Economics of Open-Source Software (OSS) Policies

Paper Learning Presentation

Chenhui Lu

University of Florida

Nov. 2025

1. Background
2. Data
3. Model
4. Estimation & Identification
5. Core Estimates
6. Policy Counterfactuals
7. Conclusion

The Broader Social Problem: Open-Source Software as a Public Good

- **Public vs. Private Provision**
  - A small group of contributors generates benefits for a global community of users.
  - This leads to **positive externalities** — private incentives fall short of social benefits.
- **Innovation–Incentive Tradeoff**
  - Private firms bear development costs but cannot fully appropriate the returns.
  - Governments face the question: Should OSS be subsidized? How and where?
- **Policy Complexity**
  - Countries differ in their approach: China emphasizes control, while the U.S. and EU emphasize openness.

Core Social Question

How can policymakers design OSS policies that balance *private underinvestment* with *large public benefits*, in order to enhance

Specific Research Questions Addressed in the Paper

1. **Firm-Level Mechanisms**
   - How do firms allocate labor between in-house and OSS development under cost minimization?
   - Are OSS investments substitutes or complements to private capital?
   - How does OSS use affect website quality and competitive dynamics?
2. **Empirical Identification**
   - How can OSS investment and usage be measured from real code repositories and web traffic data?
   - How to estimate the marginal impact of OSS on productivity and quality?
   - How to identify *home bias* and international spillover effects?
3. **Policy Counterfactuals**
   - What happens when governments impose **restrictions** (e.g., penalties on foreign OSS)?
   - What are the welfare impacts of **subsidies** (e.g., incentives for

Key Empirical Facts about Open-Source Software (OSS)

- **Pervasive Use of OSS in Commercial Software**
  - 96% of commercial codebases rely on at least one OSS component.
  - 77% of the total lines of code in production are open-source.
- **Enormous Economic Value**
  - The total value created by OSS ranges from **billions (US)** to **trillions (global)** of dollars.
  - OSS acts as a critical infrastructure input for innovation across industries.

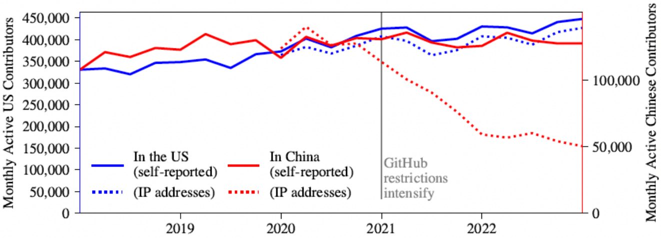
Research Gap and This Paper's Contribution

- **Existing Literature: Narrow Focus**
  - Emphasizes **individual motivations** for contribution (e.g., altruism, signaling, reputation).
  - Lacks **firm-level data** on OSS use and investment behavior.
  - Offers no quantitative framework to measure **innovation spillovers**.
- **This Paper:**
  - Shifts focus from **developer behavior** to **firm-level decisions** and **policy equilibrium**.

GitHub Contribution Trends: U.S. vs. China

- **Figure 2: Active Contributors over Time**
  - U.S. contributors (blue line): steadily increasing.
  - China contributors (red line): sharp decline since **2021**.
- **Self-reported vs. IP-based Locations**
  - Nearly **70,000** contributor gap.
  - Indicates widespread GitHub access via **VPNs**.

Figure 2: Monthly Active Contributors on GitHub in the US and China



VPNs and Policy Consequences in the OSS Ecosystem

- **VPN Regulation Facts (China)**
  - Since **2017**, unauthorized VPNs have been prohibited (MIIT, 2017).
  - Nonetheless, developers widely use VPNs to access GitHub for collaboration and code deployment.
- **Recent Case (2023)**
  - A Chinese developer reportedly earned **$140,000** in OSS income via VPN access and was penalized.
  - Highlights the coexistence of **informal institutions** (VPN use) and formal policy restrictions.

1. Background
2. Data
3. Model
4. Estimation & Identification
5. Core Estimates
6. Policy Counterfactuals
7. Conclusion

Research Context

- GitHub provides extensive data on **who contributes code**.
  - Developer activities (contributors) are observable.
- However, there is almost no data on **which firms use which OSS**.
  - Firm-level OSS **usage** is unobservable.
- **Result:** Lack of revealed preferences $\Rightarrow$ unable to identify the economic effects of OSS policies.

Why Focus on Web Development?

Reason 1: Strong Observability

- Website source code is publicly accessible (via `view source` or GitHub repos).
- Tech stacks can be automatically detected (e.g., Wappalyzer).
- $\Rightarrow$ Enables direct identification of which OSS components are used.

Reason 2: Industry Representativeness

- Web development is one of the most OSS-intensive sectors.
- JavaScript is the most widely used OSS language.
- Frameworks like React, Node.js, and Vue.js dominate the frontend ecosystem.

Reason 3: Global Industry Scale

- United States: 186,000 web developers.
- China: 926,000 web developers.

Problem Background

- **What we can observe (contributors):**
  - GitHub provides rich data on *who contributes code*.
  - $\Rightarrow$ Developer-level activity is highly observable.
- **What we cannot observe (firm use):**
  - $\Rightarrow$ Firm-level *use* of OSS is largely unobserved.

Consequence

Lack of **revealed preferences** at the firm level $\Rightarrow$ difficult to identify *causal economic effects* of OSS policies.

Data Sources and Structure

Level	Sample	Source	Variables
Website	Top 10,000 sites (2014–2022)	Alexa + Common Crawl	Traffic, source code, OSS frameworks
Firm	Owning company of each site	LinkedIn + GitHub + Orbis	#Developers, contribution records
Country	HQ country of website owner	WHOIS + Similarweb	Country-level investment & usage

Panel Construction (I)

1) Sampling

- Each month, select the global top 10,000 websites (2014–2022).
- Fetch archived HTML/code via Common Crawl.
- Get rankings and industry tags from Alexa / Similarweb.

2) Software Detection

- Detect three layers with Wappalyzer:
  - Backend frameworks
  - JavaScript frameworks (front-end)
  - UI frameworks (presentation layer)
- If no OSS is detected $\Rightarrow$ label as *in-house*.

Panel Construction (II)

3) Repository Matching

- Manually match each detected framework to its canonical GitHub repository.
- Link site-level usage to project-level contribution histories.

Linking Contributors to Firms (LinkedIn + GitHub)

- Use LinkedIn profiles to identify each contributor's **employer**.
- Combine with GitHub contribution logs (commits, PRs, issues).
- $\Rightarrow$ Map from *individual contributions* to *firm investment*.

Measures: Investment vs. Use

Concept	Definition	Source	Unit
OSS Investment	Developer time allocated to OSS projects	GitHub + LinkedIn	Hours
OSS Use	Whether a website adopts OSS frameworks	Wappalyzer detection	0/1
In-house Investment	Internal development (non-OSS) effort	LinkedIn headcount $\times$ avg hours	Hours

Summary of Data Strategy

Question	Author's Approach
How to measure firms' OSS use?	Focus on web development where source code is observable.
How to identify policy effects?	Compare OSS usage patterns across countries.
Why is web development representative?	Globally standardized languages and high OSS dependency.

Result I: Country-Level Patterns

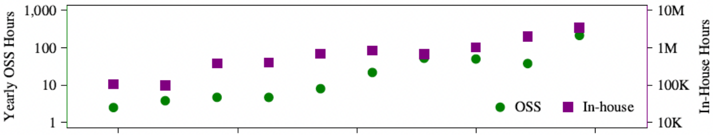
Table 1: Country-Level Web OSS Investment and Use, 2014 to 2022

Web OSS Investment				Web OSS Use			
Percent of Total		Per Web Developer-Hour		Percent of Total		Per Potential Use	
1. US	31.3%	1. Netherlands	0.0083%	1. US	40.8%	1. India	16.8%
2. China	12.9%	2. US	0.0055%	2. China	10.7%	2. Canada	14.7%
3. Germany	6.2%	3. Germany	0.0046%	3. India	6.1%	3. UK	13.8%
4. Japan	6.0%	4. Canada	0.0037%	4. UK	5.7%	4. US	13.3%
5. UK	4.9%	5. UK	0.0037%	5. France	4.7%	5. China	11.8%
6. Canada	3.4%	6. France	0.0023%	6. Russia	2.7%	6. Netherlands	11.5%
7. India	3.2%	7. Japan	0.0021%	7. Germany	1.4%	7. France	11.3%
8. France	3.0%	8. Russia	0.0006%	8. Japan	1.4%	8. Germany	10.3%
9. Russia	2.5%	9. China	0.0004%	9. Canada	1.1%	9. Russia	9.0%
10. Netherlands	2.5%	10. India	0.0001%	10. Netherlands	0.7%	10. Japan	6.1%
↪ Combined	75.9%			↪ Combined	75.3%		

Findings: The US leads in both OSS investment and usage; China ranks second. Positive externalities diffuse globally from US-centered projects.

Result II: Firm-Level Patterns

Figure 5: Web Traffic versus Web Development Investment and OSS Use



Key Facts:

- Larger firms invest more in OSS ($\uparrow$ firm size $\rightarrow \uparrow$ OSS contribution).
- OSS development hours are less than in-house development hours.

1. Background
2. Data
3. **Model**
4. Estimation & Identification
5. Core Estimates
6. Policy Counterfactuals
7. Conclusion

Step 1 | Overall Logic

Goal of the paper: To build a structural model that explains both "why firms invest in OSS" and "how OSS affects website quality."

Simply put, the model simulates how firms allocate programmers' time.

Step 2 | Core Sets and Notation

Symbol	Type	Meaning and Example
f	Firm	Company, e.g., Google, Alibaba
j	Website	Website under a firm, e.g., `youtube.com` belongs to Google
i	Consumer	Consumer type, e.g., users visiting websites
s	Software	Software framework, e.g., React, Vue, Django, Bootstrap
ℓ	Labor	Labor input (developer hours), e.g., hours invested in OSS development
t	Time	Year, e.g., 2014–2022
p	Part	Component of a website, e.g., backend, JavaScript (frontend logic), UI

Step 3 | Decision and Market Structure (Firm)

Role of Firm (f)

Each year t, firm f makes two types of decisions:

Decision Type	**Notation**	**Meaning**
Software choice	$s_{jpt} \in \mathcal{OS}_{pt} \cup \{f\}$	Which software (OSS or in-house) each website part uses
Labor input	$\ell_{fst} \geq 0$	Developer hours invested in improving each software s

Step 3 | Decision and Market Structure (Consumer)

Role of Consumer (i)
Consumer i chooses which website j to visit, based on website quality ξ_{jt}, language, country, and loading speed.

Define the traffic (or visit volume) as Q_{jit}.

Economic Meaning:

- Q_{jit} = traffic that website j receives from consumer type i at time t.
- More traffic $\Rightarrow$ higher revenue $\Rightarrow$ stronger incentive to invest.

Step 4 | Firm Profit Function (Equation 1)

$$\max_{s_{ft},\ell_{ft}} \left\{ \sum_{j \in J_f} R_{jit} Q_{jit}(\xi_t) - \sum_{s \in \mathcal{OS}_t \cup \{f\}} W_{fst}(s_{ft}, \ell_{ft}, K_t) \ell_{fst} \right\}$$

Website traffic revenue **Developer labor cost**

R_{jit}	Revenue per website visit
$Q_{jit}(\xi_t)$	Traffic, determined by website quality
$W_{fst}(\cdot)$	Wage rate paid to developers for software s
ℓ_{fst}	Firm f's labor input for software s
K_t	Industry-wide software capital stock (defined later)

Step 5 | Software Capital

This part explains the dynamic mechanism of the model.

Four Types of Capital (Equation 2):

$$K_t = \{\{K_{ft}\}_{f \in F_t}, \{K_{st}\}_{s \in \mathcal{OS}_t}, \{K_{cst}\}_{c \in C, s \in \mathcal{OS}_t}, \{K_{fst}\}_{f,s}\}$$

Type	Interpretation
K_{ft}	Firm's internal "private capital" accumulation
K_{st}	"Public capital" of each OSS framework
K_{cst}	National-level OSS contribution accumulation
K_{fst}	Firm-specific OSS investment accumulation

Step 5 | Capital Dynamics

Firm capital:

$$K_{ft} = (1 - \delta_p)K_{ft-1} + L_{ft}$$

OSS capital:

$$K_{st} = (1 - \delta_o)K_{st-1} + \sum_{f \in F_t} \ell_{fst}$$

δ_p, δ_o	Obsolescence rate (code depreciation)
L_{ft}	Firm's total labor input
ℓ_{fst}	Investment in specific OSS by firm f

Economic meaning: Software capital acts like a "knowledge stock": code depreciates over time, but new contributions accumulate.

Step 6 | Wage Equation and "DevOps" Logic

Developer wages consist of two parts:

$$\log W_{fst} = Dev_{fst} + Ops_{fst} + \bar{\omega}^{W}_{ft}$$

Component	Meaning	Economic Interpretation
Dev_{fst}	"Development" wage (improving software)	New feature / innovation development
Ops_{fst}	"Operations" wage (maintaining production)	Maintain current website operation
$\bar{\omega}^{W}_{ft}$	Firm fixed effect	Cross-firm wage level differences

Development Wage Operations Wage

$$Dev_{fst} = \frac{1}{\eta} \log L_{ft} + \gamma_L X^W_{fst}(K_t) + \kappa_L 1_{\ell_{fst-1}=0} + \omega^L_{fst}$$

- η: Labor supply elasticity
- γ_L: Marginal effect of capital characteristics
- κ_L: Switching cost
- ω^L_{fst}: Unobserved heterogeneity

$$Ops_{jp,t} = \gamma_S X^W_{fst}(K_t) + \kappa_S 1_{s_{jp,t} \neq s_{jp,t-1}} + \omega^S_{jst}$$

Step 7 | Website Quality Function

$$\Xi_{jft}(s_{jt}, K_t) = \beta_P \log K_{ft} + \beta_S \sum_p \alpha_p 1_{s_{jp,t} \in \mathcal{OS}_t} + \bar{\omega}^{\Xi}_m + \omega^{\Xi}_{jt} + \varepsilon^{\Xi}_{jt}$$

β_P	Marginal effect of firm private capital on website quality
β_S	Spillover effect of OSS usage on quality
α_p	Importance weight of each website part (backend / frontend / UI)
$\bar{\omega}^{\Xi}_m$	Market-year fixed effect
ε^{Ξ}_{jt}	Unobservable quality error

Economic Intuition: OSS usage can improve website quality—especially frontend and UI frameworks— thus increasing traffic and revenue.

Step 8 | Overall Mechanism Summary (Flowchart)

Firm labor investment $\Rightarrow$ Formation of software capital (K)

$\Downarrow$

Software capital improves website quality (ξ)

$\Downarrow$

Website quality increases traffic Q $\Rightarrow$ revenue R

$\Downarrow$

Firm updates investment decisions (path dependence, spillovers, deprecia

Transition: From Firms to Consumers

We now shift the perspective **from firms to consumers**, to analyze how website quality (ξ_{jt}) affects website visits and traffic shares.

Website Demand Model

Consumer behavior is defined as:

$$\max_{j \in \mathcal{J}_{mt} \cup \{0\}} \{ V_{ijt}(\xi_{jt}) + \varepsilon_{ijt}^{V} \}$$

i	Consumer type (language–country pair)
j	Website
t	Time
ξ_{jt}	Website quality (vertical quality)
ε_{ijt}^{V}	Preference shock
$j = 0$	Outside option (no website visit)

Utility function:

$$V_{ijt}(\xi_{jt}) = \begin{cases} 0, & j = 0 \\ \beta_V X_{ijt}^{V} + \xi_{jt}, & j \in \mathcal{J}_{mt} \end{cases}$$

Website Traffic Mechanism

Consumer choices determine the distribution of traffic:

$$Q_{jt}(\xi_t) = \sum_{i \in \mathcal{I}_t} Q_{ijt}(\xi_t)$$

where

$$Q_{ijt}(\xi_t) = \frac{\exp(V_{ijt}(\xi_{jt}))}{\sum_{k \in \mathcal{J}_{mt} \cup \{0\}} \exp(V_{ikt}(\xi_{kt}))} \cdot Q_{imt}$$

Q_{imt}	Maximum potential traffic of consumer type i in market m
Q_{ijt}	Probability of visiting website j $\times$ potential traffic
Q_{jt}	Actual total traffic aggregated across all consumers

Economic Intuition: Website quality and home bias jointly determine visit probabilities; aggregated visits form each website's market share.

Website Traffic and Firm Revenue Connection

Firm revenue comes from website traffic:

$$R_{jt} = P_{jt} \cdot Q_{jt}(\xi_t)$$

In this model, P_{jt} is not explicitly modeled, representing the marginal revenue per unit of traffic (e.g., advertising or sales revenue).

Substituting into the profit function:

$$\pi_{ft} = \sum_{j \in J_f} R_{jt}(\xi_t) - \sum_{s \in \mathcal{OS}_t \cup \{f\}} W_{fst} \ell_{fst}$$

Industry Equilibrium

Objective: Find a state where all firms and all consumers are simultaneously optimized.
Definition: For each year t, there exists a pure-strategy Nash equilibrium:

$$\{s^*_{ft}, \ell^*_{ft}\}_{f \in \mathcal{F}_t}$$

such that:

- Each firm maximizes its profit;
- Each consumer maximizes its utility.

Core Idea

- Consumer choice → identify preference parameters
- Firm choice → identify production and cost parameters
- Interaction of the two → discipline the equilibrium mechanism

Identification Logic Overview (from Table 2)

Economic question	Parameter(s)	Identification idea
Website traffic and home bias	β_V	Differences in traffic by country and language $\rightarrow$ "home bias"
Quality and capital	β_P, β_S	Response of website quality to private capital and to OSS use
Depreciation	δ_O, δ_P	Co-movement between current and past labor inputs
Wage functions	$\gamma_L, \kappa_L, \kappa_S, \rho, \sigma$	Dynamics of labor and software investment
Revenue function	R_{jt}, η	First-order condition for quality

One-sentence summary

Demand reveals preferences; Production reveals technology; Cost reveals behavior.

Step 1 – Demand Estimation (Consumer Demand)

$$V_{ijt} = \beta_V X_{ijt}^V + \xi_{jt} + \varepsilon_{ijt}$$

Estimation targets:

- β_V: strength of home bias
- ξ_{jt}: website quality (not directly observed)

Tools and data:

- Use `PyBLP` (Berry–Levinsohn–Pakes framework)
- Input Similarweb traffic shares:
  - split by country (China vs. U.S.)
  - split by language (Chinese vs. English)
- Matching: actual traffic shares ≈ model-predicted shares

Identification intuition:

- Higher local visit ratios ⇒ stronger home bias
- Under the same market, higher traffic ⇒ estimates a higher β_V or ξ_{jt}

Potential Traffic (Q_{imt}) Construction

$$Q_{jt}(\xi_t) = \sum_i \frac{\exp(V_{ijt})}{\sum_k \exp(V_{ikt})} Q_{imt}$$

Estimation relies on:

- ITU (International Telecommunication Union) internet user data
- Up to 20 websites per market/day

Step 2 – Quality Production Function

$$\Xi_{jt} = \beta_P \log K_{ft} + \beta_S OSS_{jt} + \bar{\omega}^{\Xi}_{mt} + \varepsilon^{\Xi}_{jt}$$

Parameter	Meaning	Identification source
β_P	Impact of private capital on quality	Co-movements of total labor and traffic
β_S	Impact of OSS use on quality	Co-movements of OSS use and traffic
$\bar{\omega}^{\Xi}_{mt}$	Market-year fixed effects	Address scale biases

Economic Intuition Summary

Module	**Core logic**	**Method**
Consumer choice	Traffic shares reveal home bias and quality	PyBLP (logit)
Quality production	Quality is determined by private capital + OSS	OP/LP two-step
Dynamic structure	Productivity ω follows a Markov process	One-step dynamic error model

Firm's Minimization and Maximization

we have completed:

- **Demand side:** estimation of consumer preference (β_V) and website quality (ξ_{jt})
- **Production side:** estimation of the quality production function (β_P, β_S)

Now we proceed to:

- **Infer firms' cost functions and depreciation rates** under these conditions, as well as the firms' underlying **revealed preferences**.

Step 3 | Cost Minimization

The firm's problem can be rewritten as:

$$\min_{s_{ft},\ell_{ft}} \sum_{s\in OSS\cup\{f\}} W_{fst}(s_{ft},\ell_{ft},K_t)\,\ell_{fst} \quad \text{s.t.} \quad \hat{\xi}_{jt} = \Xi_{fjt}(s_{jt},K_t) \quad (16)$$

Meaning: Given a target quality level, the firm chooses the least costly way to achieve the corresponding website traffic (Q_{jt}).

Symbol	**Meaning**
s_{ft}	Software choice (OSS or in-house R&D)
ℓ_{ft}	Labor input for each software
W_{fst}	Developer wage function
K_t	Industry-level software capital
Ξ_{fjt}	Quality production function

Statistical Structure: AR(1) Wage Process

To capture unobservable heterogeneity in wages:

$$\omega_{fst}^{L} = \rho_L \omega_{fst-1}^{L} + \nu_{fst}^{L}, \quad \nu_{fst}^{L} \sim N(0, \sigma_L^2)$$

$$\omega_{jst}^{S} = \rho_S \omega_{jst-1}^{S} + \nu_{jst}^{S}, \quad \nu_{jst}^{S} \sim N(0, \sigma_S^2)$$

Parameter	**Meaning**
ρ_L, ρ_S	Autocorrelation of unobserved wage components
σ_L, σ_S	Variance of unobserved heterogeneity
ν	Gaussian innovations

Targeted Moments

Parameter set:

$$\theta = [\delta_O, \delta_P, \alpha, \gamma_L, \gamma_S, \kappa_L, \kappa_S, \rho_L, \rho_S, \sigma_L, \sigma_S]$$

Each structural parameter corresponds to an observable empirical moment (see Table K1 for details):

Parameter	Identification source
δ_O, δ_P	Co-movement between labor input and OSS usage over time
α	Weighted importance of each website component (back-end/UI)
γ_L, γ_S	Weighted average of labor–software characteristics
κ_L, κ_S	Switching and indicator variables for software changes
ρ_L, ρ_S	Persistence (AR(1)) estimation
σ_L, σ_S	Standard deviation of traffic share for labor

Step 4 | Net Marginal Revenue Estimation

The author now turns to profit maximization (profit side).
Goal: Under a fixed quality level, estimate the website's **net marginal revenue** R_{jt}.

$$\max_{\xi_{ft}} \left\{ \sum_{j \in J_{ft}} R_{jt} Q_{jt}(\xi_t) - C_{ft}(\xi_t, L_{ft}) \right\}$$

Identification of Marginal Revenue Equation

From the first-order condition:

$$R_{ft} = \left(\frac{\partial Q_{ft}(\xi_t)}{\partial \xi_{ft}}\right)^{-1} \left(\frac{\partial C_{ft}(\xi_t, L_{ft})}{\partial \xi_{ft}} + \frac{\partial C_{ft}(\xi_t, L_{ft})}{\partial L_{ft}} \frac{\partial L_{ft}(\xi_t)}{\partial \xi_t}\right)$$

Term	Meaning
$\frac{\partial Q}{\partial \xi}$	Sensitivity of traffic to quality (from demand estimation)
$\frac{\partial C}{\partial \xi}$	Marginal cost increase from improving quality
$\frac{\partial L}{\partial \xi}$	Labor response to quality

Economic Intuition: This is a “first-order condition for quality”: at the optimum, marginal revenue equals marginal cost.

6.1 Demand Estimates (Demand Side)

Data and Methods

- Data source: **Similarweb 2014–2022** global website traffic

Main Findings

Indicator	Estimate	Meaning
Prob. of visiting domestic web. vs. foreign ones	40×	Strong country-level home bias
Prob. of visiting same-language web. vs. other-language ones	39×	Significant linguistic home bias
China's home bias strength	∼140×	China's "Great Firewall" drives extremely inward-oriented traffic structure

Net Marginal Revenue (R_{ft})

Formula

$$R_{ft} = \left(\frac{\partial Q_{ft}}{\partial \xi_{ft}}\right)^{-1} \left(\frac{\partial C_{ft}}{\partial \xi_{ft}} + \frac{\partial C_{ft}}{\partial L_{ft}} \frac{\partial L_{ft}}{\partial \xi_t}\right)$$

- Indicates the implicit marginal revenue per unit of website quality improvement

Empirical Results

Metric	Value	Reference / Source
Median $\hat{R}_{ft}$	\$0.10 per daily visitor	Estimated in paper
Range	\$0.02–\$0.57	Consistent with public firm data
Comparison	ESPN \$0.05 / small sites \$0.002	Lambrecht & Misra (2017), Ortiz-Cordova (2012)

Design of simulations

- Baseline: hold total firm labor L_{ft} fixed; iterate best responses to equilibrium.
- Two regimes:
  1. **Traffic fixed**: conservative cost-only effects.
  2. **Quality endogenous**: allow f to optimize quality.
- Uncertainty via bootstrap over traffic and OSS investment samples.

Restrictions in China (counterfactual)

- Penalty: add $\$100 \times \ell_{fst} \times (1 - K_{c(f)st}/K_{st})$ for foreign OSS.
- Short run: Chinese web OSS investment $\downarrow\sim$ 30%; domestic OSS little net gain.
- Long run: global costs rise; per \$1 penalty $\Rightarrow$ \$2 domestic loss, \$7 global deadweight.

Subsidies (China-only, China+US, Global)

- China-only: +5% domestic OSS; \$5 cost reduction per \$1 subsidy in China.
- China+US: +70–85% global OSS; \$6–\$11 cost reduction per \$1.

Product Market Competition

- When firms can improve quality, subsidies work better. Firms use the subsidy to make better websites and attract more users.
- Profits go up with subsidies. Restrictions hurt profits almost equally.
- The effect on consumers depends on how much OSS improves quality (β_S). If quality gains are small, consumer benefits are also small.

Key takeaways

- Restrictions **raise costs**; subsidies **create innovation spillovers**.
- OSS investment has big aggregate efsfects.
- Effects likely larger in sectors where quality strongly responds to OSS (AI/ML, cloud).

Thank You