The Predict–Verify–Act Trilemma in Complex Adaptive Systems: Latency and Noncommutativity

A Structural Restriction Law for One-Epoch Decision-Making

Duo Yi*
yiduo@zhidemai.com
ZHI-TECH GROUP

Ning Yang
yangning@zhidemai.com
ZHI-TECH GROUP

November 6, 2025

Abstract

This paper presents a structural basis for a *Predict* (P)–*Verify* (V)–*Act* (A) trilemma within a single decision epoch in online decision systems *and complex adaptive systems (CAS)*. We identify *one impossibility mechanism and one diagnostic dimension*: (1) **Latency** (time-constraint, hard no-go). A verification lag τ that exceeds the finite action window Δ creates a fundamental conflict between V and A. We derive quantitative lower bounds $\lambda(\tau, \Delta)$ and show that when $\tau \geq \Delta$ no strategy can achieve a zero gap (Theorem 1). (2) **Noncommutativity** (order-sensitivity, explanatory/diagnostic). We model verification V_τ, time evolution E, physical action A_{phys}, and disclosure A_{disc} as operators; nonzero commutators among them (due to drift and reflexivity *ubiquitous in CAS*) induce holonomy, *breaking symmetry across orderings* and explaining why no single ordering is obviously optimal. This motivates routing as a principled response rather than asserting a universal impossibility from noncommutativity alone.

We treat *P–V–A* as the *explicit* triangle of objectives within an epoch, while the time evolution E serves as a *latent structural* operator; in the *V–A* trade we refer to E only through *lag vs. deadline* (τ, Δ), avoiding explicit E-dependence in the proof of Theorem 1. *This framing targets online decision pipelines as instances of CAS, where predictions, verification feedback, and actions co-evolve with their environment.*

Our framework offers theoretical foundations for recent architectural innovations in AI systems (e.g., OpenAI's GPT-5 routing), while extending to policy evaluation, distributed systems, and beyond.

1 Overview

Systems that learn and act online aim to be *predictive*, *verifiable*, and *actionable* within a finite window $[t_0, t_0 + \Delta]$. Our main results establish a *hard* impossibility driven by **latency**—when the verification lag τ meets or exceeds the action window Δ, a strictly positive gap is unavoidable (Theorem 1). In contrast, **noncommutativity** provides an *explanatory/diagnostic* account of order sensitivity, clarifying why no single ordering is obviously optimal and thereby motivating routing (Theorem 2). We thus give a first-principles account in terms of two governing tensions: **latency** and **noncommutativity**.

Positioning. Our core contribution is a unified account of two fundamental constraints—*latency* (lag vs. deadline, (τ, Δ)) and *noncommutativity* (order sensitivity)—that explains classic dilemmas (Lucas critique, distribution shift, and "analysis paralysis") within a single framework and yields a design philosophy, the *Routing Doctrine*. **Latency** is our hard no-go engine: when

*Primary contact.

$\tau \geq \Delta$, Theorem 1 forces a strictly positive gap. **Noncommutativity** is *explanatory/diagnostic*: Theorem 2 (and Proposition 3) clarifies why no single ordering is obviously optimal and why simple reorderings materially change outcomes; it justifies routing but does *not* by itself claim a universal impossibility.

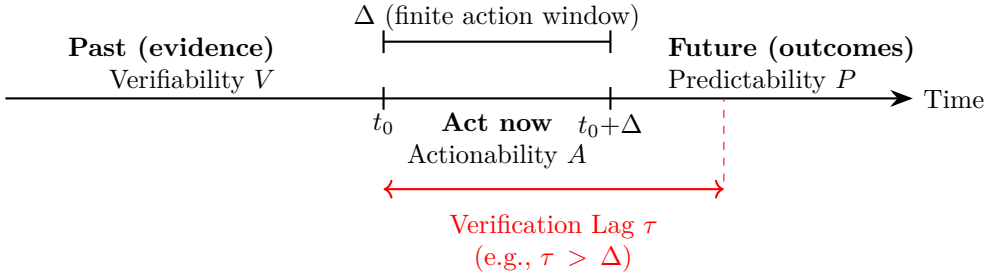

Figure 1: Decision timeline highlighting the conflict between the finite action window Δ and a potentially longer verification lag τ.

Contributions.

- **Unified lens.** A single framework that reconciles *latency* and *noncommutativity*, explaining classic dilemmas and guiding system design.
- **Hard no-go via latency.** An explicit lower bound $\lambda(\tau, \Delta)$ that becomes strictly positive when $\tau \geq \Delta$ (Theorem 1), covering many real-world pipelines.
- **Diagnostic noncommutativity.** Results that *break symmetry* across operator orderings and *motivate routing*, not a universal impossibility claim (Theorem 2, Proposition 3).
- **Routing Doctrine.** A practical rule—"*whoever dominates, obey it*"—that chooses the working edge and ordering under the present constraint profile.

1.1 From Engineering Observation to Theoretical Necessity

Fast intuition (the ice-cream trilemma). Picture a busy ice-cream shop near closing time: popular flavors may sell out soon (a short action window Δ), and getting evidence takes time (samples, asking others, reading reviews $\Rightarrow$ a verification lag τ). You want three things at once: *predict* the best flavor before buying (P), *see evidence* first (V), and *eat now* (A).
Why the three can' t be maximized together.

- **PV (know & verify) $\Rightarrow$ lose A.** Waiting for evidence consumes time; if τ eats your short window Δ, the flavor is gone or the shop closes.
- **VA (verify fast & act) $\Rightarrow$ lose P.** Quick cues (top-seller tags, what others picked) let you act now, but they can miss the globally best choice you would discover with deeper prediction.
- **PA (predict & act) $\Rightarrow$ lose V.** Acting on a confident hunch ships immediately, but you lack evidence; if you' re wrong, the error isn' t caught before you commit.

Rule of thumb. If stock-out/closing pressure dominates, go **VA**. If the flavor is unfamiliar and you are risk-averse, go **PV**. If you' re an expert with strong priors, a cautious **PA** bet is defensible.
Link to our formal setting. The tension comes from a *lag* τ to verify and a finite *window* Δ to act: when $\tau \geq \Delta$, PV kills A (our Latency No-Go); even with $\tau < \Delta$, the *order* of P,V,A still matters (our Noncommutativity Diagnostic).

By August 2025, the AI community had begun documenting OpenAI's GPT-5 routing architecture. Technical analyses (Pandit, 2025) described the system's four routing factors—conversation type, task complexity, tool needs, and explicit intent—and noted its efficiency gains. However, these accounts remained descriptive: *what* GPT-5 does, and *how* it works.

The theoretical question remained unasked: *why* must such a system exist at all?

Our framework emerged from recognizing that GPT-5's architecture is not an engineering optimization, but a response to a fundamental constraint. The catalyst came from a September 2025 discussion of the Chinese internet phrase "既要又要还要" (*jì yào yòu yào hái yào*): wanting to satisfy demands of the *past* (既), *present* (又), and *future* (还) simultaneously. This folk expression unwittingly encodes the P-V-A structure: Verification (past), Action (present), Prediction (future)—three temporal orientations that cannot be jointly optimized. This colloquial expression of frustrated ambition—ubiquitous in internet discourse—crystallized an implicit question that had been forming across disparate domains:

Why is simultaneous optimization consistently impossible?

Within days of this recognition, the pattern became unmissable:

- The Lucas critique (1976): policy evaluation fails under intervention [1].
- Goodhart's law (1975): measured metrics cease to be valid targets [2, 3].
- The CAP theorem (2002): distributed systems cannot achieve consistency and availability under partition [4, 5].
- Covariate shift (ML): training distributions diverge from deployment [6, 7].
- Analysis paralysis (psychology): deliberation consumes action windows; see the latency trade in Definition 2.
- And—yes—GPT-5's routing architecture.

These were not separate problems requiring separate solutions. They were manifestations of a single structural constraint: temporal decision systems cannot simultaneously optimize Prediction (P), Verification (V), and Action (A) within a finite epoch.

The distinction is crucial. Engineering documentation describes GPT-5's routing as addressing "efficiency" and "specialization." Our framework reveals it as addressing *impossibility*. No amount of engineering can overcome the P–V–A trilemma; routing is not an optimization—it is the only viable response.

The independent convergence between our theoretical derivation and GPT-5's empirical architecture—with the latter developed by OpenAI without knowledge of this framework and documented by the community before our formalization—validates both. When theory predicts what practice has already been discovered, it suggests we have captured a fundamental law rather than explained a particular solution.

1.2 Why Formalize the "Obvious"?

Practitioners have long recognized trade-offs in decision systems: software engineers know the "iron triangle" of cost-time-quality, economists cite the Lucas critique, ML researchers manage dataset shift. One might ask: if the P–V–A trilemma is "obvious," what is the value of formalizing it?

We answer with three contributions that go beyond informal recognition:

1. Precise impossibility conditions. Prior work describes trade-offs qualitatively ("there is tension between X and Y"). We provide exact conditions under which joint optimization is impossible:

- *Latency no-go*: When verification lag $\tau \geq \Delta$ (action window), **no strategy** can achieve zero gap (Theorem 1).
- *Noncommutativity diagnostic*: When $\varepsilon^* > 0$, **not all orderings** can be optimal (Theorem 2).

This is the difference between "voting systems have flaws" (known before Arrow) and "no voting system satisfying these axioms exists" (Arrow's impossibility theorem) [8, 9].

2. Quantitative lower bounds. We derive explicit bounds on unavoidable loss:

$$\max\{L_V(\pi), L_A(\pi)\} \geq \lambda(\tau, \Delta) = \min\{A_{\max}, L_V^{\text{time}} \cdot (\tau - \Delta)\}$$

when $\tau \geq \Delta$. This is actionable: system designers can compute whether their pipeline is in the no-go regime.

3. Unification across domains. We show that seemingly disparate phenomena—Lucas critique (economics), Goodhart's law (policy), CAP theorem (distributed systems), covariate shift (ML), software engineering trilemmas—are manifestations of the *same structural constraint.* This unification is novel.

Historical precedent. Arrow's impossibility theorem (1950) formalized what political scientists "knew": perfect voting doesn't exist. Yet Arrow's proof transformed the field by providing rigorous foundations. Similarly, while P–V–A trade-offs are "obvious" to practitioners, no prior work has: (i) formalized the constraints, (ii) quantified the bounds, or (iii) unified the phenomena. That is our contribution.

2 Model and Notation

Methodological Note. The objectives P, V, and A are not ad hoc; they emerge from the temporal structure of decision-making itself. Any system that (i) performs long-horizon prediction P_H about the future, (ii) validates against past/present evidence via V_τ, and (iii) acts in the present through A_{phys}, necessarily exhibits this triadic structure. Our contribution is not to define these objectives, but to formalize why operational constraints—verification lag vs. deadline (τ, Δ) and noncommutativity measured by $\varepsilon_\star$—create *max-type* shortfalls that preclude a trivial joint optimum within a single epoch: when $\tau \geq \Delta$ no joint maximizer exists, and under $\varepsilon_\star > 0$ not all orderings can be optimal. The impossibility/diagnostic phenomena arise from the system dynamics E (drift/reflexivity) and the sequential nature of V_τ and A_{phys}, not from how the objectives are defined.

Windowed Verifiability. Let $V^\Delta(\pi)$ denote the verifiability score that only credits evidence produced no later than $t_0 + \Delta$. Evidence arriving after the deadline does not contribute to V^Δ.

Assumption 1 (Sequentiality and Non-parallelism)**.** Within the decision epoch $[t_0, t_0 + \Delta]$, verification and action cannot be parallelized. Any policy π must realize one of the sequential schedules: $\mathsf{verify} \to \mathsf{act}$ or $\mathsf{act} \to \mathsf{verify}$, where verification consumes wall-clock lag $\tau > 0$ to become decision-relevant. Missing the action deadline yields zero actionable reward within the epoch.

Assumption 2 (Temporal Regularity of V^{Δ})**.** There exists $L_V^{\text{time}} > 0$ such that for all $\tau \geq 0$,

$$V_{\max} - V^{\Delta}(\tau) \;\geq\; L_V^{\text{time}} \cdot (\tau - \Delta)_+,$$

with $V^{\Delta}(0) = V_{\max}$ and $(x)_+ = \max\{0, x\}$.

Assumption 3 (Pairwise Swap-Sensitivity)**.** For each operator pair (X, Y) drawn from the feasible library and for a non-null set of distributions $\mathcal{D}_{X,Y} \subseteq \mathcal{P}$, there exists $K_{X,Y} > 0$ such that for all $\mu \in \mathcal{D}_{X,Y}$,

$$\begin{aligned}\max\big\{&|P(XY\mu) - P(YX\mu)|,\ |V(XY\mu) - V(YX\mu)|,\\ &|A(XY\mu) - A(YX\mu)|\big\} \geq K_{X,Y} \cdot d(XY\mu, YX\mu).\end{aligned}$$

Operators and Objectives. Let (Ω, Σ) be a measurable space and $\mathcal{P}$ the set of probability laws on it, equipped with a metric d (e.g., TV or W_1). We fix a start time t_0, window $\Delta > 0$, and lag $\tau > 0$. We define:

- Operators (acting on $\mu \in \mathcal{P}$): E (evolution), V_τ (verification), A_{phys} (physical action), A_{disc} (disclosure), P_H (prediction operator, mapping $\mu \to \mu_f$).
- Objectives (Lipschitz continuous mappings $\mathcal{P} \to \mathbb{R}$): $\mathsf{P}(\mu)$, $\mathsf{V}(\mu)$, $\mathsf{A}(\mu)$.
- $\mathsf{P}(\mu)$ is a shorthand for $\mathsf{Score}(P_H(\mu), \mu_{\text{real}})$.

Maxima convention. We fix a baseline law μ_0. For $F \in \{P, V, A\}$ define $F_{\max} := \sup_{\pi \in \Pi} F(\pi\mu_0)$. We also write $V^{\Delta}(\pi) := V^{\Delta}(\tau(\pi))$ where $\tau(\pi)$ denotes the verification lag induced by policy π.

Commutators. For operators X, Y define the commutator norm:

$$\|[X, Y]\| \;:=\; \sup_{\mu \in \mathcal{P}} d\big(X(Y\mu),\, Y(X\mu)\big). \tag{1}$$

3 Why Separate Predictability and Verifiability?

P and V evaluate *different time slices*. V aligns a model to the past law $\mathbb{P}_{\text{past}}$ (window $[t_0 - \tau, t_0]$), while P evaluates forward accuracy on the future law $\mathbb{P}_{\text{future}}$ (at $t_0 + H$). Under drift (Assumption 4), $\mathbb{P}_{\text{past}} \neq \mathbb{P}_{\text{future}}$, so P and V are distinct objectives.

4 Lower Bounds via Two Independent Tensions

We formalize the trilemma's two distinct sources.

4.1 Tension 1: Noncommutativity (Path Dependence)

Assumption 4 (Drift)**.** There exists $\varepsilon_V > 0$ with $\|[V_\tau, E]\| \geq \varepsilon_V$.

Assumption 5 (Reflexivity)**.** There exists $\varepsilon_A > 0$ with $\|[A_{\text{phys}}, E]\| \geq \varepsilon_A$.

Assumption 6 (Disclosure)**.** There exists $\varepsilon_P > 0$ with $\|[P_H, A_{\text{disc}}]\| \geq \varepsilon_P$.

Definition 1 (Noncommutativity Gap)**.** We define the gap from noncommutativity as $\varepsilon_\star := \max\{\varepsilon_V, \varepsilon_A, \varepsilon_P\}$.

Assumption 7 (Joint Sensitivity)**.** We assume the objectives are jointly sensitive to changes in the underlying distribution, required to create a quantitative gap from noncommutativity. That is, there exists a constant $K > 0$ such that for any $\mu, \nu \in \mathcal{P}$:

$$\max\left\{|\mathsf{P}(\mu) - \mathsf{P}(\nu)|, |\mathsf{V}(\mu) - \mathsf{V}(\nu)|, |\mathsf{A}(\mu) - \mathsf{A}(\nu)|\right\} \geq K \cdot d(\mu, \nu).$$

Remark 1. Assumption 7 is a formalization of "objectives that matter". It posits that a significant change in the underlying probability distribution must be detectable by at least one of the core objectives. If $K = 0$, the objectives are blind to system changes, making the trilemma trivial.

4.2 Tension 2: Latency Constraint (Time Dependence)

Assumption 8 (Latency Conflict)**.** A verification lag τ and an action window Δ exist. A conflict occurs if $\tau \geq \Delta$.

Definition 2 (Latency Gap)**.** We define the *Latency Gap* $\lambda(\tau, \Delta)$ as the minimal unavoidable loss to either V or A induced by Assumption 8. This gap satisfies:

- $\lambda(\tau, \Delta) \geq \lambda_{\min} > 0$ if $\tau \geq \Delta$ (Latency Conflict).
- $\lambda(\tau, \Delta) = 0$ if $\tau < \Delta$ (No Conflict).

This formalizes the loss from the A-V temporal mismatch.

5 Main Results

Positioning of Results. Theorem 1 (Latency) is a *universal* no-go when $\tau \geq \Delta$. Theorem 2 (Noncommutativity) is a *diagnostic theorem*: it explains why orderings create asymmetry and routing is hard; it does not by itself establish impossibility of jointly attaining $(P_{\max}, V_{\max}, A_{\max})$ without further global conditions.

Theorem 1 (Latency No-Go with Explicit Lower Bound)**.** *Assume verification and action are sequential and non-parallelizable within a finite window $[t_0, t_0 + \Delta]$, with verification lag τ. For any policy π, let $L_V(\pi) := V_{\max} - V^{\Delta}(\pi)$ and $L_A(\pi) := A_{\max} - A(\pi)$. Then there exists a function $\lambda(\tau, \Delta) \geq 0$ such that*

$$\max\{L_V(\pi),\, L_A(\pi)\} \;\geq\; \lambda(\tau, \Delta).$$

In particular, if $\tau \geq \Delta$ then $\lambda(\tau, \Delta) > 0$, so no joint maximizer exists within the epoch.

Proof. By Assumption 1 there are two schedules.

Case (i): verify→act. If $\tau \geq \Delta$, the action is postponed past $t_0 + \Delta$ and thus yields $A(\pi) = 0$ within-epoch. Hence $A_{\max} - A(\pi) \geq A_{\max}$.

Case (ii): act→verify. Verification becomes decision-relevant only after $t_0 + \tau > t_0 + \Delta$, so only windowed verifiability V^{Δ} counts. By Assumption 2, $V_{\max} - V^{\Delta}(\pi) \geq L_V^{\text{time}} \cdot (\tau - \Delta)_+$ (or $V_{\max}$ under the hard-window variant).

Define

$$\lambda(\tau, \Delta) := \begin{cases} \min\{A_{\max}, L_V^{\text{time}} \cdot (\tau - \Delta)\} & \text{if } \tau \geq \Delta, \\ 0 & \text{if } \tau < \Delta. \end{cases}$$

Since any policy must choose one of these schedules, we have

$$\max\{L_V(\pi), L_A(\pi)\} \geq \lambda(\tau, \Delta).$$

For $\tau < \Delta$, neither schedule forces a positive gap from latency, hence $\lambda(\tau, \Delta) = 0$. When $\tau \geq \Delta$, we have $\lambda(\tau, \Delta) > 0$, precluding a joint maximizer within the epoch. □

Theorem 2 (Noncommutativity Diagnostic: Symmetry Breaking and Routing)**.** *Under Assumption 7 (joint sensitivity) —or alternatively Assumption 3 —for any noncommuting pair* (X, Y) *there exists an ordering* $\pi' \in \{XY, YX\}$ *such that*

$$\max_{F\in\{P,V,A\}} \big(F_{\max} - F(\pi')\big) \;\geq\; c\,\|[X,Y]\|\,, \qquad c := \tfrac{K}{2}.$$

Proof. Let $\pi_1 = XY$ and $\pi_2 = YX$. By the definition of the commutator norm, for any $\eta > 0$ there exists μ with $d(\pi_1\mu, \pi_2\mu) \geq \|[X,Y]\| - \eta$. By Assumption 7, there exists $F^\star \in \{P, V, A\}$ such that

$$\big|F^\star(\pi_1\mu) - F^\star(\pi_2\mu)\big| \;\geq\; K \cdot d(\pi_1\mu, \pi_2\mu) \;\geq\; K\big(\|[X,Y]\| - \eta\big).$$

Hence

$$\max\{F^\star_{\max} - F^\star(\pi_1),\; F^\star_{\max} - F^\star(\pi_2)\} \;\geq\; \tfrac{K}{2}\big(\|[X,Y]\| - \eta\big).$$

Letting $\eta \downarrow 0$ gives the claim. □

Remark 2. Among the concrete operator pairs from Assumptions 4–6, the one with maximal commutator $\varepsilon_\star := \max\{\varepsilon_V, \varepsilon_A, \varepsilon_P\}$ yields

$$\max_{F\in\{P,V,A\}} \big(F_{\max} - F(\pi')\big) \;\geq\; c\,\varepsilon_\star$$

for some $\pi' \in \{XY, YX\}$ by Theorem 2.

Proposition 3 (Qualitative No-Go via Contradiction)**.** *Suppose there exists a policy* $\pi^\star$ *that simultaneously attains* $(P_{\max}, V_{\max}, A_{\max})$ *within the epoch for all* μ *in a set* $\mathcal{D}$*. Then for every feasible pair* (X, Y) *and all* $\mu \in \mathcal{D}$ *one must have* $F(XY\mu) = F(YX\mu) = F_{\max}$ *for* $F \in \{P, V, A\}$*, hence* $XY\mu = YX\mu$ *whenever objectives distinguish distributions. Therefore* $[X, Y] = 0$ *on* $\mathcal{D}$*. If any commutator is provably nonzero on* $\mathcal{D}$*, such a universal policy cannot exist.*

Proof. This is the contrapositive of Theorem 2: if a joint maximizer $\pi^\star$ exists on $\mathcal{D}$, then all orderings must achieve the maximal objective values, which by Assumption 7 (or 3) implies all commutators vanish on $\mathcal{D}$. Thus, if any commutator is provably nonzero, no such policy can exist. □

Corollary 1 (Consequences of Latency and Noncommutativity)**.** (i) *Latency (hard no-go). If* $\tau \geq \Delta$*, then no joint maximizer exists within the epoch* $[t_0, t_0 + \Delta]$*.*

(ii) *Noncommutativity (diagnostic). If* $\varepsilon_\star > 0$ *and Assumptions 4–6 hold, then not all orderings can be optimal; in particular, there exists a pair* $(X, Y) \in \{(V_\tau, E),\, (A_{phys}, E),\, (P_H, A_{disc})\}$ *and an ordering* $\pi' \in \{XY, YX\}$ *with*

$$\max_{F\in\{P,V,A\}} \big(F_{\max} - F(\pi')\big) \;\geq\; c\,\varepsilon_\star,$$

where $c = \frac{K}{2}$*.*

Proof. (i) By Theorem 1, when $\tau \geq \Delta$ we have $\lambda(\tau, \Delta) > 0$, hence a joint maximizer cannot exist within the epoch.

(ii) Let (X, Y) be the pair among $\{(V_\tau, E), (A_{\text{phys}}, E), (P_H, A_{\text{disc}})\}$ attaining $\varepsilon_\star = \max\{\varepsilon_V, \varepsilon_A, \varepsilon_P\}$. Applying Theorem 2 to this pair yields the stated bound, so at least one ordering is suboptimal for at least one objective $F \in \{P, V, A\}$. □

Remark 3. Theorems 1 and 2 formalize the two independent mechanisms of the trilemma. The Latency Gap (λ) is a *universal* constraint on all strategies, while the Noncommutativity Gap ($\varepsilon_\star$) is an *existential* constraint proving that *some* strategies are provably suboptimal, breaking the symmetry required for a joint optimum.

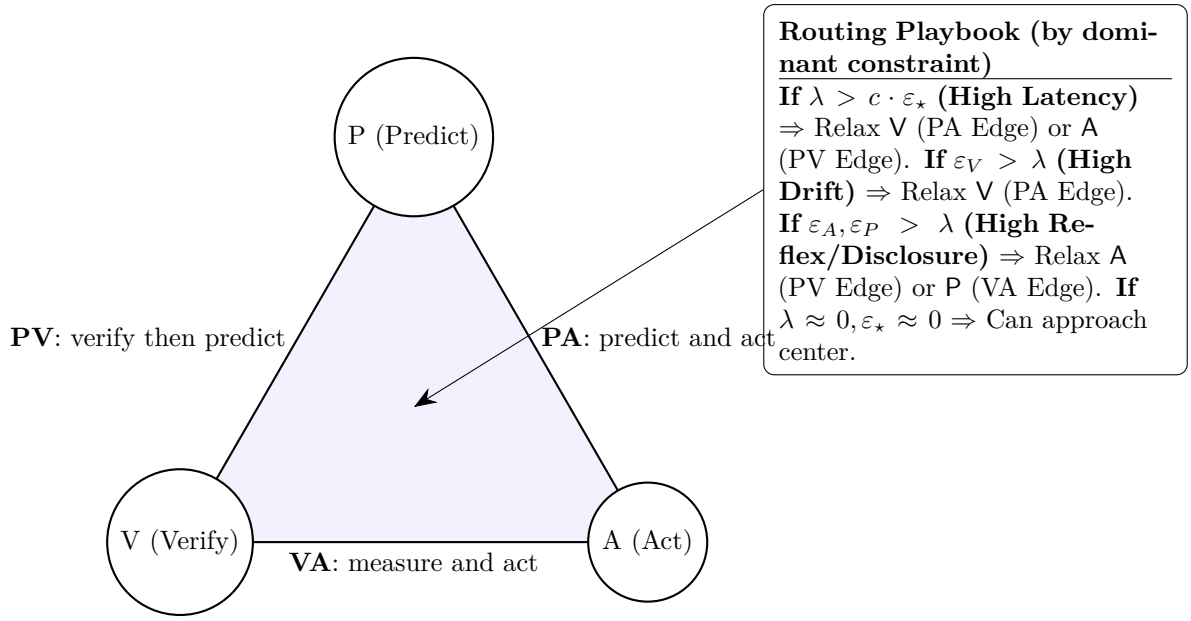

Figure 2: P–V–A routing triangle. The dominant constraint (either high latency λ or a large commutator $\varepsilon_\star$) dictates which edge (relaxing a vertex) is the optimal strategy.

6 Minimal Counterexample

Consider a two-state system $\Omega = \{0, 1\}$. Let E be a non-stationary kernel K_t (drift δ). Let A_{phys} flip the state (reflexivity α). Let V_τ require a time lag τ (proxy bias η), while the action window is Δ. **Case 1: Latency Conflict.** If $\tau \geq \Delta$, impossibility holds by Theorem 1. **Case 2: Noncommutativity Conflict.** If $\tau < \Delta$, we test noncommutativity. Assume Assumption 7 holds. For suitable (α, δ, η) one has $\varepsilon_\star \geq \max(|\delta - \eta|, \alpha)$. By Theorem 2, there exists a strategy π' with a gap of $\Omega(\max\{|\delta - \eta|, \alpha\})$.

In Case 1, impossibility holds by latency. In Case 2, not all orderings can be optimal—at least one incurs a positive gap per Theorem 2; absent further global conditions, we do not assert the nonexistence of a joint maximizer. (*See Appendix A for explicit bounds and constructions.*)

7 Routing Doctrine (Edges + Guardrails)

Doctrine. The optimal strategy is to identify the *dominant constraint* (from Assumptions 4–8) and choose the operating edge that relaxes the corresponding objective. The constraints are measured by the gap components $\lambda(\tau, \Delta)$ and $\varepsilon_\star = \max(\varepsilon_V, \varepsilon_A, \varepsilon_P)$.

PV edge (relax A).

- *When?* When reflexivity/disclosure is dominant (ε_A or ε_P are large).
- *Guardrails.* Time-shift or batch disclosure; limit disclosure radius; shadow traffic.
- *Effect.* Shrinks $\varepsilon_A, \varepsilon_P$, reducing the noncommutativity gap from Theorem 2.

VA edge (relax P).

- *When?* When disclosure is potent (ε_P large) *and* latency is low ($\tau \approx 0$).
- *Guardrails.* Shorten horizon H; use conformal bands; enforce abstention.
- *Effect.* Reduces actionable prediction impact, lowering ε_P.

PA edge (relax $\vee$).

- *When?* **EITHER** under high drift (ε_V large) **OR** high latency ($\tau \geq \Delta$).
- *Guardrails.* Canary + rollback; streaming/online calibration; shrink τ (if possible).
- *Effect.* Lowers ε_V. If τ is irreducible, this edge is forced by $\lambda > 0$ (from Theorem 1).

Playbook (three steps). *(1)* Measure all constraints: $\lambda(\tau, \Delta)$ and $\varepsilon_V, \varepsilon_A, \varepsilon_P$; *(2)* pick edge by the *dominant* gap source (e.g., if $\lambda > c \cdot \varepsilon_\star$, latency is the main problem); *(3)* apply guardrails to mitigate.

Portfolio strategies. In practice, organizations often adopt blended portfolios: e.g., internet companies allocate $\sim 80\%$ capacity to **VA** loops (rapid iteration, A/B testing) and $\sim 20\%$ to longer **PA/PV** bets, with **PV** reserved for safety-/regulatory-critical changes (see Section C).

Single-epoch vs. multi-epoch. Our results apply *per* decision epoch $[t_0, t_0 + \Delta]$ and impose hard, max-type shortfalls when latency or noncommutativity is active (e.g., Theorem 1). In practice, pipelines chain *multiple* epochs: multi-round routing reallocates capacity across VA and PA/PV tracks and updates P, V, A as fresh evidence arrives, allowing aggregate improvement over time despite the per-epoch no-go. See Section C for VA-centric workflows and Appendix C for an illustrative multi-epoch case.

8 Case Study: GPT-5 as a Deployed Routing System

8.1 GPT-5 Architecture Overview

Public materials describe GPT-5 as a *unified system* comprising a *smart and fast* model that answers most questions, a *deeper reasoning* model for harder problems, and a *real-time router* that decides which model to use for a given request.[1] This query-level routing has been emphasized in OpenAI's product pages and system card, and widely discussed by independent reports.[36, 37, 38, 39, 40]

8.2 Mapping to the P–V–A Framework

While internal details are undisclosed, externally observable behavior is *consistent* with the following alignment between components and our edges:

GPT-5 Component	**P–V–A Role**	**Expected Trade-off Signature (qualitative)**	**Theoretical Anchor**
Fast "smart" model	PA edge (action-feasible)	Lower latency, higher A within window; may accept smaller V^Δ on hard cases	Thm. 1
Deeper reasoning model	PV edge (verification-oriented)	More deliberation time, higher V on difficult prompts; reduced A if $\tau\uparrow$	Thm. 1
Real-time router	Meta-controller	Chooses ordering/edge under current constraint profile (lag vs. deadline, task hardness)	Thm. 2

Table 1: A cautious alignment of GPT-5 components with our P–V–A edges based on public descriptions and observed behavior.

[1] We are not affiliated with OpenAI; this section synthesizes official documentation and independent analyses.

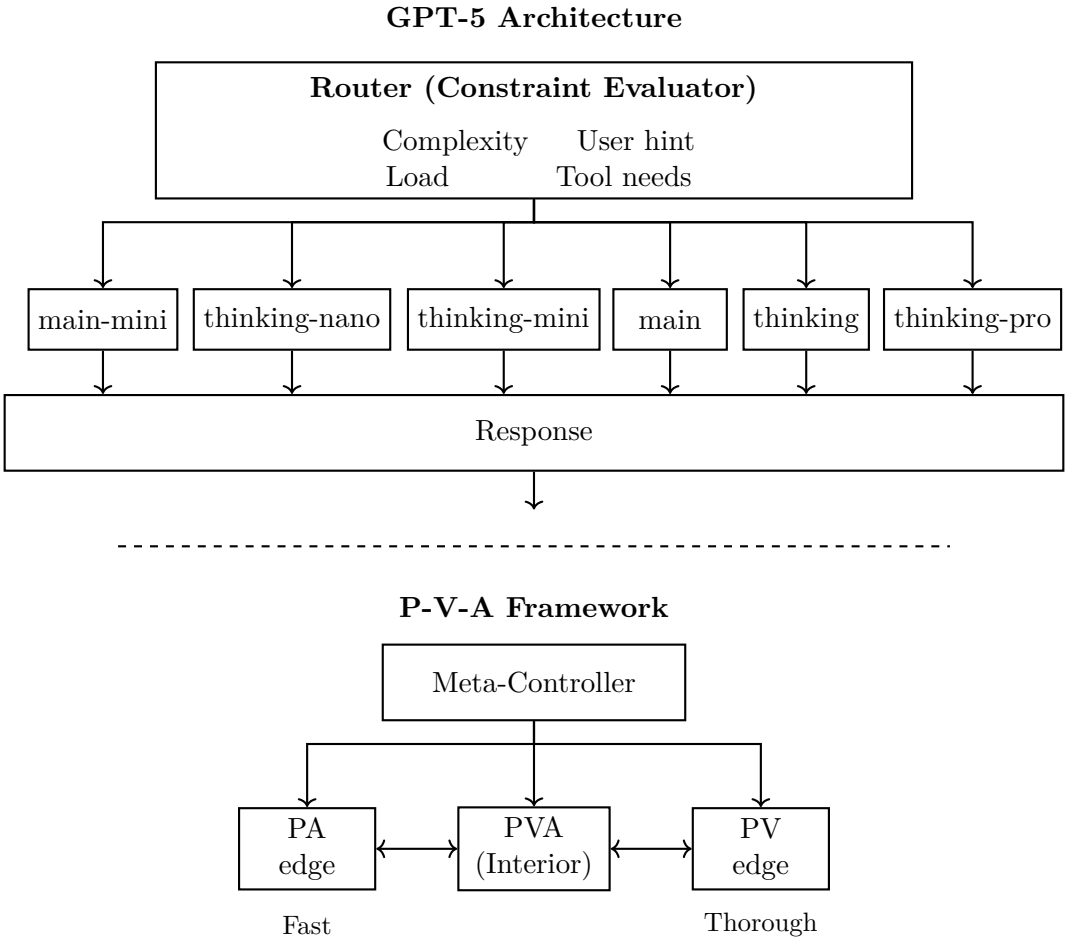

Figure 3: GPT-5 architecture (top) as an instantiation of P-V-A routing framework (bottom). The router acts as meta-controller, specialized models correspond to triangle edges, demonstrating independent convergence to theoretical predictions.

8.3 Validation of Theoretical Predictions

Prediction 1 (Specialization > Single Monolith). A system with specialized models should outperform a single undifferentiated model at a fixed budget. GPT-5 materials explicitly distinguish a fast model and a deeper reasoning model, consistent with specialization benefits [36, 37].

Prediction 2 (Routing as a Core Capability). Our framework predicts that routing—rather than uniform inference—is core to performance under varying constraints. OpenAI documents a real-time router that selects which model to use; public reports describe this as a defining feature [37, 38, 40].

Prediction 3 ("Fast vs. Slow" Corresponds to Edges). Choosing a fast path (higher A, potentially lower V^{Δ} on hard items) or a slow path (higher V, lower within-window A) matches our PA vs. PV edge intuition and the latency bound in Theorem 1 [37].

Prediction 4 (Constraint-Aware Behavior is Necessary). Performance should improve when the router is constraint-aware (task hardness, tool needs, lag/window). Public accounts describe query-level routing conditioned on such factors [38, 39].

Prediction 5 (Multi-epoch Routing and User Override). While our analysis applies *per* decision epoch $[t_0, t_0 + \Delta]$, GPT-5 natively operates over multi-turn interactions (multi-epoch decisions), allowing the router to incorporate fresh evidence each turn and continually improve P, V, A under the per-epoch no-go constraints. In addition, GPT-5 exposes a *model picker* that

lets users manually select a specific model, functioning as an exogenous routing override when constraints or intent are known a priori [37, 40]. This aligns with the portfolio view (VA loops + PA/PV tracks) and the single-epoch vs. multi-epoch discussion in Section C.

8.4 Limitations and Future Directions

We do not claim knowledge of GPT-5 internal parameters or full component count. Public information evolves and some reports are third-party. Within these limits, the P–V–A view suggests: (i) explicit modeling of $(\tau, \Delta, \varepsilon_\star)$ in router policy; (ii) multi-scale routing (within-epoch vs. multi-epoch); (iii) reflexivity-aware adjustments when disclosure/actions shift future evidence laws.

8.5 Discussion

Engineering practice (query-level routing across at least two model classes) and theory (latency no-go; noncommutative diagnosis) *converge independently.* This strengthens the claim that the P–V–A trilemma is structural in complex adaptive systems (CAS), and that routing is a principled response rather than a mere optimization.

9 Scope and Falsifiability

Scope. The results apply to online systems constrained by (i) noncommutativity ($\varepsilon_\star > 0$) **or** (ii) latency ($\tau \geq \Delta$). **Falsifiable predictions.** (a) If all commutators ≈ 0 **and** $\tau < \Delta$, simultaneous near-attainment should be observed (i.e., both $\lambda \approx 0$ and $\varepsilon_\star \approx 0$). (b) Reducing ε_P, ε_V, or τ should monotonically shrink the gap.

10 Meta-Theoretic Statement

The present proof method elevates the P–V–A trilemma to a *structural meta-law.* It is not a universal trilemma from *arbitrary* noncommutativity, but a specific one arising from the temporal operators $\{V_\tau, A_{\text{phys}}, A_{\text{disc}}, P_H\}$ and their interaction with time evolution (E) and time constraints (Δ, τ).

Meta-theoretic framing (hypothesis). We propose that the P–V–A trilemma is a *structural meta-law* governing decision-making under time: its force arises from temporal operators and order effects rather than domain-specific artifacts. Convergent evidence comes from three independent strands:

1. **Mathematical necessity.** Theorems 1–2 show (i) a latency no-go when $\tau \geq \Delta$, and (ii) a quantified shortfall under noncommutativity, tying gaps to commutator norms.

2. **Independent empirical convergence.** Recent routing architectures developed independently of this framework align with a P–V–A decomposition and edge selection (see discussion in the main text).

3. **Cross-domain recurrence.** Decades-old software-engineering trade-offs and modern internet A/B practice instantiate the same routing logic under different constraint regimes (Appendix B, Appendix C).

When theory predicts what practice has independently rediscovered across domains and decades, the pattern is better understood as domain-invariant structure rather than a particular solution.

Temporal morphism and falsifiability. Each vertex fixes a distinct temporal commitment (past validation V, present execution A, future anticipation P). The trilemma arises when:

1. **Noncommutativity** (e.g., $\|[A_{\text{phys}}, E]\| > 0$) creates path-dependent residue (holonomy).
2. **Latency** ($\tau \geq \Delta$) creates a direct temporal conflict between V_τ and A.

If all commutators and the latency conflict are neutralized (i.e., $\varepsilon_\star \to 0$ and $\lambda \to 0$), the trilemma *degenerates*.
(*See Appendix B for the software-engineering trilemma (P–V–A mapping) and Appendix C for internet A/B practice and the VA/PA 80/20 portfolio.*)

11 Related Work (Brief)

11.1 Connecting the Dots: A Unified P–V–A Lineage Across Disciplines

(i) Lucas critique and performative prediction (P vs. A). Intervention- and disclosure-induced law changes are formalized by *Reflexivity* (Assumption 5, $\|[A_{\text{phys}}, E]\| > 0$) and *Disclosure* (Assumption 6, $\|[P_H, A_{\text{disc}}]\| > 0$). See [1].

(ii) Goodhart's law (P vs. A). A (P,A) projection in which optimization/targeting induces path dependence is captured by *Disclosure* (Assumption 6); see also *Reflexivity* (Assumption 5). See [2, 3].

(iii) Dataset shift / covariate drift (P vs. V). Distributional non-stationarity affecting verification is formalized by *Drift* (Assumption 4, $\|[V_\tau, E]\| > 0$). See [6, 7].

(iv) Analysis paralysis (A vs. V). Finite-window conflict is formalized by *Latency Conflict* (Assumption 8, $\tau \geq \Delta$) and Definition 2.

(v) CAP/PACELC. Spatial consistency–availability tradeoffs are orthogonal but compatible with our temporal constraints [4, 5].

(vi) Software engineering "iron triangle" (triadic P–V–A instance). The classic trade-off among time/schedule, quality, and cost predates modern ML yet matches a P–V–A mapping (prediction/commitment, verification, action/expenditure); see [41, 42, 43] and Appendix B for a case study.

(vii) Brooks's Law (A vs. V, P). The observation that "adding manpower to a late software project makes it later" [41] is a *routing failure*: increasing the physical action capacity A_{phys} (headcount) under high noncommutativity $\varepsilon_\star$ (communication/training overhead) degrades both long-horizon prediction P_H (via context switching) and verification V_τ (via integration/testing burden). See Appendix B.4 for the full structural analysis.

Our contribution is to *unify* these pairwise tensions (P–A, P–V, A–V) and to show that *triadic* instances (e.g., the engineering iron triangle) arise as concrete P–V–A realizations within a single framework.

11.2 Multi-epoch optimization: how P/V/A each improves

One-liner. Multi-cycle turns P/V/A from a zero-sum trade-off into an intertemporal investment: each cycle allocates time and attention to raise future P (better models & calibration), V (shorter τ_{eff}, higher power), and A (faster, safer execution with richer options), consistent with trustworthy experimentation practices and modern software delivery research [24, 33, 34].
Conceptual shift. Multi-epoch optimization resolves the apparent paradox between our impossibility results and real-world improvement. The key insight is **temporal scale separation**:

- **Within-epoch (micro)**: Theorems 1–2 impose hard constraints. Any policy π must sacrifice at least one vertex.

- **Across-epochs (macro)**: Iterative routing reallocates capacity dynamically. Early epochs may emphasize VA (fast feedback) to stabilize the system; later epochs shift to PA/PV (long bets) once τ_{eff} is compressed and $\varepsilon^\star$ is reduced.

This multi-scale view reconciles two truths:

1. **No single epoch** can achieve $(P_{\max}, V_{\max}, A_{\max})$ [Theorem 1].
2. **Chained epochs** can drive all three toward their limits via:
   - Data/model accumulation ($\uparrow P$)
   - Infrastructure/instrumentation ($\downarrow \tau$, $\uparrow V$ power)
   - Delivery automation ($\uparrow A$ speed, reversibility, options)

The portfolio heuristic (80% VA, 20% PA/PV; see Appendix C.3) operationalizes this at the organizational level.

11.2.1 P — Predictability (prediction quality)

Core levers.

1. **Data accumulation & model iteration.** Deep learning/LM scaling laws show that increasing data and model size along power-law regimes systematically lowers generalization error and improves out-of-sample (OOS) performance [11, 12, 13].
2. **Counterfactual replay / IPS.** Offline evaluation/learning with logged data using IPS, self-normalized IPS, and doubly-robust estimators reduces bias/variance for bandits, recommenders, and ads [16, 14, 15, 17, 18].
3. **Purposeful exploration.** Multi-armed bandits (UCB/Thompson sampling) systematically collect informative samples over cycles, yielding lower long-run regret and more stable extrapolation [19, 20].

Typical signals.

- OOS/holdout improvements & cross-validation quality [21].
- Calibration error (ECE) and Brier score decrease [22, 23].
- Uncertainty intervals shrink across cycles (via standard sampling theory) [21].
- Stronger correlation with long-term OEC (Overall Evaluation Criterion) and long-holdout [24].

11.2.2 V — Verifiability (identification & trust)

Core levers.

1. **Pre-positioned/parallel checks.** *Always-valid* inference enables continuous monitoring without inflating Type I error, supporting online/parallel checks [25, 26].
2. **Blocking/stratified designs.** Stratification and blocking planned at design-time materially increase power and reduce variance; public-sector experimentation guidance provides practical patterns [27].
3. **Proxy calibration.** Variance reduction with pre-experiment covariates (e.g., CUPED) and delayed-feedback modeling enable earlier reads and higher sensitivity [28, 29].

Typical signals.

- Effective lag τ_{eff} decreases via delayed-feedback modeling / survival-style estimators [29].
- Statistical power $(1-\beta)$ increases from blocking/stratification and CUPED variance reduction [27, 28].
- SRM (Sample Ratio Mismatch) rate decreases; A/A stability and repeatability improve via rigorous data-quality guardrails [30, 24].

11.2.3 A — Actionability (execution speed, reversibility, option richness)

Core levers.

1. **Phased rollout & guardrails.** Canary releases and error-budget/SLO practices enable fast, reversible increments and safe rollbacks [31, 32].
2. **Option set engineering.** Modeling structured/combination actions (linear/combinatorial/slate bandits) enlarges the feasible action set and improves per-time decision capacity with reversibility [19, 35].
3. **Delivery infrastructure.** CI/CD capabilities measured by the DORA Four Keys—Deployment Frequency, Lead Time for Changes (mean-time-to-ship), Change Failure Rate, and Time to Restore (rollback/repair latency)—quantify "faster, safer, more reversible" delivery [33, 34].

Typical signals.

- Mean-time-to-ship (lead time) ↓; rollback latency ↓; incident (change-failure) rate ↓; CI/CD throughput ↑—mapping directly to the Four Keys [33, 34].

12 Limitations and Future Work

(i) Lipschitz constants L_P, L_V, L_A and the Joint Sensitivity constant K must be estimated, which can be difficult. Sharpening these bounds is future work. (ii) Modelling A_{phys} and A_{disc} for complex multi-agent feedback is open. (iii) Extending the gap inequality to multi-epoch compositions. (iv) Measuring commutators in highly nonstationary settings is challenging. (v) Toward a unified space-time framework. The P–V–A trilemma may constitute the temporal face of a broader Consistency–Verification–Action–Prediction (C–V–A–P) tetrahedron, where spatial separation yields CAP-style constraints [4, 5] and temporal boundedness yields the P–V–A constraints. A full C–V–A–P treatment with quantified space–time trade-offs is left for future work.

13 Philosophical Implications and Meta-Theoretical Insights

13.1 Epistemic Limits

Our impossibility result makes a basic cognitive dilemma explicit. *Predictability* (P), *Verifiability* (V), and *Actionability* (A) line up with the maximalist aspirations of rationalism, empiricism, and pragmatism, respectively. The theorem shows that no single epistemic paradigm can monopolize truth—a conclusion that resonates with canonical debates on the sources of knowledge. In our framing, the latency constraint supplies a robust no-go, while noncommutativity explains why order sensitivity undermines any "obvious" path to optimality.

13.2 Structural Constraints of Temporality

From an existential perspective, the P–V–A triangle mirrors Heidegger's temporality of "being-in-the-world":

- **Past** (the domain of V): settled yet unchangeable;
- **Future** (the domain of P): conceivable yet unsettled;
- **Present** (the domain of A): unfolding yet evanescent.

Our results formalize that a time-bound agent must live within the tension of these three temporal dimensions. Temporality enters only latently through the lag–deadline pair (τ, Δ), which drives the latency no-go and shapes feasible trade-offs to be routed.

13.3 The Ethical Basis of Freedom and Responsibility

If P–V–A could be jointly maximized, moral choice would collapse into mere technical calculation. The very impossibility (with latency as the hard engine and noncommutativity as a second axis when latency is mild) safeguards a space where ethical choice matters: we must act under partial information and own responsibility for decisions. This echoes Kant's practical reason: genuine freedom consists in choosing deliberately *within* constraints.

13.4 Ontological Hints about Reality

From quantum uncertainty to reflexivity in social systems, many strata of reality exhibit triangular tensions akin to P–V–A. Our results suggest that such a triadic structure may be a generic feature of the real: neither fully determined nor purely random, neither purely objective nor purely subjective.

13.5 Reframing Wisdom

The theorem invites a redefinition of wisdom: not "finding the perfect solution," but knowing which vertex to prioritize—and which to sacrifice—given the situation. This context-sensitive discernment (*phronesis*) is not fully algorithmizable. The *Routing Doctrine* operationalizes this stance: read the dominant constraint (latency vs. order sensitivity), then route accordingly along $PV/VA/PA$.

14 Conclusion

We offered a unified lens for *latency* and *noncommutativity*. Latency alone already yields a hard P–V–A trilemma in many practical regimes; when latency is not the bottleneck ($\tau < \Delta$), noncommutativity becomes the second key axis that explains difficulty and *guides routing*. This honest positioning improves both *predictive power* (when a gap is unavoidable) and *design power* (when routing can still reduce the gap).

Toward a Practical Philosophy of Bounded Rationality. The P–V–A framework ultimately prescribes a discipline of *situated decision-making*. In confronting any decision problem, three questions become essential:

1. **Identify the dominant constraint**: Which vertex of the triangle—Prediction, Verification, or Action—constitutes the primary objective under the present constraint profile $(\tau, \Delta, \varepsilon^\star)$?

2. **Align strategy with priority**: Does the chosen routing strategy advance the primary objective while *hedging* secondary objectives to the extent feasible?

3. **Ensure acceptable trade-offs**: Does the necessarily sacrificed vertex remain within acceptable bounds? Are *guardrails* in place to prevent catastrophic failures on the deprioritized dimension?

To interrogate decisions through this triadic lens—identifying what to prioritize, verifying alignment with that priority, and bounding downside risk—is not merely pragmatic accommodation to constraint. It is, we argue, the essence of *rationality under finitude*. The aspiration to simultaneously optimize all objectives (the "既要又要还要" fallacy) is not ambitious; it is incoherent. True rationality lies not in denying impossibility, but in *navigating* it with clarity about trade-offs and courage in making them.

The P–V–A framework does not promise perfect decisions. It promises *honest* ones—grounded in structural constraints, transparent about sacrifices, and disciplined in execution. As AI systems, policy frameworks, and human institutions confront increasingly complex temporal decision problems, this clarity may prove more valuable than any algorithm.

Acknowledgments

We thank colleagues for discussions on noncommutative operators, information geometry, and online decision systems.

A Explicit bounds for the minimal counterexample

We consider $\Omega = \{0, 1\}$ with law $\mu = \text{Bernoulli}(p)$. Let d be either TV or W_1 on Ω. Define E by a non-stationary kernel K_t that maps $p \mapsto p + \delta$ if $p < \frac{1}{2}$ and $p \mapsto p - \delta$ if $p \geq \frac{1}{2}$, with $0 < \delta \leq \frac{1}{2}$. Let A_{phys} flip the state with probability $\alpha \in (0, 1)$ so that $p \mapsto (1-\alpha)p + \alpha(1-p) = p + \alpha(1-2p)$. Let V_τ (with lag τ) replace p by an estimate $\hat{p}$ from the past window with $|\hat{p} - p| \leq \eta$. Let the action window be Δ.

Case 1: Latency conflict. If $\tau \geq \Delta$, impossibility holds by Theorem 1; the within-epoch gap satisfies $\text{Gap} \geq \lambda(\tau, \Delta) > 0$.

Case 2: Noncommutativity conflict. If $\tau < \Delta$ (so $\lambda = 0$), we test noncommutativity. Assume Assumption 7 holds with constant $K > 0$. For suitable p near $\frac{1}{2}$ one has

$$d(V_\tau E\mu,\, EV_\tau\mu) \geq |\delta - \eta|,$$
$$d(A_{\text{phys}} E\mu,\, EA_{\text{phys}}\mu) \geq \alpha.$$

Hence $\varepsilon_\star \geq \max(|\delta - \eta|, \alpha)$. By Theorem 2, there exists a strategy π' with a gap $\Omega(\max\{|\delta - \eta|, \alpha\})$.

Explicit $\lambda(\tau, \Delta)$ in toy models. (*Hard window*) If $V^\Delta(\tau) = 0$ whenever $\tau \geq \Delta$ and $A_{\max} = 1$, then $\lambda(\tau, \Delta) \geq \min\{1, V_{\max}\}\, \mathbf{1}\{\tau \geq \Delta\}$. (*Soft window*) If $V^\Delta(\tau) = V_{\max} - L_V^{\text{time}}(\tau - \Delta)_+$ with $L_V^{\text{time}} > 0$, then $\lambda(\tau, \Delta) \geq \min\{A_{\max}, L_V^{\text{time}}(\tau - \Delta)\}$.

B Software Engineering Trilemma: a P–V–A case study

The software engineering "iron triangle" —the trade-off between cost, time, and quality —has been a cornerstone of project management since at least the 1970s [41, 42, 43]. We demonstrate that this classical trade-off instantiates the P–V–A structure.

B.1 P–V–A mapping

The traditional formulation ("good, fast, cheap — pick two") obscures temporal structure and conflates distinct dimensions. We reframe it within the P–V–A lens:

Table 2: Software Engineering Iron Triangle re-mapped to P–V–A (direction vs. validation vs. execution)

Traditional	Reframed	PVA	Operator	Detailed interpretation
Speed	**Predictability**: direction & timing correctness	P	P_H	Prediction about future outcomes: *are we building the right thing, at the right time*? Includes feature-effectiveness hypotheses, market timing, and strategic alignment. Not merely execution velocity.
Quality	**Verifiability**: standards & evidence	V	V_τ	Validation against evidence: code quality (tests, review), requirement satisfaction, and hypothesis testing (e.g., online experiments). Captures both technical QA and product validation.
Cost	**Actionability**: execution capacity & resource use	A	A_{phys}	Implementation under resource constraints (time, money, personnel): coding and deployment speed, operational effort, and runtime budgets. Fast execution lives here, *distinct* from predictive correctness.

Clarification on "Speed" as Predictability (P). The traditional "speed" term conflates *directional correctness* (what/when to build) with *execution velocity* (how fast we build). In the P–V–A decomposition, directional correctness maps to P (via P_H), while execution velocity belongs to A (via A_{phys}). This separation is crucial in AI-accelerated settings: when A_{phys} is dramatically faster, bottlenecks shift upstream to P (strategic clarity) and to V (assurance), rather than disappearing.

Example: Online A/B testing as P–V–A. A hypothesis such as "variant X outperforms Y" is a **prediction** (P). Deploying variants and allocating traffic is the **action** (A). Measuring user response and deciding whether to ship is **verification** (V). If V contradicts P, the feature is not shipped, showing that predictive accuracy (P) and verification (V) *gate* execution (A).

B.2 Impossibility mechanisms

Disclosure noncommutativity ($[P_H, A_{\mathbf{disc}}] \neq 0$). Premature disclosure (A_{disc}) before completing long-horizon prediction (P_H) induces path dependence: the public commitment alters the environment/state μ, which then feeds back into both subsequent predictions and feasible actions. Under Assumptions 4–6 (nonzero commutators) and Assumption 7 (or Assumption 3), Theorem 2 and Corollary 1(ii) imply that there exists an ordering $\pi' \in \{A_{\text{disc}} P_H,\ P_H A_{\text{disc}}\}$ with

$$\max_{F \in \{P,V,A\}} \left(F_{\max} - F(\pi')\right) \ \geq \ c\,\varepsilon_\star,$$

so *not all* orderings can be optimal. *Industry pattern.* Announcing a fixed launch date well in advance (i.e., executing A_{disc} while P_H is incomplete) typically increases coordination pressure and scope volatility, leading to positive gaps on V (quality) or A (delivery), even if more resources are allocated. Notable examples include large-scale software projects with premature public commitments that subsequently required quality compromises or schedule extensions (cf. Goodhart/performative effects and the discussion in §11).

Latency conflict ($\tau \geq \Delta$). When verification/QA (V_τ) and action (A_{phys}) are sequential and non-parallelizable within the finite window $[t_0, t_0 + \Delta]$, Theorem 1 yields, for any policy π,

$$\max\{L_V(\pi),\, L_A(\pi)\} \;\geq\; \lambda(\tau, \Delta), \qquad L_V(\pi) := V_{\max} - V^{\Delta}(\pi), \;\; L_A(\pi) := A_{\max} - A(\pi).$$

In particular, if $\tau \geq \Delta$ then $\lambda(\tau, \Delta) > 0$ and *no joint maximizer* exists within the epoch. *Scheduling check.* For instance, with $d = 8$, $\tau = 4$, and $\Delta = 6$, we have $8 + 4 > 6$, which triggers the latency no-go.

Drift and reflexivity. Temporal evolution and intervention amplify both diagnostic and latency effects:

$$[V_\tau, E] \neq 0 \quad \text{(requirements drift)}, \qquad [A_{\text{phys}}, E] \neq 0 \quad \text{(reflexivity/technical debt)}.$$

Practically, coordination/communication overhead can make the *effective* τ superlinear in team size (Brooks' Law [41]), while platform and market changes enlarge $\varepsilon_\star$. Since $\lambda(\tau, \Delta)$ increases with τ, and the noncommutative lower bounds scale with $\varepsilon_\star$, both mechanisms raise the minimal *max-type* shortfall, tightening the P–V–A trade even when additional resources are applied (cf. [42]).

B.3 Routing as methodological choice

Different methodologies represent edge selections:

Methodology	Edge	Strategy	Trade-off
Waterfall	PV	Planning + verification	Sacrifices iteration
Agile	PA	Rapid delivery	Partial/rolling verification
A/B Testing	VA	Quick measure + rapid deploy	Limited long-horizon prediction (relaxes P)

From Waterfall (PV) to Agile (PA): drivers and transition. Historically, software moved from *PV-centric* waterfall to *PA-centric* agile for structural reasons aligned with the P–V–A constraints:

- **Shrinking action windows (Δ).** Market cycles and release cadences shortened; long upfront verification cannot fit within the window, increasing the likelihood that $\tau \geq \Delta$ and triggering the latency no-go (Theorem 1).
- **Rising noncommutativity ($\varepsilon_\star$).** Volatile requirements, user feedback, and platform changes amplify $[V_\tau, E]$ and $[A_{\text{phys}}, E]$, making rigid PV plans brittle and favoring edges that adapt trajectory (PA/VA).
- **Lower cost of change.** CI/CD, cloud, microservices, and feature flags reduce the marginal cost of action/rollback (A_{phys}), tilting the calculus toward *earlier* action with *rolling* verification (PA).
- **Operational observability.** Telemetry and monitoring shorten *effective* verification lag (τ), making near-real-time feedback feasible and enabling incremental validation *post* deployment.

Under large $\varepsilon_\star$ relative to feasible $\lambda(\tau, \Delta)$, PA-style flexible routing tends to dominate PV-style rigid planning: it accepts bounded verification debt in exchange for preserving the action window and adaptability. This methodological shift is not merely cultural—it is a *structural response* to temporal and path-dependence constraints in the decision epoch.

Bridge to VA-centric internet development. As deployment automation and observability matured, *VA*-centric workflows (e.g., online experiments, progressive rollouts) became viable defaults for internet applications, where measurement can be embedded in production. Section C leverages the same P–V–A lens: abundant telemetry compresses τ, allowing frequent *VA* iterations; long-horizon P is maintained via periodic retraining and offline simulation, yielding a pragmatic routing doctrine for large-scale online systems.

B.4 Brooks's Law as a Routing Pathology

Brooks's Law [41] states: "Adding manpower to a late software project makes it later." We show this is a consequence of routing under *misdiagnosed constraints*.

Failure mode. When a project is late, the intuitive response is to increase the action capacity A_{phys} (add developers). However, this ignores:

1. **Reflexivity:** $[A_{\text{phys}}, E] \neq 0$. Adding developers alters the evolution operator E (communication topology, onboarding/knowledge transfer), *reducing the marginal effectiveness of* A_{phys} via coordination overhead.
2. **Verification burden:** $[V_\tau, A_{\text{phys}}] \neq 0$. More code and interfaces increase the testing/review load (V_τ), enlarging the *effective* lag τ. If τ approaches or exceeds the remaining window Δ, the latency conflict is triggered and, by Theorem 1,

$$\max\{L_V(\pi),\, L_A(\pi)\} \;\geq\; \lambda(\tau, \Delta) \;>\; 0,$$

   so no joint maximizer exists within the epoch.

Correct routing. Diagnose the active bottleneck and select the edge accordingly:

- **P** *(unclear requirements)* $\Rightarrow$ **PV** edge: pause A_{phys}, clarify design/spec (P_H and V_τ).
- **V** *(technical debt / poor testability)* $\Rightarrow$ strengthen V_τ (refactor, tests), not A_{phys}.
- $\lambda(\tau, \Delta)$ *(time pressure)* $\Rightarrow$ **PA** edge: cut scope, ship an MVP within $[t_0, t_0 + \Delta]$.

Blindly increasing A_{phys} is optimal only when $\varepsilon_\star \approx 0$ (negligible coordination/noncommutativity) *and* $\lambda(\tau, \Delta) \approx 0$ (no binding time constraint)—conditions rarely met in practice.

C Internet Application Development: VA-centric Iteration and Online A/B Testing

Why the VA edge. In internet application development, rapid iteration under drift and reflexivity makes the VA edge (V_τ then A_{phys}) particularly effective. Predictions are often brittle or stale ($[V_\tau, E] \neq 0$), so teams prefer *small, quickly verifiable changes* over heavy upfront modeling.

C.1 A/B testing as a VA loop

Online controlled experiments operationalize a short VA cycle:

1. **Hypothesize** a minimal change with a measurable metric.
2. **Act** by rolling out to a small, randomized treatment group.

3. **Verify** by measuring outcomes within the window $[t_0, t_0+\Delta]$ (guarding against SRM, novelty, peeking).
4. **Decide** (ship, iterate, or revert) and **repeat**.

This minimizes reliance on P_H and emphasizes timely V^Δ under latency constraints.

C.2 Observed failure rates in online experiments

Company / Product	Failure rate	Success rate	Notes
Microsoft (overall)	66%	34%	Kohavi career aggregate
Bing	85%	15%	Mature, highly optimized surface
Airbnb (Search)	92%	8%	Highest rate reported in talks
Booking.com[†]	80–90%	10–20%	Public talks / recaps
Google Ads[‡]	80–90%	10–20%	Cross-company anecdote

Table 3: Reported outcomes for online A/B tests ("success" = shipped positive impact or statistically significant uplift on the OEC; "failure" = negative or non-significant, thus not shipped). Figures compiled from public talks and practitioner sources; actual rates vary with metric choice, product maturity, guardrails, and sample size. Sources include [44].

[†] **Booking.com:** The oft-quoted "∼90% of experiments fail / only ∼10% significant uplift" comes from public talks and conference recaps (e.g., Weigel 2017 talk and post-event summaries) [45, 46]. Practitioner observation, *not an official stance.*

[‡] **Google Ads:** Kohavi (Lenny's Newsletter, 2023) notes that *other companies* (e.g., Booking, Google Ads) have published 80–90% failure rates [44]. Cross-company anecdote / third-party reference, *not an official stance.*

C.3 Portfolio heuristic in practice (80/20)

Field-proven split. Frontline teams often adopt a pragmatic capacity allocation that aligns with the VA edge (a pattern also noted in practitioner interviews, e.g., [44]):

- **80% incremental (VA loops).** Prioritize known-effective directions: small, quickly verifiable changes, short VA cycles ($V^\Delta \to A_{\text{phys}}$), fast rollouts and reversions, strict guardrails, and metric hygiene (SRM checks, no peeking).
- **20% high-risk long bets (default PA edge; PV for safety-critical exceptions).** Allocate a minority of capacity to longer-horizon, higher-variance bets that typically follow a prediction→action route ($P_H \to A_{\text{phys}}$ / A_{disc}), with staged V^Δ checkpoints and multi-epoch evaluation. For safety-/regulatory-critical or irreversible changes (e.g., payments compliance, privacy requirements, large data migrations), prefer a PV-heavy process (planning/proofs → pre-deployment verification) before broad action.

This 80/20 split reflects that predictions (P_H) are often brittle in fast-moving products, while timely verification (V^Δ) and decisive action (A_{phys}) compound value under drift and reflexivity.

Implication for routing. High empirical failure rates are consistent with a world where (i) effects are small on already-optimized products, (ii) nonstationarity and reflexivity are significant, and (iii) latency τ must be managed within delivery windows Δ. Thus internet apps gravitate to the VA edge (fast verification → action), using P_H mainly for scoping and guardrails rather than as a single-shot decision engine. See the 80/20 portfolio heuristic above (80% VA, 20% PA by default; PV for safety-/regulatory-critical exceptions).

D AI-era Software Engineering — Revisiting Brooks's Law

D.1 Motivation and scope

Brooks's Law—"adding manpower to a late software project makes it later" [41]—exposes structural tensions that our P–V–A framework renders explicit at the *per-epoch* level. The rise of AI-assisted development appears to collapse implementation latency for many tasks, prompting the question: do our impossibility mechanisms (latency and noncommutativity) change in kind, or only in degree? This appendix synthesizes how AI shifts the active constraints while preserving the trilemma's structure. For the mechanistic treatment of Brooks's Law as a routing pathology, see Appendix B.4.

D.2 How AI shifts the P–V–A constraints

(i) Action acceleration (surface effect). AI-assisted coding raises the effective capacity of A_{phys} and compresses implementation time for many units of work. This *reduces* the fraction of the epoch $[t_0, t_0 + \Delta]$ consumed by action, but does not alter the *sequential* nature of V_τ and A_{phys}.

(ii) Verification paradox (mixed effect). Automation can shorten τ via better tooling (tests, static analysis, CI), yet also increases verification *complexity* (e.g., hallucination patterns, interface proliferation, security surface). In Theorem 1, the bound

$$\max\{L_V(\pi),\, L_A(\pi)\} \;\geq\; \lambda(\tau, \Delta)$$

remains binding whenever the *effective* τ approaches Δ; shrinking raw coding time alone does not remove the max-type shortfall.

(iii) Prediction becomes a bottleneck (deep effect). When A_{phys} accelerates, upstream specification and context—our P_H—often dominate the wall-clock critical path. Under noncommutativity, Assumptions (nonzero commutators) together with Theorem 2 diagnose that *not all* orderings can be optimal when $\varepsilon_\star > 0$: premature A_{phys} or disclosure A_{disc} can degrade V_τ and future P_H.

D.3 Why small teams gain structural advantage

Let the team size be n. Coordination overhead makes both the noncommutative sensitivity and effective lag grow with n:

$$\varepsilon_\star(n) \;\uparrow \quad \text{and} \quad \tau_{\text{eff}}(n) \;=\; \tau_{\text{base}} \;+\; \tau_{\text{coord}}(n),$$

where $\tau_{\text{coord}}(n)$ aggregates integration/testing/review and communication costs (often superlinear in n). Even if A_{phys} scales with n, the *per-epoch* bound $\max\{L_V, L_A\} \geq \lambda(\tau_{\text{eff}}, \Delta)$ remains, and the diagnostic asymmetry from $\varepsilon_\star(n)$ intensifies. Hence, small teams—with lower τ_{coord} and typically smaller $\varepsilon_\star$—can realize larger *aggregate* throughput across chained epochs, consistent with the portfolio and multi-epoch routing view in Section C.

D.4 Why "AI programming feels exhausting": a routing diagnosis

Empirically reported fatigue is consistent with *constraint mismatch* and *edge misrouting*.

1. **Latency misread.** Treating problems as PA (fast delivery) when τ is binding forces repeated rework. By Theorem 1, if $\Delta < d + \tau$ (development d plus verification lag τ), then $\lambda(\tau, \Delta) > 0$ and no joint maximizer exists in-epoch; the correct mitigation is to either compress τ (test design, observability) or relax scope (PA with explicit V debt), not to amplify A_{phys} alone.

2. **Order asymmetry ignored.** With $\varepsilon_\star > 0$, Theorem 2 implies that not all orderings are optimal; committing (A_{disc}) or implementing (A_{phys}) before stabilizing P_H/V_τ can lock in loss on at least one objective.

Design implication. Route short-feedback work to VA loops, long-horizon/platform work to PV tracks, and scope-constrained delivery to PA—and rebalance across epochs (Section C; see also GPT-5's multi-epoch behavior in Section 8.3).

D.5 Testable predictions (operational hypotheses)

We list falsifiable hypotheses for organizational measurement; each targets a specific mechanism.

H1 (Small-team advantage under high $\varepsilon_\star$). For tasks with high integration/coordination complexity, teams of size 1–5 achieve lower *per-epoch* max-gap and higher feature-throughput per unit time than larger teams, holding A_{phys} budget fixed.

H2 (τ-driven routing shift). As the effective τ is reduced by observability and automated tests, the fraction of work routed to VA loops increases and correlates with improved lead time without a rise in post-release defect rates.

H3 (P_H as critical path). With AI assistance, the share of total cycle time attributable to upstream specification (context engineering) rises; interventions that reduce P_H ambiguity (templates, checklists) improve end-to-end time more than equal effort invested in A_{phys}.

D.6 Evidence snapshot (data provenance)

External reports indicate large-scale adoption of AI-assisted coding. Public interviews and articles in mid-to-late 2025 report that a substantial share of production code in some teams/products is AI-generated (around 90%, with some products at 90–95%), with earlier reports citing $\geq 70\%$ at prior timepoints [49, 50, 51, 52]. These sources support the assumption that A_{phys} has been materially accelerated in practice; they do not, however, negate the per-epoch bounds in Theorem 1 nor the diagnostic asymmetries in Theorem 2.

D.7 Summary

AI shifts *where* the P–V–A triangle binds: A_{phys} accelerates; V_τ becomes a paradox of speed versus assurance; P_H often becomes the bottleneck. Small teams benefit structurally through lower τ_{coord} and smaller $\varepsilon_\star$. The per-epoch impossibility is unaffected: when $\tau \geq \Delta$, max-type shortfalls are unavoidable; when $\varepsilon_\star > 0$, not all orderings can be optimal. Multi-epoch routing (portfolio rebalancing across VA/PA/PV) remains the pragmatic doctrine (Section C), with GPT-5 providing a deployed case of multi-turn (multi-epoch) operation and user override (Section 8.3).

References

[1] Lucas, R. E., Jr. (1976). "Econometric Policy Evaluation: A Critique." *Carnegie–Rochester Conference Series on Public Policy.*

[2] Goodhart, C. A. E. (1975). "Problems of Monetary Management: The U.K. Experience." *Papers in Monetary Economics.*

[3] Manheim, D., & Garrabrant, S. (2018). "Categorizing Variants of Goodhart's Law." arXiv:1803.04585.

[4] Brewer, E. A. (2000). "Towards robust distributed systems." Keynote, ACM Symposium on Principles of Distributed Computing (PODC).

[5] Gilbert, S., & Lynch, N. (2002). "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services." *ACM SIGACT News*, 33(2), 51–59.

[6] Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (eds.) (2009). *Dataset Shift in Machine Learning*. MIT Press.

[7] Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., & Herrera, F. (2012). "A unifying view on dataset shift." *Pattern Recognition*, 45(1), 521–530.

[8] K. J. Arrow. *Social Choice and Individual Values*. Wiley, 1951. (2nd ed., 1963).

[9] K. J. Arrow. "A Difficulty in the Concept of Social Welfare." *Journal of Political Economy*, 58(4):328–346, 1950.

[10] Strathern, M. (1997). "'Improving ratings': audit in the British University system." *European Review*.

[11] J. Hestness, S. Narang, N. Ardalani, G. Diamos, et al. Deep Learning Scaling is Predictable, Empirically. *arXiv:1712.00409*, 2017.

[12] J. Kaplan, S. McCandlish, T. Henighan, et al. Scaling Laws for Neural Language Models. *arXiv:2001.08361*, 2020.

[13] J. Hoffmann, S. Borgeaud, A. Mensch, et al. Training Compute-Optimal Large Language Models. *arXiv:2203.15556*, 2022.

[14] L. Bottou, J. Peters, J. Quiñonero-Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard, E. Snelson. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *Journal of Machine Learning Research*, 14:3207–3260, 2013.

[15] M. Dudík, J. Langford, L. Li. Doubly Robust Policy Evaluation and Learning. In *Proceedings of ICML*, 2011.

[16] L. Li, W. Chu, J. Langford, R. E. Schapire. Unbiased Offline Evaluation of Contextual Bandit-based News Article Recommendation Algorithms. In *Proceedings of WSDM*, 2011.

[17] A. Swaminathan, T. Joachims. Counterfactual Risk Minimization: Learning from Logged Bandit Feedback. In *Proceedings of ICML*, 2015. (See also *JMLR* 16:1731–1755, 2015.)

[18] A. Swaminathan, T. Joachims. The Self-Normalized Estimator for Counterfactual Learning. In *NeurIPS*, 2015.

[19] T. Lattimore, Cs. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[20] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, Z. Wen. A Tutorial on Thompson Sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018.

[21] T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning* (2nd ed.). Springer, 2009.

[22] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger. On Calibration of Modern Neural Networks. In *Proceedings of ICML*, 2017.

[23] G. W. Brier. Verification of Forecasts Expressed in Terms of Probability. *Monthly Weather Review*, 78(1):1–3, 1950.

[24] R. Kohavi, D. Tang, Y. Xu. *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing*. Cambridge University Press, 2020.

[25] R. Johari, L. Pekelis, D. J. Walsh. Always Valid Inference: Bringing Sequential Analysis to A/B Testing. *arXiv:1512.04922*, 2015.

[26] R. Johari, P. Koomen, L. Pekelis, D. J. Walsh. Always Valid Inference: Continuous Monitoring of A/B Tests. *Operations Research*, 70(3):1806–1821, 2022.

[27] Office of Evaluation Sciences (OES). Blocking in Randomized Evaluations (Practice Guide). U.S. GSA, methods note, online resource, accessed 2025.

[28] A. Deng, Y. Xu, R. Kohavi, T. Walker. Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data (CUPED). In *Proceedings of WSDM*, 2013.

[29] O. Chapelle. Modeling Delayed Feedback in Display Advertising. In *Proceedings of KDD*, 2014.

[30] N. Fabijan, J. Gupchup, A. Gupta, P. Omhover, L. Vermeer, P. Dmitriev. Diagnosing Sample Ratio Mismatch in Online Controlled Experiments: A Taxonomy and Rules of Thumb for Practitioners. In *Proceedings of KDD*, 2019.

[31] B. Beyer, C. Jones, J. Petoff, N. R. Murphy (eds.). *The Site Reliability Workbook*. O' Reilly, 2018. (See Chapter "Canarying Releases"; also Google SRE online workbook.)

[32] B. Beyer, C. Jones, J. Petoff, N. R. Murphy (eds.). *Site Reliability Engineering: How Google Runs Production Systems*. O' Reilly, 2016. (Error Budgets & SLO chapters; Google SRE online resources.)

[33] DORA. The Four Keys (DORA software delivery metrics): Deployment Frequency, Lead Time for Changes, Change Failure Rate, Time to Restore Service. dora.dev (official), accessed 2025.

[34] DORA. *2022 Accelerate State of DevOps Report*. Google Cloud / dora.dev, 2022.

[35] B. Kveton, Z. Wen, A. Ashkan, Cs. Szepesvári. Cascading Bandits: Learning to Rank in the Cascade Model. In *Proceedings of ICML*, 2015. (See also NeurIPS 2015 on combinatorial cascading bandits.)

[36] OpenAI (2025). *GPT-5*. OpenAI.

[37] OpenAI (2025). *GPT-5 System Card*. OpenAI.

[38] InfoQ (2025). "OpenAI's GPT-5 Debuts with Router and New Model Sizes."

[39] Latent Space (2025). "GPT-5 Router: Mixture of Models."

[40] TechCrunch (2025). "ChatGPT's model picker is back."

[41] Brooks, F. P. (1975). *The Mythical Man–Month: Essays on Software Engineering*. Addison–Wesley.

[42] Boehm, B. W. (1981). *Software Engineering Economics*. Prentice–Hall.

[43] Pollack, J., Helm, J., & Adler, D. (2018). "What is the Iron Triangle, and how has it changed?" *International Journal of Managing Projects in Business*, 11(2), 527–547. doi:10.1108/IJMPB-09-2017-0107

[44] Ronny Kohavi and Lenny Rachitsky. The Ultimate Guide to A/B Testing (*Lenny's Newsletter* podcast + transcript). July 2023. Available at: https://www.lennysnewsletter.com/p/the-ultimate-guide-to-ab-testing. Discusses typical fail rates; notes that other companies (e.g., Booking, Google Ads) have published 80–90% failure rates (cross-company anecdote). Accessed November 6, 2025.

[45] Erin Weigel. *Concept != Execution* — Digital Elite Camp 2017 (conference talk video). 2017. Available at: https://www.youtube.com/watch?v=3O2D0C5fxd0. Frequently cited when quoting "90% of experiments fail" at Booking.com. Accessed November 6, 2025.

[46] NightMonkey Blog. Digital Elite Camp — Day 2 (conference recap with quotes from Erin Weigel). July 2017. Available at: https://www.nightmonkey.nl/blog/digital-elite-camp-day-2/. Quote: "90% of their experiments fail. Only 10% seem to provide a statistically significant uplift." Accessed November 6, 2025.

[47] Luke Wroblewski. Conversions@Google 2017 — Building a Testing Culture (notes from Stuart Frisby's talk). March 2017. Available at: https://www.lukew.com/ff/entry.asp?1965=. Background on Booking.com's experimentation culture (no explicit fail-rate percentage). Accessed November 6, 2025.

[48] Kohavi, R. (2017). "Trustworthy A/B Tests: Pitfalls in Online Controlled Experiments." eMetrics Summit talk. PDF.

[49] L. Rachitsky. "Anthropic' s CPO on what comes next | Mike Krieger." *Lenny' s Newsletter*, Jun 5, 2025. Available at: Anthropic' s CPO on what comes next | Mike Krieger.

[50] Lenny' s Podcast. "Anthropic' s CPO on what comes next | Mike Krieger (co-founder of Instagram)." Apple Podcasts, Jun 5, 2025. Available at: Anthropic' s CPO on what comes next.

[51] *WIRED*. "Inside Anthropic' s First Developer Day, Where AI Agents Took Center Stage." May/Jun 2025. Available at: Inside Anthropic' s First Developer Day.

[52] *Business Insider*. "Anthropic CEO says 90% of code written by teams at the company is done by AI." Oct 2025. Available at: Anthropic CEO says 90% of code ... by AI.