GPUs as GPTs: Innovation, Propagation, and Downstream-to-Upstream Feedback

Avi Goldfarb* Yuval Lidany†

July 30, 2026

Abstract

We study how innovation in the GPU industry is shaped by the cross-market value of its technologies: a technology released in one application market can later propagate to the others—the downstream-to-upstream feedback that makes GPUs a general-purpose technology. We construct a panel that links each GPU technological advancement to its origin market and to the markets it propagated to. NVIDIA, AMD, and Intel compete in personal computing, data centers, and autonomous vehicles over 2008–2023. We estimate a leader–follower dynamic game, with NVIDIA as the leader, in which R&D builds firm capability, which drives technology release and cross-market propagation, and technology accumulated in one market raises entry into the others. The cross-market value of GPU innovation concentrates in the data center. Rivalry governs both releases and propagation: a firm's release hazard responds twelve times more strongly to rivals' recent releases than to its own history, its propagation six times. R&D spending alone reveals none of this: the industry's largest spender originates none of its technologies. We then use the model to examine how a merger between the two followers affects releases and propagation. Removing a competitor slows every firm. With the combination's pipeline gains, however, the merged firm nearly doubles its releases and raises its propagation by 28 percent; the leader accelerates on both margins: releases by 30 percent, propagation by 8.

*University of Toronto and NBER, agoldfarb@rotman.utoronto.ca.

†Harvard University, ylidany@hbs.edu.

"The programmable nature of our architecture allows us to make leveraged investments in research and development: we can support several multi-billion-dollar end markets with shared underlying technology" (NVIDIA's 2026 10-K).

1 Introduction

Graphics processing units (GPUs) have shifted from graphic focus to become the central a computing force input across multiple end-markets. The same underlying GPU technology that facilitate capabilities in one downstream applications, propagate to otherwise independent markets usages, such as the personal computing (PC), autonomous vehicles (AV), and data centers (DC). In this case, GPUs can be considered as a general-purpose technology (GPT), which is "characterized by pervasiveness (they are used as inputs by many downstream sectors), inherent potential for technical improvements, and innovational complementarities, meaning that the productivity of R&D in downstream sectors increases as a consequence of innovation in the GPT. Thus, as GPT's improve they spread throughout the economy, bringing about generalized productivity gains" (Bresnahan and Trajtenberg, 1995). In GPUs this mechanism also runs in reverse: the value of producing a technology depends on the ability to later propagate it across application markets, so downstream applications feed back into upstream innovation. Therefore, this paper asks two questions: how downstream application markets shape upstream GPU innovation through shared technology propagation, and how a change in market structure reshapes the technology race. Because GPUs are the enabling hardware of artificial intelligence (AI), how this industry innovates and where its technologies propagate shape the pace at which AI compute advances (Cockburn et al., 2019; Goldfarb et al., 2023).

To answer these questions we combine data on GPU manufacturers' R&D effort, costs, and technology release and adoption dates for NVIDIA, AMD, and Intel from 2008 through 2023 across the PC, AV, and data-center markets. The key identifying feature of the data

is that each underlying technology is linked both to the market in which it is first released and to the markets in which it later appears. This allows us to distinguish firms' productive capability, the timing with which firms release technologies in particular markets, and the speed with which technologies propagate across markets.

To analyze GPUs as a GPT, we construct and estimate an infinite-horizon dynamic model of technology release and cross-market propagation. In the model, firms choose R&D effort to develop GPU technologies that enable capabilities in downstream markets, while internalizing the future value of propagating those technologies to other markets. This future value reflects the value of additional downstream applications, which allows firms to spread the high fixed costs of R&D across multiple markets. Because NVIDIA is the first mover for most GPU technologies in the data, we use an asymmetric model in which NVIDIA is the leader and AMD and Intel are the followers.

The key state variables that allow the model to capture GPUs as a GPT are productive capability and market-specific technology stocks. Productive capability is a firm-level measure that evolves with R&D effort and a firm-specific drift and governs the speed of firm releases and propagation. Market-specific technology stocks summarize the firm's accumulated technological position in each downstream end market and grow with deployment. These market-specific stocks enter the model in two places. First, the stock in a given market enters the firm's per-period flow profit in that market, reflecting that a stronger technological position raises the revenue the firm earns from operating there. Second, stocks accumulated in *other* markets enter the firm's entry probability index for any given market, raising the probability that the firm enters it. This within-firm cross-market spillover gives GPUs their GPT character in the model: market-specific technology stocks accumulated in one market raise the value of entering and operating in other markets. Because accumulated technology stocks also feed into future productive capability and the firm's R&D-effort decision—a form of learning by doing—such accumulation raises the rate at which the firm releases and propagates new technologies. This learning is within-market but also spills across markets,

since capability is shared across them.

The model estimation highlights two main findings. First, GPU technologies propagate unevenly across downstream end-markets: the estimated propagation hazards are economically meaningful only in the data center, against baselines near zero in PC and AV, so the cross-market value of GPU innovation is currently concentrated in data centers.

Second, and central to the paper, competition shapes the timing of both releases and propagations. The strongest response is at the release stage. A firm's per-quarter release probability responds far more strongly to the recent flow of its rivals' releases than to its own release history: twelve times as strongly for the followers and nine times for the leader. Both rival coefficients, and both rival–own differences, are significant at the 1 percent level. The rivalry is two-sided: the leader responds to the followers' releases nearly as strongly as the followers respond to the leader's. Rivalry also operates at the propagation stage. A unit increase in rivals' recent propagation into a destination market raises a firm's own propagation hazard into that market 5.9 times as much as the firm's own prior propagation there; in the data center this amounts to about 18 percentage points per quarter, against 3 for the firm's own activity. The propagation coefficient is significant at the 1 percent level conditionally and the rival–own difference at the 5 percent level, with a bootstrap interval well away from zero; the result survives full re-estimation of the model on synthetic panels, where its finite-sample bias, if anything, runs toward zero. Firms, in short, release in races and propagate toward their rivals rather than toward their own past positions.

We use the estimated model to evaluate the effect of a counterfactual merger between the two followers, AMD and Intel. Theoretically, the effect of a merger between the followers is ambiguous. On one hand, the merger may increase deployment and propagation, since the merged entity may benefit from combining the firms' technology portfolios, keeping both development pipelines producing, and paying only once for the technologies both firms were developing separately. On the other hand, the merger decreases competitive pressure across downstream markets. The counterfactual separates the merger's two forces. The loss of a

competitor alone slows the technology race for every firm; these combination gains more than offset it, so the merger nearly doubles the merged firm's releases, raises its cross-market propagation by 28 percent, and accelerates the leader by 30 percent through its estimated racing coefficient. Varying how much of Intel's development pipeline the merged firm keeps traces the surface of this effect: keeping about a seventh of it already turns the leader from slowed to accelerated; the acceleration survives shedding the development the two firms duplicated; and keeping none of it, the merged firm gains through propagation rather than releases.

The paper makes two contributions to the literature. First, the paper adds to the understanding of GPT dynamics and innovation. Bresnahan and Trajtenberg (1995) propose a mechanism based on innovational complementarities between GPTs and downstream sectors, showing how improvements in a GPT raise the productivity of downstream R&D and generate broader economy-wide gains. Helpman and Trajtenberg (1998) model a GPT that diffuses sequentially because each application sector must invest before adopting it (see also Jovanovic and Rousseau, 2005); our propagation stage is the firm-level version of that margin, and the downstream-to-upstream feedback we add is the direction their framework leaves open. Using the information technology (IT) sector as a GPT, Bresnahan et al. (2002) add to the literature by showing that the productivity gains from technology adoption depend on organizational change, complementary investments, and firm-level reconfiguration that allow firms to fully benefit from technological improvements. This paper contributes to the literature by providing a missing GPT dynamic: feedback from end-market application opportunities to upstream innovation. Competition and opportunities to enable downstream capabilities in PC, data center, and AV market shape firms' GPU innovation incentives.

Second, the paper contributes to the literature on mergers and innovation. A large body of work shows that competition intensity shapes innovation incentives, with Aghion et al. (2005) documenting an inverted-U relationship between competition and innovation. Bloom et al. (2013) adds by identifying mechanisms through which market structure and spillovers

affect R&D incentives. The innovation effects of mergers depend on the trade-off between reduced competitive pressure and efficiency, spillover, or appropriability gains from mergers. In recent work, Federico et al. (2018) directly measure merger effects on innovation outcomes and find that the net effect is context-dependent. Studying Intel and AMD in microprocessors, Goettler and Gordon (2011) find that Intel would innovate faster without AMD, because competition there erodes the margin innovation earns; here competition enters the timing of release and propagation across markets, so removing a rival removes a stimulus and slows every firm. Their exercise removes a competitor outright—pure competition removal—while our merger also combines the firms' technologies, and the counterfactual varies how much of the combination survives. This paper contributes to the literature on market structure and innovation by modeling two margins of innovation—the release of new technologies and their propagation across downstream markets—that respond to firms' productive capability and market-specific technology stocks, on the one hand, and to changes in competitive pressure, on the other, and by applying this framework to GPUs as a GPT.

The remainder of the paper proceeds as follows. Section 2 describes the data and the motivating evidence on GPU innovation. Section 3 develops the model. Section 4 presents the estimation method, the estimates, and the model's fit. Section 5 evaluates the AMD–Intel merger.

2 Data

2.1 Industry Background and Data Sources

Graphics processing units are massively parallel processors that originated as accelerators for personal-computer graphics and became a general-purpose computing platform: the same architectural generation that a firm designs for PC graphics can subsequently be applied to data-center computation and to autonomous-vehicle platforms. Our unit of technology is the architectural *advancement*—a discrete technological capability such as programmable

shading, unified shaders, or tensor cores—rather than the hardware generation that carries it: a single generation may introduce several advancements, and successive advancements may refine the same underlying technology across generations. Commercial products implement these advancements in specific end-markets. The industry that designs these architectures is concentrated in three firms—NVIDIA, AMD, and Intel—which together account for effectively all GPU shipments throughout the sample period.[1] We study their innovation behavior across the three end-markets in which GPU architectures are applied: PC, DC, and AV.

We combine two panels. The first is a quarterly financial panel assembled from the firms' quarterly and annual filings, Forms 10-Q and 10-K: segment revenue, operating income by segment, research and development expenditure, operating expenses, and employment. The second is a quarterly technology-event panel that records, for every technology of every firm, its origin market, its *release* date, defined as the technology's first commercial application, and the date of each subsequent *propagation*, defined as its first application in an end-market it had not previously reached. A technology enters the panel when it introduces a distinct hardware capability, and every event is dated to its first documented commercial application in the relevant market. Sources are vendor announcements, product documentation, and contemporaneous trade coverage, recorded for every event.

2.2 Sample and Variable Construction

The estimation panel is quarterly, with firm-specific windows determined by the availability of segment reporting: NVIDIA is observed from 2008Q1, AMD from 2009Q1, and Intel from 2010Q1, all through 2023Q2. From the financial panel we construct three variables. Effort, a_t^f, is measured from R&D expenditure as the standardized quarterly growth of R&D spending,

[1] Quarterly unit-shipment data from Jon Peddie Research (Jon Peddie Research, 2023b). In the overall PC-GPU market (integrated and discrete), the three firms exhaust reported vendor shares in every quarter of the sample, with Intel supplying integrated-GPU architectures throughout. In the discrete add-in-board segment, NVIDIA and AMD were the only suppliers until Intel entered with its Arc A-series in 2022 (the A380 in mid-2022, the A770/A750 in 2022Q4); thereafter the three firms exhaust reported shares, with NVIDIA at 80.2% of add-in-board shipments at the end of the sample (Jon Peddie Research, 2023a; Shilov, 2023).

normalized so that $a_t^f = 1$ corresponds to the industry-average pace of R&D expansion.[2] R&D cost, quarterly R&D expenditure in dollars, fixes the scale of each firm's effort-cost function; revenue, total segment revenue, disciplines the model's revenue equation through its pooled mean and dispersion.[3]

From the event panel we construct, for each firm and market, the quarterly history of release and propagation events, the dates of first events and of market entry, and deployment status, and, for each market, the quarterly advance of its technology frontier; first-event dates are expressed as fractions of each firm's observation window for estimation.

2.3 Summary Statistics

Table 1 summarizes the panel. Its central contrast is between resources and technological output: Intel's quarterly R&D budget, $3.1 billion, is more than six times NVIDIA's or AMD's, yet over the full technology history Intel originates none of the panel's 22 underlying technologies. NVIDIA originates 10 outright and shares 3 more—the industry-founding advancements: rasterization and the programmable and unified shader—with AMD, which alone originates 2; the remaining 7 enter the industry from vendors outside the panel. A technology is originated once but generates many events: the 50 release and 25 propagation events inside the estimation windows include every later adoption of an advancement by rival firms and its carriage across markets. Activity within the windows follows the same ordering as origination: NVIDIA leads in both releases and propagations, and while every firm releases in every market, Intel's propagations are confined to the data center.

The rival-exposure correlations measure the co-movement of a firm's events with the recency-weighted flow of rivals' events of the same type in the same market: releases with rivals' releases, propagations with rivals' propagations. They are positive for all three firms and largest for the firms that originate least. Intel's release correlation (0.39) far exceeds

[2]We take the quarterly log-change in R&D expenditure, standardize it in the pooled panel, and rescale it to mean 1 and standard deviation 0.35.

[3]The model's revenue equation is log-linear (Section 3), so revenue is anchored by its average in logs—the pooled geometric mean, $3.47 billion—and its dispersion is likewise measured in logs, 0.34.

Table 1: The panel: firms, activity, and technology events.

	NVIDIA	AMD	Intel
Sample start (all end 2023Q2)	2008Q1	2009Q1	2010Q1
Quarters	62	58	54
Mean quarterly R&D ($B)	0.49	0.47	3.08
Release events	22	15	13
Propagation events	14	6	5
Release freq. per quarter (PC/AV/DC)	.13/.10/.13	.07/.03/.16	.13/.02/.09
Propagation freq. per quarter (PC/AV/DC)	.08/.06/.08	.05/.02/.03	.00/.00/.09
First AV entry	2015Q1	2023Q2	2022Q1
Rival-exposure corr. (release, propagation)	.13, .02	.26, .33	.39, .25

Notes. A release is a technology's first commercial application in its origin market; a propagation is its first application in another end-market. Event counts and frequencies are within each firm's estimation window. Frequencies are events per firm-quarter by market. Rival-exposure correlations measure the co-movement of a firm's events with the recency-weighted flow of its rivals' same-stage events in the same market. All three firms are active in PC and DC throughout their panels; AV is the only market entered in-sample.

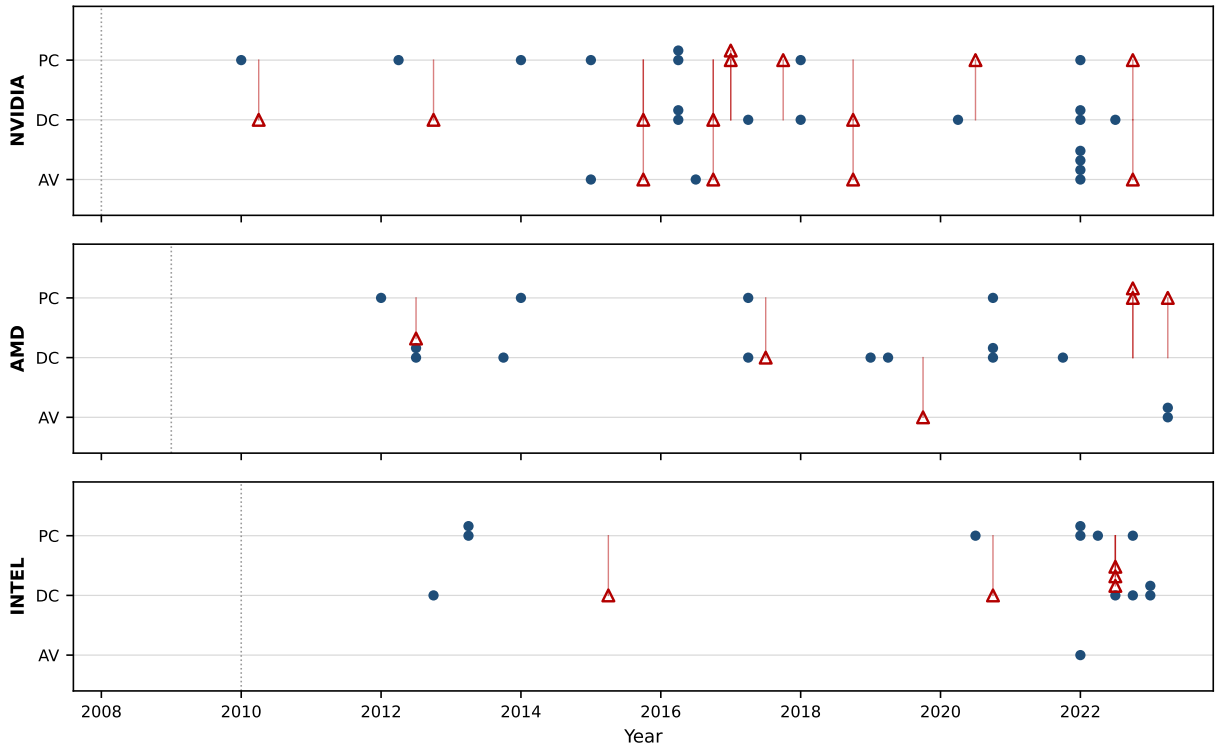
NVIDIA's (0.13), with AMD in between; on the propagation side AMD's is the largest (0.33), and NVIDIA's, whose propagations lead rather than follow, is near zero. Section 2.4 develops these patterns, together with the timing evidence in Figure 1, into five facts.

2.4 Motivating Evidence: GPU Innovation

Five features of Table 1 and Figure 1 organize the modeling choices in Section 3. First, a technology's life has two stages—release and propagation—and the second is both selective and delayed. It is selective because propagation follows half of releases, making cross-market application a central margin of a technology's life yet far from automatic. It is delayed because propagation arrives a median of three quarters after release (Figure 2), with an interquartile range of 1 to 4 quarters. The median lag is two quarters into the data center, three into PC, and four into autonomous vehicles. Propagation almost never coincides with release: of the thirty propagations with directly documented origin dates, only one occurs in its release quarter. The delay reflects the complementary adaptation a new application requires (David, 1990). Innovation and diffusion are therefore distinct margins with distinct timing, and the model treats them as such: a two-stage hazard, a release stage followed by a

propagation stage, with an estimated release-to-propagate onset lag (Sections 3.5–3.6).

Figure 1: GPU technology releases and cross-market propagations.

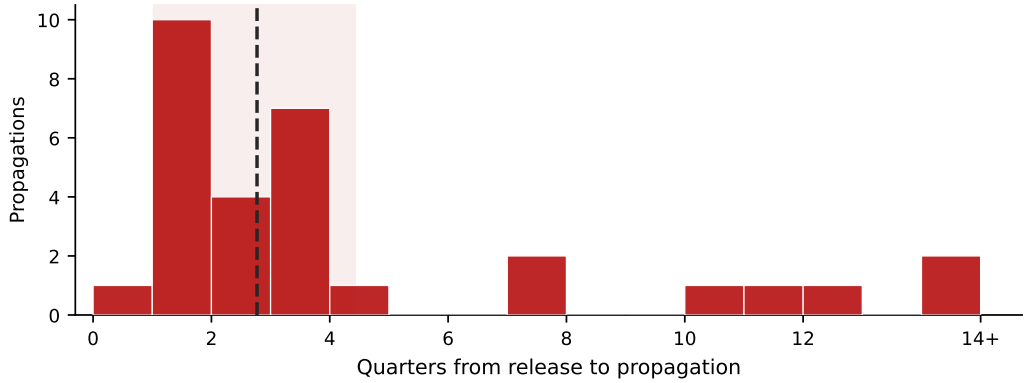


Notes. One panel per firm; lanes are the three end-markets (PC, DC, AV). *Legend*: blue filled circles (●) mark a technology's *release*, placed in its origin market at the release quarter; red open triangles (△) mark a propagation, placed in the destination market; the thin red vertical connector links each propagation to its origin lane, so the horizontal distance between a circle and the triangle it feeds is the release-to-propagation lag documented in the text. Events sharing a market and quarter are stacked vertically. Dotted vertical lines mark each firm's estimation-window start; all events shown fall within the windows and match the counts in Table 1.

Second, firms act when and where their rivals have just acted. Every rival-exposure correlation in Table 1 is positive, strongly for the followers and detectably for the leader, so a firm's releases and propagations cluster just after its rivals' recent same-stage activity in the same market. Timing co-movement of this kind is the signature of strategic response. It motivates rival-flow terms in both hazard stages—for the followers and for the leader—and an equilibrium in which all three firms best-respond to one another's projected release and propagation paths (Section 3.8).

Third, the industry has one technological leader and two followers. Origination is

Figure 2: Distribution of release-to-propagation lags.



Notes. The lag from a technology's release to its propagation into another end-market, in quarters. The figure covers all 30 propagations with directly documented origin dates over the full technology history; the estimation uses the 25 propagation events inside the estimation windows, which are limited by the coverage of the firms' financial reports. The dashed line marks the median (2.8 quarters); the shaded band is the interquartile range (1.0 to 4.4).

concentrated in NVIDIA, which originates or co-originates 13 of the panel's 22 technologies, against AMD's 5 and Intel's none; adoption runs the other way—Intel, inventing nothing, adopts 11. This asymmetry motivates leader-specific coefficients in the two hazards and the leader-to-follower spillover in market entry, and it disciplines the productivity ordering imposed in estimation (Section 4.1).

Fourth, markets differ in maturity and in which margins move. All three firms are active in PC and the data center from the start of their panels, so those extensive margins never move; autonomous vehicles is the one market entered in-sample, and the entries are staggered across eight years (2015Q1, 2022Q1, 2023Q2). Propagation, meanwhile, is most frequent into the data center, though the completed record shows technologies flowing into all three markets; AV events of any kind are scarce. Market heterogeneity of this size calls for market-specific hazard and entry intercepts, and it makes AV entry timing the extensive margin with the power to identify cross-firm differences.

Fifth, R&D input fluctuates from quarter to quarter, but deployment does not follow it. The volatility of quarterly R&D growth ranges from 4 percent (Intel) to 9 percent (NVIDIA and AMD), with top-decile quarterly changes of 8 to 15 percent; yet the correlation between

deployment and lagged effort is essentially zero for every firm (NVIDIA 0.05, AMD 0.02, Intel −0.04)—even the larger movements in R&D input leave no trace in contemporaneous deployment. This pattern is consistent with effort building a slow-moving capability stock that raises the hazards of both stages—the release of new technologies and their subsequent propagation into additional markets—rather than triggering events directly, which is precisely the model's capability mechanism (Section 3.5).

Together the five facts dictate the model's architecture: forward-looking firms choose costly R&D effort under private shocks (the fifth fact) that builds firm-specific capability (the third); capability, market conditions, and rivals' recent activity (the second) drive two sequential event hazards (the first) whose intercepts and entry margins differ by market (the fourth). Each estimated parameter class corresponds to one of these documented margins.

3 Model

We model a discrete-time, infinite-horizon dynamic game, with time indexed by t and a common discount factor $\beta \in (0, 1)$, in which a finite set of firms $f \in \mathcal{F}$ competes across a finite set of markets $m \in \mathcal{M}$.

In each period, firm f observes its state s_t^f and draws a private productivity shock ξ_t^f, then chooses R&D effort a_t^f. Effort is costly now and pays off in the future, building the firm's productive capability x_t^f (the firm's sole source of productivity differences) and, through it, the rate at which the firm enters markets and deploys technologies. Market entry (a firm's first deployment into a market) and subsequent deployment are not chosen directly but realize stochastically. The entry indicator $e_{t,m}^f \sim \text{Bernoulli}(p_{f,m}^e)$ and the deployment indicator $d_{t,m}^f \sim \text{Bernoulli}(p_{f,m}^d)$ have probabilities $p_{f,m}^e(s_t^f, a_t^f)$ and $p_{f,m}^d(s_t^f, a_t^f)$ that depend on effort and the current state. Capability accumulates stochastically with deployment. The firm therefore trades a deterministic effort cost against probabilistically realized entry, deployment, and capability gains. The structure is that of Ericson and Pakes (1995)—costly investment

stochastically advances a firm state that determines profits—with the state driving the timing of technology events across markets rather than market structure alone.

A deployment takes one of two forms: a *release*, when the firm puts a technology into its origin market, or a *propagation*, when an already-released technology propagates to another market. We track the two separately.

Notation. The auxiliary indices and laws of motion below are affine in their continuous inputs, and we suppress their coefficients: each block appears as a named function, name$(\cdot)$, subscripted by m where its coefficients vary by market and by f where any varies by firm. The same variable may enter several functions, once for each channel through which it acts. Explicit symbols are reserved for the reported structural objects: the hazard coefficients—notably the racing coefficients $\beta^{\text{rel}}_{rr,F}$ and $\beta^{\text{rel}}_{rr,L}$ and the pursuit coefficient β^{prop}_{r}—the onset center τ_0, and the productivity drift μ^f_x.

3.1 State and Actions

Every choice the firm makes and every event probability it faces in quarter t depends on its state,

$$s^f_t = \big(x^f_t,\ (z^f_{t,m})_{m\in\mathcal{M}},\ (A^f_{t,m})_{m\in\mathcal{M}},\ (D^f_{t-1,m})_{m\in\mathcal{M}},\ (n^{f,\text{rel}}_{t,m})_{m\in\mathcal{M}},\ (n^{f,\text{prop}}_{t,m})_{m\in\mathcal{M}} \big)$$

where x^f_t is the firm-level productive capability defined above and the remaining components are market-specific. $z^f_{t,m} \in [0,1]$ is firm f's accumulated technology stock in market m, which raises flow profit and, through a spillover, also the probability of entering markets $m' \neq m$.

The indicator $A^f_{t,m} \in \{0,1\}$ records whether the firm is active in market m and is irreversible once set. $D^f_{t-1,m} \in \{0,1\}$ is the previous-period deployment indicator: a firm that deployed in market m last period is likely to deploy again. $n^{f,\text{rel}}_{t,m} \in \mathbb{Z}_+$ and $n^{f,\text{prop}}_{t,m} \in \mathbb{Z}_+$ count the firm's own past releases in, and past propagations into, market m; the superscript names the event type. Rivals' activity enters the firm's problem not as a state variable but

as market-specific time paths taken as given: the recency-weighted flows of rivals' release and propagation events that shift the two hazards (Section 3.6), and, for followers, the leader's projected stock path in the entry index (Section 3.4).

3.2 Timing

Within quarter t, events resolve in four stages:

1. **Effort.** Firm f observes its state s_t^f, draws a private productivity shock ξ_t^f, and chooses effort a_t^f, which determines the entry and deployment probabilities $p_{f,m}^e$ and $p_{f,m}^d$.

2. **Entry/deployment.** For each inactive market, entry $e_{t,m}^f \sim \text{Bernoulli}(p_{f,m}^e)$ realizes and updates the activity indicator $A_{t,m}^f$. In each active market, deployment $d_{t,m}^f \sim \text{Bernoulli}(p_{f,m}^d)$ realizes. Deployment builds its technology stock $z_{t,m}^f$ and earns revenue.

3. **Release and propagation.** In each origin market, unreleased technologies are released at the release hazard (3). In each destination market, released technologies become progressively eligible to propagate as they mature, through the endogenous onset ramp (5), and propagate each quarter with the probability given by the propagation hazard (4). Each release and propagation is a dated technology event. Together these events form the firm's own event counts and, when fired by its rivals, the recency-weighted rival flows—the count of rival same-stage events in the market over the preceding eight quarters, divided by four so that a flow of one corresponds to a rival event every two quarters—that enter both hazards.

4. **Transition to $t+1$.** The technology stock $z_{t,m}^f$ makes a discrete jump of size Δ_m only under deployment, with a probability that rises in capability and deploy intensity. Capability x_{t+1}^f evolves deterministically from $(x_t^f, \bar{z}_t^f, a_t^f)$ and the firm's accumulated propagation counts (where $\bar{z}_t^f$ is the firm's average stock across markets) plus a Gaussian shock, with a single additive bump Δ_m^e for markets entered for the first time this period. The game then repeats at $t+1$ from the updated state s_{t+1}^f.

Although the model resolves stages 2 and 3 with separate mechanisms, they are two readings of the same underlying actions, the firm putting technologies into markets: the model reads those actions as firm-market activity (entry and deployment) that builds the stock $z^f_{t,m}$ and earns revenue, and as dated technology events (release and propagation) that form the event counts and rival flows entering the hazards.

Figure 3 summarizes the model as two interlocking engines. Panel (a) traces the accumulation engine: effort builds capability, capability drives entry and deployment, and deployment builds technology stocks that feed back into capability through $\bar{z}^f$—so success in one market raises the propensity to succeed in the others, the self-reinforcing logic of a general-purpose technology. Panel (b) traces the competition engine: each release and propagation enters the firm's own event record and its rivals' recent-activity flows, which shift every subsequent hazard, and the response of a firm's releases and propagations to rivals' recent releases and propagations is the object the model is built to recover. The remainder of this section develops each component.

3.3 Flow Profit

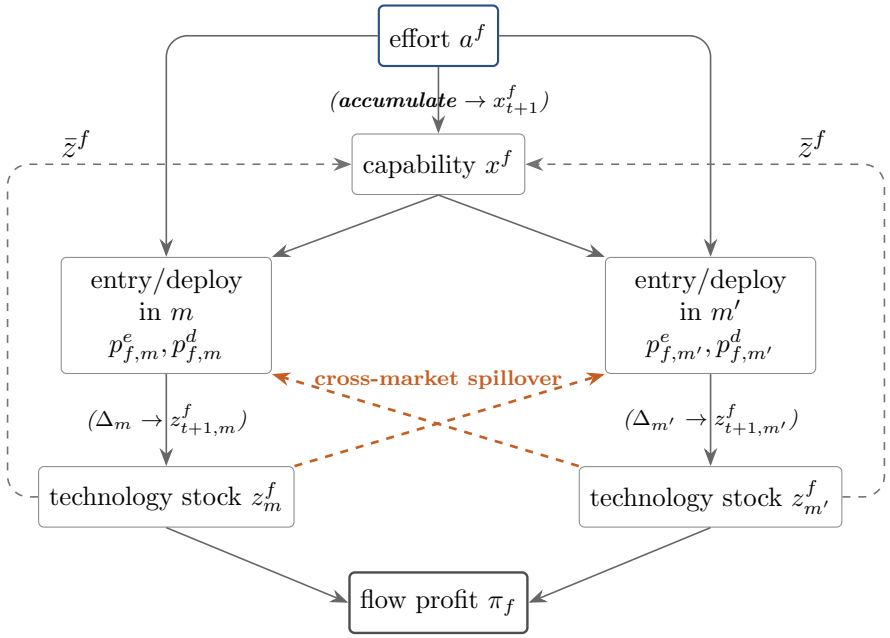
Per-period flow profit decomposes into operating revenue earned on market activity and performance revenue on a new deployment:

$$\begin{aligned}\pi_f(s^f_t, a^f_t, \xi^f_t) \;=\; & \sum_{m\in\mathcal{M}} \Big[\, p^A_{f,m} \cdot R^{\text{op}}_{f,m}(x^f_t, z^f_{t,m}) + p^d_{f,m} \cdot R^{\text{perf}}_m(a^f_t) \,\Big] \\ & - C_{\text{op}} \sum_m A^f_{t,m} - c(a^f_t) - \rho_d \sum_m p^d_{f,m} + \xi^f_t \sum_m p^d_{f,m}.\end{aligned}$$

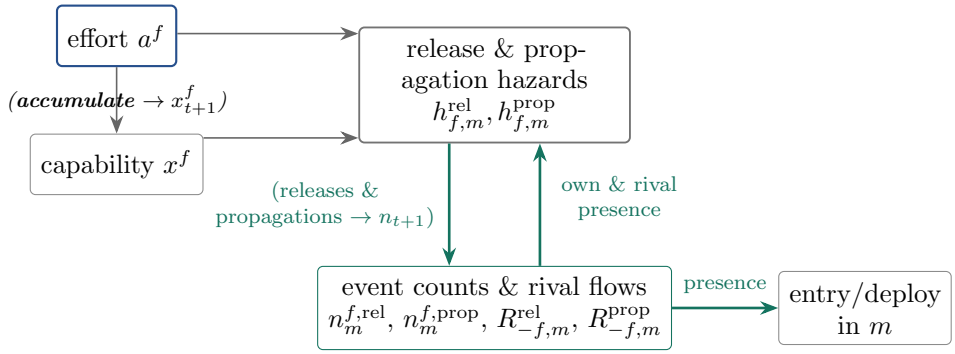
The first bracketed block is *operating* revenue, earned on activity: $R^{\text{op}}_{f,m}(x^f_t, z^f_{t,m})$ is affine in capability x^f_t and the market technology stock $z^f_{t,m}$, with a market intercept and a firm-specific shift (legacy revenue and pricing power, distinct from productivity). This block is scaled by $p^A_{f,m} = A^f_{t,m} + (1 - A^f_{t,m})\, p^e_{f,m}$, the probability of being active in market m (entry and deployment probabilities $p^e_{f,m}$, $p^d_{f,m}$ are defined in Section 3.4). The second block is

Figure 3: **Two "GPUs as GPTs" engines: capability accumulation and competition.**

(a) Capability, stock, and the GPT spillover



(b) Cross-market competition through the hazards



Notes. A causal map whose nodes are firm-level variables. Arrows out of a stock (x^f, z^f_m, n^f_m) read its current level; only the laws of motion, tagged ($\cdot \to t{+}1$), write next-period state. *Panel (a).* Within the period, effort a^f and current capability x^f set the entry/deploy probabilities p^e, p^d in both markets; entry/deploy activity earns revenue (flow profit π_f) and, via the jumps Δ_m, builds next period's technology stocks. Across periods, capability follows one law of motion in effort and the firm's *average* technology stock $\bar{z}^f$: today's effort (*accumulate*) and average stock (the dashed $\bar{z}^f$ feedback) build next period's capability—a learning-by-doing channel that spans markets because $\bar{z}^f$ pools them. Not drawn: first entry into a market gives capability a one-time bump Δ^e_m. The orange arrows are the GPT spillover: technology stock built in one market raises the propensity to enter the others. *Panel (b)* uses the same effort–capability spine: effort and capability now shift the release and propagation hazards, whose firings build the firm's own event counts and its rivals' recent event flows; these are read back as own presence and rivals' recent activity, feeding both hazards—rival releases accelerate a firm's own releases (racing), rival propagations its propagation (pursuit).

performance revenue, earned on a new deployment: $R_m^{\text{perf}}(a_t^f)$ is affine in effort a_t^f.

On the cost side, C_{op} is the operating cost per active market, $c(a_t^f)$ is the quadratic effort cost, and ρ_d is a per-deployment cost. Separately, the private productivity shock $\xi_t^f \sim \mathcal{N}(0, \sigma_{\xi,f}^2)$ shifts profit, scaling with deployment intensity.

3.4 Entry and Deployment Probabilities

Entry and deployment are not chosen but realize stochastically, with probabilities driven by effort, capability, and two cross-market spillover channels. For an inactive market ($A_{t,m}^f = 0$), entry occurs with probability

$$p_{f,m}^e = \Lambda\big(b_{f,m}^e / \sigma_{e,m}\big),$$

where $\Lambda(\cdot)$ is the standard logistic CDF and $\sigma_{e,m} > 0$ is the estimated market-specific scale of the private entry shock $\varepsilon_{f,m}^e \sim \text{Logistic}(0, \sigma_{e,m})$. The linear entry index $b_{f,m}^e$, which summarizes the firm's propensity to enter, is

$$\begin{aligned} b_{f,m}^e \;&= I_m^e\big(x_t^f, z_{t,m}^f, a_t^f\big) + \Theta^{\text{prop}}\big(n_{t,m}^{f,\text{prop}}\big) \\ &\quad + \Gamma_f^z\Big(\textstyle\sum_{m' \neq m} z_{t,m'}^f\Big) + \Gamma_m^L\big(z_{t,m,L}\big)\,\mathbf{1}[f \notin L] + \mathbf{1}[m = \text{AV}]\,\delta_{f,\text{AV}}^{\text{shift}}(y). \end{aligned}$$

The baseline index $I_m^e(x_t^f, z_{t,m}^f, a_t^f)$ is affine in capability x_t^f, the own-market stock $z_{t,m}^f$, and effort a_t^f, plus a market entry intercept. The function $\Theta^{\text{prop}}(\cdot)$ loads the firm's own cumulative propagation count into m.[4]

The cross-market z-spillover $\Gamma_f^z(\cdot)$ loads firm f's stock accumulated in its other markets, $\sum_{m' \neq m} z_{t,m'}^f$. The leader-stock spillover $\Gamma_m^L(\cdot)$ acts on $z_{t,m,L}$, the leader firm's accumulated stock in market m, taken as given by followers (Section 3.8) and capturing market development. Receipt is restricted to followers by the indicator $\mathbf{1}[f \notin L]$, so the leader does not spill over

[4]Earlier specifications also carried a rival-presence heuristic in the entry and deployment indices, by which a follower read rival presence off its own stock through a calibrated map. The term is quantitatively inert—setting it to zero changes no simulated moment by more than 2×10^{-4}—and is dropped from the reported model; rival activity operates through the event flows in the two hazards.

onto itself. The term $\delta^{\text{shift}}_{f,\text{AV}}(y)$ is a logit shift on AV entry that switches sign at a fixed onset date: before the firm's documented platform introduction it lowers the entry index by the estimated magnitude, and from the introduction onward it raises the index by the same amount.[5] On a firm's first entry into market m, capability receives an estimated one-time increment $\Delta^e_m \geq 0$: $x^f_t \leftarrow \min(x^f_t + \Delta^e_m, x_{\max})$. The increment is not a driver of that entry's probability but instead a capability gain *from* entering. Because deployment, the release and propagation hazards, and entry into the firm's *other* markets all rise in x^f_t, this gain lets a firm build on a foothold in one market rather than stay stuck at low capability.

In each active market, the deployment probability is

$$p^d_{f,m} \;=\; D^f_{t-1,m}\,\delta + (1 - D^f_{t-1,m})\,\Lambda\big(b^d_{f,m}/\sigma_{d,m}\big),$$

where $D^f_{t-1,m} \in \{0,1\}$ indicates whether the firm deployed in m last period: a firm that deployed last period redeploys with the deploy-persistence probability δ, while one that did not deploys with probability $\Lambda(b^d_{f,m}/\sigma_{d,m})$. Here $\Lambda(\cdot)$ is the standard logistic CDF and $\sigma_{d,m} > 0$ the deploy-index scale. The deploy index reuses the affine baseline of the entry index, $I^d_m(x^f_t, z^f_{t,m}, a^f_t)$, and adds a firm-specific own-propagation channel: $b^d_{f,m} = I^d_m(x^f_t, z^f_{t,m}, a^f_t) + \Theta^d_f\big(n^{f,\text{prop}}_{t,m}\big)$. It carries none of the entry index's cross-market or leader-stock channels, and no calibrated-date shift, since those forces govern the extensive margin; it responds to the firm's own readiness: capability, own-market stock, effort, and accumulated own-market propagation.

[5]The onset dates are the firms' documented AV-platform introductions—NVIDIA in 2015, AMD in 2019, and Intel in 2022; the estimated Intel and NVIDIA magnitudes appear in Table 3. The dates themselves are not estimable: each firm enters autonomous vehicles at most once, so the objective carries no information about a change point between adjacent quarters, while the shift magnitudes are identified from entry timing against the fixed windows.

3.5 Capability, Technology-Stock Dynamics, and Propagation

Productive capability evolves linearly in effort, the cross-market stock feedback, and the first-entry bump:

$$x^f_{t+1} = G_f\big(x^f_t,\, a^f_t,\, \bar{z}^f_t,\, \textstyle\sum_m n^{f,\text{prop}}_{t,m}\big) + \mu^f_x + \varepsilon^x_t + \textstyle\sum_m \Delta^e_m \big(A^f_{t,m} - A^f_{t-1,m}\big), \tag{1}$$

where $\varepsilon^x_t \sim \mathcal{N}(0, \sigma^2_x)$, $\bar{z}^f_t$ is the firm's average stock across its M markets, and $G_f(\cdot)$ is affine, carrying a firm subscript because its effort loading is firm-specific; its other coefficients (capability persistence, the average-stock feedback, the propagation-count loading, and a constant depreciation) are common across firms. The drift constant μ^f_x is kept explicit: it carries each firm's productivity level and pins down the ordering. The final term applies the first-entry bump: since $A^f_{t,m}$ is irreversible (Section 3.4), the difference $A^f_{t,m} - A^f_{t-1,m}$ equals 1 in the single quarter the firm first enters market m and 0 in every other quarter, so capability receives the one-time increment Δ^e_m exactly once per market entered. Each market stock advances by a discrete jump at the transition from t to $t+1$,

$$z^f_{t+1,m} = z^f_{t,m} + \Delta_m J^f_{t,m}, \qquad \mathbb{P}(J^f_{t,m} = 1) = p^d_{f,m}\, \Lambda\big(\Psi_m(x^f_t, p^d_{f,m})\big), \tag{2}$$

where the jump $J^f_{t,m} \in \{0,1\}$ adds $\Delta_m > 0$ to the stock, $\Lambda(\cdot)$ is the logistic function, and $\Psi_m(x^f_t, p^d_{f,m})$ is affine in capability and deploy intensity, with a market intercept. The jump is contingent on the period's deployment[6]: only a firm that deploys in m can accumulate stock, so it requires both that the firm deploys, with probability $p^d_{f,m}$, and that the deployment then translates into stock, with probability $\Lambda(\Psi_m(x^f_t, p^d_{f,m}))$. Deploy intensity $p^d_{f,m}$ thus enters twice, as the multiplicative gate and inside Ψ_m. This probability rises in capability x^f_t and deploy intensity, so a more capable firm is more likely to turn a given deployment into stock. Both laws of motion are truncated to their state grids, $x^f_t \in [0, x_{\max}]$ and $z^f_{t,m} \in [0,1]$.

[6]Equation (2) is the leader's stock law of motion. For followers the stock instead jumps with calibrated probability $r_m(1 - z^f_{t,m})$: catch-up is fast when the stock is low and fades as it approaches the common ceiling, and requires neither capability nor deployment; the stock-to-capability feedback below applies to all firms.

Capability and the market-specific stocks form a single self-reinforcing process through two channels. The first is deliberate R&D: effort raises $p^e_{f,m}$ and $p^d_{f,m}$ contemporaneously and, by accumulating into x^f_{t+1}, raises them in all future periods. The second runs from accumulated stock back to capability: stock built in any one market enters x^f_{t+1} through $\bar{z}^f_t$, so past deployment raises future productive capability and thereby speeds deployment and stock-building in every market. This stock-to-capability feedback is a *learning-by-doing* mechanism (Arrow, 1962; Besanko et al., 2010), operating alongside the R&D-effort channel rather than in place of it. Because $\bar{z}^f_t$ averages stock across markets, the learning is cross-market: doing in one downstream market raises the capability the firm brings to the others, reinforcing the GPT character of GPUs.

3.6 Two-Stage Propagation Hazard

Each technology has a single origin market. *Release* (Stage 1) puts the technology into its origin market; *cross-market propagation* (Stage 2) propagates an already-released technology into a market other than its origin. Each stage fires a discrete event in the market it acts on: releases accumulate into the firm's release count $n^{f,\text{rel}}_{t,m}$ and propagations into its own-propagation count $n^{f,\text{prop}}_{t,m}$. Rivals' dated events enter the hazards separately, through the recency-weighted flows defined below. The two stages have separate hazards. Rather than gating Stage 2 by an exogenous lag, we let a released technology's readiness to propagate build *endogenously* with its tenure (the onset ramp (5) below). The per-quarter release hazard is

$$h^{\text{rel}}_{f,m} = \sigma\big(\alpha^{\text{rel}}_{f,m} + \beta^{\text{rel}}_a a^f_t + \beta^{\text{rel}}_x x^f_t + \beta^{\text{rel}}_n n^{f,\text{rel}}_{t,m} + \beta^{\text{rel}}_{rr,\tau(f)} R^{\text{rel}}_{-f,m,t}\big), \qquad \alpha^{\text{rel}}_{f,m} = \alpha^{\text{rel}}_m + \delta^{\text{rel}}_{f,m}, \quad \delta^{\text{rel}}_{\text{INTEL},m} \equiv 0, \tag{3}$$

and the destination-market propagation hazard is

$$h^{\text{prop}}_{f,m} = \sigma\big(\alpha^{\text{prop}}_m + \beta^{\text{prop}}_a a^f_t + \beta^{\text{prop}}_x x^f_t + \beta^{\text{prop}}_n n^{f,\text{prop}}_{t,m} + \beta^{\text{prop}_r} R^{\text{prop}}_{-f,m,t}\big). \tag{4}$$

In both equations $\sigma(\cdot)$ is the standard logistic function, so each hazard is a per-quarter probability in $(0, 1)$. The own-history regressors are cumulative counts: $n_{t,m}^{f,\text{rel}}$ is the firm's past releases in origin market m and $n_{t,m}^{f,\text{prop}}$ its past propagations into destination market m. The rival regressors are recency-weighted flows, not stocks: $R_{-f,m,t}^{\text{rel}}$ (respectively $R_{-f,m,t}^{\text{prop}}$) is the count of rivals' release (propagation) events in market m over the preceding eight quarters—roughly one GPU architecture cycle—scaled by one fourth and pooled across both rivals, so a firm responds to rivals' recent moves rather than to their accumulated history. The asymmetry is deliberate. The rival term measures pressure, which is current: a decaying flow rises when rivals move and subsides when they stop, so the model can represent a race that ends—with a cumulative rival stock, measured rivalry could only ever grow. The own term measures experience: accumulated releases and propagations proxy a demonstrated ability to place technologies in a market, and experience does not decay. The distinction also does econometric work: cumulative counts trend upward for every firm and are confounded with calendar time, whereas the rival flow varies at event frequency—exactly the variation the rival-exposure moments of Section 4.1 are built to match.

The release-stage rival coefficient is firm-type-specific, $\tau(f) \in \{F, L\}$: the two followers share $\beta_{rr,F}^{\text{rel}}$ and the leader carries its own $\beta_{rr,L}^{\text{rel}}$, so release rivalry is two-sided by specification rather than imposed one-way. The propagation-stage rival coefficient β_r^{prop} operates for the followers; the leader's propagation-stage rival response, and the cross-stage terms the general specification admits (rival propagations in the release hazard, rival releases in the propagation hazard), carry no statistical support and are set to zero in the reported specification (Section 4.2).

The rival coefficients are the paper's central objects. In the release hazard, $\beta_{rr,F}^{\text{rel}}$ and $\beta_{rr,L}^{\text{rel}}$ measure release racing: how strongly a firm's release hazard rises just after rivals' recent releases in the same market, holding fixed its own release history. In the propagation hazard, β_r^{prop} measures competitive pursuit across markets: how the recent flow of rivals' propagations into m shifts firm f's own propagation hazard into that market, holding fixed f's own prior

propagations there ($n_{t,m}^{f,\text{prop}}$).

The propagation intercept, by contrast, is market-level (α_m^{prop}) with no per-firm deviations: cross-firm differences in propagation timing run through capability x_t^f and the activity counts, not through firm-specific intercepts.[7]

In the data, a technology does not propagate immediately after its release: propagation typically begins several quarters later (Figure 2), a delay that a propagation hazard ignoring the technology's tenure cannot produce. The delay is the adoption lag of the diffusion literature—a technology spreads only as it is adapted to each new use (Comin and Hobijn, 2010). We estimate this delay, τ_0. A released technology of tenure τ (quarters since release) counts as ready to propagate with weight

$$w(\tau) = \sigma\big(\kappa_\tau\,(\tau - \tau_0)\big), \tag{5}$$

which is near 0 when the technology is young and near 1 once it is mature. The propagation hazard (4) is scaled by the average of $w(\tau)$ across the firm's released technologies—the share of them ready to propagate. The center τ_0 is the tenure at which readiness typically arrives; the steepness $\kappa_\tau > 0$ is fixed and sets only how sharply readiness turns on.

3.7 Value Function and Policy

Each firm chooses effort to maximize expected discounted profit. The value function solves

$$V_f(s,\xi) \;=\; \max_{a\in[a_{\min},a_{\max}]} \Big\{\pi_f(s,a,\xi) + \beta\,\mathbb{E}_{s'|s,a}\,V_f(s',\xi')\Big\},$$

where the inner expectation integrates over the entry and deployment draws, the stock jumps (2), and the capability shock in (1), and the next-period shock ξ' is integrated by Gauss–Hermite quadrature. We solve the firm's problem by rolling-horizon backward induction. The

[7] Per-firm propagation deviations are not identified in this panel: they improve propagation timing only by driving two effort-shock variances to their bounds, and with the variances held interior the improvement disappears and the productivity ordering breaks. We therefore keep the propagation intercept market-level.

firm's projected release and propagation dates—projected from the two hazards evaluated along the rival flow paths, so anticipated rival activity shifts the firm's own expected event calendar— partition the next $\hat{H}$ quarters into event-free blocks; within each block of length T_i the value is computed by ordinary backward induction, and the blocks are chained by $V_i(T_i, \cdot) = V_{i+1}(0, \cdot)$, solving the last block first. The terminal value of the final block is the stationary myopic-policy value (the continuation beyond the foresight horizon $\hat{H}$, defined next). The procedure is re-run each quarter as the firm re-projects events—hence "rolling-horizon."

The foresight horizon $\hat{H}$ is the number of future quarters the firm looks ahead; beyond $\hat{H}$, the continuation value reverts to the stationary value under the myopic policy. $\hat{H}$ is set identified: varying it alone over $[1, 60]$ quarters, with the other parameters held fixed, moves the estimation objective by a fraction of one simulation standard error, so the data fit about equally well at any horizon in this range and single out none.

Because the objective is nearly flat across the set, we do not estimate $\hat{H}$ but instead *fix* it at a single interior value ($\hat{H} \approx 21.5$ q) and verify by an invariance sweep that the reported estimates do not change as $\hat{H}$ ranges over the set. This mirrors the standard treatment of the discount factor in dynamic discrete choice and dynamic games, which is generically not point-identified from choice data and is therefore fixed rather than estimated (Rust, 1987; Magnac and Thesmar, 2002). Fixing $\hat{H}$ is thus a restriction on a set-identified parameter that the data do not pin down, not an external calibration.

The leader role carries two asymmetries. First, the leader-stock spillover runs one way: the leader's accumulated stock $z_{t,m,L}$ enters only followers' entry indices, through Γ_m^L, so the leader's build-out develops markets for the followers but not conversely. Second, the rival-flow responses are leader-specific: the leader carries its own release-racing coefficient $\beta_{rr,L}^{\text{rel}}$—the leader is not oblivious to the followers. The leader's stock path is a different object from the rival flows: R^{rel} and R^{prop} record rivals' recent discrete events, while $z_{t,m,L}$ is a continuous stock, projected over calendar time and taken as given by followers (Section 3.8),

capturing the development of the market that followers then enter. Because $z_{t,m,L}$ varies over calendar time, the follower's problem is non-stationary within the panel and stationary beyond. Table 2 collects the rival channels: for each, the firms that respond, the rival object entering the hazard or index, its treatment in estimation and in counterfactuals, and the associated parameters.

Table 2: Rival channels: who responds, through what, and how rival activity is treated.

Channel	Admitted for	Rival object	Treatment of rival activity	Parameters
Release hazard (3): rival releases	All firms	Recency flow $R^{\text{rel}}_{-f,m,t}$	Rivals' release paths taken as given: the observed history in estimation (beliefs correct on the path by construction); mutually consistent projections in counterfactuals	$\beta^{\text{rel}}_{rr,F}$, $\beta^{\text{rel}}_{rr,L}$
Propagation hazard (4): rival propagations	All firms	Recency flow $R^{\text{prop}}_{-f,m,t}$	Same construction as above	β^{prop}_{r}; leader's coefficient set to zero (Section 4.2)
Cross-stage terms in both hazards	All firms	Recency flows	Same construction as above	Set to zero: no statistical support (Section 4.2)
Entry index: leader-stock spillover	Followers only	Leader's stock $z_{t,m,L}$	The leader's projected build-out path, anticipated along the follower's planning calendar and taken as given	$\gamma^{\text{leader}}_{\text{AV}}$; DC structurally unidentified, dropped
Leader flow profit: strategic deployment cost	Leader only	None (no rival state tracked)	No belief: a constant competitive cost per deployment	cost scale

Notes. "Admitted for" describes the model's structure. Parameters are estimated unless noted as set to zero; values are in Table 3. Recency flows are pooled counts of rival same-stage events over the trailing eight quarters, scaled by one fourth. The belief column states the information structure: on the hazard channels firms take rivals' event paths as given—observed play in estimation, the mutually consistent fixed point in counterfactuals (Section 3.8); on the entry margin, followers track the leader's projected path explicitly, while the leader tracks no rival state.

3.8 Equilibrium and the Treatment of Rivals

Firms are linked through paths of rival activity that each takes as given, not through any contingency of strategies on rivals' particular realized draws—realized releases and propagations are stochastic outcomes of the two hazards throughout the model. The rival path has one construction in each exercise. In estimation it is the rivals' observed event history: each firm best-responds to the play its rivals actually realized—the standard two-stage treatment of rivals in empirical dynamic games (Bajari et al., 2007)—so beliefs are correct along the observed path by construction, and the rival coefficients are identified from the covariation of a firm's events with the path of its rivals' releases and propagations (Section 4.1).

In counterfactuals, where observed rival play does not exist, each firm instead forecasts its rivals' release and propagation timing from the same hazards that generate those events, evaluated along the rivals' capability, effort, and event histories, and the forecasts are determined jointly: all firms' projected dates are iterated against one another, the leader's included, until each firm's projected calendar is a best response to its rivals'—a fixed point in event paths, in the restricted-information tradition of oblivious and moment-based equilibrium (Weintraub et al., 2008; Ifrach and Weintraub, 2017). Given its rival path, a firm's problem is a single-agent dynamic program over its own state: the path enters its two hazards as the recency-weighted flows $R^{\text{rel}}_{-f,m,t}$ and $R^{\text{prop}}_{-f,m,t}$ in (3)–(4) and, for followers, as the leader's projected stock path in the entry index. What we do not compute is a Markov-perfect equilibrium in strategies: rivals enter as deterministic time paths rather than state-contingent policies, so the fixed point is in paths rather than in policy functions. These two objects exhaust the rival channels: firms carry no other beliefs about rivals.

The solution concept sits between two poles of the literature. The oblivious element is the leader's stock path $z_{t,m,L}$ (Weintraub et al., 2008; Benkard et al., 2015): a deterministic trajectory, projected from the leader's own problem and taken as given by followers, with no feedback from follower play. The event side asks more of firms than obliviousness: the

forecast calendars are time-varying and iterated until each firm's is a best response to its rivals', so event rivalry is mutually consistent rather than taken as given. In a three-firm industry this matters, because a single rival event moves the recency flow by the order of its mean. The concept demands less than a Markov-perfect or perfect Bayesian equilibrium, in which strategies are contingent on the realized joint state and beliefs update by Bayes' rule. That concept is not attainable in this class of models: each firm's discretized state (capability, three per-market technology stocks, activity indicators) has roughly 900 points, so strategies over the three-firm joint state require a fixed point on $900^3 \approx 7 \times 10^8$ states—and because a rival's capability is latent, genuinely Bayesian play would further require filtering beliefs over it from discrete event histories, a state space of distributions.

Our middle ground—single-agent dynamic programs against mutually consistent, policy-consistent event paths, with beliefs correct along the path—follows the logic of restricted-information equilibrium concepts developed precisely for such settings: the moment-based Markov equilibrium of Ifrach and Weintraub (2017), in which firms condition on a low-dimensional summary of rivals rather than the full industry state, and the experience-based equilibrium of Fershtman and Pakes (2012), in which optimality is required only on outcomes players actually experience. In estimation the concept is on its firmest ground: conditioning each firm on rivals' realized play is the standard two-stage treatment of dynamic games (Bajari et al., 2007), and beliefs are correct by construction because they are the data. In counterfactuals the concept is unchanged—each firm best-responds to taken-as-given rival paths consistent with its rivals' behavior—but the consistent path is no longer observable and must be computed, by iterating all firms' projections to the fixed point, as in counterfactual analyses built on path-based equilibrium concepts (Benkard et al., 2015). Taking rival paths as given rather than solving for state-contingent rival strategies is the counterfactual solution's only approximation; it is accurate when behavior loads on a firm's own capability and event history rather than on anticipation of rivals' future moves. The flatness of the fit in the foresight horizon $\hat{H}$ provides direct evidence for this condition: if behavior barely responds to

how far ahead the firm looks, anticipated future events—rivals' moves included—contribute little to it.

4 Estimation and Results

4.1 Two-Step Method of Simulated Moments

The structural parameter vector $\theta \in \Theta \subseteq \mathbb{R}^P$ is estimated to match a vector of $J = 80$ empirical moments in four families. At the firm level we match the mean and dispersion of effort, the mean cost, the deploy–effort lag correlation, and the normalized quarter of first entry. At the firm-by-market level we match the entry and deploy means and, where the data identify them, the release and propagation frequencies and the normalized quarters of first release and first propagation. At the market level we match the per-market frequency of technology-stock jumps: the rate at which firms' stocks $z^f_{t,m}$ advance (Eq. (2)), pooled across firms within each market. Finally, we match six rival-exposure co-movements: for each firm, the covariation of its release and propagation events with the cumulative path of its rivals' releases and propagations. These co-movements identify the rival-response coefficients in the two hazards. Table 11 in Appendix D verifies this claim directly with Andrews–Gentzkow–Shapiro sensitivities (Andrews et al., 2017). Appendix B lists all 80 moments with their data values, and Appendix A collects the calibrated constants and the pre-panel state.

Let $g(\theta) = \frac{1}{S}\sum_{s=1}^{S} g(\theta, \omega_s)$ be the simulated-moment vector averaged over S Monte Carlo replications, with empirical counterpart g^{data}. We estimate $\hat{\theta}$ by two-step method of simulated moments (MSM) (McFadden, 1989; Pakes and Pollard, 1989), minimizing $Q(\theta; W) = \big(g(\theta) - g^{\text{data}}\big)^\top W \big(g(\theta) - g^{\text{data}}\big)$. The first step uses a diagonal weight matrix. The second uses the efficient weight $\hat{W} = \hat{\Omega}^{-1}$ (Hansen, 1982), where $\hat{\Omega}$ is the full covariance matrix of the simulated moment vector across 100 panel draws. In both steps the objective is minimized by Powell's derivative-free search from multiple starting values. Appendix C

details the computation: the state grids, the quadrature, and the solution of the dynamic program.

$\hat{\Omega}$ is model-implied—simulated at the first-step estimate—because the single observed panel cannot deliver a data-based estimate of an 80×80 moment covariance. With $J = 80$ moments and 100 draws the sample covariance is nearly singular, and a weight built on its raw inverse would load on moment combinations whose variance is estimated near zero, the small-sample pathology of efficient GMM documented by Altonji and Segal (1996). We therefore shrink before inverting: the correlation matrix is replaced by $(1 - \lambda)\hat{R} + \lambda I$—off-diagonal correlations pulled toward zero, each moment's variance untouched—with the intensity set by the analytic rule of Schäfer and Strimmer (2005), building on Ledoit and Wolf (2004), rather than tuned; on our draws the rule yields $\hat{\lambda} = 0.51$.[8] Before shrinkage nine eigenvalues of $\hat{R}$ are exactly zero, traceable to eight moments that are constant across draws and one exact duplicate; after shrinkage the eigenvalues lie in $[0.51, 2.88]$, a condition number of 5.7. A trimmed re-estimation that removes the degenerate moments at the source and rebuilds $\hat{\Omega}$ on 1,000 draws, reported at the end of this subsection, confirms the estimates do not depend on them.

Inference proceeds in three layers, whose joint conclusion is that the standard errors in Table 3 hold with every estimated parameter free; Appendix D details each layer. First, the standard errors in Table 3 are two-step-MSM sandwich estimates at $\hat{\theta}$, computed with a numerical Jacobian (central differences under common random numbers) and conditional on the non-reported parameters held at their estimated values (cf. Cocci and Plagborg-Møller, 2025).[9] Conditioning is a requirement of the objective's shape, not a convenience: it is nearly flat in several auxiliary directions, and an unconditional sandwich would invert that

[8] The intensity is the closed-form minimizer of the estimated mean-squared error of the shrunk correlation matrix: the ratio of the sampling variance of the off-diagonal correlations to their squared magnitude, clipped to $[0, 1]$. It involves no tuning parameter. At $\lambda = 0$ the raw correlations are kept; at $\lambda = 1$ they are deleted and $\hat{W}$ reduces to diagonal inverse-variance weighting.

[9] The variance is the general sandwich $(1+1/S)\,(\hat{G}^\top \hat{W} \hat{G})^{-1} \hat{G}^\top \hat{W} \hat{\Omega} \hat{W} \hat{G}\,(\hat{G}^\top \hat{W} \hat{G})^{-1}$ with $S = 12$ simulation replications, evaluated at the estimation weight $\hat{W}$. Because $\hat{W}$ is a shrunk inverse rather than $\hat{\Omega}^{-1}$ itself, the efficient-GMM shortcut $(\hat{G}^\top \hat{\Omega}^{-1} \hat{G})^{-1}$ does not apply.

near-zero curvature, contaminating every reported standard error with noise from directions the moments barely measure; the second and third layers below verify that conditioning hides no uncertainty.

The reported specification first fixes the parameters the moments cannot identify: a rank analysis of the weighted moment Jacobian at $\hat{\theta}$ finds nine parameters—all auxiliary constants, none reported in Table 3—that have no effect on any moment, and these are held at their estimated values as normalizations. The Jacobian of the remaining 71 estimated parameters has rank 70 against the 80 moments: one flat combination remains, mixing the leader spillover $\gamma_{\text{AV}}^{\text{leader}}$ with the auxiliary entry and flow intercepts it trades off against on the sparse AV entry margin—the same weak identification Table 3 flags for that coefficient. No other reported coefficient loads on the flat direction. The curvature of the objective remains weak in nine directions; these involve auxiliary parameters and, among the reported coefficients, only the same $\gamma_{\text{AV}}^{\text{leader}}$—not the hazard coefficients, and not the propagation-stage rival coefficient. The capability drifts μ_x^f are estimated subject to the leader–follower productivity ordering documented in Section 2.4, imposed through a penalty that maintains a minimum margin between adjacent firms.

Second, a one-step parametric bootstrap (Andrews, 2002)—500 synthetic panels are simulated at $\hat{\theta}$, and on each the estimate is updated by a single Newton step of the MSM first-order condition, reusing the estimation Jacobian and weight in place of a full re-optimization—closely reproduces the sandwich standard errors (median bootstrap-to-sandwich ratio 0.96 across the reported parameters). Third, a full re-estimation parametric bootstrap re-runs the estimator itself—the same objective, weight matrix, optimizer, and bounds, with all 71 parameters free—on 25 synthetic panels generated at $\hat{\theta}$. The dispersion of the re-estimated coefficients reproduces the conditional standard errors—for the propagation rival coefficient, 1.35 against the sandwich 1.30—so conditioning does not understate uncertainty; and the re-estimated competition coefficients center below the values that generated the panels, so the estimator's finite-sample bias runs toward zero and the reported estimates are, if

anything, conservative. The conditional estimates, validated by both bootstraps, are the paper's inference.

A trimmed-moment re-estimation verifies that none of this hinges on the moment set or the shrinkage. An audit of $\hat{\Omega}$ traces its nine singular directions to eleven degenerate moments: the nine activity shares, which are constant at 1.000 in the PC and data-center markets for every parameter value and nearly collinear with the deployment shares in autonomous vehicles, and Intel's two never-firing propagation frequencies. Dropping them ($J = 69$), rebuilding $\hat{\Omega}$ on 1,000 draws, and re-running the full two-step estimator from multiple starting values leaves every reported coefficient essentially unchanged: the release stage and the leader spillover are identical to four decimals, the propagation slopes move by at most 0.12 standard errors, and the shrinkage intensity falls from $\hat{\lambda} = 0.51$ to 0.11—the heavy shrinkage of the full moment set was a symptom of the degenerate moments, not of the informative ones. The sandwich standard errors and the one-step bootstrap reproduce on the trimmed set (median bootstrap-to-sandwich ratio 0.94), with the follower racing coefficient, if anything, sharper ($t = 8.0$). Table 10 in Appendix D reports the main and trimmed estimates side by side.

4.2 Results

Table 3 reports the estimates. The release stage carries the model's strongest competitive result. A firm's per-quarter release probability responds to the recency-weighted flow of its rivals' releases with coefficient $\hat{\beta}^{\text{rel}}_{rr,F} = 2.88$ for the followers and $\hat{\beta}^{\text{rel}}_{rr,L} = 2.19$ for the leader, against $\hat{\beta}^{\text{rel}}_{n} = 0.24$ on the firm's own release history. At the point estimates, rival releases move the hazard roughly twelve and nine times as much as the firm's own past releases for the followers and the leader, respectively.

In probability terms, at the followers' sample average release rate of 0.08 per quarter per market, a one-unit increase in rivals' recent release flow—one rival release every two quarters—raises the per-quarter release probability by about 22 percentage points, and the leader's response, evaluated at its higher average rate of 0.12, is nearly identical at about 23;

the firm's own release history moves the same probability by about 2 percentage points. These coefficients are the structural counterpart of the second fact of Section 2.4—the uniformly positive rival-exposure correlations of Table 1—and they preserve its ordering: the correlations are strongest for the firms that originate least, and the estimated response is likewise larger for the followers than for the leader.

Both racing coefficients remain significant under the full re-estimation bootstrap of Section 4.1—the leader's at the 5 percent level, the followers' at the 10. Release rivalry runs in both directions: the leader's response to the followers' releases (2.19) is close to the followers' response to their rivals' releases (2.88), and the difference between the two coefficients is insignificant. This symmetry distinguishes the racing channel from the leader-stock spillover (Γ_m^L), which runs only from the leader to the followers. Of the remaining slopes, $\hat{\beta}_x^{\text{rel}} = 0.12$ on capability is small, and $\hat{\beta}_a^{\text{rel}} = -0.52$ on contemporaneous effort is negative and significant at the 5 percent level: effort in a quarter lowers the probability of a release in that same quarter, consistent with the fifth fact of Section 2.4—effort builds capability rather than triggering releases.[10]

At the propagation stage, the coefficient on the recency-weighted flow of rivals' propagations into the destination market is $\hat{\beta}_r^{\text{prop}} = 3.47$, significant at the 1 percent level conditionally, with a bootstrap 90 percent interval of [1.45, 5.45]; under the full re-estimation bootstrap of Section 4.1 its sampling dispersion is 1.35, and its finite-sample bias runs toward zero. The coefficient on the firm's own prior propagations, $\hat{\beta}_n^{\text{prop}} = 0.59$, is significant at the 5 percent level. At the point estimates the rival channel moves the propagation hazard 5.9 times as much as the firm's own prior activity, and the rival–own difference is significant at the 5 percent level: the asymmetry that defines the release stage—rivals' recent activity outweighing the firm's own history—carries over to propagation. The two remaining slopes, $\hat{\beta}_x^{\text{prop}} = 0.27$ on capability and $\hat{\beta}_a^{\text{prop}} = 1.44$ on current effort, are positive but insignificant.

[10] Rival-flow channels without statistical support—from rival propagations to releases, from rival releases to the follower's propagation, and from rival flows to the leader's propagation—are excluded (set to zero) in the reported specification; their exclusion leaves the fit essentially unchanged.

Table 3: Structural parameter estimates.

	Parameter	Estimate (s.e.)
Release hazard: release racing		
Rival recent releases, followers	$\beta^{\text{rel}}_{rr,F}$	2.88 (0.51)
Rival recent releases, leader	$\beta^{\text{rel}}_{rr,L}$	2.19 (0.42)
Own prior releases	β^{rel}_{n}	0.24 (0.11)
Capability	β^{rel}_{x}	0.12 (0.08)
Effort	β^{rel}_{a}	−0.52 (0.21)
Rival/own ratio (followers, leader)	$\beta^{\text{rel}}_{rr}/\beta^{\text{rel}}_{n}$	12.1, 9.2 [†]
Propagation hazard (4)		
Rival recent propagation, followers	β^{prop}_{r}	3.47 (1.30)
Own prior propagation	β^{prop}_{n}	0.59 (0.29)
Capability	β^{prop}_{x}	0.27 (0.28)
Effort	β^{prop}_{a}	1.43 (1.22)
Rival/own ratio	$\beta^{\text{prop}}_{r}/\beta^{\text{prop}}_{n}$	5.9 [†]
Productivity drift μ^f_x		
NVIDIA	μ^{NVIDIA}_{x}	0.10 [d]
AMD	μ^{AMD}_{x}	0.00 [d]
Intel	μ^{INTEL}_{x}	−0.41 [d]
Effort-shock standard deviation $\sigma_{\xi,f}$		
NVIDIA	$\sigma_{\xi,\text{NVIDIA}}$	0.91 (1.05)
AMD	$\sigma_{\xi,\text{AMD}}$	0.37 (0.53)
Intel	$\sigma_{\xi,\text{INTEL}}$	0.20 (0.32)
Other structural objects		
Release-to-propagate onset (quarters)	$\hat{\tau}_0$	6.2 (0.41)
Leader spillover, AV	$\gamma^{\text{leader}}_{\text{AV}}$	6.80 (18.09) [c]
AV entry shift at platform onset, Intel	$\delta^{\text{shift}}_{\text{INTEL,AV}}$	3.67 (1.61)
AV entry shift at platform onset, NVIDIA	$\delta^{\text{shift}}_{\text{NVIDIA,AV}}$	1.55 (0.43)

Notes. Standard errors (in parentheses) are two-step-MSM sandwich estimates conditional on the non-reported parameters held at their estimated values (cf. Cocci and Plagborg-Møller, 2025). A one-step parametric bootstrap with 500 replications (Andrews, 2002) closely reproduces these standard errors (median bootstrap-to-sandwich ratio 0.96); bootstrap 90% percentile intervals for the release rival/own ratios are [5.3, 37.1] (followers) and [3.9, 27.5] (leader), and the interval for β^{prop}_r is [1.45, 5.45]. A full re-estimation bootstrap (25 replications, all parameters free) reproduces these standard errors, with finite-sample bias toward zero for the competition coefficients (Section 4.1). [†]Ratios are descriptive; inference is by the rival–own difference: releases $t = 4.66$ and $t = 4.14$ (significant at the 1 percent level), propagation $t = 2.14$ (significant at the 5 percent level). [c]Weakly identified; the corresponding data-center spillover is structurally unidentified—all three firms are active in the data center from the panel start, so the entry margin it operates on never fires—and is dropped. [d]Estimated subject to the ordering restriction of Section 4.1; μ^{NVIDIA}_x and μ^{AMD}_x rest on the restriction, so no standard errors apply. Intel's drift is interior.

Because the slopes are common across markets and the estimated intercepts leave the baseline hazard economically meaningful only in the data center (the data-center intercept

of -2.85 implies a baseline propagation probability of 0.055 per quarter; the PC and AV intercepts imply baselines near zero), the coefficients bind where cross-market propagation concentrates. In the data center, as average marginal effects at the point estimates, a one-unit increase in rivals' recent propagation flow raises the per-quarter propagation probability by about 18 percentage points, against 3 for the firm's own prior activity, 1 for a unit of capability, and 7 for a unit of effort; the effects are proportionally smaller in PC and AV. The identifying variation mirrors the descriptive record: in the rival-exposure row of Table 1, the propagation-stage correlations are 0.33 for AMD and 0.25 for Intel but near zero for NVIDIA, whose propagations lead rather than follow—the same split the specification carries, with β_r^{prop} operating for the followers and the leader's propagation-stage response set to zero. The release-to-propagate onset is $\hat{\tau}_0 = 6.2$ quarters (s.e. 0.41): a released architecture becomes ready to propagate after about a year and a half.[11]

Firm-level differences in the speed of technology output are carried by a single productivity object: each firm's drift μ_x^f in the law of motion for productive capability, its baseline rate of capability accumulation. Because capability enters both hazards, the drift governs how quickly a firm releases technologies and propagates them across markets. The estimated drifts are 0.10 for NVIDIA, 0.00 for AMD, and -0.41 for Intel; the first two rest on the ordering restriction of Section 4.1, so no standard errors apply to them, and the data-determined magnitude is Intel's drift, well below AMD's. The ordering matches the third fact: NVIDIA, which originates or co-originates 13 of the panel's 22 technologies, carries the highest drift, while Intel, which originates none and adopts most, carries a negative one—the drift is the model's currency for the origination asymmetry. The private productivity shock that the firm observes before choosing effort has firm-specific dispersion $\sigma_{\xi,f}$, estimated interior for all three firms. These dispersions absorb the fifth fact, the quarter-to-quarter R&D fluctuations of 4 to 9 percent that deployment does not follow, and they preserve its firm ordering: Intel,

[11]The onset is the least stable object across the re-estimation bootstrap's synthetic panels, with per-panel scatter of about three quarters: a single panel supplies roughly twenty-five propagation timings, so per-dataset information about τ_0 is limited. The reported standard error is conditional on the remaining parameters.

whose R&D path is smoothest in the data, carries the smallest dispersion.

Two further objects fix timing and the cross-firm structure. The onset $\hat{\tau}_0$ is recovered from the timing of first propagations rather than imposed. The foresight horizon $\hat{H}$ is set-identified (Section 3) and fixed at $\hat{H} = 21.5$ quarters, and the remaining estimates do not move with it. The leader's influence runs one way: its accumulated stock in a market raises followers' propensity to enter it, as the leader's build-out develops the market for the followers. The estimated spillover in autonomous vehicles, $\gamma_{\text{AV}}^{\text{leader}} = 6.80$, is insignificant. The corresponding data-center spillover is structurally unidentified and is dropped: every firm is active in the data center from the panel start, so the entry margin it operates on never fires.

Finally, the AV entry shifts translate the fourth fact's staggered entries into the entry index: a firm's propensity to enter autonomous vehicles rises from its documented platform introduction onward, with estimated magnitudes of 3.67 for Intel and 1.55 for NVIDIA, both significant. In probability terms, the simulated per-quarter AV entry hazard rises from below 0.01 percent before Intel's window to 12 percent after it, and from 2 to 41 percent around NVIDIA's. Intel's shift is more than twice NVIDIA's, yet its entry hazard after the onset is far lower, 12 against 41 percent. A larger shift producing a smaller hazard means the rest of Intel's entry index, built from its capability, stocks, and spillover, is much weaker; the shift compensates for that.

4.3 Model Fit

Table 4 reports the fit to the firm-level moments. Across the full set of 80 moments, 59 percent have an absolute residual below 0.05 and 72 percent below 0.10. Simulated mean effort matches the data to two decimals for every firm, and the market-level technology-stock jump rates are matched at 0.01 in every market. The deploy–effort lag correlation is near zero in both data and model for all three firms, consistent with the fifth fact of Section 2.4, though the model dates NVIDIA's slightly negative (-0.09 against 0.05). The remaining firm-level slack is in effort dispersion, where simulated effort is more volatile than observed

for AMD (0.47 against 0.38), with NVIDIA and Intel close (0.33 against 0.34 and 0.20 against 0.20). The one extensive margin with power to discriminate across firms is autonomous vehicles—both PC and the data center were entered by all three firms before the sample window and, since entry is irreversible, their market shares there equal 1 throughout in data and model alike. In autonomous vehicles the model matches Intel's market share to within 0.05 (0.06 against 0.11) and overstates NVIDIA's and AMD's (0.62 against 0.55 and 0.22 against 0.02).

Table 4: Fit of firm-level moments.

	NVIDIA		AMD		Intel	
Moment	Data	Model	Data	Model	Data	Model
Effort, mean	1.07	1.07	0.97	0.97	0.95	0.95
Effort, s.d.	0.34	0.33	0.38	0.47	0.20	0.20
Deploy–effort corr.	0.05	−0.09	0.02	−0.08	−0.04	−0.03
AV market share	0.55	0.62	0.02	0.22	0.11	0.06

Notes. Firm-level moments, data versus simulated. Effort is the standardized quarterly growth of R&D spending (Section 2); the deploy–effort correlation is between deployment and lagged effort. PC and data-center market shares are 1.00 in both data and model for every firm and are omitted; the informative extensive margin is autonomous vehicles.

The two hazards are identified by timing moments: for each firm and market, the normalized quarter at which the firm first releases any technology there, and the normalized quarter at which it first propagates any technology into that market from another, with 0 denoting the start of the firm's panel and 1 its end. Table 5 reports, for each firm-market cell, the simulated first-release (Panel A) and first-propagation (Panel B) timings alongside their data counterparts. Release timing is defined for all nine cells, since every firm releases in every market. The model matches AMD's data-center release timing exactly and Intel's PC and data-center timings to within 0.10, about five quarters. The residual release-timing slack now runs in both directions: the model dates the followers' AV releases and NVIDIA's AV and data-center releases somewhat earlier than observed (gaps of eight to fourteen quarters), and the one large miss is NVIDIA's first PC release, which the model places far later than the data (0.85 against 0.13): NVIDIA's completed PC release pool is dominated by late-panel

technologies, and a single common release hazard cannot date its first PC release early while matching the pool's overall frequency.

Table 5: Fit of first-event timing.

		Panel A: First release		*Panel B: First propagation*	
Firm	Market	Data	Model	Data	Model
NVIDIA	PC	0.13	0.85	0.58	0.69
	AV	0.45	0.27	0.50	0.64
	DC	0.53	0.30	0.00	0.11
AMD	PC	0.21	0.34	0.95	0.75
	AV	0.98	0.80	0.74	0.83
	DC	0.24	0.24	0.00	0.13
Intel	PC	0.24	0.29	—	—
	AV	0.89	0.75	—	—
	DC	0.20	0.30	0.00	0.14

Notes. Entries are the normalized quarter of the first event (0 = panel start, 1 = panel end), one row per firm-market cell. Dashes in Panel B indicate no documented cross-market propagation—the firm provides no single technology across those two markets.

Cross-market propagation is structurally sparser, arising only where a firm applies the same architecture in two markets. With the completed registry the fit tightens considerably: the remaining gaps are 0.20 or less in every cell, and the data-center cells—the systematic miss of earlier fits—are now within 0.11 to 0.14 of their targets. The residual data-center gap has the same structural source as before: the data place the first data-center propagations at the very start of each firm's panel (their technologies were released before the panel begins), while a released technology in the model becomes ready to propagate only after about six quarters ($\hat{\tau}_0$), so the model cannot fire a propagation in a firm's opening quarters. The miss is the price of a single economy-wide onset parameter.

5 Counterfactual: An AMD–Intel Merger

5.1 Design

We evaluate a merger of the two followers at the estimated parameters. Intel exits the firm set, and the merged firm occupies AMD's position, inheriting AMD's estimated firm-specific parameters and AMD's panel window.[12] Its assets are constructed from those of the two firms by explicit rules. The technology registry is the union of the two registries; where both firms hold the same advancement with the same origin market, the entry with the earlier release date is kept—this removes 13 of the 48 pooled entries, a duplicated share of 27 percent—and distinct advancements of the same underlying technology are all retained. The propagation pathways are the union of the two firms' documented origin-to-destination pathways. The initial state is built market by market: the merged firm has entered a market if either firm had entered it, and its initial capability, per-market technology stocks, and per-market own-event counts are the maximum of the two firms' values.

Two synergies are added, both with magnitudes fixed by the event registry rather than chosen. The first keeps both development pipelines producing. Before the merger, AMD and Intel run two separate release processes; the merged firm runs both. In any quarter it releases unless both pipelines fail to deliver, so its release probability is $p^{\text{merged}} = 1 - (1 - p^{\text{AMD}})(1 - p^{\text{INTEL}})$. Because the estimated hazards are small, this is approximately the sum $p^{\text{AMD}} + p^{\text{INTEL}}$ (when both pipelines deliver in the same quarter, the firm records one release, not two), and it is implemented by shifting the merged firm's release intercepts so that its hazard equals that sum. Both pipelines are evaluated at the merged firm's own state, so releases between the merging parties no longer feed each other's racing: post-merger, the rival flow the merged firm faces is the leader's alone. The second synergy is on the cost

[12] A merger counterfactual requires a stand on the merged firm's type. Treatments in the literature range from combining the merging firms' assets (Perry and Porter, 1985) to merger-generated productivity synergies in innovation industries (Igami and Uetake, 2020). We adopt the convention that the merged firm operates on AMD's estimated parameters, and measure the efficiency this confers through the blended-cost design of Table 7.

side: duplicated development is paid for once. The merged firm's effort-cost coefficients are scaled by $1 - s$, where $s = 0.27$ is the duplicated share of the pooled registry; the benefit side—capability, stocks, propagation opportunities, profit—is unchanged. The merged firm's cost coefficients are not re-estimated: a merged AMD–Intel generates no data on which to identify them, so the exercise holds the estimated primitives fixed and takes its stand on post-merger costs through the explicit transformations above; the cost-blend design of Table 7 shows that the findings do not hinge on that stand.

The merger's effect depends on these design choices, so we report a surface of counterfactuals rather than a single number. One dial, $\lambda \in [0, 1]$, indexes pipeline integration: it scales the merged firm's release hazard from AMD's pipeline alone at $\lambda = 0$ to the sum of both firms' pipelines at $\lambda = 1$, and the cost saving scales in step as λs, so one dial measures one economic object. The dial spans exactly the margin Cunningham et al. (2021) identify: full discontinuation of the acquired pipeline at $\lambda = 0$, both pipelines producing at $\lambda = 1$. We call this endowed hazard the firm's release capacity; how many releases it yields is an equilibrium outcome. Four variants pin down the remaining assumptions. The *endogenous-only* design grants the pooled portfolio and the full cost saving but assumes nothing about release capacity, so any gain in activity must come from the firm's own effort and capability. The *deduplicated* design thins the merged release flow by the same duplicated share that generates the saving, to $(1 - s)(p^{\text{AMD}} + p^{\text{INTEL}})$, and we evaluate it with and without the saving itself. The *cost-blend* design sets the merged firm's cost per effort to the weighted average of the two firms', each weighted by its contribution to the pooled registry, measuring the efficiency that lineage alone confers. The *exit* benchmark removes one of the two firms and pools nothing. The featured merger, $\lambda = 1$ with both synergies, is the surface's upper bound.

Both worlds, the three-firm baseline and the two-firm merged market, are solved under the identical counterfactual solution of Section 3.8, so the merger effect—the difference between the two worlds—reflects the change in the market, not the way either world is solved.

Because the model is stochastic even at fixed parameters—the event hazards and shocks are drawn each quarter—reported outcomes are means over 300 simulated histories of each world, and the two worlds share their random draws replication by replication, so that the shared simulation noise cancels in the difference. The model has no pricing or demand side: the exercise isolates the merger's effect on the technology race and is silent on market-power harm.[13]

5.2 Results

Table 6 reports the featured merger—the full design with the pooled portfolio and both synergies. Three findings stand out. First, this merger produces a substantially more active technology firm. The merged follower generates 14.0 release events over the panel against 7.5 for AMD and Intel operating separately, and 10.6 propagation events against 8.3, with the gains concentrated in the data center (9.9 propagations merged against 7.9 apart; a final data-center technology stock of 10 against 7). Its first data-center release comes in 2011.5 against AMD's 2014.1, its first PC release in 2013.0 against 2015.1, and it enters autonomous vehicles in 2019.6 against AMD's 2020.2, with the probability of AV entry rising from 0.68 to 0.78. In autonomous vehicles the merger's effect runs through entry rather than propagation: propagation into AV is rare in the data, and the estimated baseline propagation probability into AV is 0.03 percent per quarter (Section 4.2). The cost saving raises the merged firm's chosen effort from 0.97 to 1.10: cheaper development translates into more effort and faster building through the firm's own effort choice.

Second, the leader accelerates. NVIDIA's release count rises from 6.4 to 8.3—a 30 percent increase, concentrated in the data center (2.8 to 4.4), with its first data-center release moving about half a year earlier—and its propagations rise from 4.8 to 5.2. Nothing about NVIDIA is changed by the merger design: the response is an equilibrium outcome of its estimated racing coefficient $\beta^{\text{rel}}_{rr,L}$ facing the merged firm's larger release flow.

[13] On the demand-side margin of cross-market mergers, see Lidany (2026).

Table 6: AMD–Intel merger: baseline versus merged market.

	Followers				NVIDIA	
Outcome	AMD	Intel	Apart	Merged	Baseline	Merger
Effort (mean)	0.97	0.95	—	1.10	1.08	1.08
Capability (final)	5.15	4.69	—	5.45	5.59	5.60
First AV entry (year)	2020.2	2022.5	—	2019.6	2013.4	2013.4
AV entry (share of seeds)	0.68	0.54	—	0.78	1.00	1.00
Propagation events (total)	3.9	4.4	8.3	10.6	4.8	5.2
into DC	3.5	4.4	7.9	9.9	3.1	3.3
First DC propagation (year)	2010.7	2011.7	—	2010.5	2009.8	2009.8
Release events (total)	5.4	2.1	7.5	14.0	6.4	8.3
in PC	0.8	0.3	1.1	3.3	0.3	0.4
in DC	4.2	1.6	5.8	9.8	2.8	4.4
First PC release (year)	2015.1	2017.2	—	2013.0	2015.1	2015.6
First DC release (year)	2014.1	2016.7	—	2011.5	2013.9	2013.4

Notes. Means over 300 simulated histories of each world, with random draws shared across worlds replication by replication; $\hat{\theta}$ held fixed; every mechanism of Section 3 unchanged. "Apart" sums AMD and Intel where counts aggregate; dashes mark levels and dates that do not. The merged follower inherits AMD's firm-specific estimated parameters and panel window, combines the two firms' registries and pathways under the rules of Section 5, and carries both registry-implied synergies. Both worlds are solved under the identical policy-consistent event-path fixed point of Section 3.8; NVIDIA is re-solved in both worlds and responds through its estimated racing coefficient $\beta^{\text{rel}}_{rr,L}$. AV outcomes appear on the entry margin because cross-market propagation into AV is rare: three in-sample events across the three firms, none for Intel, and the estimated propagation baseline is economically meaningful only in the data center (Section 4.2).

Third, the design surface of Table 7 shows that the merger's sign depends on what the combination creates, and it locates the turning points. Moving the integration dial from $\lambda = 0$ to $\lambda = 1$ raises the merged firm's releases monotonically from 7.3 to 14.0 and the leader's from 6.2 to 8.3. On the full grid the merged firm's releases reach the two firms' joint pre-merger level already at $\lambda^{**} = 0.04$, and the leader's response turns positive at $\lambda^{*} = 0.14$: preserving a seventh of the second pipeline is enough to flip the leader from slowed to accelerated, so only near-total shutdown of Intel's pipeline makes the merger slow the race.

The four variants then show what each assumption contributes. The *endogenous-only* design, which grants the pooled portfolio and the full cost saving but holds release capacity at AMD's pipeline alone, is the surface's floor: the merged firm's gains come through propagation rather than releases—propagation rises 17 percent while releases (6.9) fall slightly below the two firms apart (7.5) and the leader slows mildly (6.1 against 6.4). The pattern is the direct

Table 7: The merger design surface.

	Merged rel.	Merged prop.	NVIDIA rel.
Pre-merger baseline (AMD and Intel apart)	7.5	8.3	6.4
Exit only (AMD remains)	4.7	3.8	5.9
Exit only (Intel remains)	1.4	4.1	5.5
Pooled portfolio only ($\lambda = 0$)	7.3	9.6	6.2
Endogenous only (portfolio + cost saving)	6.9	9.7	6.1
Pipeline integration $\lambda = 1/3$	9.6	10.0	6.7
Pipeline integration $\lambda = 2/3$	12.3	10.3	7.4
Deduplicated union, no cost saving	10.5	10.0	6.9
Deduplicated union, with cost saving	10.1	10.1	6.8
Full union, blended cost structure	14.0	10.1	8.2
Featured: full union, both synergies ($\lambda = 1$)	14.0	10.6	8.3

Notes. Each row is one merger design, solved and simulated identically to Table 6 (300 simulated histories per world, shared random draws, $\hat{\theta}$ fixed). The baseline row is the estimated three-firm market: the two follower columns sum AMD and Intel apart, and comparisons of any design against it give the merger's effect. Pipeline integration λ scales the merged firm's release capacity from AMD's pipeline alone to the full two-pipeline sum, with the cost saving scaled as λs; on the full grid the merged firm's releases reach the two firms' joint pre-merger level at $\lambda^{**} = 0.04$ and the leader's response turns positive at $\lambda^{*} = 0.14$. "Deduplicated" thins the merged release flow by the duplicated share $s = 0.27$. "Blended cost structure" sets the merged firm's cost per effort to the average of the two firms', taken in logs and weighted by their raw contributions to the pooled registry (27:21), implemented as a single rescaling of the merged firm's effort-cost coefficients. The capability-pooling rule (maximum versus capped sum of initial capabilities) is immaterial. The featured design is the surface's upper bound.

imprint of the estimated hazards, in which effort builds capability and propagation but enters the release hazard with a small negative coefficient: with no capacity granted, the cost saving flows into effort and effort into building, while the release margin stays at AMD's inherited capacity.

The *deduplicated* design asks whether the result depends on keeping the duplicated capacity. It thins the merged release flow by the same duplicated share that generates the cost saving, to $(1 - s)$ of the two-pipeline sum. Even with no cost saving at all, releases are 10.5—40 percent above the two firms apart—and the leader still accelerates by 9 percent. The decomposition is visible in Table 7: the pooled portfolio alone lifts releases from 4.7 under exit to 7.3 by strengthening the state at which the release hazard is evaluated, and the thinned capacity adds the rest, from 7.3 to 10.5—the design gives up only the duplicated slice of the capacity gain, not the portfolio. The cost saving itself trims releases slightly in

both designs (7.3 to 6.9 at $\lambda = 0$; 10.5 to 10.1 under deduplication)—the imprint of that same negative effort coefficient: the extra effort the saving buys builds capability (its mean rises from 5.19 to 5.56 in the deduplicated pair) while mildly delaying releases.

In the featured merger the merged firm inherits AMD's cost structure, so even the Intel side of its portfolio is developed at AMD's lower cost per effort. The *cost-blend* design removes that advantage by setting the merged firm's cost per effort to the registry-weighted average of AMD's and Intel's. Development becomes more expensive: chosen effort falls from 1.10 to 0.60, below either firm's pre-merger level (0.97 and 0.95). The comparison is not a cost ranking: effort reflects a firm's returns to effort as well as its costs. Intel chose 0.95 despite a cost per effort well above AMD's because its own estimated returns to effort supported it; the merged firm has AMD's returns, not Intel's, so the blended cost pushes its effort below both. Propagation slips to 10.1, still above the 8.3 the two firms generate apart. Releases, by contrast, stay at 14.0—nearly double the 7.5 apart—and the leader's response stays at 28 percent. This is the same mechanism read from the other side: the blend degrades effort, the margin that builds capability and propagation, and leaves untouched the pipeline capacity that generates releases—so the release surge, and with it the merger's sign, runs on capacity, while the cost structure governs only how much the firm builds.

The *exit* benchmark anchors the surface from below: removing either follower with nothing pooled slows every surviving firm. With Intel gone, AMD's releases fall from 5.4 to 4.7 and the leader's from 6.4 to 5.9; with AMD gone, Intel's fall from 2.1 to 1.4 and the leader's to 5.5. This is the competition-removal force that any merger must first overcome.

6 Conclusion

This paper studies the innovation process of a GPT: firms innovate, release technology advancements in an origin market, and propagate them across the other application markets. Propagation, in turn, builds the firm's capability, reinforcing innovation in every market, the

origin included. The process runs in both directions—conditions in the application markets feed back into upstream innovation, as competition and opportunity downstream shape what firms create and what technology propagates. The GPU industry gives this lens its sharpest available test case.

Viewed as a GPT the GPU industry runs on two competitive margins. Firms release in races: a firm's release hazard responds about ten times more strongly to rivals' recent releases than to its own history, and the leader races its followers as hard as they race it. Firms also propagate toward their rivals: propagation into a destination market is pulled about six times more strongly by rivals' propagation there than by the firm's own, arriving about a year and a half after a technology's release and concentrating where application value currently lies, the data center. R&D spending alone reveals none of this—the industry's largest spender originates none of its technologies; the industry's dynamics are visible in the cross-market record of technology events, not in R&D expenditure.

The same lens bears on antitrust. A merger between the two followers changes the technology race itself, and the direction of the change is not obvious ex ante. As pure competition removal, the merger slows the race for every firm, the leader included. With the synergies implied by the technology registry, it instead nearly doubles the merged firm's release capacity and, through the racing channel, accelerates the leader by thirty percent. The number of competitors therefore matters—removing one slows the race—but it is not a sufficient statistic: what the combination adds to the race can reverse the merger's sign, and the leader's equilibrium response is the observable that separates the two cases.

The broader lesson is about how to analyze a GPT. Its application markets are not separate objects of study: application opportunities in one market shape the technologies available to all, and a change in competitive conditions in one market transmits to the others through the upstream race. Standard practice—estimating each market in isolation, or treating technology as an exogenous arrival process—misses both directions of this transmission. The apparatus developed here, release and propagation hazards disciplined by a documented

registry of technology events, makes the transmission estimable, and its data requirements are modest: a record of what was created, where it propagates, and when. As AI repeats the pattern—foundation models created upstream, released in races, and propagated across application markets whose returns feed back into the direction of the technology itself—the questions this paper asks of GPUs are the questions that industry will pose at far larger scale. GPUs are the enabling technology of AI: the race studied here is the upstream of the GPT that follows, and its data center—where this paper finds application value concentrating—is where that technology is being built.

References

Aghion, P., Bloom, N., Blundell, R., Griffith, R., and Howitt, P. (2005). Competition and innovation: An inverted-u relationship. *Quarterly Journal of Economics*, 120(2):701–728.

Altonji, J. G. and Segal, L. M. (1996). Small-sample bias in GMM estimation of covariance structures. *Journal of Business & Economic Statistics*, 14(3):353–366.

Andrews, D. W. (2002). Higher-order improvements of a computationally attractive k-step bootstrap for extremum estimators. *Econometrica*, 70(1):119–162.

Andrews, I., Gentzkow, M., and Shapiro, J. M. (2017). Measuring the sensitivity of parameter estimates to estimation moments. *The Quarterly Journal of Economics*, 132(4):1553–1592.

Arrow, K. J. (1962). The economic implications of learning by doing. *The review of economic studies*, 29(3):155–173.

Bajari, P., Benkard, C. L., and Levin, J. (2007). Estimating dynamic models of imperfect competition. *Econometrica*, 75(5):1331–1370.

Benkard, C. L., Jeziorski, P., and Weintraub, G. Y. (2015). Oblivious equilibrium for concentrated industries. *The RAND Journal of Economics*, 46(4):671–708.

Besanko, D., Doraszelski, U., Kryukov, Y., and Satterthwaite, M. (2010). Learning-by-doing, organizational forgetting, and industry dynamics. *Econometrica*, 78(2):453–508.

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

Bresnahan, T. F., Brynjolfsson, E., and Hitt, L. M. (2002). Information technology, workplace organization, and the demand for skilled labor: Firm-level evidence. *The quarterly journal of economics*, 117(1):339–376.

Bresnahan, T. F. and Trajtenberg, M. (1995). General purpose technologies 'engines of growth'? *Journal of Econometrics*, 65(1):83–108.

Cocci, M. D. and Plagborg-Møller, M. (2025). Standard errors for calibrated parameters. *Review of Economic Studies*, 92(5):2952–2978.

Cockburn, I. M., Henderson, R., and Stern, S. (2019). The impact of artificial intelligence on innovation: An exploratory analysis. In Agrawal, A., Gans, J., and Goldfarb, A., editors, *The Economics of Artificial Intelligence: An Agenda*, pages 115–146. University of Chicago Press.

Comin, D. and Hobijn, B. (2010). An exploration of technology diffusion. *American Economic Review*, 100(5):2031–2059.

Cunningham, C., Ederer, F., and Ma, S. (2021). Killer acquisitions. *Journal of Political Economy*, 129(3):649–702.

David, P. A. (1990). The dynamo and the computer: An historical perspective on the modern productivity paradox. *American Economic Review*, 80(2):355–361.

Ericson, R. and Pakes, A. (1995). Markov-perfect industry dynamics: A framework for empirical work. *The Review of Economic Studies*, 62(1):53–82.

Federico, G., Langus, G., and Valletti, T. (2018). Horizontal mergers and product innovation. *International Journal of Industrial Organization*, 59:1–23.

Fershtman, C. and Pakes, A. (2012). Dynamic games with asymmetric information: A framework for empirical work. *The Quarterly Journal of Economics*, 127(4):1611–1661.

Goettler, R. L. and Gordon, B. R. (2011). Does AMD spur Intel to innovate more? *Journal of Political Economy*, 119(6):1141–1200.

Goldfarb, A., Taska, B., and Teodoridis, F. (2023). Could machine learning be a general purpose technology? A comparison of emerging technologies using data from online job postings. *Research Policy*, 52(1):104653.

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

Helpman, E. and Trajtenberg, M. (1998). A time to sow and a time to reap: Growth based on general purpose technologies. In Helpman, E., editor, *General Purpose Technologies and Economic Growth*, pages 55–83. MIT Press, Cambridge, MA.

Ifrach, B. and Weintraub, G. Y. (2017). A framework for dynamic oligopoly in concentrated industries. *The Review of Economic Studies*, 84(3):1106–1150.

Igami, M. and Uetake, K. (2020). Mergers, innovation, and entry-exit dynamics: Consolidation of the hard disk drive industry, 1996–2016. *Review of Economic Studies*, 87(6):2672–2702.

Jon Peddie Research (2023a). AIB shipments climb in Q2 2023, with unit sales increasing from quarter to quarter. Desktop add-in-board shipments and supplier shares, Q2 2023.

Jon Peddie Research (2023b). Market watch: a quarterly report on the graphics processor unit market. https://www.jonpeddie.com/store/market-watch-a-report-on-the-graphics-processor-unit-market/. Quarterly PC graphics shipments and supplier shares (desktop and notebook), 2008–2023.

Jovanovic, B. and Rousseau, P. L. (2005). General purpose technologies. In Aghion, P. and Durlauf, S. N., editors, *Handbook of Economic Growth*, volume 1B, pages 1181–1224. Elsevier.

Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411.

Lidany, Y. (2026). Cross-product compatibility, lock-in, and market power: The case of smartphones and laptops. Working paper.

Magnac, T. and Thesmar, D. (2002). Identifying dynamic discrete decision processes. *Econometrica*, 70(2):801–816.

McFadden, D. (1989). A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica: Journal of the Econometric Society*, pages 995–1026.

Pakes, A. and Pollard, D. (1989). Simulation and the asymptotics of optimization estimators. *Econometrica: Journal of the Econometric Society*, pages 1027–1057.

Perry, M. K. and Porter, R. H. (1985). Oligopoly and the incentive for horizontal merger. *American Economic Review*, 75(1):219–227.

Rust, J. (1987). Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. *Econometrica*, 55(5):999–1033.

Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4(1).

Shilov, A. (2023). GPU market 'healthy and vibrant' in Q2 2023: report. Tom's Hardware, https://www.tomshardware.com/news/

gpu-market-healthy-and-vibrant-in-q2-2023-report. Reporting Jon Peddie Research Q2 2023 add-in-board data.

Weintraub, G. Y., Benkard, C. L., and Van Roy, B. (2008). Markov perfect industry dynamics with many firms. *Econometrica*, 76(6):1375–1411.

A Empirical Mapping and Calibrated Constants

Pre-panel state. At $t = 0$ the stocks and counts are seeded from documented product history: NVIDIA enters with $z_{0,\mathrm{PC}} = 0.95$, $z_{0,\mathrm{DC}} = 0.70$, $n_{0,\mathrm{PC}} = 4$, and $n_{0,\mathrm{DC}} = 3$ (GeForce since 1999, Tesla/Fermi from 2006), while AMD and INTEL enter with zero pre-panel stocks and counts. Initial capability is set to each firm's mean observed effort: $x_0 = 1.073$ for NVIDIA, 0.971 for AMD, and 0.953 for INTEL. Panel starts: NVIDIA 2008Q1, AMD 2009Q1, INTEL 2010Q1.

Cost-era adjustment. AMD's observed R&D outlays accelerate sharply after 2020, with mean quarterly cost rising almost threefold, which a single stationary cost function cannot fit. AMD's cost therefore includes an era term: from the breakpoint $y^{\mathrm{era}}_{\mathrm{AMD}} = 2020$ onward, an estimated per-year amount is added to its cost, and AMD's cost moments are split at the same year (Section 4.1). INTEL and NVIDIA show no comparable break, and their era terms are zero.

Calibrated constants. Table 8 collects the constants fixed outside the estimation, with the role each plays in the model; each choice is discussed at its point of use in Sections 3 and 4.1.

Table 8: Calibrated constants.

Object	Role	Value
Discount factor β	quarterly discounting	0.92
Foresight horizon $\hat{H}$	look-ahead span in quarters, set identified	21.5
Onset steepness κ_τ	sharpness of the readiness weight $w(\tau)$	1.5
Recency window	rival-flow lookback in quarters	8
Recency scaling	unit normalization of the rival flow	1/4
Follower stock step	stock-jump probability at zero stock	0.0105
AV shift onset dates	AV-platform introductions (NVIDIA, AMD, Intel)	2015, 2019, 2022
Productivity-ordering margin	minimum gap in μ_x^f	0.10
Effort bounds	admissible effort choice set	[0, 3]
Effort scaling	effort measure, pooled mean and s.d.	1.00, 0.35

Notes. The foresight horizon is set-identified (Section 3). The recency window counts rival same-stage events over the trailing eight quarters, scaled by one fourth, so one unit of the rival flow corresponds to one rival event every two quarters. The follower stock step equals the observed market technology-jump frequency (0.011 per quarter, Table 9), identical across markets in the data, and applies per quarter in every market. The AV shift dates are NVIDIA 2015, AMD 2019, and Intel 2022; their magnitudes are estimated (Section 4.2). The productivity-ordering margin is imposed through a penalty with weight 1000.

B Targeted Moments

Table 9 lists all $J = 80$ estimation moments with their data values; the moment families are described in Section 4.1.

Table 9: The 80 targeted moments and their data values.

Moment	Data	Moment	Data
AMD_m_a_mean	0.971	AMD_m_cutoff_freq_pc	0.052
AMD_m_a_sd	0.381	AMD_m_cutoff_freq_av	0.017
AMD_m_log_cost_pre2020	5.784	AMD_m_cutoff_freq_datacenter	0.034
AMD_m_d_corr_a_lag	0.017	INTEL_m_cutoff_freq_pc	0.000
AMD_m_A_mean_pc	1.000	INTEL_m_cutoff_freq_av	0.000
AMD_m_A_mean_av	0.017	INTEL_m_cutoff_freq_datacenter	0.093
AMD_m_A_mean_datacenter	1.000	NVIDIA_m_cutoff_freq_pc	0.081
AMD_m_D_mean_pc	1.000	NVIDIA_m_cutoff_freq_av	0.065
AMD_m_D_mean_av	0.017	NVIDIA_m_cutoff_freq_datacenter	0.081
AMD_m_D_mean_datacenter	1.000	AMD_m_first_cutoff_norm_pc	0.948
AMD_m_log_cost_post2020	6.809	AMD_m_first_cutoff_norm_av	0.741
AMD_m_first_av_entry_norm	0.847	AMD_m_first_cutoff_norm_datacenter	0.000
AMD_m_log_cost_per_effort	6.172	INTEL_m_first_cutoff_norm_datacenter	0.000
INTEL_m_a_mean	0.953	NVIDIA_m_first_cutoff_norm_pc	0.581
INTEL_m_a_sd	0.195	NVIDIA_m_first_cutoff_norm_av	0.500
INTEL_m_cost_mean	3084.4	NVIDIA_m_first_cutoff_norm_datacenter	0.000
INTEL_m_d_corr_a_lag	−0.045	AMD_m_release_freq_pc	0.069
INTEL_m_A_mean_pc	1.000	AMD_m_first_release_norm_pc	0.207
INTEL_m_A_mean_av	0.111	AMD_m_release_freq_av	0.034
INTEL_m_A_mean_datacenter	1.000	AMD_m_first_release_norm_av	0.983
INTEL_m_D_mean_pc	1.000	AMD_m_release_freq_datacenter	0.155
INTEL_m_D_mean_av	0.111	AMD_m_first_release_norm_datacenter	0.241
INTEL_m_D_mean_datacenter	1.000	INTEL_m_release_freq_pc	0.130
INTEL_m_first_av_entry_norm	0.778	INTEL_m_first_release_norm_pc	0.241
INTEL_m_log_cost_per_effort	8.074	INTEL_m_release_freq_av	0.019
NVIDIA_m_a_mean	1.073	INTEL_m_first_release_norm_av	0.889
NVIDIA_m_a_sd	0.336	INTEL_m_release_freq_datacenter	0.093
NVIDIA_m_cost_mean	493.2	INTEL_m_first_release_norm_datacenter	0.204
NVIDIA_m_d_corr_a_lag	0.053	NVIDIA_m_release_freq_pc	0.129
NVIDIA_m_A_mean_pc	1.000	NVIDIA_m_first_release_norm_pc	0.129
NVIDIA_m_A_mean_av	0.548	NVIDIA_m_release_freq_av	0.097
NVIDIA_m_A_mean_datacenter	1.000	NVIDIA_m_first_release_norm_av	0.452
NVIDIA_m_D_mean_pc	1.000	NVIDIA_m_release_freq_datacenter	0.129
NVIDIA_m_D_mean_av	0.548	NVIDIA_m_first_release_norm_datacenter	0.532
NVIDIA_m_D_mean_datacenter	1.000	AMD_m_rival_exposure_release	0.260
NVIDIA_m_first_av_entry_norm	0.389	AMD_m_rival_exposure_prop	0.333
NVIDIA_m_log_cost_per_effort	5.997	INTEL_m_rival_exposure_release	0.390
m_jump_mean_pc	0.011	INTEL_m_rival_exposure_prop	0.248
m_jump_mean_av	0.011	NVIDIA_m_rival_exposure_release	0.126
m_jump_mean_datacenter	0.011	NVIDIA_m_rival_exposure_prop	0.016

Notes. cutoff denotes propagation events; first_*_norm are first-event timings normalized to each firm's panel (0 = start, 1 = end); cost moments are in millions of dollars per quarter or logs thereof; jump moments are per-quarter market technology-stock jump frequencies.

C Computation

The firm's dynamic program is solved on a discrete state grid: capability on 7 points over $[0, 6]$, per-market stocks on 4 points over $[0, 1]$, with the activity and deployment indicators and own-event counts as discrete states; the next-period private shock is integrated by Gauss–Hermite quadrature with 5 nodes. The value function is computed by rolling-horizon backward induction: the firm's projected release and propagation dates partition the next $\hat{H}$ quarters into event-free blocks, each block is solved by backward induction, blocks are chained, and beyond $\hat{H}$ the continuation value is the stationary value under the myopic policy; the procedure is re-run each quarter as projections update. In counterfactuals, all firms' projected event calendars are iterated to the fixed point of Section 3.8. The simulated moment vector is an average over $S = 12$ replications whose random draws are fixed once and reused at every parameter value, so the objective varies only through θ. The two-step estimator minimizes the objective by Powell's method from three starting values; the second-step weight is the inverse of the simulated moment covariance from 100 panel draws, with the correlation matrix shrunk toward the identity (Schäfer–Strimmer intensity 0.51) and the result normalized to a mean-one diagonal.

D Inference Details

Sandwich standard errors. The Jacobian of the simulated moments is computed by central differences under common random numbers (48 seeds per side; relative step 5 percent with a floor of 0.02). Simulation noise enters the covariance through the factor $1 + 1/S$. The nine parameters with no effect on any moment (Section 4.1) are fixed as normalizations; standard errors for the 25 reported parameters are computed jointly, conditional on the 46 estimated but non-reported parameters, and the contrasts and ratio intervals use the full 25×25 covariance.

One-step parametric bootstrap. Each of 500 replications draws two objects at $\hat{\theta}$: one synthetic panel, standing in for the observed data, and an independent average of $S = 12$ simulated panels, constructed exactly as the estimator computes its simulated moment vector. Their difference enters the one-step map (Andrews, 2002), so each replication carries the estimator's own simulation noise and the bootstrap dispersion matches the sandwich's $(1 + 1/S)$ convention; seed offsets are spaced to avoid shared firm-level random streams. The median bootstrap-to-sandwich ratio across reported parameters is 0.96.

Full re-estimation bootstrap. Each of 25 replications draws a synthetic panel at $\hat{\theta}$ and re-runs the entire estimator—the same objective, weight matrix, penalties, and optimizer, with all 71 parameters free—warm-started from the replication's one-step draw so that incomplete optimization inflates rather than understates the measured dispersion. Standard errors are the dispersion of the re-estimated coefficients scaled by $\sqrt{1 + 1/S}$, which restores the simulation-noise component of the MSM variance (McFadden, 1989) that the dispersion omits because every re-estimation reuses the estimator's frozen simulation draws; the dispersion reproduces the conditional standard errors, and the re-estimated competition coefficients center below the values that generated the panels, so the reported estimates are, if anything, conservative.

Trimmed-moment robustness. Table 10 reports the re-estimation of Section 4.1 on the trimmed moment set: the eleven degenerate moments dropped ($J = 69$), $\hat{\Omega}$ rebuilt on 1,000 draws ($\hat{\lambda} = 0.11$), the estimator re-run from multiple starting values, and standard errors recomputed under the same conventions as Table 3: the sandwich conditional on the non-reported parameters, and the one-step bootstrap (median bootstrap-to-sandwich ratio 0.94); the full re-estimation bootstrap is not repeated.

Table 10: Robustness: published versus trimmed-moment estimates.

	Parameter	Published (s.e.)	Trimmed (s.e.)
Release hazard: release racing			
Rival recent releases, followers	$\beta^{\text{rel}}_{rr,F}$	2.88 (0.51)	2.88 (0.36)
Rival recent releases, leader	$\beta^{\text{rel}}_{rr,L}$	2.19 (0.42)	2.19 (0.45)
Own prior releases	β^{rel}_{n}	0.24 (0.11)	0.24 (0.10)
Capability	β^{rel}_{x}	0.12 (0.08)	0.12 (0.10)
Effort	β^{rel}_{a}	−0.52 (0.21)	−0.52 (0.20)
Rival/own ratio (F, L)	$\beta^{\text{rel}}_{rr}/\beta^{\text{rel}}_{n}$	12.1, 9.2 [†]	12.1, 9.2 [†]
Propagation hazard			
Rival recent propagation, followers	β^{prop}_{r}	3.47 (1.30)	3.64 (1.64)
Own prior propagation	β^{prop}_{n}	0.59 (0.29)	0.58 (0.23)
Capability	β^{prop}_{x}	0.27 (0.28)	0.29 (0.23)
Effort	β^{prop}_{a}	1.43 (1.22)	1.46 (0.80)
Rival/own ratio	$\beta^{\text{prop}}_{r}/\beta^{\text{prop}}_{n}$	5.9 [†]	6.2 [†]
Productivity drift μ^{f}_{x}			
NVIDIA	μ^{NVIDIA}_{x}	0.10 [d]	0.11 [d]
AMD	μ^{AMD}_{x}	0.00 [d]	0.00 [d]
Intel	μ^{INTEL}_{x}	−0.41 [d]	−0.54 [d]
Effort-shock standard deviation $\sigma_{\xi,f}$			
NVIDIA	$\sigma_{\xi,\text{NVIDIA}}$	0.91 (1.05)	0.93 (1.17)
AMD	$\sigma_{\xi,\text{AMD}}$	0.37 (0.53)	0.35 (8.34)
Intel	$\sigma_{\xi,\text{INTEL}}$	0.20 (0.32)	0.42 (1.14)
Hazard intercepts and AV entry shifts			
Propagation intercepts (PC, AV, DC)	—	−8.76, −8.02, −2.85	−8.76, −8.02, −2.85
Release intercepts (PC, AV, DC)	—	−7.30, −5.89, −6.02	−7.30, −5.89, −6.02
AV entry shifts (Intel, NVIDIA)	—	3.67, 1.55	3.67, 1.55
Other structural objects			
Release-to-propagate onset (quarters)	$\hat{\tau}_0$	6.2 (0.41)	6.2 (0.40)
Leader spillover, AV	$\gamma^{\text{leader}}_{\text{AV}}$	6.80 (18.09) [c]	6.80 (10.93) [c]

Notes. The trimmed specification drops the eleven degenerate moments identified in Section 4.1 ($J = 69$), rebuilds $\hat{\Omega}$ on 1,000 draws (shrinkage intensity $\hat{\lambda} = 0.11$ against 0.51), and re-runs the two-step estimator from multiple starting values; standard errors follow the identical conditional sandwich convention as Table 3. Estimates agree to four decimals except the propagation-hazard slopes, which move by at most 0.12 standard errors; the intercept and shift estimates are identical across the two columns and their standard errors move comparably. The trimmed weight leaves the objective flatter in the effort-shock dispersions $\sigma_{\xi,f}$—nuisance parameters, insignificant under either weighting. [†]Ratios are descriptive. [c]Weakly identified (see Table 3). [d]Estimated subject to the ordering restriction of Section 4.1.

Moment sensitivity of the estimates. Following Andrews et al. (2017), the sensitivity of $\hat{\theta}$ to the moments is $\Lambda = (\hat{G}^{\top}\hat{W}\hat{G})^{-1}\hat{G}^{\top}\hat{W}$, evaluated at $\hat{\theta}$ with the estimation weight.

Table 11 reports, for each headline coefficient, the moments with the largest standardized sensitivities: the effect of a one-standard-deviation perturbation of the moment on the coefficient, in units of its standard error.

The pattern matches the stated identification. The followers' racing coefficient is most sensitive to the followers' rival-exposure moment and the leader's to the leader's; the propagation rival coefficient responds most to propagation timing and to the propagation exposure moment; and the onset $\hat{\tau}_0$ responds most to the timing of first data-center propagations, the moments it is recovered from. The signs admit a direct reading. A one-standard-deviation delay in AMD's first propagation into PC would lower the estimated β_r^{prop} by 0.50 standard errors, as the model can generate a later first propagation only through a weaker response to rivals. Conversely, a one-standard-deviation increase in the propagation exposure co-movement would raise the estimate by 0.39 standard errors, as stronger clustering of propagations after rivals' activity requires a stronger response to match. The estimate is therefore disciplined from both directions. The leader's negative entry follows the same logic. A one-standard-deviation rise in the AV stock-jump rate would lower $\hat{\beta}_{rr,L}^{\text{rel}}$ by 0.24 standard errors: a more active NVIDIA position in autonomous vehicles by itself raises its simulated release rate, leaving less for racing to explain.

Table 11: Andrews–Gentzkow–Shapiro sensitivity to moments: headline coefficients.

Coefficient	Most sensitive moments	Sensitivity (s.e. per s.d.)
$\beta^{\text{rel}}_{rr,F}$	AMD rival-exposure (release)	0.52
	AMD release frequency, AV	0.37
	Intel release frequency, DC	0.33
$\beta^{\text{rel}}_{rr,L}$	NVIDIA rival-exposure (release)	0.63
	NVIDIA release frequency, DC	0.53
	Stock-jump frequency, AV	−0.24
β^{prop}_{r}	AMD first propagation timing, PC	−0.50
	AMD rival-exposure (propagation)	0.39
	AMD propagation frequency, AV	0.25
β^{rel}_{a}	Intel rival-exposure (release)	0.28
	AMD first propagation timing, PC	0.24
	AMD effort dispersion	−0.22
$\hat{\tau}_0$	NVIDIA first propagation timing, DC	0.62
	AMD first propagation timing, DC	0.24
	AMD release frequency, AV	0.24

Notes. Each entry answers a perturbation question: if the moment were one standard deviation higher in the data, the coefficient would shift by the stated number of its own standard errors, holding the other moments fixed. Formally, entries are $\Lambda_{kj}\,\mathrm{sd}(g_j)/\mathrm{se}(\hat{\theta}_k)$, with moment standard deviations taken from the $\hat{\Omega}$ draws at $\hat{\theta}$. The top three moments by absolute sensitivity are shown per coefficient.