The Task Frontier Race
Horizon, Forgiveness, and Dynamic AI Competition

June 2026

Abstract

This paper develops a theory of AI progress in which the primitive object is the value of raising a task frontier. The frontier has two economically distinct bottlenecks: horizon, the serial depth of the required action path, and forgiveness, the extent to which mistakes can be observed, corrected, retried, or absorbed before they create irreversible loss. A system therefore has two capabilities: one for long dependent tasks and one for fragile tasks. The model yields three sufficient statistics: the value of relaxing the horizon boundary, the value of relaxing the fragility boundary, and the value of relaxing both at once. These statistics explain why AI firms race on different technologies in different task markets. Runtime, tools, search, and test-time compute are most valuable where tasks are long but forgiving. Base-model reliability, verification, and monitoring are most valuable where tasks are fragile. This race inversion is the paper's central result: a scalar quality ladder can fit either region, but it cannot discipline the reversal without adding region-specific wedges. The dynamic application separates productive data flywheels from pure lock-in. Deployment scale generates durable advantage only when user traces move the scarce task boundary; when they do not, scale mainly reallocates rents. The model explains why low-cost imitation can be powerful in forgiving workflows while frontier premia persist in fragile deployment domains.

JEL: O31, O33, L13, L40, D24.

1 Introduction

Frontier AI progress is usually described as movement along a scalar quality ladder: larger models, lower loss, higher benchmark scores, or more capable agents. That language is useful but incomplete. It cannot explain why two tasks can look equally difficult on a benchmark and yet have very different economic adoption thresholds. Coding agents can make mistakes, run tests, inspect logs, and try again. Product recommendation is often short and forgiving. Self-driving may involve simple local controls, but a local error can cause large irreversible harm. Scientific discovery, drug development, and autonomous organizations can require both long sequences of dependent steps and extremely low tolerance for hidden error.

The paper's first claim is that the economically relevant task frontier has two dimensions. The first is horizon: how much a task depends on a long serial chain of state-dependent actions. The second is forgiveness: how much the task environment allows mistakes to be retried, monitored, corrected, sandboxed, or reversed. I write the second coordinate as fragility f, the inverse of forgiveness. A high-f task is not simply a task with many steps. It is a task with a narrow acceptable path.

This primitive keeps the model small. Retryability, reversibility, stakes, human review, monitoring, and required error rates are not extra dimensions. They are mechanisms that move a task's effective fragility. Tests and rollback make coding more forgiving. Real-time externalities make driving less forgiving. Clinical harm, unobserved errors, and one-shot decisions make some medical and scientific tasks less forgiving. The two primitive task coordinates are horizon and forgiveness.

An AI system has capability vector $q = (q_H, q_F)$. Horizon capability shifts the frontier over long serial tasks. Forgiveness capability shifts the frontier over fragile tasks that require reliability, discipline, monitoring, and low irreversible error. Task (h, f) succeeds with probability

$$p(q, h, f) = G_H(q_H - h)G_F(q_F - f).$$

The multiplicative form says that a task must clear both gates. Long context and planning do not solve an unforgiving task if reliability is too low. Low local error rates do not solve a task that is beyond the system's serial depth.

The coordinates are not only labels. In a sequential task, longer chains mechanically raise the required local reliability, while observability, retries, verification, rollback, and sandboxing lower it. This gives a formal reason why coding can be long but forgiving and why driving can be short but fragile. The task frontier is a reduced form for serial error, recovery, and irreversibility, not just another benchmark score.

Let $a(h, f)$ be the value density of tasks. Total task surplus is

$$W(q) = \iint a(h, f)G_H(q_H - h)G_F(q_F - f)\, dh\, df.$$

The finite-step value of raising the frontier is

$$A(q, \Delta) = W(q + \Delta) - W(q), \qquad \Delta = (\Delta_H, \Delta_F) \geq 0.$$

The central sufficient statistics are

$$B_H(q) = \iint a(h, f)g_H(q_H - h)G_F(q_F - f)\, dh\, df,$$

$$B_F(q) = \iint a(h, f) G_H(q_H - h) g_F(q_F - f)\, dh\, df,$$

and

$$C(q) = \iint a(h, f) g_H(q_H - h) g_F(q_F - f)\, dh\, df.$$

B_H is the value of relaxing the horizon boundary. B_F is the value of relaxing the fragility boundary. C is the value mass at the corner where both constraints bind. The exact interaction surplus from improving both dimensions is

$$\int_0^{\Delta_H} \int_0^{\Delta_F} C(q_H + u, q_F + v)\, dv\, du.$$

Thus horizon and reliability are complements exactly when valuable task mass sits near both boundaries.

The paper's second claim is that firms race over technologies, not directly over q_H and q_F. Base-model scaling b is the foundation-model race: model size, pretraining, post-training, data mixture, accelerator clusters, and distributed training systems. Runtime/harnessing r is the deployed-agent-stack race: test-time compute, search, tools, memory, verifiers, retrieval, retry loops, agent-computer interfaces, and workflow integration. Deployment data d and distillation m make runtime cumulative. More users create more task traces, corrections, and successful trajectories; distillation compresses those trajectories into cheaper prompts, policies, verifiers, smaller models, and task routines.

This distinction changes the race. Base scaling mostly raises core competence and per-step reliability. Runtime/harnessing mostly extends effective horizon and can raise reliability when errors are observable and recoverable. It is not a generic substitute for a better base model: retries and search help most when there is feedback, a verifier, or a safe sandbox. Deployment data matters most for runtime because it tells the firm which tool calls, retries, verifiers, retrieval routines, and workflows actually work. Distillation is the bridge: runtime discovers expensive successful trajectories, and distillation turns them into cheaper future capability.

The paper's central result is therefore not that AI quality is two-dimensional in the abstract. It is that task regions induce different races. Where the scarce boundary is serial depth and errors are recoverable, firms should race on search, tools, workflows, and test-time compute. Where the scarce boundary is fragile reliability, firms should race on base-model quality, verification, monitoring, and rollback. A scalar quality ladder can fit either pattern in isolation, but it cannot discipline the reversal without adding region-specific technology or adoption wedges.

The resulting theory speaks to several current AI patterns. Low-cost followers can be highly valuable where task environments are forgiving because runtime, tools, retries, and distillation can close much of the gap. Frontier systems retain value where small base-reliability gaps are multiplied by fragile, high-value deployment environments. Full-stack tipping is most likely in tasks that are both long and unforgiving, because the leader's base model makes its harness better, its deployment gives it more task traces, and distillation lowers future runtime cost. Private racing can also persist when productive frontier value is small, because firms value market-share shifts over already installed task value.

The paper is organized around three results. Sections 2–3 define the frontier and derive the sufficient statistics B_H, B_F, and C. Sections 4–5 use those statistics to derive race inversion across task regions and the scalar benchmark. Sections 8 and 8.1 apply the same objects to data-distillation tipping and pure lock-in. The remaining sections collect supporting implications for imitation, stopping, and frontier-surplus wedges. The appendix verifies each derivation step by step.

2 Task Primitives

The task space is $\mathcal{T} \subseteq \mathbb{R}^2$. A task is indexed by (h, f). The coordinate h is horizon: the serial depth of the required action path. The coordinate f is fragility, the inverse of forgiveness. Higher f means that a task has less slack for retry, correction, observation, or reversible error.

Let an AI system have capability vector

$$q = (q_H, q_F) \in \mathbb{R}^2.$$

The task success probability is

$$p(q, h, f) = G_H(q_H - h)G_F(q_F - f), \tag{1}$$

where $G_H, G_F : \mathbb{R} \to [0, 1]$ are continuously differentiable increasing functions with densities $g_H = G'_H$ and $g_F = G'_F$. The shifted form says that capability moves a boundary through task space.

The multiplicative specification is the baseline separable case. It is useful because it lets the sufficient statistics be written exactly and transparently. It should not be read as a claim that engineering systems always separate horizon from reliability. In applications, long trajectories can amplify error, verifiers can trade latency for reliability, and scaffolds can introduce new failure modes. Those forces can be modeled as task-specific shifts in effective h and f, or as an additional interaction term around the separable benchmark. The baseline result below says what the boundary statistics are before adding those engineering interactions.

Let $a(h, f) \geq 0$ be an integrable task-value density. Total task surplus is

$$W(q) = \iint_{\mathcal{T}} a(h, f)G_H(q_H - h)G_F(q_F - f)\, dh\, df. \tag{2}$$

For a finite project step $\Delta = (\Delta_H, \Delta_F) \geq 0$, define

$$A(q, \Delta) = W(q + \Delta) - W(q). \tag{3}$$

This is the agentic raising statistic. It is finite-step by construction because frontier training, infrastructure, data, evaluation, and deployment projects are lumpy.

Operational content. The primitives are meant to be measurable. Horizon h can be proxied by trajectory length, dependency depth, number of state-contingent subtasks, context span, and the number of external actions that must be coordinated before payoff is realized. Fragility f can be proxied by required error rate, cost of a bad action, observability of mistakes, reversibility, rollback availability, verification coverage, and whether failure creates harm before correction is possible. Capability q_H is measured by success on long-horizon tasks holding the error environment fixed. Capability q_F is measured by success in low-tolerance tasks holding horizon fixed. This makes the theory falsifiable: the same model should predict different adoption and race patterns for coding, product recommendation, driving, clinical support, and scientific discovery because these tasks occupy different regions of the (h, f) space.

Identification protocol. The primitives can be disciplined without turning the paper into an empirical study. Construct benchmark or deployment cells r with known value weights a_r and observed success rates $\hat{p}_{ir}$. Two controlled task families identify the coordinates. In the horizon family, chain length or dependency depth varies while the verifier, retry budget, and failure tolerance are held fixed; this shifts h_r but not f_r. In the fragility family, retryability, observability, or tolerated failure probability varies at fixed chain length; this shifts f_r but not h_r. Normalizations set one baseline task to $h = f = 0$ and one log-odds unit of difficulty to one unit of each coordinate. The measurement equation is the success equation

$$\hat{p}_{ir} = G_H(q_{iH} - h_r) G_F(q_{iF} - f_r) + \varepsilon_{ir},$$

with cell error ε_{ir}. Given the fitted kernels and task values, the regional scarcity ratio is recovered by plug-in moments,

$$\widehat{\Theta}_R = \frac{\sum_{r \in R} a_r g_H(\hat{q}_H - \hat{h}_r) G_F(\hat{q}_F - \hat{f}_r)}{\sum_{r \in R} a_r G_H(\hat{q}_H - \hat{h}_r) g_F(\hat{q}_F - \hat{f}_r)}.$$

Equivalently, B_H^R and B_F^R can be recovered from finite differences in regional value after projects known to move mostly H or mostly F. The theory is falsified, not merely renamed, if race direction does not move with $\widehat{\Theta}_R$, or if a scalar model with region-invariant project productivities matches the same adoption and investment rankings.

2.1 A Sequential Microfoundation

The two task coordinates can be derived from a simple sequential production problem. Consider a task with n essential stages. Each stage requires one correct local action. The local probability of a correct action is

$$\pi(q_F, \varphi) = G_F(q_F - \varphi),$$

where φ is the underlying local reliability burden. If a mistake is observable and recoverable, the system can make $\ell \geq 0$ additional attempts before the stage is lost. The stage success probability is therefore

$$S_\ell(q_F, \varphi) = 1 - \{1 - G_F(q_F - \varphi)\}^{1+\ell}.$$

The task also requires enough coordination capacity to handle the serial depth n. Write this horizon gate as $G_H(q_H - \log n)$. Conditional on clearing the horizon gate, the probability that all stages succeed is $S_\ell(q_F, \varphi)^n$. Thus the exact sequential success probability is

$$P(q, n, \ell, \varphi) = G_H(q_H - \log n)\left[1 - \{1 - G_F(q_F - \varphi)\}^{1+\ell}\right]^n.$$

Writing $h = \log n$, this is

$$P(q, n, \ell, \varphi) = G_H(q_H - h)\mathcal{G}_F(q_F - \varphi; h, \ell),$$

where

$$\mathcal{G}_F(u; h, \ell) = \left[1 - \{1 - G_F(u)\}^{1+\ell}\right]^{\exp(h)}.$$

The baseline specification in (1) is the common-kernel version of this exact product gate. In the exact sequential model, the reliability gate can vary with task horizon and retry technology.

Theorem 1 (Sequential reduction to horizon and fragility)**.** *Fix a required task-level failure probability $\varepsilon \in (0, 1)$, and suppose G_F is strictly increasing. Define*

$$p^*(n, \ell, \varepsilon) = 1 - \{1 - (1 - \varepsilon)^{1/n}\}^{1/(1+\ell)}$$

and

$$f(n, \ell, \varepsilon, \varphi) = \varphi + G_F^{-1}(p^*(n, \ell, \varepsilon)).$$

Then the reliability condition

$$S_\ell(q_F, \varphi)^n \geq 1 - \varepsilon$$

holds if and only if $q_F \geq f(n, \ell, \varepsilon, \varphi)$. Moreover f is increasing in n and φ, decreasing in the retry budget ℓ, and increasing as the tolerated failure probability ε falls.

The theorem gives the primitives economic content. Horizon is serial depth, $h = \log n$.

Fragility is the reliability threshold f. The two coordinates need not be statistically independent: long tasks can also require higher local reliability. This is not double counting because h is the coordination gate and f is the reliability gate. Long tasks can still be forgiving when errors are observable and retries are cheap. Short tasks can be fragile when the tolerated failure probability is extremely low or mistakes are irreversible. Verification and rollback lower effective fragility by raising ℓ or by making failures observable; unsafe deployment raises it.

3 Sufficient Statistics

Define the horizon boundary value

$$B_H(q) = \iint_{\mathcal{T}} a(h, f) g_H(q_H - h) G_F(q_F - f) \, dh \, df, \tag{4}$$

the forgiveness boundary value

$$B_F(q) = \iint_{\mathcal{T}} a(h, f) G_H(q_H - h) g_F(q_F - f) \, dh \, df, \tag{5}$$

and the cross-boundary statistic

$$C(q) = \iint_{\mathcal{T}} a(h, f) g_H(q_H - h) g_F(q_F - f) \, dh \, df. \tag{6}$$

Theorem 2 (Task-frontier sufficient statistics)**.** *Suppose $a \in L^1_+(\mathcal{T})$, G_H and G_F are continuously differentiable, and g_H, g_F are bounded and continuous. Then W is continuously differentiable and*

$$\frac{\partial W(q)}{\partial q_H} = B_H(q), \qquad \frac{\partial W(q)}{\partial q_F} = B_F(q).$$

Moreover B_H is differentiable in q_F, B_F is differentiable in q_H, and

$$\frac{\partial B_H(q)}{\partial q_F} = \frac{\partial B_F(q)}{\partial q_H} = C(q) \geq 0.$$

Thus forgiveness capability raises the marginal value of horizon capability by exactly $C(q)$, and horizon capability raises the marginal value of forgiveness capability by the same statistic.

Theorem 3 (Finite-step value and cross-boundary complementarity)**.** *Under the assumptions of Theorem 2, for every $\Delta = (\Delta_H, \Delta_F) \geq 0$,*

$$\begin{aligned} A(q, \Delta) &= \int_0^{\Delta_H} B_H(q_H + u, q_F) \, du + \int_0^{\Delta_F} B_F(q_H + \Delta_H, q_F + v) \, dv \qquad (7) \\ &= \int_0^{\Delta_F} B_F(q_H, q_F + v) \, dv + \int_0^{\Delta_H} B_H(q_H + u, q_F + \Delta_F) \, du. \qquad (8) \end{aligned}$$

The extra value from doing both improvements rather than adding the two stand-alone improvements is

$$\begin{aligned} I(q,\Delta) &= W(q_H+\Delta_H, q_F+\Delta_F) - W(q_H+\Delta_H, q_F) \\ &\quad - W(q_H, q_F+\Delta_F) + W(q_H, q_F) \\ &= \int_0^{\Delta_H}\int_0^{\Delta_F} C(q_H+u, q_F+v)\,dv\,du \geq 0. \end{aligned} \tag{9}$$

As $\|\Delta\| \downarrow 0$*,*

$$A(q,\Delta) = \Delta_H B_H(q) + \Delta_F B_F(q) + o(\|\Delta\|).$$

These two theorems are the paper's core. They replace a vague claim that a model is "better" with three sufficient statistics. B_H says whether the scarce dimension is serial depth. B_F says whether the scarce dimension is reliability in unforgiving environments. C says whether the economy is waiting for systems that can do both.

4 Classification and the One-Dimensional Special Case

The two primitives classify tasks without adding more state variables. Relative to the current frontier, the four regions are:

	Low fragility / high forgiveness	High fragility / low forgiveness
Low horizon	Product recommendation, ad copy, search help	Driving maneuvers, medical dosage, payments control
High horizon	Coding, debugging, data analysis, research assistance	Drug discovery, critical infrastructure, autonomous organizations

The classification is not fixed by nature. Task design moves tasks in this space. Tests, sandboxing, human review, monitoring, rollback, and staged deployment lower effective fragility. Real-time externalities, hidden errors, one-shot decisions, and delayed feedback raise it.

The same boundary logic gives a task-level interpretation of the sufficient statistics. A task matters most for horizon progress when it lies near the horizon boundary and has already cleared the fragility gate. It matters most for reliability progress when it has cleared the horizon gate and lies near the fragility boundary. It matters most for interaction when it sits near both boundaries. This is why the model treats tasks near the corner differently from tasks that are difficult along only one dimension.

The one-dimensional agentic frontier is a limiting case of this model. If the forgiveness margin is fixed, already solved, or economically invariant, define

$$\tilde{a}(h) = \int_{\mathcal{T}_f} a(h, f) G_F(q_F - f)\, df.$$

The relevant surplus for horizon progress becomes

$$\tilde{W}(q_H) = \int \tilde{a}(h) G_H(q_H - h)\, dh,$$

with boundary statistic

$$\tilde{B}(q_H) = \int \tilde{a}(h) g_H(q_H - h)\, dh.$$

Thus the original agentic racing model is not wrong. It is the right reduced form when the scarce margin is horizon alone. The merged paper uses the richer primitive because frontier AI deployment varies along both margins. The appendix gives the corresponding task-level derivative and reduction formally.

5 Race Technologies: Base, Runtime, Data, and Distillation

The task frontier is described by $q = (q_H, q_F)$, but firms do not purchase q directly. They choose technologies that produce it. Let firm i have technology vector

$$x_i = (b_i, s_i, \tau_i, v_i, d_i, m_i).$$

The component b_i is base-model stock. The runtime components are search or test-time compute s_i, tools, retrieval, memory, and workflow integration τ_i, and verification, monitoring, and rollback v_i. The component d_i is deployment-data stock, and m_i is distillation stock. Search expands the number of candidate paths. Tools and workflows expand the action space and make long tasks executable. Verification and monitoring determine whether runtime effort is reliable rather than merely longer. Deployment data improves these runtime components because the firm observes real tasks, failures, corrections, and successful workflows. Distillation makes expensive runtime behavior cheaper and more automatic, but only when the traces being distilled are informative and sufficiently clean.

Task capability is produced by

$$q_{iH} = H(x_i), \qquad q_{iF} = F(x_i).$$

Base scaling is the foundation-model race: model size, pretraining, post-training, data mixture, accelerator clusters, and distributed training systems. Runtime/harnessing is the deployed-agent-stack race: test-time compute, search, tools, memory, verifiers, retry loops, retrieval, agent-computer interfaces, and workflow integration. The maintained technology restrictions are

$$H_b \geq 0, \quad F_b > 0, \quad H_s, H_\tau > 0, \quad F_v > 0, \quad H_d, F_d, H_m, F_m \geq 0.$$

Search and tools are especially important for horizon, so H_s and H_τ are large. Base scaling and verification are especially important for reliability, so F_b and F_v are large. Unverified search and tool use mainly raise H and can even create engineering failure modes outside the baseline monotone case. The technologies may be complements:

$$H_{bs}, H_{b\tau}, H_{bv} \geq 0, \qquad F_{bs}, F_{b\tau}, F_{bv} \geq 0.$$

The economic value of each technology is determined by the boundary it moves. Let $q_i = (H(x_i), F(x_i))$. Ignoring strategic rent shifting to isolate task-frontier creation, the marginal value of any technology component $y \in \{b, s, \tau, v, d\}$ is

$$V_y = B_H(q_i) H_y(x_i) + B_F(q_i) F_y(x_i).$$

If firm i serves task mass M_i and distillation also reduces per-task runtime cost $c_R(m_i)$, with $c_R'(m_i) \leq 0$, then the marginal value of distillation is

$$V_m = B_H(q_i) H_m(x_i) + B_F(q_i) F_m(x_i) - M_i c_R'(m_i).$$

This chain rule connects the CS mechanisms to the economic task frontier. Base scaling is valuable when it moves the scarce task boundary through H_b or F_b. Runtime is valuable when search, tools, or verification move the boundary through their H or F derivatives. Deployment data is valuable because it raises the productivity of runtime. Distillation is valuable because it both raises effective capability and lowers future runtime cost. The same formula also explains why the object of the race can shift across markets: the technology with the highest derivative is not the technology with the highest economic value unless it moves the boundary that is scarce in that task region.

This technology layer is deliberately close to computer-science language. Scaling laws and compute-optimal training motivate b [Hoffmann et al.(2022)]. Test-time compute and verifier-guided search motivate s and v [Snell et al.(2024)]. Tool use, retrieval, memory, reflection, and agent interfaces motivate τ and v [Schick et al.(2023), Yang et al.(2024)]. Software-agent benchmarks show that interface design changes measured capability [Jimenez et al.(2023)].

Distillation transfers expensive runtime behavior into cheaper systems.

Market capture matters because it generates data. Let M_i be the task mass served by firm i. A simple data law of motion is

$$d_{i,t+1} = (1 - \delta_d) d_{i,t} + \lambda_d M_{i,t}.$$

Let successful or corrected runtime traces be

$$T_{i,t} = \iint_{\mathcal{R}_{i,t}} a(h, f)p(q_i, h, f)\omega(h, f)\, dh\, df,$$

where $\mathcal{R}_{i,t}$ is the region of tasks served by firm i, and $\omega(h, f)$ measures how informative a trace is for training, debugging, or distillation. This weight is high when outcomes are observed, failures are recoverable, and corrections identify the source of error; it is low when mistakes are hidden, catastrophic, or legally unusable. Distillation evolves as

$$m_{i,t+1} = (1 - \delta_m)m_{i,t} + \lambda_m T_{i,t}.$$

Thus runtime/harnessing creates current capability and a future data asset. Distillation turns that asset into lower cost and more durable task-specific capability. The data flywheel is strongest for forgiving tasks, where failures are observable, recoverable, and informative rather than catastrophic.

The model therefore has forward prediction power. First, open or low-cost systems should catch up fastest on high-horizon but forgiving tasks where tools, tests, retrieval, and retry loops generate useful feedback. Second, frontier base models should retain larger premia on fragile tasks unless a reliable external verifier can turn the task into a forgiving one. Third, deployment scale should matter most when user interactions produce high-quality traces, not merely when they produce many tokens. Fourth, distillation should first compress the cost of successful workflows before it fully erases frontier reliability gaps. Fifth, full-stack tipping should appear where base quality, runtime harnessing, deployment traces, and distillation are complements.

The sharp prediction is a race inversion across task regions. Let $R \subseteq \mathcal{T}$ be a task region and define regional boundary values

$$B^R_H(q) = \iint_R a(h, f)g_H(q_H - h)G_F(q_F - f)\, dh\, df,$$

$$B^R_F(q) = \iint_R a(h, f)G_H(q_H - h)g_F(q_F - f)\, dh\, df.$$

The ratio $\Theta_R(q) = B^R_H(q)/B^R_F(q)$ is the local scarcity of horizon relative to fragile reliability in that region.

Theorem 4 (Task-region race inversion and scalar failure)**.** *Consider two same-cost unit projects y and z. Project y is more horizon-intensive and project z is more reliability-intensive:*

$$d_{yH} > d_{zH}, \qquad d_{yF} < d_{zF},$$

where $d_y = (d_{yH}, d_{yF})$ and $d_z = (d_{zH}, d_{zF})$ are the induced local capability steps. Suppose

$B_H^R(q), B_F^R(q) > 0$. *Ignoring strategic rent shifting, project* y *has higher local frontier value than project* z *on region* R *if and only if*

$$\Theta_R(q) > \frac{d_{zF} - d_{yF}}{d_{yH} - d_{zH}}.$$

Project z *has higher local frontier value if the inequality is reversed. Hence, if two task regions lie on opposite sides of this threshold, the privately efficient race direction reverses across regions.*

No scalar capability model with region-invariant same-cost project increments can generate this reversal at a fixed state. In a scalar model, the local ranking is proportional to a single regional boundary value times $\gamma_y - \gamma_z$*, whose sign does not change across regions. This is a restricted scalar benchmark: if the scalar model is allowed to choose region-specific project productivities or adoption wedges, it can mimic the reversal, but only by putting the missing task heterogeneity back into the technology or cost side of the model.*

This theorem is the disciplining content of the two-dimensional primitive. A runtime project such as search, tools, and workflow integration can dominate in long forgiving work because it mainly relaxes the horizon boundary. A base-model or verification project can dominate in short fragile work because it mainly relaxes the reliability boundary. A scalar quality ladder can fit either ranking, but not the reversal without making the technology or costs task-region-specific. That move is exactly the two-dimensional task frontier in disguised form.

The same inversion can be stated for actual private incentives rather than frontier value alone. Let regional current revenue be $\sigma(r)W_R(q)$, where

$$W_R(q) = \iint_R a(h,f)p(q,h,f)\, dh\, df,$$

and consider small same-cost projects y and z. Write

$$D = (D_H, D_F) = d_y - d_z, \qquad D_H > 0 > D_F.$$

Theorem 5 (Private race inversion with rent shifting)**.** *At a fixed rival state, the first-order private value difference between project* y *and project* z *on region* R *is*

$$\sigma(r)\{B_H^R(q)D_H + B_F^R(q)D_F\} + \sigma'(r)W_R(q)\,\beta \cdot D.$$

Hence project y *is privately preferred to project* z *on region* R *if and only if*

$$\Theta_R(q) > \frac{-D_F}{D_H} - \frac{\sigma'(r)W_R(q)}{\sigma(r)B_F^R(q)}\frac{\beta \cdot D}{D_H}.$$

Project z *is privately preferred if the inequality is reversed. Thus rent shifting does not eliminate* *the task-region inversion; it shifts the threshold by an observable market-share term. If two regions lie on opposite sides of their adjusted thresholds, private investment rankings reverse across regions.*

The economic intuition is that private competition adds a business-stealing motive on top of frontier creation. A project can be attractive because it creates new feasible tasks, because it protects installed revenue, or both. But the business-stealing term does not erase the task frontier. It only tilts the cutoff. A runtime project still wins where the scarce boundary is horizon, and a reliability project still wins where the scarce boundary is fragility, unless market-share incentives are strong enough to move the region across the cutoff.

Corollary 1 (Race direction)**.** *Holding strategic rent shifting fixed, firms race on base scaling when V_b is high relative to the marginal cost of base investment. They race on runtime/harnessing when V_x is high for some $x \in \{s, \tau, v\}$ relative to runtime cost. They race for deployment and market capture when V_d is high. They distill when V_m is high. Full-stack racing is most valuable when base and runtime are technologically complementary and task mass lies near both boundaries, so that $C(q)$ is large.*

The corollary is the investment interpretation of the model. It maps the same sufficient statistics into observable races: GPU-intensive base scaling, inference-time harnessing, market capture for data, and distillation.

6 Low-Cost Imitation and Frontier Reliability

Let L be a frontier system and M a lower-cost follower. Suppose

$$q^L \geq q^M$$

component by component. For a task region $R \subseteq \mathcal{T}$, define delivered task value

$$V_i(R) = \iint_R a(h,f)p(q^i,h,f)\,dh\,df, \qquad i \in \{L, M\}.$$

Let $c_i(R)$ be the cost of serving the region and

$$\kappa(R) = c_L(R) - c_M(R)$$

be the follower's cost advantage.

For any task region R, the lower-cost follower is preferred on delivered net value exactly when its cost advantage exceeds the frontier system's performance advantage:

$$\kappa(R) \geq \iint_R a(h,f)\{p(q^L,h,f) - p(q^M,h,f)\}\,dh\,df. \tag{10}$$

If the performance gap on R is uniformly bounded above by ε, then the follower is preferred whenever

$$\kappa(R) \geq \varepsilon \iint_R a(h, f)\, dh\, df.$$

If the performance gap on R is uniformly bounded below by $\eta > 0$, then the frontier system is preferred whenever

$$\eta \iint_R a(h, f)\, dh\, df > \kappa(R).$$

This is the economics of imitation. Cheap followers win where the remaining performance gap is small relative to cost savings. That is most likely on forgiving tasks or tasks well behind both boundaries, because runtime tools, retries, and distillation can substitute for some base-model gap. Frontier systems retain value where the performance gap lies near a valuable fragile boundary, because runtime cannot fully repair errors that are irreversible, hidden, or costly. The appendix verifies the sorting condition step by step.

Current AI markets illustrate the distinction without being necessary for the theorem. Open or lower-cost models can reproduce workflows and interfaces and perform strongly on many agentic and coding benchmarks [GLM-4.5 Team(2025), Z.AI(2026)].

Runtime work shows that inference procedures and harness design can change task performance without changing the base model [Wei et al.(2022)]. Self-consistency, reasoning-and-acting agents, and agent-computer interfaces push the same point in more agentic settings [Wang et al.(2022a), Yao et al.(2022), Yang et al.(2024)]. Distillation can compress expensive runtime traces into cheaper future systems [Hinton et al.(2015), Hsieh et al.(2023)]. Self-generated reasoning and instruction data provide a related feedback channel. The model's point is not to rank specific systems. It is to distinguish the GPU-cluster base race from the deployment-data runtime race.

7 A Two-Period Race Decomposition

There are two firms. Firm i has capability q_i, and firm j has capability q_j. Relative market appeal is

$$r_i = \beta \cdot (q_i - q_j), \qquad \beta = (\beta_H, \beta_F) \geq 0.$$

Let $\sigma(r) \in (0, 1)$ be firm i's market share, increasing in r. Current revenue is

$$\pi_i(q_i, q_j) = \sigma(r_i) W(q_i).$$

Firm i's project is $\Delta_i \geq 0$. The rival either invests in project Δ_j or does not, $a_j \in \{0, 1\}$. The gross private benefit from investing rather than not investing, holding a_j fixed, is

$$\begin{aligned} P_i = &\ \sigma(r_i + \beta \cdot \Delta_i - a_j\beta \cdot \Delta_j)W(q_i + \Delta_i) \\ &- \sigma(r_i - a_j\beta \cdot \Delta_j)W(q_i). \end{aligned}$$

For every $q_i, q_j, \Delta_i, \Delta_j$, and $a_j \in \{0, 1\}$, the gross private benefit decomposes into task-frontier creation and rent shifting:

$$\begin{aligned} P_i = &\ \sigma(r_i + \beta \cdot \Delta_i - a_j\beta \cdot \Delta_j)A(q_i, \Delta_i) \\ &+ \{\sigma(r_i + \beta \cdot \Delta_i - a_j\beta \cdot \Delta_j) - \sigma(r_i - a_j\beta \cdot \Delta_j)\}W(q_i). \end{aligned} \tag{11}$$

The first term is task-frontier creation. The second term is rent shifting over the installed task stock. Locally, the productive component is governed by

$$\Delta_{iH}B_H(q_i) + \Delta_{iF}B_F(q_i).$$

The joint value of a project that raises both dimensions includes the cross-boundary interaction $I(q_i, \Delta_i)$.

Let $\alpha_i = \beta \cdot \Delta_i$ and $\alpha_j = \beta \cdot \Delta_j$. The current payoff incentive for firm i to invest is higher when the rival invests than when the rival does not invest exactly when

$$\begin{aligned} &\{\sigma(r_i + \alpha_i - \alpha_j) - \sigma(r_i + \alpha_i) - \sigma(r_i - \alpha_j) + \sigma(r_i)\}W(q_i) \\ &\quad \geq \{\sigma(r_i + \alpha_i) - \sigma(r_i + \alpha_i - \alpha_j)\}A(q_i, \Delta_i). \end{aligned} \tag{12}$$

The left side is the change in defensive rent preservation over $W(q_i)$. The right side is the lost preemption value from no longer being the only firm to move. Defensive racing is most likely when the installed task stock is large relative to the new frontier increment.

8 The Infinite-Horizon Markov Race

The formal race is a finite-state discounted stochastic game. The state is

$$s = (b_1, r_1, d_1, m_1, b_2, r_2, d_2, m_2, z),$$

where z is a cost state. The induced capability vector is

$$q_i(s) = \{H(b_i, s_i, \tau_i, v_i, d_i, m_i), F(b_i, s_i, \tau_i, v_i, d_i, m_i)\}.$$

Here $r_i = (s_i, \tau_i, v_i)$. Each firm chooses a project $a \in \mathcal{M}$. Project 0 is no investment. A project can be a base-scaling project, runtime/harness project, deployment/data project, or distillation project. It changes the firm's technology state and therefore induces a capability step

$$\Delta_{i,a}(s) = q_i(T_i(s,a)) - q_i(s),$$

where $T_i(s,a)$ denotes firm i's own post-project technology state before rival updates. Project a costs $K_a(z)$. Current payoff is

$$u_i(s,a,n) = \sigma(\beta \cdot [q_i + \Delta_{i,a}(s) - q_j - \Delta_{j,n}(s)])\, W(q_i + \Delta_{i,a}(s)) - K_a(z),$$

where n is the rival's project and $q_i, \Delta_{i,a}, \Delta_{j,n}$ are evaluated at state s. A transition rule $T(s,a,n)$ updates base stocks, runtime stocks, data stocks, distillation stocks, and the cost state. The discount factor is $\delta \in (0,1)$.

Theorem 6 (Dynamic race pressure)**.** *On any finite state grid and finite project menu, the discounted stochastic game has a stationary mixed Markov-perfect equilibrium. For a continuation value V_i, the payoff gain from project a rather than no investment against rival project n is*

$$\Phi_i^V(s,a,n) - K_a(z),$$

where

$$\begin{aligned}\Phi_i^V(s,a,n) = {} & \sigma(\beta \cdot [q_i + \Delta_{i,a} - q_j - \Delta_{j,n}])\, A(q_i, \Delta_{i,a}) \\ & + \{\sigma(\beta \cdot [q_i + \Delta_{i,a} - q_j - \Delta_{j,n}]) - \sigma(\beta \cdot [q_i - q_j - \Delta_{j,n}])\}\, W(q_i) \\ & + \delta\{V_i(T(s,a,n)) - V_i(T(s,0,n))\}. \end{aligned} \tag{13}$$

In equilibrium, every project used with positive probability maximizes the expected net gain over the rival's mixed project choice.

This theorem is the Markov version of the two-period decomposition. It keeps the economic objects visible. The first line is frontier creation. The second line is rent shifting. The third line is continuation value. The result matters because it shows that the same sufficient statistics still organize a fully dynamic race: dynamics add continuation values, but they do not change which task boundary a project moves.

For binary investment, write $\Phi_L(a_F)$ and $\Phi_F(a_L)$ for the race pressure of the leader and follower against the rival action. Let K be the project cost. At any state, holding continuation values fixed, the pure-action continuation game has the following equilibrium regimes:

$$\begin{aligned}
(0,0) \text{ is an equilibrium} &\iff \Phi_L(0) \leq K \text{ and } \Phi_F(0) \leq K,\\
(1,0) \text{ is an equilibrium} &\iff \Phi_L(0) \geq K \text{ and } \Phi_F(1) \leq K,\\
(0,1) \text{ is an equilibrium} &\iff \Phi_L(1) \leq K \text{ and } \Phi_F(0) \geq K,\\
(1,1) \text{ is an equilibrium} &\iff \Phi_L(1) \geq K \text{ and } \Phi_F(1) \geq K.
\end{aligned}$$

The preemption region in which a firm wants to invest only if the other does not is characterized by $\Phi_i(0) \geq K \geq \Phi_i(1)$.

The economic point is that race intensity is not monotone in technological opportunity alone. It also depends on whether investment is defensive, preemptive, or productive at the current state.

8.1 A Data-Distillation Tipping Threshold

The Markov formulation above is general. A simple one-dimensional restriction gives the core comparative static. Consider a symmetric market with total task mass M. Let $x_t = \beta\cdot(q_{1t}-q_{2t})$ be the market-relevant appeal gap and let firm 1 serve share $\sigma(x_t)$, where $\sigma(0) = 1/2$, $\sigma'(0) > 0$, and $\sigma(-x) = 1 - \sigma(x)$. Suppose ordinary catch-up shrinks gaps at rate $\rho \in [0, 1]$. At the symmetric state, a marginal unit of served task mass raises next period data by λ_d and raises next period distillation stock by $\lambda_m\bar{\omega}(q)$, where

$$\bar{\omega}(q) = \frac{1}{M}\iint_{\mathcal{R}} a(h,f)p(q,h,f)\omega(h,f)\,dh\,df$$

is the average usable trace intensity in the served task region. Let

$$q_d = (H_d, F_d), \qquad q_m = (H_m, F_m)$$

be the capability derivatives with respect to deployment data and distillation at the symmetric technology state. One unit of served mass therefore changes next-period appeal by

$$\chi(q) = \lambda_d\,\beta\cdot q_d + \lambda_m\bar{\omega}(q)\,\beta\cdot q_m. \tag{14}$$

The gap evolves as

$$x_{t+1} = (1-\rho)x_t + \chi(q)M\{2\sigma(x_t) - 1\}. \tag{15}$$

Theorem 7 (Data-distillation tipping threshold)**.** *The symmetric state $x = 0$ is locally stable if*

$$2\chi(q)M\sigma'(0) < \rho$$

and locally unstable if

$$2\chi(q)M\sigma'(0) > \rho.$$

For logit demand $\sigma(x) = 1/(1+\exp(-\kappa x))$*, the tipping threshold is*

$$\chi(q)M > \frac{2\rho}{\kappa}.$$

This is the structural data flywheel. Market scale alone does not guarantee tipping. Tipping requires market scale M, trace productivity $\bar{\omega}(q)$, data and distillation productivities (λ_d, λ_m), and demand sensitivity $\sigma'(0)$ to overcome ordinary catch-up ρ. The condition predicts tipping in tasks where deployment produces useful corrected trajectories and distillation can compress them into future capability. It predicts weaker tipping in fragile tasks where failures are hidden, catastrophic, or legally unavailable for learning, even if the task market is large.

The same condition also distinguishes productive learning from pure share lock-in. On a served region R, one marginal unit of served task mass creates local task-frontier value

$$\zeta_R(q) = \lambda_d\{B_H^R(q)H_d + B_F^R(q)F_d\} + \lambda_m\bar{\omega}_R(q)\{B_H^R(q)H_m + B_F^R(q)F_m\},$$

where $\bar{\omega}_R$ is the average usable trace intensity on R. If the tipping inequality holds but $\zeta_R(q)$ is below the opportunity cost of frontier learning, scale mainly reallocates market share. If both the tipping inequality and the productive learning condition below hold, the leader's market share also creates frontier value. For finite data or distillation steps, the extra joint value is given by the cross-boundary statistic C in Theorem 3.

Formally, let

$$\chi_R(q) = \lambda_d\,\beta\cdot q_d + \lambda_m\bar{\omega}_R(q)\,\beta\cdot q_m$$

and define the region-level lock-in index

$$L_R(q) = 2\chi_R(q)M_R\sigma'(0) - \rho.$$

Let $\nu \geq 0$ be the opportunity cost of one marginal unit of frontier learning on region R.

Proposition 1 (Productive tipping versus pure lock-in)**.** *On region R, the symmetric state is locally stable if $L_R(q) < 0$ and locally unstable if $L_R(q) > 0$. Conditional on local instability, scale creates productive tipping if $\zeta_R(q) > \nu$ and pure share lock-in if $\zeta_R(q) \leq \nu$. If $L_R(q) < 0$ but $\zeta_R(q) > \nu$, deployment creates productive learning without local market tipping.*

This proposition separates two mechanisms that are often conflated. A large installed base can make a leader harder to displace without improving the frontier. Conversely, a deployment channel can be socially productive even without local monopoly tipping if it generates traces that raise the scarce horizon or fragility boundary.

9 Tipping and Stopping

The vector model needs a scalar notion of market-relevant lead. Let

$$x = \beta \cdot (q_L - q_F) \geq 0$$

be the leader's appeal gap. Suppose catch-up compresses the inherited gap by fraction $\rho \in [0, 1]$. If the leader invests, its appeal rises by $\alpha_L = \beta \cdot \Delta_L$. If the follower invests, its appeal rises by $\alpha_F = \beta \cdot \Delta_F$. On a region where labels do not switch, the next appeal gap is

$$x' = (1 - \rho)x + \alpha_L a_L - \alpha_F a_F.$$

On the identity-preserving region,

$$\mathbb{E}[x' - x \mid s] = -\rho x + \alpha_L p_L(s) - \alpha_F p_F(s),$$

where $p_L(s)$ and $p_F(s)$ are the equilibrium investment probabilities. If the selected pure regime is leader-only racing, $(a_L, a_F) = (1, 0)$, the appeal gap expands iff $\alpha_L > \rho x$. If the selected pure regime is mutual racing and $\alpha_L = \alpha_F$, the gap contracts at rate ρx. If the selected pure regime is follower-only racing, the gap contracts by $\rho x + \alpha_F$. Leader tipping is therefore possible only when the leader-only regime conditions hold and the leader's market-relevant project step exceeds catch-up.

This expression gives the economic content of tipping. A lead does not grow simply because a leader is ahead. It grows when the equilibrium race selects leader-only investment and the leader's market-relevant step is large enough to dominate ordinary catch-up.

Stopping has a different logic. Suppose σ is Lipschitz with constant L_σ. Consider a sequence of states and projects with project step Δ_n, costs $K_n \geq \underline{K} > 0$, and continuation differences bounded by

$$|V_i(T(s_n, m, n_j)) - V_i(T(s_n, 0, n_j))| \leq L_V \|\Delta_n\|_1.$$

Let $\bar{B}_n$ bound $B_H + B_F$ on all own-capability rectangles crossed by the project, and let $\bar{W}_n$ bound current task surplus. If

$$\|\Delta_n\|_1 \{\bar{B}_n + L_\sigma \|\beta\|_\infty \bar{W}_n + \delta L_V\} \to 0,$$

then the maximum dynamic race pressure from the project converges to zero. Hence, for all sufficiently large n, no positive project is played in any best response.

Conversely, productive frontier value can vanish while private racing persists. If $A(q, \Delta)$ is arbitrarily small but the rent-shifting term in (11) exceeds the project cost, investment is privately profitable even though productive frontier creation is negligible.

Stopping is not caused by scaling slowdown alone. It occurs when the finite-step frontier value, the rent-shifting value, and the continuation value are all small relative to lumpy costs. Private racing can persist after productive frontier value has become small if the installed stock $W(q)$ is large and market share is sensitive to relative capability.

10 Private and Social Race Incentives

In a two-dimensional capability space, the best model need not dominate in every coordinate. A social planner can route each task to the model that gives the highest success probability. Define frontier task surplus for two systems by

$$S(q_1, q_2) = \iint_{\mathcal{T}} a(h, f) \max\{p(q_1, h, f), p(q_2, h, f)\}\, dh\, df. \tag{16}$$

This isolates task-frontier surplus. It omits consumer surplus from competition, safety externalities, data externalities, and compute-market incidence; those can be added around the frontier component.

The result in this section is deliberately narrow. It is not a complete welfare theorem for AI competition. It separates the frontier-surplus channel from other channels that may make follower entry socially valuable, including price competition, diffusion, resilience, and safety externalities.

Suppose the leader dominates the follower component by component, $q_L \geq q_F$, and the leader project is $\Delta \geq 0$. Then the social frontier gain from a leader project is

$$A(q_L, \Delta).$$

If the follower project Δ_F leaves the follower dominated,

$$q_F + \Delta_F \leq q_L,$$

then its social frontier gain, as measured by S, is zero. Yet the follower's private current gain can be positive:

$$\sigma(\beta \cdot [q_F + \Delta_F - q_L])A(q_F, \Delta_F) + \{\sigma(\beta \cdot [q_F + \Delta_F - q_L]) - \sigma(\beta \cdot [q_F - q_L])\}W(q_F) > 0.$$

Thus non-frontier catch-up can be privately profitable even when it does not raise the best available task frontier.

This wedge does not say that follower catch-up is socially worthless. Diffusion, resilience, price competition, and consumer surplus may be valuable. It says that those benefits are distinct from raising the task frontier. The decomposition separates frontier creation, business stealing, and socially valuable diffusion.

11 Conclusion

AI progress is not a scalar quality ladder. The economically relevant frontier moves through a two-dimensional task space: horizon and forgiveness. Those coordinates can be microfounded by serial tasks with error accumulation, retries, monitoring, rollback, and irreversibility. The sufficient statistics for frontier movement are B_H, B_F, and C. They say whether the next valuable step is longer agency, higher reliability in unforgiving environments, or a joint advance at the corner.

The central economic implication is race inversion. The same economy can rationally race on runtime, tools, and test-time compute in one task region and on base reliability or verification in another. Market scale then has two distinct effects: it can create pure share lock-in, or it can create productive learning when served tasks produce usable traces that move the scarce boundary. The one-dimensional agentic-racing model is the special case in which horizon is the only scarce task coordinate and the technology layer can be collapsed into a single capability step.

A Step-by-Step Mathematical Checks

A.1 Sequential Microfoundation

Let $\pi = G_F(q_F - \varphi)$. With ℓ retries, a stage fails only if all $1 + \ell$ attempts fail, so

$$S_\ell(q_F, \varphi) = 1 - (1 - \pi)^{1+\ell}.$$

The reliability requirement is

$$S_\ell(q_F, \varphi)^n \geq 1 - \varepsilon.$$

Taking the nth root and rearranging,

$$1 - (1 - \pi)^{1+\ell} \geq (1 - \varepsilon)^{1/n},$$

$$(1 - \pi)^{1+\ell} \leq 1 - (1 - \varepsilon)^{1/n},$$

and therefore

$$\pi \geq 1 - \{1 - (1 - \varepsilon)^{1/n}\}^{1/(1+\ell)} = p^*(n, \ell, \varepsilon).$$

Since G_F is strictly increasing, this is equivalent to

$$q_F \geq \varphi + G_F^{-1}(p^*(n, \ell, \varepsilon)).$$

The monotonicity follows from the expression for p^*. Increasing n lowers $1-(1-\varepsilon)^{1/n}$, which raises p^*. Increasing ℓ lowers p^*. Lowering the tolerated failure probability ε also raises p^*. The effect of φ is direct.

A.2 Sufficient Statistics

Start with

$$W(q) = \iint a(h,f) G_H(q_H - h) G_F(q_F - f)\, dh\, df.$$

The assumptions dominate the integrand and the derivative integrands. Differentiating under the integral sign,

$$\frac{\partial}{\partial q_H}\{G_H(q_H - h) G_F(q_F - f)\} = g_H(q_H - h) G_F(q_F - f),$$

so

$$\frac{\partial W(q)}{\partial q_H} = B_H(q).$$

Similarly,

$$\frac{\partial W(q)}{\partial q_F} = B_F(q).$$

Differentiating B_H in q_F gives

$$\frac{\partial B_H(q)}{\partial q_F} = \iint a(h,f) g_H(q_H - h) g_F(q_F - f)\, dh\, df = C(q).$$

Differentiating B_F in q_H gives the same expression. Since all terms are nonnegative, $C(q) \geq 0$.

A.3 Finite-Step Value

Move first in the horizon direction:

$$W(q_H + \Delta_H, q_F) - W(q_H, q_F) = \int_0^{\Delta_H} B_H(q_H + u, q_F)\, du.$$

Then move in the forgiveness direction:

$$W(q_H + \Delta_H, q_F + \Delta_F) - W(q_H + \Delta_H, q_F) = \int_0^{\Delta_F} B_F(q_H + \Delta_H, q_F + v)\, dv.$$

Adding gives (7). Reversing the path gives (8).

For the interaction term,

$$\begin{aligned} I(q,\Delta) = {} & \{W(q_H + \Delta_H, q_F + \Delta_F) - W(q_H, q_F + \Delta_F)\} \\ & - \{W(q_H + \Delta_H, q_F) - W(q_H, q_F)\}. \end{aligned}$$

Using the horizon derivative,

$$I(q, \Delta) = \int_0^{\Delta_H} \{B_H(q_H + u, q_F + \Delta_F) - B_H(q_H + u, q_F)\} \, du.$$

Using the forgiveness derivative of B_H,

$$B_H(q_H + u, q_F + \Delta_F) - B_H(q_H + u, q_F) = \int_0^{\Delta_F} C(q_H + u, q_F + v) \, dv.$$

Substituting gives (9). The local expansion follows from differentiability of W.

A.4 Local Salience

The task-level integrand is

$$a(h, f)G_H(q_H - h)G_F(q_F - f).$$

Differentiating with respect to q_H and q_F gives the two first-order terms in the local salience expression in Section 4. The cross derivative of the integrand is

$$a(h, f)g_H(q_H - h)g_F(q_F - f),$$

which gives the small-rectangle interaction.

A.5 One-Dimensional Reduction

If q_F is fixed, then

$$W(q_H, q_F) = \int \left[\int a(h, f)G_F(q_F - f) \, df \right] G_H(q_H - h) \, dh.$$

The bracketed term is $\tilde{a}(h)$. Differentiating this one-dimensional surplus with respect to q_H gives $\tilde{B}(q_H)$. This proves the one-dimensional reduction in Section 4.

A.6 Technology Race Values

The induced capability vector is

$$q_i(b_i, s_i, \tau_i, v_i, d_i, m_i) = \{H(b_i, s_i, \tau_i, v_i, d_i, m_i), F(b_i, s_i, \tau_i, v_i, d_i, m_i)\}.$$

By Theorem 2, the gradient of W with respect to q is

$$\nabla_q W(q_i) = (B_H(q_i), B_F(q_i)).$$

For any technology component $y \in \{b, s, \tau, v, d\}$, the chain rule gives

$$\frac{\partial W(q_i)}{\partial y_i} = B_H(q_i)H_y(x_i) + B_F(q_i)F_y(x_i).$$

For distillation,

$$\frac{\partial}{\partial m_i}\{W(q_i) - M_i c_R(m_i)\} = B_H(q_i)H_m(x_i) + B_F(q_i)F_m(x_i) - M_i c'_R(m_i).$$

This proves the technology value formulas in Section 5. Corollary 1 follows by comparing each marginal value with the corresponding marginal project cost. The statement about full-stack racing follows because technological complementarity raises the relevant derivatives of H and F, while Theorem 3 shows that task-side complementarity is governed by the cross-boundary statistic $C(q)$.

A.7 Task-Region Race Inversion

On region R, the local frontier value of project y is

$$B_H^R(q)d_{yH} + B_F^R(q)d_{yF},$$

and the corresponding value of project z is

$$B_H^R(q)d_{zH} + B_F^R(q)d_{zF}.$$

Project y is preferred if and only if

$$B_H^R(q)(d_{yH} - d_{zH}) > B_F^R(q)(d_{zF} - d_{yF}).$$

Since $d_{yH} > d_{zH}$ and $B_F^R(q) > 0$, division gives

$$\frac{B_H^R(q)}{B_F^R(q)} > \frac{d_{zF} - d_{yF}}{d_{yH} - d_{zH}}.$$

The reverse inequality gives preference for z. If two regions have ratios on opposite sides of the threshold, the ranking reverses.

In a scalar model with same-cost region-invariant increments, the local value difference between y and z on region R is

$$\tilde{B}^R(\bar{q})(\gamma_y - \gamma_z).$$

Because $\tilde{B}^R(\bar{q}) \geq 0$, the sign is the sign of $\gamma_y - \gamma_z$ and cannot reverse across regions. This proves Theorem 4.

A.8 Private Race Inversion

Regional private revenue after a small project d is

$$\sigma(r + \beta \cdot d)W_R(q + d).$$

The first-order expansion around $d = 0$ is

$$\sigma(r)W_R(q) + \sigma(r)\{B_H^R(q)d_H + B_F^R(q)d_F\} + \sigma'(r)W_R(q)\beta \cdot d.$$

Subtracting the expansion for project z from that for project y gives

$$\sigma(r)\{B_H^R(q)D_H + B_F^R(q)D_F\} + \sigma'(r)W_R(q)\beta \cdot D.$$

Project y is preferred if this expression is positive. Rearranging and dividing by

$$\sigma(r)B_F^R(q)D_H > 0$$

gives the stated threshold. The reverse inequality gives preference for z. This proves Theorem 5.

A.9 Cost-Performance Sorting

The follower is preferred iff

$$V_M(R) - c_M(R) \geq V_L(R) - c_L(R).$$

Rearranging gives

$$c_L(R) - c_M(R) \geq V_L(R) - V_M(R),$$

which is (10). The uniform upper and lower gap conditions follow by bounding $p(q^L, h, f) - p(q^M, h, f)$ on R.

A.10 Race Decomposition

By definition,

$$\begin{aligned} P_i = {} & \sigma(r_i + \beta \cdot \Delta_i - a_j\beta \cdot \Delta_j)W(q_i + \Delta_i) \\ & - \sigma(r_i - a_j\beta \cdot \Delta_j)W(q_i). \end{aligned}$$

Add and subtract

$$\sigma(r_i + \beta \cdot \Delta_i - a_j\beta \cdot \Delta_j)W(q_i).$$

This gives (11) because

$$A(q_i, \Delta_i) = W(q_i + \Delta_i) - W(q_i).$$

A.11 Strategic Interaction

Let $\alpha_i = \beta \cdot \Delta_i$ and $\alpha_j = \beta \cdot \Delta_j$. The incentive when the rival invests is

$$P_i(1) = \sigma(r_i + \alpha_i - \alpha_j)W(q_i + \Delta_i) - \sigma(r_i - \alpha_j)W(q_i).$$

The incentive when the rival does not invest is

$$P_i(0) = \sigma(r_i + \alpha_i)W(q_i + \Delta_i) - \sigma(r_i)W(q_i).$$

Subtracting and writing $W(q_i + \Delta_i) = W(q_i) + A(q_i, \Delta_i)$ gives

$$\begin{aligned} P_i(1) - P_i(0) = {}& \{\sigma(r_i + \alpha_i - \alpha_j) - \sigma(r_i + \alpha_i)\}A(q_i, \Delta_i) \\ & + \{\sigma(r_i + \alpha_i - \alpha_j) - \sigma(r_i + \alpha_i) - \sigma(r_i - \alpha_j) + \sigma(r_i)\}W(q_i). \end{aligned}$$

Rearranging gives (12).

A.12 Dynamic Race Pressure and Regimes

For fixed rival project n, the current payoff difference between project a and no investment is

$$\begin{aligned} &\sigma(\beta \cdot [q_i + \Delta_{i,a} - q_j - \Delta_{j,n}])W(q_i + \Delta_{i,a}) - K_a \\ &\quad - \sigma(\beta \cdot [q_i - q_j - \Delta_{j,n}])W(q_i). \end{aligned}$$

Apply the race decomposition and add the continuation difference

$$\delta\{V_i(T(s, a, n)) - V_i(T(s, 0, n))\}.$$

This proves Theorem 6. The regime theorem follows by applying the two best-response inequalities to the four pure action profiles.

A.13 Data-Distillation Tipping

The gap map is

$$\Psi(x) = (1 - \rho)x + \chi(q)M\{2\sigma(x) - 1\}.$$

Because $\sigma(0) = 1/2$, $x = 0$ is a steady state. Differentiating,

$$\Psi'(0) = 1 - \rho + 2\chi(q)M\sigma'(0).$$

Since $\rho \in [0, 1]$ and $\chi(q) \geq 0$, the derivative is nonnegative. The standard one-dimensional local stability condition $|\Psi'(0)| < 1$ is therefore equivalent to $\Psi'(0) < 1$, which is

$$2\chi(q)M\sigma'(0) < \rho.$$

If the inequality is reversed, the symmetric state is locally unstable: a small positive lead grows rather than shrinks. For logit demand,

$$\sigma'(0) = \frac{\kappa}{4},$$

so instability requires $\chi(q)M > 2\rho/\kappa$.

A.14 Productive Tipping

The region-level map replaces $\chi(q)$ and M by $\chi_R(q)$ and M_R. Its derivative at the symmetric state is

$$1 - \rho + 2\chi_R(q)M_R\sigma'(0) = 1 + L_R(q).$$

Thus the symmetric state is locally stable when $L_R(q) < 0$ and locally unstable when $L_R(q) > 0$. The local frontier value created by one marginal unit of served mass is $\zeta_R(q)$. Comparing this value with the opportunity cost ν gives the productive-learning classification. This proves Proposition 1.

A.15 Tipping

On the identity-preserving region,

$$x' = (1 - \rho)x + \alpha_L a_L - \alpha_F a_F.$$

Taking expectations gives

$$\mathbb{E}[x' - x \mid s] = -\rho x + \alpha_L p_L(s) - \alpha_F p_F(s).$$

Substituting pure regimes gives the cases stated in Section 9.

A.16 Stopping

The finite-step frontier value satisfies

$$A(q, \Delta_n) \leq \|\Delta_n\|_1 \bar{B}_n.$$

The market-share change is bounded by

$$L_\sigma \|\beta\|_\infty \|\Delta_n\|_1.$$

Thus the rent-shifting term is bounded by

$$L_\sigma \|\beta\|_\infty \|\Delta_n\|_1 \bar{W}_n.$$

The continuation term is bounded by $\delta L_V \|\Delta_n\|_1$. Adding the three bounds proves that race pressure converges to zero under the stopping condition stated in Section 9. Positive costs then rule out positive projects in best response. Private persistence follows immediately if the rent-shifting term alone exceeds cost while $A(q, \Delta)$ is arbitrarily small.

A.17 Private-Social Frontier Wedge

If $q_L \geq q_F$, then monotonicity of G_H and G_F implies

$$p(q_L, h, f) \geq p(q_F, h, f)$$

for every task. Hence $S(q_L, q_F) = W(q_L)$. A leader project changes social frontier surplus by $W(q_L + \Delta) - W(q_L) = A(q_L, \Delta)$. If the follower remains dominated after its project, then $S(q_L, q_F + \Delta_F) = W(q_L)$, so its frontier social gain is zero. The private gain expression is the race decomposition with $q_i = q_F$, $q_j = q_L$, and no rival investment.

References

[Acemoglu and Autor(2011)] Acemoglu, Daron, and David Autor. 2011. “Skills, Tasks and Technologies: Implications for Employment and Earnings.” *Handbook of Labor Economics* 4B: 1043–1171.

[Acemoglu and Restrepo(2018)] Acemoglu, Daron, and Pascual Restrepo. 2018. “The Race between Man and Machine: Implications of Technology for Growth, Factor Shares, and Employment.” *American Economic Review* 108(6): 1488–1542.

[Acemoglu and Restrepo(2019)] Acemoglu, Daron, and Pascual Restrepo. 2019. “Automation and New Tasks: How Technology Displaces and Reinstates Labor.” *Journal of Economic Perspectives* 33(2): 3–30.

[Aghion et al.(2001)] Aghion, Philippe, Christopher Harris, Peter Howitt, and John Vickers. 2001. “Competition, Imitation and Growth with Step-by-Step Innovation.” *Review of Economic Studies* 68(3): 467–492.

[Anthropic(2026)] Anthropic. 2026. “Claude Sonnet.” `https://www.anthropic.com/claude/sonnet`.

[Bresnahan and Trajtenberg(1995)] Bresnahan, Timothy F., and Manuel Trajtenberg. 1995. "General Purpose Technologies: Engines of Growth?" *Journal of Econometrics* 65(1): 83–108.

[DeepSeek-AI(2025)] DeepSeek-AI. 2025. "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning." arXiv:2501.12948.

[Fink(1964)] Fink, A. M. 1964. "Equilibrium in a Stochastic n-Person Game." *Journal of Science of the Hiroshima University, Series A-I* 28: 89–93.

[Gadre et al.(2024)] Gadre, Samir Yitzhak, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Jenia Jitsev, Luca Soldaini, Alexandros G. Dimakis, Gabriel Ilharco, Pang Wei Koh, Shuran Song, Thomas Kollar, Yair Carmon, Achal Dave, Reinhard Heckel, Niklas Muennighoff, and Ludwig Schmidt. 2024. "Language Models Scale Reliably with Over-Training and on Downstream Tasks." arXiv:2403.08540.

[GLM-4.5 Team(2025)] GLM-4.5 Team. 2025. "GLM-4.5: Agentic, Reasoning, and Coding (ARC) Foundation Models." arXiv:2508.06471.

[Hinton et al.(2015)] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. 2015. "Distilling the Knowledge in a Neural Network." arXiv:1503.02531.

[Hsieh et al.(2023)] Hsieh, Cheng-Yu, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. "Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes." arXiv:2305.02301.

[Jimenez et al.(2023)] Jimenez, Carlos E., John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?" arXiv:2310.06770.

[Hoffmann et al.(2022)] Hoffmann, Jordan, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, John Rae, Oriol Vinyals, and Laurent Sifre. 2022. "Training Compute-Optimal Large Language Models." arXiv:2203.15556.

[Kwa et al.(2025)] Kwa, Thomas, Ben West, Joel Becker, Ann Deng, Kiara Garcia, Megan Kinniment, Nick Rush, Sevon Von Arx, Ryan Bloom, Tobias Broadley, Huy Du, Ben Goodrich, Nikola Jurkovic, Lawrence Miles, Sawyer Nix, Tony Lin, Nikhil Parikh, David Rein, Lucas Sato, Hanna Wijk, Daniel Ziegler, Elizabeth Barnes, and Lawrence Chan. 2025. "Measuring AI Ability to Complete Long Software Tasks." arXiv:2503.14499.

[Pindyck(1991)] Pindyck, Robert S. 1991. "Irreversibility, Uncertainty, and Investment." *Journal of Economic Literature* 29(3): 1110–1148.

[Schick et al.(2023)] Schick, Timo, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. "Toolformer: Language Models Can Teach Themselves to Use Tools." arXiv:2302.04761.

[Shapley(1953)] Shapley, Lloyd S. 1953. "Stochastic Games." *Proceedings of the National Academy of Sciences* 39(10): 1095–1100.

[Shinn et al.(2023)] Shinn, Noah, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. "Reflexion: Language Agents with Verbal Reinforcement Learning." arXiv:2303.11366.

[Snell et al.(2024)] Snell, Charlie, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. "Scaling LLM Test-Time Compute Optimally Can Be More Effective than Scaling Model Parameters." arXiv:2408.03314.

[Vesely et al.(1981)] Vesely, W. E., F. F. Goldberg, N. H. Roberts, and D. F. Haasl. 1981. *Fault Tree Handbook*. U.S. Nuclear Regulatory Commission, NUREG-0492.

[Wang et al.(2022a)] Wang, Xuezhi, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. "Self-Consistency Improves Chain of Thought Reasoning in Language Models." arXiv:2203.11171.

[Wang et al.(2022b)] Wang, Yizhong, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. "Self-Instruct: Aligning Language Models with Self-Generated Instructions." arXiv:2212.10560.

[Wei et al.(2022)] Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." arXiv:2201.11903.

[Wijk et al.(2024)] Wijk, Hanna, Tony Lin, Joel Becker, Kiara Garcia, Megan Kinniment, Lucas Sato, and coauthors. 2024. "RE-Bench: Evaluating Frontier AI R&D Capabilities of Language Model Agents." arXiv:2411.15114.

[Yang et al.(2024)] Yang, John, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. "SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering." arXiv:2405.15793.

[Yao et al.(2022)] Yao, Shunyu, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. "ReAct: Synergizing Reasoning and Acting in Language Models." arXiv:2210.03629.

[Zelikman et al.(2022)] Zelikman, Eric, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. "STaR: Bootstrapping Reasoning With Reasoning." arXiv:2203.14465.

[Z.AI(2026)] Z.AI. 2026. "Claude Code: Methods for Using the GLM Coding Plan in Claude Code." https://docs.z.ai/scenario-example/develop-tools/claude.