hausman p-val	<0.001	<0.001

Table OA-8: Network Centrality and Risk Transmission within Social Network

This table reports 2SLS estimates examining how the impact of peer risk is moderated by the focal token's network position, proxied by Eigenvector Centrality (EC), serving as a supplementary analysis for Table 6. This table interacts *SocNet Peer Y* with the token's centrality in the Social Network (*SocNet EC*) to test for structural heterogeneous effects within the developer network. All specifications include the full set of controls, token fixed effects, and time fixed effects as in previous tables. T-statistics are reported in parentheses based on Driscoll-Kraay standard errors with lags corresponding to the dependent variable's window (5 or 15 weeks). $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.10$.

Dependent (Y):	$IVOL_{i,t+1}$ (1)	$DSVOL_{i,t+1}$ (2)	$MAX_{i,t+1}$ (3)	$ES5_{i,t+1}$ (4)	$DSBETA_{i,t+1}$ (5)	$USBETA_{i,t+1}$ (6)
DevNet Peer Y	-0.660***	-0.186***	-0.201***	-0.799***	0.020	0.473**
	(-4.657)	(-3.120)	(-4.957)	(-4.093)	(0.125)	(2.296)
SocNet EC	0.057	-0.057***	-0.390***	-0.105	-2.194***	-0.584*
	(0.158)	(-4.578)	(-4.316)	(-1.207)	(-6.463)	(-1.655)
SocNet Peer Y	0.846***	0.119***	0.142***	0.998***	0.455***	0.927***
	(6.931)	(8.020)	(10.291)	(6.112)	(4.438)	(6.331)
SocNet Peer Y × SocNet EC	0.102	0.745***	0.585***	0.826	1.623***	0.535
	(0.265)	(9.080)	(7.705)	(1.625)	(5.253)	(1.580)
Controls	✓	✓	✓	✓	✓	✓
Token FE	✓	✓	✓	✓	✓	✓
Time FE	✓	✓	✓	✓	✓	✓
Observations	209,752	195,040	209,752	209,752	209,752	209,752
Adjusted R^2	0.385	0.569	0.519	0.286	0.055	0.072

Online Appendix References

Bali, T. G., Cakici, N., & Whitelaw, R. F. (2011). Maxing out: Stocks as lotteries and the cross-section of expected returns. *Journal of Financial Economics*, *99*(2), 427–446.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), P10008.

Burt, R. S. (1992). *Structural holes: The social structure of competition*. Harvard University Press.

Cattaneo, M. D., Crump, R. K., Farrell, M. H., & Feng, Y. (2024). On binscatter. *American Economic Review*, *114*(5), 1488–1514.

Chava, S., Hu, F., & Paradkar, N. (2025). Gambling on crypto tokens? *Journal of Financial and Quantitative Analysis*, *60*(6), 2819–2846. https://doi.org/10.1017/S0022109024000681

Chen, J., & Roth, J. (2024). Logs with zeros? some problems and solutions. *The Quarterly Journal of Economics*, *139*(2), 891–936.

Da, Z., Engelberg, J., & Gao, P. (2011). In search of attention. *The journal of finance*, *66*(5), 1461–1499.

Hirshleifer, D., Peng, L., & Wang, Q. (2025). News diffusion in social networks and stock market reactions. *The Review of Financial Studies*, *38*(3), 883–937.

Humphries, M. D., & Gurney, K. (2008). Network 'small-world-ness': A quantitative method for determining canonical network equivalence. *PLoS One*, *3*(4), e0002051.

Latora, V., & Marchiori, M. (2001). Efficient behavior of small-world networks. *Physical Review Letters*, *87*(19), 198701.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, *45*(2), 167–256.

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*(2), 026113.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, *103*(23), 8577–8582.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442.