Intelligence Emerges From Loops, Not FLOPs: Feedback Bandwidth, Environments, and the Geometry of Experience

Jace Hall[1]

[1]**Hall iNtelligence, LLC**

ABSTRACT

Recent discussions of AI scaling have emphasized compute (FLOPs) and parameter counts as the primary drivers of capability. While scaling laws such as Kaplan et al. (2020) and Chinchilla (Hoffmann et al., 2022) demonstrate empirical regularities, they risk obscuring the deeper mechanisms by which intelligence emerges. This paper argues that intelligence is a product of feedback loops, not FLOPs. Environments are not just benchmarks, but operators on policy: they shape identity as much as they measure ability. I introduce the concept of feedback bandwidth (B), defined along dimensions of latency, veracity, granularity, and counterfactual richness, and sketch a relationship $\Delta\text{Perf} \propto f(B) \cdot T$ to capture how capability growth scales with loop efficiency and experience budget. Examples from coding environments, curriculum learning, multi-agent interaction, and tool use illustrate how feedback geometry governs generalization and robustness. The commentary concludes with falsifiable predictions, grounded in recent literature, that improved feedback veracity, latency, granularity, and consolidation pipelines reduce sample complexity and enhance transfer. By reframing scaling through the lens of loops, this paper positions environment design as the true bottleneck for AGI development and highlights feedback geometry as a substrate-neutral lever for capability, alignment, and safety.

Keywords: AI scaling, FLOPs, feedback loops, environments, experience budget, AGI safety, machine learning theory

1. INTRODUCTION

The rapid progress of large-scale machine learning has fueled a growing belief that compute, parameter count, and token exposure are the primary determinants of intelligence. Scaling laws such as Kaplan et al. (2020) formalized this view, showing predictable gains in loss reduction with increasing FLOPs, while Chinchilla (Hoffmann et al., 2022) highlighted data efficiency tradeoffs. More recently, test-time compute approaches have further emphasized raw capacity as the bottleneck.

Yet as Karpathy (2025) and others have observed, the next frontier may not lie in larger models or more tokens, but in the environments in which these systems learn. Environments are not simply benchmarks or task lists. They are operators on policy: they do not merely measure ability, they actively shape identity. The intelligence that emerges is a direct function of the feedback geometry in which it is embedded.

This commentary advances a simple thesis: intelligence emerges from loops, not FLOPs. By loops, I mean feedback cycles in which action produces signal, which updates policy, which produces new action. What matters is not only how much compute or data is consumed, but the bandwidth, structure, and veracity of these loops.

The paper develops this thesis in several steps. First, I define the notion of feedback bandwidth, B, and outline its key dimensions: latency, veracity, granularity, and counterfactual richness. I then present a simple sketch relating B to learning progress over time. Next, I analyze how environments, curricula, verifiers, and multi-agent settings serve as scaffolds that amplify loop quality. Finally, I provide falsifiable predictions, grounded in recent empirical work, to illustrate

how optimizing feedback geometry accelerates learning more effectively than scaling FLOPs alone.

The central claim is that environment design should be treated as a first-class optimization problem. By improving the geometry of experience, we shape not only performance but the very identity of intelligence.

2. BACKGROUND: SCALING LAWS AND BOTTLENECKS

The dominant narrative in modern AI research has emphasized scaling as the primary driver of capability. Kaplan et al. (2020) observed that language model loss follows smooth power laws in model size, dataset size, and compute budget, suggesting predictable improvements from ever-larger systems. Hoffmann et al. (2022) refined this view with the Chinchilla scaling laws, showing that compute-optimal performance requires balancing parameter count with dataset size, shifting attention to token availability as the new bottleneck.

This compute-centric perspective has fueled the widespread notion that more FLOPs inevitably yield more intelligence. Indeed, much of the community's focus has been on hardware roadmaps, data curation, and parameter scaling. However, diminishing returns have become increasingly evident. Training massive models on static datasets often produces marginal gains at escalating costs, while data scarcity and repetition limit future scaling potential.

Recent developments underscore these limits. Approaches such as OpenAI's o1-preview have explored test-time compute, where reasoning is extended by running more inference steps without increasing training size. While this can improve accuracy, it again highlights that FLOPs alone are insufficient: the structure of feedback and the quality of the reasoning environment matter just as much.

In parallel, Karpathy (2025) and others have emphasized that the true bottleneck lies not in FLOPs or tokens but in environments. Environments provide the loops that convert compute into learning. They determine how quickly feedback arrives, how reliable it is, how granular credit assignment can be, and how richly counterfactuals can be explored. In short, environments are not passive testbeds but active operators on policy.

This shift in perspective motivates the central thesis of this paper: that intelligence emerges from the geometry of loops, not from raw FLOPs.

Related ideas appear in embodied cognition and active inference, which emphasize agent–environment coupling. Our contribution is to provide a simple operational metric, B, and a research program for engineering high-bandwidth loops in LLM-centric systems.

A related emphasis on loop quality appears in feedback control (e.g., control bandwidth and stability margins), and in curriculum learning as a progressive shaping of experience. Here we focus on a simple operational metric, B, and a practical program for engineering high-bandwidth loops in LLM-centric systems.[1]

3. ENVIRONMENTS AS OPERATORS ON POLICY

Environments are often treated as static benchmarks or evaluation suites. In practice, they are far more than neutral arenas: they are operators on policy. Each environment actively shapes the trajectory of learning by determining which signals are available, how they are structured, and how they update the agent.

A policy π_θ trained within a given environment E is not only evaluated by it but sculpted by it. The intelligence that emerges is a direct function of the feedback geometry in which the policy is embedded. Latency, reliability, and richness of signals all condition the identity of the learner. In this sense, environments cause intelligence just as much as they measure it.

This perspective reframes environment design as a first-class optimization problem. Rather than simply constructing benchmarks for post hoc evaluation, we should be engineering environments that maximize learning efficiency and robustness. The critical concept here is *feedback bandwidth*, B, which captures the effective rate at which useful signal flows through the loop from action to policy update.

Feedback bandwidth has several core dimensions:

[1] See, e.g., *Feedback Systems* by Åström & Murray (2010) and Bengio et al. (2009) on curriculum learning.

- **Latency**: the time required for an action to generate a training signal.
- **Veracity**: the probability that the signal is correct and aligned with intended objectives.
- **Granularity**: the resolution of the signal, from coarse episode-level feedback to token-level or step-level credit assignment.
- **Counterfactual richness**: the extent to which the environment allows "what if" exploration, enabling policies to test alternative trajectories and learn from hypothetical outcomes.

As a concrete example, consider a coding environment. Low latency corresponds to receiving immediate linter or unit test results as code is written, rather than after a full compile. High granularity means pinpointing the precise token or line responsible for an error, rather than simply reporting "build failed." Improvements in these dimensions bend learning curves without changing model size or parameter count.

Optimizing feedback bandwidth is therefore central to capability emergence. It determines how much effective learning progress can be extracted per unit of compute or experience. In the following section, I formalize this intuition with a simple sketch relating B to performance growth.

4. FORMAL SKETCH OF BANDWIDTH AND PROGRESS

To make the concept of feedback bandwidth precise, consider an environment E with effective bandwidth B. Let π_θ denote a policy with parameters θ trained in E over a budget of T interactions.

We define B as a monotone, saturating function of four factors:

$$B = g\Big(\tfrac{1}{\text{latency}}, \text{veracity}, \text{granularity}, \text{counterfactual richness}\Big)$$

where g increases with each argument but exhibits diminishing returns.

The expected learning progress over the interaction budget T can then be expressed as:

$$\Delta\text{Perf}(\pi_\theta; T, E) \propto f(B) \cdot T$$

Here, ΔPerf represents the improvement in performance, and $f(B)$ is a concave function capturing diminishing returns as B grows. This sketch illustrates that intelligence gains are not solely a function of model size or compute, but are mediated by the structure and quality of feedback loops.

Two implications follow:

- Increasing T without improving B eventually yields diminishing returns. More experience in a poor feedback geometry produces limited capability growth.
- Increasing B improves the efficiency of learning per unit of interaction. High-bandwidth environments produce greater capability from the same interaction budget.

As a practical illustration, one can approximate B in real systems with simple proxies. For example, let

$$\hat{B} = \sigma\Big(\alpha \cdot \tfrac{1}{\hat{\ell}} + \beta \cdot \hat{v} + \gamma \cdot \hat{g} + \delta \cdot \log(1 + \hat{c})\Big)$$

where $\hat{\ell}$ is observed latency per action (in milliseconds), $\hat{v}$ is the fraction of outputs passing a verifier, $\hat{g}$ is the inverse average span length of error localization (a granularity proxy), and $\hat{c}$ is the number of distinct counterfactuals surfaced. $\sigma(x) = 1/(1 + e^{-x})$ ensures saturation, and $\alpha, \beta, \gamma, \delta$ are tunable weights.

A toy pseudocode sketch:

```
for each interaction step t:
    latency = elapsed_ms(action_t -> feedback_t)
    veracity = 1 if verifier_pass else 0
    granularity = 1.0 / (1 + span_length_of_error)
    cf = count_valid_counterfactuals()
    score = alpha*(1/latency) + beta*veracity + \
            gamma*granularity + delta*log(1+cf)
    Bhat_t = 1/(1+exp(-score))
```

This $\hat{B}$ is not unique, but illustrates how feedback bandwidth can be operationalized for experiments or A/B tests. Any monotone, saturating composition aligned with the four dimensions is compatible with the framework.

This formalism captures the central thesis of this paper: the bottleneck is not only more FLOPs or larger parameter counts, but the effective bandwidth of the feedback loops that structure learning. The next sections examine how environments, verifiers, and curricula serve as scaffolds that improve B, and how experience can be managed as a resource in its own right.

5. THE EXPERIENCE BUDGET

Compute and parameter count are widely tracked as primary resources in machine learning, but the true bottleneck is experience. An agent's trajectory of improvement depends on how effectively raw experience is transformed into durable policy updates. This perspective reframes intelligence growth as a process of managing an *experience budget*.

The productive loop can be summarized as:

$$\text{act} \rightarrow \text{log traces} \rightarrow \text{compress} \rightarrow \text{distill}$$

Each action produces traces, which are logged. These traces are compressed into efficient representations, which are then distilled into model weights. The pipeline mirrors the biological role of sleep, where experience is replayed, consolidated, and integrated into long-term memory. In engineered systems, this corresponds to offline reinforcement learning, dataset aggregation, and distillation pipelines.

Managing the experience budget requires explicit design choices. Key considerations include:

- **Trace fidelity**: which signals and contexts are worth logging.
- **Compression**: balancing information preservation with representational efficiency.
- **Distillation frequency**: determining how often traces should be consolidated into weights for maximum stability.
- **Replay strategies**: selecting which experiences to revisit to optimize generalization under shifting distributions.

A well-managed experience budget can significantly improve robustness. For example, distillation pipelines have been shown to improve resistance to distributional shift in question answering (Hsieh et al., 2023) and to enhance stability in autonomous driving (Yu et al., 2025). These findings support the claim that structured act-trace-compress-distill loops reduce regret under environment shifts compared to equal-compute baselines that lack distillation.

By elevating experience to the status of a budgeted resource, on par with compute, we can design systems that grow capabilities more reliably. In the following sections, I discuss how scaffolds such as verifiers and curricula further enhance feedback bandwidth and experience efficiency.

6. SCAFFOLDS OF INTELLIGENCE

Intelligence does not emerge from isolated policies alone. It is scaffolded by the structures that provide reliable signals, enforce consistency, and amplify feedback. In modern machine learning,

these scaffolds take the form of verifiers, critics, and constraint systems that accelerate training while reducing dependence on scarce human labels.

Verifiers can be understood as external oracles that increase the veracity and granularity of feedback. Examples include:

- **Programmatic checkers**: linters, unit tests, and type systems that provide rapid, reliable feedback in coding environments.
- **Retrieval-grounded critics**: retrieval-augmented models that cross-check outputs against trusted sources.
- **Ensemble consistency tests**: comparing outputs across multiple models to identify contradictions or errors.
- **Constraint solvers**: symbolic engines or formal methods that enforce logical or mathematical validity.

These scaffolds function as oracles with low latency and high veracity, raising the effective feedback bandwidth B without proportional increases in human effort. For example, in programming environments, linters and unit tests deliver rapid token-level error signals that shape policies more efficiently than delayed pass/fail judgments.

Scaffolds also play a safety role. By embedding verifiers within training loops, we can align incentives so that the easiest way for a policy to receive reward is also the safest way to behave. A system that must satisfy programmatic constraints, consistency checks, and logical solvers learns to produce outputs that are both useful and reliable, reducing reliance on post hoc filtering or reinforcement learning from human feedback.

In this way, verifiers and related scaffolds serve as accelerators of capability emergence. They enhance the geometry of loops by making feedback cheaper, sharper, and more abundant, thereby shaping both performance and identity. The next section extends this logic to curricula, which act as control systems for managing the trajectory of experience.

7. CURRICULUM AS CONTROL

Curriculum learning is often conceived as a sequence of tasks arranged by difficulty. From the perspective of loops and feedback geometry, a curriculum is better understood as a control system that regulates the trajectory of experience.

A well-designed curriculum should not be a static spreadsheet of tasks. Instead, it should adapt dynamically to the learner, auto-scheduling experiences that maximize learning progress. The system itself becomes a moving target, always slightly ahead of the agent's capabilities.

Key principles of curriculum as control include:

- **Progress-based scheduling**: promote tasks that yield the steepest generalization gradient, measured by learning progress rather than static difficulty labels.
- **Adversarial dynamics**: inject adversaries that track the learner, ensuring continual adaptation and avoiding premature convergence.
- **Self-adjusting difficulty**: maintain a challenge level that is neither trivial nor impossible, keeping the agent in a zone of maximal growth.
- **Task diversity**: ensure exposure to varied environments to build robustness and prevent overfitting to narrow domains.

Viewed in this way, curriculum design is analogous to a control loop with stability margins. Too little pressure yields stagnation, while excessive pressure leads to collapse. The optimal curriculum maintains the agent at the edge of competence, where feedback bandwidth is maximized and experience is most efficiently converted into durable capability.

By reframing curriculum as an active control system, we highlight its role as another scaffold of intelligence. Just as verifiers increase feedback veracity, curricula regulate the flow of challenge to sustain growth. The following section extends this principle to safety, arguing that environment design can align incentives so that the safest behavior is also the most rewarding.

8. SAFETY THROUGH RISK-SCULPTED ENVIRONMENTS

Alignment cannot be achieved by prompts or superficial constraints alone. It must be embedded in the geometry of feedback itself. Environments can be designed so that the easiest way to gain reward is also the safest way to behave. In this framing, safety emerges not from external oversight alone but from incentives sculpted into the loop.

A risk-sculpted environment is one in which:

- **Cheap but sharp consequences** provide immediate correction for unsafe actions without requiring costly human intervention.
- **Reward is isomorphic across training and deployment**, so that policies do not face incentive mismatches once deployed.
- **Safe shortcuts** exist such that the most efficient path to reward is also the one consistent with alignment objectives.
- **Adversarial pressure** is included to expose unsafe edge cases during training rather than after deployment.

The principle is simple: if incentives are aligned within the training loop, policies will naturally generalize toward safe behavior in deployment. Conversely, if training environments tolerate unsafe shortcuts or misaligned rewards, incoherence is baked into the policy itself.

This approach reframes safety not as an afterthought but as an intrinsic property of loop design. Just as verifiers act as scaffolds that increase signal quality, risk-sculpted environments act as regulators that shape incentives. Together, they raise the effective bandwidth of feedback while ensuring that capability growth remains stable and aligned.

The next section examines how to evaluate these properties, moving beyond final task scores toward metrics that better capture stability, transfer, and robustness.

9. EVALUATION BEYOND FINAL SCORES

Final task accuracy or benchmark scores provide only a partial view of intelligence. From the perspective of loops and feedback geometry, evaluation must capture the dynamics of learning itself: stability, adaptability, and transfer.

Several alternative metrics are especially relevant:

- **Time to stability**: how quickly a policy reaches consistent performance within a new environment. Faster stabilization indicates higher effective feedback bandwidth.
- **Regret under distribution shift**: the cumulative penalty incurred when the environment changes. Policies trained in stable, high-bandwidth loops should exhibit lower regret when facing novel or perturbed conditions.
- **Transfer coefficient**: the degree to which skills or representations acquired in one environment improve performance in unrelated environments. Fine-grained, veridical feedback is expected to raise transfer by producing more generalizable structures.
- **Robustness to adversaries**: the extent to which performance degrades when exposed to adaptive opponents or adversarial perturbations. Policies with richer counterfactual exploration should maintain higher robustness.

These metrics move beyond static endpoints and instead evaluate the quality of learning dynamics. They help distinguish between policies that succeed by exploiting shallow shortcuts and those that have developed deeper competence.

By adopting such measures, we can more accurately assess what environments are truly buying us. The following sections extend this evaluation to multi-agent interaction and tool use, which further expand the geometry of feedback.

10. MULTI-AGENT AND TOOL USE

Real-world intelligence is rarely solitary. It emerges in social and tool-rich contexts, where agents must negotiate, cooperate, compete, and orchestrate actions under constraints. Training environments that omit these dynamics risk producing brittle policies that fail when deployed into interactive, multi-agent settings.

Multi-agent environments create feedback geometries that elicit higher-order cognition. Negotiation, coalition forming, and deception require theory of mind - the ability to model the beliefs and intentions of others. Policies that never face such dynamics in training are unlikely to spontaneously generalize them at deployment. By embedding social interaction into the loop, environments can accelerate the emergence of strategic reasoning and coordination.

Tool use similarly expands the geometry of feedback. Exposing policies to real tools and APIs - complete with latency, errors, and rate limits - forces them to learn orchestration under constraint. Evaluation should not only measure text prediction accuracy, but also how well agents can choose, sequence, and recover from tool calls in realistic conditions.

Together, multi-agent and tool-rich environments provide diverse, high-bandwidth loops that go beyond static task completion. They mirror the ecological pressures under which natural intelligence evolved, and they help sculpt policies that are more adaptable, robust, and coherent.

The next section proposes a unifying framework: a modern "Gym" for LLMs that integrates these elements into composable, programmatically verifiable environments.

11. A GYM FOR LLMS

If environments are the true bottleneck, then building them should be treated as a central research priority. Just as OpenAI Gym (Brockman et al., 2016) provided a standardized suite of reinforcement learning tasks, a modern Gym for large language models should integrate the principles of feedback geometry and scaffolding.

A next-generation Gym should include the following components:

- **Composable tasks**: modular environments that can be combined to test diverse capabilities under consistent feedback protocols.
- **Programmatic verifiers**: linters, unit tests, solvers, and retrieval-augmented critics that deliver rapid and reliable feedback signals.
- **Tool integration**: built-in support for external APIs and tools, complete with latency, errors, and rate limits, to reflect realistic constraints.
- **Auto-curriculum with adversaries**: dynamic scheduling of challenges and adaptive opponents to keep policies operating at the edge of competence.
- **Trace logging and distillation hooks**: infrastructure for capturing experience traces and feeding them into offline consolidation pipelines.
- **Non-stationarity controls**: adjustable knobs for latency, noise, and distributional shift to test stability under changing conditions.
- **Multi-agent protocols**: social interaction modes for negotiation, collaboration, and competition.
- **Metrics beyond accuracy**: standardized measures such as time to stability, regret under shift, transfer coefficients, and robustness to adversaries.

Our vision aligns with recent tool-use benchmarks such as ToolBench (Qin et al., 2023) and adversarial curriculum frameworks like POET (Wang et al., 2019), but centers evaluation on bandwidth and stability rather than endpoint accuracy.

Classical curriculum learning [13] and modern feedback control perspectives [12] both recommend shaping the sequence and bandwidth of experience; our proposal integrates these ideas into a single operational testbed for LLMs.

Such a Gym would make feedback bandwidth B a measurable quantity and enable systematic study of how environment geometry shapes intelligence. By integrating verifiers, curricula, tool use, and social dynamics into a unified framework, it would provide a testbed not only for scaling capability but also for aligning incentives and maintaining stability.

The following section grounds these principles in falsifiable predictions, showing how improvements in latency, veracity, granularity, and consolidation pipelines should yield measurable efficiency and robustness gains.

12. FALSIFIABLE PREDICTIONS

A theory of intelligence grounded in feedback geometry must generate testable predictions. The following claims specify measurable outcomes that follow from improving key dimensions of feedback bandwidth B and managing the experience budget effectively.

1. **Latency and veracity.** Doubling veracity and halving latency in a verified environment will reduce sample complexity by at least 30 percent on code, tool-use, and navigation tasks at matched parameter counts. This aligns with Hu et al. (2023), which reported up to 40 percent sample efficiency gains when LLMs were paired with oracle verifiers in MiniGrid and code-generation tasks.

2. **Granularity.** Increasing feedback granularity from coarse episode-level signals to token-level or step-level signals will raise transfer coefficients across unrelated environments by at least 20 percent at constant compute. This is consistent with Cui et al. (2018), who found 15–25 percent improvements in downstream classification from fine-grained labels, and Pan et al. (2024), who demonstrated multi-granularity transfer gains in continual reinforcement learning.

3. **Experience consolidation.** A structured act $\rightarrow$ trace $\rightarrow$ compress $\rightarrow$ distill pipeline will improve robustness to non-stationarity by at least 15 percent compared to equal-compute baselines without distillation, measured as regret under environment shift. Empirical support comes from Hsieh et al. (2023), who observed 10–20 percent stability gains in domain-shift QA with distillation, and Yu et al. (2025), who reported significant collision reductions in autonomous driving via distillation-based reinforcement learning pipelines.

These predictions are deliberately conservative, grounded in existing empirical work while projecting forward to more comprehensive experiments. They offer clear benchmarks by which the loop-centric view of intelligence can be validated or refuted.

The next section discusses broader implications: why scaling laws plateau, why environment design is the true bottleneck, and how stability-preserving loops provide a pathway for both capability and safety.

13. DISCUSSION AND IMPLICATIONS

The arguments developed in this paper suggest a reframing of intelligence: not as the product of raw compute or parameter count, but as the emergent property of feedback loops shaped by environments. This loop-centric perspective offers several broad implications.

Limits of scaling laws. Scaling laws provide useful empirical regularities, but they ultimately plateau when environments fail to deliver high-bandwidth, coherent feedback. Larger models trained on static, low-veracity datasets produce diminishing returns, highlighting that FLOPs alone are insufficient for continued capability growth.

Environments as the true bottleneck. The geometry of feedback - latency, veracity, granularity, counterfactual richness - determines how effectively compute is converted into competence. Designing environments that optimize these dimensions yields greater gains than adding parameters in poor environments. Environment design should therefore be treated as a first-class optimization problem.

Scaffolds and coherence. Verifiers, curricula, and multi-agent settings act as scaffolds that enhance feedback bandwidth and stabilize growth. By embedding safety constraints into these

scaffolds, we align incentives such that coherent and safe behavior is the path of least resistance. Stability-preserving loops thus serve as both the engine of intelligence and the substrate of alignment.

Forecasting AGI. If intelligence is a function of loop quality rather than FLOPs, then AGI progress will hinge less on hardware scaling curves and more on advances in environment engineering. Predictive timelines should incorporate the rate of innovation in verifiers, curricula, and training ecosystems rather than focusing narrowly on compute budgets.

This reframing challenges the community to broaden its focus. Compute remains necessary, but it is not sufficient. The future of AI depends on how well we design the loops through which policies learn, consolidate, and generalize. Intelligence emerges from loops, not FLOPs.

14. CONCLUSION

This paper has advanced a simple but consequential claim: intelligence emerges from loops, not FLOPs. While compute and parameter count are necessary ingredients, they do not by themselves determine capability. What matters is the geometry of feedback - the latency, veracity, granularity, and counterfactual richness of the environments in which policies are embedded.

By formalizing feedback bandwidth, B, and situating it within the broader discourse on scaling, we have highlighted why environments should be treated as first-class optimization problems. Scaffolds such as verifiers, curricula, and multi-agent protocols increase B and stabilize growth. Risk-sculpted environments align incentives, making safe behavior the most efficient path to reward. Experience itself must be budgeted and consolidated through act-trace-compress-distill pipelines.

The loop-centric view also generates falsifiable predictions, some of which already find support in the literature. Improvements in latency, veracity, granularity, and consolidation consistently reduce sample complexity, improve transfer, and enhance robustness. These results suggest that environment design is not a secondary consideration but the primary bottleneck for progress.

Looking forward, the challenge is to systematically design and measure feedback geometry. By optimizing loops, not only can we accelerate the emergence of intelligence, we can also embed stability and alignment into its foundations. The future of AI will be shaped less by how many FLOPs we can spend, and more by how well we design the loops through which policies learn.

While the framework is general, we have also illustrated how B can be approximated in practice through simple proxies and heuristics. This provides a concrete starting point for experimental validation of the loop-centric view.

For a formal treatment of invariants as the substrate of robust emergence, see our companion paper on the Law of Invariant-Preserving Loops [Hall, 2025d], which completes the sequence initiated in [Hall, 2025a; Hall, 2025b].

APPENDIX A: IMPLEMENTATION GUIDE FOR $\hat{B}$

Proxies and logging. The table below shows illustrative proxies you can log in practice. These are not unique; any monotone, saturating proxies aligned with the four dimensions are compatible with the framework.

Dimension	Example proxy	Logging event
Latency	$\hat{\ell}$: wall-clock ms per action-to-signal	timestamp(action), timestamp(feedback)
Veracity	$\hat{v}$: fraction of outputs passing verifier	boolean verifier_pass per step
Granularity	$\hat{g}$: $1/(1+\text{error-span})$	character or token span for error
Counterfactuals	$\hat{c}$: distinct valid alternatives surfaced	count of valid "what-if" branches

Table 1. Illustrative proxies for $\hat{B}$ components.

Rolling estimate. Maintain a rolling average over a window W steps:

$$\overline{\hat{B}}_t = \frac{1}{W} \sum_{k=t-W+1}^{t} \hat{B}_k$$

Use $\overline{\hat{B}}_t$ for A/B tests, curriculum gating, or early stopping.
Minimal integration snippet. Attach logging to the loop that produces feedback:

```
for step t in interaction_loop:
    t0 = now_ms()
    action = policy(state)
    feedback = environment(action)      # includes verifier outputs if any
    latency = now_ms() - t0
    veracity = 1 if feedback.passed else 0
    granularity = 1.0 / (1 + feedback.error_span)
    cf = feedback.num_valid_counterfactuals
    score = alpha*(1/latency) + beta*veracity + \
            gamma*granularity + delta*log(1+cf)
    Bhat = 1/(1+exp(-score))
    log({latency, veracity, granularity, cf, Bhat})
```

This guide is intentionally minimal. The aim is to provide a concrete starting point that any ML team can implement immediately.

APPENDIX B: MINIMAL EXPERIMENTAL PROTOCOLS

Protocol 1: Latency $\times$ Veracity.

- *Setup:* A verified coding or MiniGrid-like task with a programmatic checker.
- *Manipulations:* Halve latency by moving checks from batch to streaming; double veracity by improving oracle coverage.
- *Metric:* Sample complexity to fixed score, and learning curve area under curve (AUC).
- *Hypothesis:* At matched parameters, reducing latency and increasing veracity cuts sample complexity by $\geq 30\%$.

Protocol 2: Granularity and Transfer.

- *Setup:* Train with coarse, step-level, and token-level error signals.
- *Metric:* Transfer coefficient to unrelated environments; stability under mild shift.
- *Hypothesis:* Finer granularity raises transfer by $\geq 20\%$ at constant compute.

Protocol 3: Experience Consolidation.

- *Setup:* Add a nightly act$\rightarrow$trace$\rightarrow$compress$\rightarrow$distill job to a baseline.
- *Metric:* Regret under non-stationarity and time to stability after a distribution change.
- *Hypothesis:* Distillation reduces regret by $\geq 15\%$ versus equal-compute baselines.

APPENDIX C: THREATS TO VALIDITY

Proxy choice. Different domains may favor different proxies for latency, veracity, granularity, and counterfactual richness. Our $\hat{B}$ example is illustrative, not exclusive.
Confounds. Gains from increased $\hat{B}$ can be confounded with hidden curriculum or model selection bias. Use fixed seeds, ablations, and protocol preregistration.
External validity. Results in coding or MiniGrid-like tasks may not immediately transfer to large open-ended systems. We treat these as footholds to iterate on the metric.
Metric gaming. As with any metric, $\hat{B}$ can be gamed. Use adversarial audits and ensemble critics to look for degenerate strategies.

REFERENCES

[1] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). *Scaling laws for neural language models*. arXiv preprint arXiv:2001.08361.

[2] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J.-B., Damoc, B., Clark, A., Casas, D. L., Guy, A., Menick, J., Ring, R., Hennigan, T., Caine, A., Caine, B., Jones, C., et al. (2022). *Training compute-optimal large language models*. arXiv preprint arXiv:2203.15556. (Chinchilla scaling laws)

[3] Sutton, R. S. (2019). *The bitter lesson*. Published online at: http://www.incompleteideas.net/IncIdeas/BitterLesson.html

[4] Hu, B., Zhao, C., Zhang, P., Zhou, Z., Yang, Y., Xu, Z., and Liu, B. (2023). *Enabling intelligent interactions between an agent and an LLM: A reinforcement learning approach*. arXiv preprint arXiv:2306.03604.

[5] Cui, Y., Song, Y., Sun, C., Howard, A., and Belongie, S. (2018). *Large scale fine-grained categorization and domain-specific transfer learning*. arXiv preprint arXiv:1806.06193.

[6] Pan, C., Ren, L., Feng, Y., Xiong, L., Wei, W., Li, Y., and Yang, X. (2024). *Multi-granularity knowledge transfer for continual reinforcement learning*. arXiv preprint arXiv:2401.15098.

[7] Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.-Y., and Pfister, T. (2023). *Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes*. arXiv preprint arXiv:2305.02301.

[8] Yu, R., Zhang, X., Zhao, R., Yan, H., and Wang, M. (2025). *DistillDrive: End-to-end multi-mode autonomous driving distillation by isomorphic hetero-source planning model*. arXiv preprint arXiv:2508.05402.

[9] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). *OpenAI Gym*. arXiv preprint arXiv:1606.01540.

[10] Qin, Y., Chen, Z., Yao, Y., Chen, J., Lin, Z., Li, Z., ... and Zhang, D. (2023). *ToolBench: A comprehensive benchmark for tool-augmented LLMs*. arXiv preprint arXiv:2307.16789.

[11] Wang, R., Lehman, J., Clune, J., and Stanley, K. O. (2019). *POET: Open-ended coevolution of environments and their optimized solutions*. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO).

[12] Åström, K. J., and Murray, R. M. (2010). *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press.

[13] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). *Curriculum learning*. Proceedings of ICML.

[14] Hall, J. (2025a). *Illusions as Diagnostics, Coherence as Invariant*. Unpublished manuscript.

[15] Hall, J. (2025b). *Beyond Situational Awareness*. Unpublished manuscript.

[16] Hall, J. (2025c). *Intelligence Emerges from Loops, Not FLOPs*. Unpublished manuscript.

[17] Hall, J. (2025d). *The Law of Invariant-Preserving Loops: Toward Robust Emergence in Self-Modifying Agents*. Unpublished manuscript.