

Amortized Deliberation: Practice and the Dual-Process Partition

Flyxion
Independent Researcher

October 2026

Abstract

The dual-process vocabulary of a fast, automatic “System 1” and a slow, deliberate “System 2” is routinely applied to individual thoughts, tasks and, more recently, to classes of machine models. The observable basis for the division is speed, effort and reportability. This essay argues that these cannot by themselves identify two kinds of thought, because each is a property of a task together with a history of practice and not of the task alone. A reversal argument, with the keypad as its standard example, shows that no history-independent set of tasks separates easy from hard across a pair whose cost order reverses with practice. A one-process model in which practice amortizes earlier evaluation, with retrieval, applicability and fallback costs stated explicitly, reproduces the continuous practice curve and the apparent fast and slow regimes. A routing argument shows that a categorical sorter does no further explanatory work in that model, though it is not excluded. The shape of a cost distribution, including bimodality, is shown to underdetermine the number of mechanisms. Two illustrations are used. The first is a cached decision architecture, whose cost is an explicit continuous function of how much computation has been precomputed while its outputs are unchanged. The second is a progress-aligned recurrent model, in which practice supports reusable sequential computation that extends beyond the training length without externalized text. Together they indicate that learning history, representation, execution and reuse explain variation in cost, and that a division into two systems is neither needed for that explanation nor identified by the evidence. What survives of the distinction is a set of distinguishable dimensions.

1 The Content of a Partition Claim

The inference from differences in speed to differences in cognitive kind requires additional assumptions, and the dual-process vocabulary is used with at least three different commitments. On one interpretation, tasks themselves belong to history-independent classes, so that a thought is fast or slow whoever thinks it. On a second, the classes are mechanisms whose

engagement depends on the agent and the situation, and a given thought is handled by one or the other according to circumstance. On a third, the vocabulary is merely descriptive of speed and effort and specifies no architecture. These interpretations must be distinguished, because evidence against an intrinsic partition of tasks does not by itself establish that only one mechanism operates. The popular form of the doctrine adds a further element, which is that the slow mechanism is the one associated with awareness, effort and the self, and the fast mechanism is what remains once those are subtracted, and a dispatcher that assigns each thought to its mechanism is often posited. The vocabulary was given its widest currency by Kahneman (2011), and the original two-systems reading has been criticized as a conflation of several independent dimensions (Keren and Schul 2009; Melnikoff and Bargh 2018).

The most careful current statement of the dual-process view is that of Evans and Stanovich (2013), who separate the defining characteristics of the two types of processing from the attributes that merely tend to accompany them. On their account Type 1 processing is defined by autonomy, meaning that it does not require working memory or controlled attention, and Type 2 processing is defined by its dependence on working memory and by cognitive decoupling, the capacity for hypothetical thinking. Speed, effort, high capacity and association with consciousness are treated as correlates and not as definitions. The target of this essay is narrower than the whole of that position. The reversal argument below bears directly on classifications inferred from cost, since a ranking by speed or effort is a ranking by correlated attributes and the argument shows that such rankings depend on history. It leaves the proposed defining characteristics, autonomy and the use of working memory for decoupling, as questions for further investigation. Two cautions apply to them. A response that begins as a decoupled, working-memory-dependent computation can become autonomous through practice, so autonomy is itself history-dependent. And the defining characteristics are properties of processes. Attributing these characteristics to tasks independently of history would require a further argument; the reversal result concerns cost separation rather than autonomy or working-memory dependence directly.

The difficulty with inferring a partition from behaviour is that the behaviour in question, which is speed, depends on history. A thought is a computation performed by a particular agent at a particular time, and its cost depends on what that agent has already computed. The same sum is hard for a child and effortless for an adult, and this is not a fact about the sum. The partition claim treats the endpoints of a practice curve as if they were the outputs of two different machines. The present essay formalizes a single-process alternative in which cost falls with accumulated experience, and shows that the observations usually cited in favour of the partition are what such a process predicts, so that they cannot by themselves discriminate between the accounts. The claim defended is deliberately of two strengths. The formal results establish that a partition of tasks or of mechanisms cannot be inferred from speed alone. The further position, that no partition of mechanisms is needed, is argued from parsimony and

from worked models, and it is not offered as a theorem.

2 Setting

Let $\mathcal{T}$ be a set of tasks and $\mathcal{H}$ a set of histories, where a history $h \in \mathcal{H}$ records everything an agent has previously done that bears on its performance. Let $C : \mathcal{T} \times \mathcal{H} \rightarrow \mathbb{R}_{>0}$ assign to each task and history the resource cost of producing an adequate response, measured in whatever units are convenient, such as time, error-corrected attempts or arithmetic operations. A history is a partial record of past episodes, and for a task t we write $n_h(t)$ for the number of prior episodes in h that are relevant to t .

Definition 1 (History-independent cost separation). A set $E \subseteq \mathcal{T}$ is *cost-separating at h* if $C(t, h) < C(t', h)$ for all $t \in E$ and $t' \notin E$. It is *intrinsic* if it is cost-separating at every $h \in \mathcal{H}$.

A partition of tasks into fast and slow that is read off from observed speed would need to be a set of this kind, since otherwise the grouping changes when the agent’s history changes. A theory might instead classify processes, or classify tasks relative to an agent and a history, and the proposition below does not touch such theories directly. Its target is the inference from observed speed to a partition of tasks.

3 Difficulty Is a Relation

Definition 2 (Reversal pair). Tasks t, t' form a *reversal pair* if there are histories h, h' with $C(t, h) < C(t', h)$ and $C(t, h') > C(t', h')$.

Proposition 1 (No intrinsic partition across a reversal). *If E contains exactly one member of a reversal pair, then E is not intrinsic.*

Proof. Say $t \in E$ and $t' \notin E$. At h' we have $C(t, h') > C(t', h')$, which violates the requirement that every member of E be cheaper than every non-member. Hence E is not cost-separating at h' . The argument is symmetric if $t' \in E$ and $t \notin E$ and one uses h . $\square$

The proposition does not say that every set is non-intrinsic. A set containing both members of every reversal pair, or neither, escapes it, and a population whose histories are drawn from a common practice distribution may contain a set that is cost-separating for every history in that population. Such a set would be a fact about the population’s training, and it could fail when the distribution of histories changed. Nor does the proposition rule out two mechanisms whose relative engagement changes with practice. What it establishes is narrower: a partition of tasks inferred from speed cannot establish a partition of mechanisms, because speed depends on history. Showing that there is only one mechanism requires further evidence, and the sections below supply an argument and models, not a proof.

Example 1 (The keypad). A telephone arranges its digits with 1 2 3 in the top row, and a calculator arranges them with 7 8 9 in the top row. Let t be entering a fixed four-digit number on the telephone layout and t' be entering the same number on the calculator layout. Let h be the history of someone who has used telephones for years and calculators rarely, and h' the converse. Under these stipulated histories, which are plausible and not measured, $C(t, h) < C(t', h)$ and $C(t, h') > C(t', h')$. The task is identical in every respect that bears on its content, namely the same digits in the same order, and the ordering of cost flips with the history. Any set E that contains one of the two entries and not the other is therefore history-dependent by Proposition 1.

The same structure appears with a door that pushes or pulls, an appliance whose controls are laid out differently from the one previously used, or a language whose word order differs from the speaker's own. In each case the cost of the task is a joint property of the task and the practice behind it. The reversal argument applies wherever practice is task-specific. Where two tasks differ mainly in how much they have been rehearsed, and not in irreducible demands or attainable performance, their cost order can be exchanged by exchanging the rehearsal, and not every pair of tasks is of this kind.

4 Practice as Amortization

The standard empirical regularity is the power law of practice (Newell and Rosenbloom 1981), according to which cost after n relevant episodes is well described by

$$c(n) = a + b n^{-\beta}, \quad a \geq 0, b > 0, \beta > 0. \quad (1)$$

Other forms, such as exponentials, fit some data better, and Heathcote, Brown and Mewhort (2000) argue that a power law in averaged curves can arise from averaging over participants and items whose individual curves are exponential, so that the aggregate form does not identify the form of individual learning. The curve is therefore a stipulated illustration and not a universal empirical law, and nothing below depends on the particular form. What matters is that c is continuous, strictly decreasing and without a distinguished point.

Proposition 2 (No privileged threshold). *For any threshold $\theta > a$, the number of episodes at which the stipulated cost first falls to θ is $n^*(\theta) = (b/(\theta - a))^{1/\beta}$. The map $\theta \mapsto n^*(\theta)$ is a continuous bijection from (a, ∞) onto $(0, \infty)$, and the derivative $c'(n) = -b\beta n^{-\beta-1}$ is continuous and nonvanishing on $(0, \infty)$.*

Proof. Solve $a + b n^{-\beta} = \theta$ for n . The remaining claims are direct computations. $\square$

No value of θ corresponds to a distinguished point of the curve, since c' is continuous and nonvanishing, and the choice of θ belongs to the observer and not to c . Any statement

of the form “the task became automatic after n^* repetitions” is therefore a statement about the threshold θ chosen by the observer, since different choices move n^* over the entire positive half-line, and the stipulated curve itself contains no step. This bears on a particular family of hypotheses, namely those that posit a discontinuous and behaviourally observable switch from one mechanism to another, which would predict a gap in the cost or practice axis. It does not bear on all two-mechanism accounts. Two mechanisms whose probabilities of engagement change gradually, or which operate concurrently with changing relative contributions, also produce continuous performance curves, and a single mechanism can show abrupt transitions. Continuity therefore does not exclude two mechanisms. What it shows is that crossing a threshold is not evidence of a switch.

The mechanism behind the curve can be made explicit with a memoization model, in the spirit of the instance theory of automatization (Logan 1988), in which automaticity is the retrieval of stored episodes. Anderson (1982) supplies a complementary account, in which knowledge compilation, through proceduralization and composition, converts knowledge that is executed interpretively into productions that run directly, so that practice can change a response by compiling a procedure and not only by storing an instance. Neither account, taken alone, establishes that cognition uses one mechanism. The model has to be executable and not merely an envelope of attainable costs, because an agent cannot take the cheaper of two routes without first paying to find out which is cheaper. Let M be a store of solved instances. Let $S(t)$ be the cost of solving t from scratch, including the checking of the result. Retrieving the nearest stored case costs ℓ , and assessing whether that case applies to the present task costs α . If it applies, which happens with probability $\pi(t, M)$, the case is adapted at expected cost $\bar{\delta}(t, M)$, where adaptation cost is a nondecreasing function $\delta(d)$ of the distance d between the task and the stored case, with $\delta(0) = 0$, and δ is assumed continuous. If it does not apply, the agent falls back on solving from scratch after having paid for retrieval and assessment. The expected cost is

$$C(t \mid M) = \ell + \alpha + \pi(t, M) \bar{\delta}(t, M) + (1 - \pi(t, M)) S(t). \quad (2)$$

Equation (2) explicitly includes the overhead of unsuccessful reuse attempts, namely retrieval and assessment that end in fallback, and so it is not the minimum of the costs of the two routes.

Proposition 3 (Gradual improvement under growing coverage). *Fix $\ell, \alpha, \bar{\delta}$ and $S > \ell + \alpha + \bar{\delta}$, so that the stored route is cheaper than solving afresh, and let π_n be the probability that the nearest stored case applies at the n th encounter, with π_n nondecreasing in n and $\pi_n \rightarrow 1$. Then the expected cost at the n th encounter is nonincreasing in n and converges to $\ell + \alpha + \bar{\delta}$, which is the cost when a stored case always applies. The limit is forced by $\pi_n \rightarrow 1$, and the improvement is gradual exactly to the extent that π_n is, so that if π_n is continuous and strictly increasing in a continuous measure of experience, the expected cost is continuous and strictly decreasing in it.*

Proof. From Equation (2), the expected cost is $\ell + \alpha + \bar{\delta} + (1 - \pi_n)(S - \bar{\delta})$, which decreases as π_n increases, and $S > \bar{\delta}$ follows from the stated inequality. The limit follows from $\pi_n \rightarrow 1$. $\square$

Remark 1. In the special case where the first encounter pays S and every later encounter retrieves the stored result at cost ℓ , the cumulative mean cost per episode is $\ell + (S - \ell)/n$. This decreases smoothly even though the per-episode costs are $S, \ell, \ell, \dots$, so it describes average expenditure and not the current performance of the agent. Gradual improvement in current performance requires an assumption of the kind in Proposition 3, such as growing coverage, falling retrieval latency or rising applicability.

Item-level evidence bears on the coverage assumption. Rickard (2004) reports item-level tests of candidate models of the transition from algorithmic computation to memory retrieval in skill learning, and such tests treat learning history as acting on particular items and not on a task category as a whole. That supports treating the applicability probability as a property of items and cases, and it supports history-sensitive change in which strategy is used. It concerns which strategy is used on which item, and it does not by itself settle whether the strategies are separate architectural mechanisms.

Equation (2) describes one process with one store and two ways of producing an answer, in which the route taken is determined by how close the nearest stored case is and whether it applies. Where applicability remains reliable and adaptation cost varies continuously with distance, small differences from a stored case can incur small additional costs. Continuity of adaptation cost does not by itself ensure continuity of the applicability probability, and an assessment that fails can trigger an expensive fallback, so the cost of a task near a stored case can change abruptly where applicability does. A task far from every stored case costs approximately $\ell + \alpha + S(t)$.

The verification attributed to stored responses requires care. A stored answer was verified for the situation in which it was produced. Reuse is warranted only when the present instance preserves the conditions on which that verification depended, and a similar but different instance may fail them, which is what the assessment cost α pays for and what the fallback covers. Habitual responses can also be acquired without any verified deliberate solution, by imitation or by reinforcement of outcomes, and nothing here denies that. The claim is conditional. Where a stored response derives from an evaluated solution and the conditions of that evaluation are preserved, its present speed is no evidence that evaluation never occurred, and the idea that a habitual response was never checked cannot be inferred from its speed.

5 The Routing Argument

A two-systems theory needs an assignment of tasks to systems. Let $\rho : \mathcal{T} \rightarrow \{1, 2\}$ be the assignment and let $r(t)$ be the cost of computing it. Suppose the routed agent pays $r(t)$ and

then pays $C_{\rho(t)}(t)$ under the selected system, so that its total cost is

$$C_{\text{routed}}(t) = r(t) + C_{\rho(t)}(t). \quad (3)$$

Proposition 4 (Routing must be cheaper than the saving). *If the speed advantage of System 1 on a task t is to survive routing, then $r(t) < C_2(t) - C_1(t)$.*

Proof. The routed cost on a System 1 task is $r(t) + C_1(t)$, which beats the System 2 cost $C_2(t)$ only if $r(t) < C_2(t) - C_1(t)$. $\square$

The proposition permits cheap routing between distinct mechanisms, and it does not show that every cheap router is a memory lookup. The argument for parsimony is narrower. In the model of Equation (2), the operations that determine whether a stored response is used are retrieval and applicability assessment, whose cost $\ell + \alpha$ is paid on every encounter and which already settle whether the fast route succeeds. A routing policy could use a cheap cue, a task identifier, or a learned estimate of retrieval success without performing retrieval itself, so a separate dispatcher is not impossible. The model shows only that a dispatcher is unnecessary for its specified execution policy, in which the success of the fast route is settled by steps already paid for. A router added to it would need to be justified by evidence that its cue carries information the retrieval steps do not, and the speed evidence discussed here does not supply that. A serious alternative is the expected value of control account (Shenhav, Botvinick and Cohen 2013), in which control is allocated by weighing the expected benefit of a control signal against its effort cost, and the signals selected vary in intensity as well as in identity. This is a specified mechanism of control allocation and not an unexplained supervisor, so it is not open to the objection below, and it fits the position taken here to the extent that the allocation it describes is graded. It also shows the shape that a defensible two-mechanism account would take, namely a stated cost-benefit policy whose inputs and outputs can be tested, with the retrieval and assessment policy of Equation (2) as one competitor, in place of a categorical sorter justified by the speed of the outcome. This is also where the supernatural reading can enter, though only under one condition. Two mechanisms need not be coordinated by an agent-like supervisor, and a learned mechanical routing policy raises no difficulty of this kind. The objection applies when the sorter is taken to be a supervisor that surveys each thought, judges it easy or hard, and dispatches it, while its own intelligence is left unexplained, because the supervisor is then the figure the dichotomy was meant to explain. A retrieval and assessment step against a store of prior solutions requires no such supervisor, and it is itself a candidate routing policy, of the graded kind discussed above.

6 Why the Partition Looks Real

Three features of the evidence make a continuum look like two groups, and they induce three different kinds of error: the first is an error of selection, the second of salience, and the third of inference.

The first is the selection of examples. Studies of fast intuitive judgement use tasks that are easy for a typical adult, meaning tasks for which $n_h(t)$ is large in the sampled population, and studies of deliberate reasoning use tasks for which it is small. If the label “System 1” is then applied to the tasks with large n , the label predicts low cost by construction, and the finding that System 1 tasks are cheap is a restatement of how the tasks were chosen.

The second is the observer’s attention to the ends of the practice curve. A person remembers the first, halting attempt at a new appliance and the thousandth effortless one, and no episode in between is marked. Naming the two ends gives the impression of two machines, whereas the intermediate points exist and are unremarkable.

The third is that the shape of a cost distribution underdetermines the architecture that produced it. Suppose practice counts, treated as a continuous approximation to what are in fact discrete counts, are distributed with density f_N and cost is $Y = c(N)$ with c as in Equation (1). Since c is strictly decreasing and differentiable, the density of Y at $c(n)$ is

$$f_Y(c(n)) = \frac{f_N(n)}{|c'(n)|} = \frac{f_N(n) n^{\beta+1}}{b\beta}. \quad (4)$$

The factor $n^{\beta+1}$ is nonconstant, so it moves the locations of the density maxima and can change their number. A unimodal distribution of practice counts can therefore yield a bimodal distribution of costs, and a bimodal distribution of practice counts can yield a unimodal one. A mixture of two mechanisms can likewise appear unimodal. Neither the presence nor the absence of bimodality in cost, taken by itself, identifies the number of mechanisms, and it follows that a bimodal histogram of response times does not, by itself, identify two systems. This conclusion serves the broader argument better than a claim that a single process cannot produce bimodality, since it shows that distributional appearance underdetermines architecture in both directions.

7 An Illustration from a Cached Decision Architecture

A concrete case in which the split between reusable and request-specific computation is explicit is a recent family of “typed decision” models, which choose among a set of options, judge a condition, or assign a score, and have been described by their authors with the language of System 1, a description this essay does not adopt (Li and Wang 2026). The architecture is used here only as an instance of the amortization in Equation (2), and no claim is made

about its performance or about the authors’ evaluations. In it, each of K options is encoded independently of the request into a vector together with per-layer key and value projections, and these entries are stored. A request is encoded fresh, and a small decision stage refines the request state by attending over the stored entries. The reported property relevant here is that substituting cached entries for recomputed ones leaves every output unchanged in exact arithmetic, because an entry depends only on the option.

Let c_{opt} be the cost of encoding and projecting one option, and C_{req} the cost of encoding the request and running the decision stage over prepared entries. If a fraction $\rho \in [0, 1]$ of the K options is already in the store,

$$C(\rho) = C_{\text{req}} + (1 - \rho) K c_{\text{opt}}. \quad (5)$$

At $\rho = 0$ every option is computed afresh, and at $\rho = 1$ only the request-specific work remains. The two ends correspond to what would be called slow and fast operation, and they are produced by the same weights, the same procedure and the same decisions, differing only in how much of the work has been done in advance. The cost is linear in ρ , so there is no value of ρ at which the model changes into another kind of model. Unlike the memory of Equation (2), retrieval here is exact, since an option is either in the store or it is not, and the applicability step is trivial because an entry depends only on the option. The decision stage is also run on every request even at full reuse, which is the architectural counterpart of the observation that a habitual response is still a computation on the present situation, with some of its inputs precomputed.

The illustration separates two things that the label runs together. The low latency at high reuse comes from the cache, and it is a cost property. The accuracy of the model on its decision tasks comes from training on a very large number of decisions of that type, which is practice, and the architecture exposes the amortized part as an explicit memory. The sensible name for such a model describes what it does, namely typed-output decision with cached option representations. The name of a supposed cognitive system adds a commitment, that the category exists independently of the training and the caching that made the tasks cheap, which the construction does not support.

8 Practice That Teaches a Procedure

The cached architecture shows practice acting through stored components. A second case shows practice acting differently, and it matters because it blocks the inference that all learned speed is retrieval. Progress-aligned recurrence (Wang et al. 2026) takes a pretrained language model, repeats a block of its layers a variable number of times, and trains it on sequential state-tracking tasks, such as following a series of swaps among five boxes and answering

which item ends in a queried box. Its central training choice is that iteration k is trained to predict the state after the first k operations, so that each iteration has a defined job, and its second is to rescale the recurrent state to a fixed norm after every iteration, which controls the magnitude of the state so that a drifting scale does not change what the repeated transformation computes. A learned halting signal lets the model decide when to answer, and a variant trained on prefixes of problems obtains the intermediate targets from final answers alone.

The reported results are specific. Trained on at most 32 moves, a small model achieves 95.5% full-state accuracy at 512 moves, sixteen times the training length, against 55.7% for the best prior recurrent method and 0.8% for an explicit chain of thought on the same backbone. Removing the norm anchoring reduces the extrapolation factor from 16 to below 2.2, and replacing the progress-aligned targets with final-answer supervision reduces it to at most 8. A model with four times the core depth, and a recurrent model with additional iterations but without the alignment, each extrapolate less than twofold. These are the authors' reports on synthetic tasks and mostly on one small backbone, and the paper itself notes that results on larger backbones vary far more across seeds, that accuracy does not rise monotonically with model size, and that on graph reachability every recipe stays below 3.5-fold extrapolation, which the authors attribute to the absence of a canonical intermediate state at each iteration, an explanation this essay does not rely on.

Four consequences bear on the argument. First, practice here does not act by retrieving complete solved instances, which is a distinction of the kind that Anderson (1982) draws between storing and compiling. It supports reusable sequential computation that can be applied more times than it was applied in training, so an account of practice limited to amortized answers is too narrow and should be broadened to the acquisition of computational competence. The result does not show that the latent state alone implements a sufficient transition state, since the full prompt remains available at every iteration. Second, the amount of computation applied is graded, being the number of iterations, an integer-valued and variable depth, and a longer problem simply receives more of it. Third, the availability of extra computation is not what makes it useful. A model that can iterate without the learned organization gains little from more iterations, so the depth of computation depends on training history and not on the architecture's capacity to loop. Fourth, the extended computation runs without any generated intermediate text, and the explicit chain of thought on the same backbone collapses on longer problems, so externalization is neither what makes extended computation possible nor what distinguishes it.

The paper also bears on the claim that a correct answer, once reached, simply persists. The paper's own models, trained with the number of iterations matched to problem length, answer correctly at the matching iteration and degrade rapidly if run beyond it, which is why a halting rule is needed, and the learned stopping rule that works best, which requires six of eight consecutive iterations to confirm completion, reaches 92.9% at sixteen times the training

length against 98.7% at eight times, with wide variation across seeds. Obtaining an adequate answer and preserving it through further computation are therefore different achievements, and a theory of cost that identifies the first with the second will misdescribe the second.

9 What Remains of the Distinction

The partition claim is not supported by the evidence usually offered for it, and the observations that motivated it remain. Some responses are produced more cheaply than others, some are accompanied by report and some are not, and some can be interrupted and revised while others run to completion. These are real differences, and they are best treated as distinguishable dimensions that fall into three families. The first family concerns history. One parameter is the number of relevant prior episodes, $n_h(t)$. A second is the distance $d(t, m)$ from the present task to the nearest stored case, together with whether the stored result carries a warrant that applies. A third is whether the practiced competence takes the form of stored instances or of a reusable procedure (Logan 1988; Anderson 1982). The second family concerns execution, and its parameter is the depth of computation actually executed on the present task. The third family concerns access to what was computed, and it comprises three properties that are often run together, although they are conceptually distinguishable and need not be coupled. Externalization is whether intermediate states are written into a context where they can be reread. Decodability is whether intermediate states can be recovered from the system by a readout of a stated class, which in the recurrent case holds because the training objective makes it so. Reporting is whether the agent produces an account of its process, and an account need not correspond to what was computed. The recurrent model above is deep, decodable by its trained readout and unexternalized, and a person's practiced arithmetic may be shallow and readily reported after the fact. These dimensions permit combinations that a single fast-slow axis obscures.

On this view the central thesis is that cost, automaticity and reportability do not by themselves establish cognitive kinds, and that explaining their variation requires an account of learning history, representation, execution and reuse. A practiced response can be a compiled form of an earlier deliberate one, and the cached architecture and the amortization model show how. Practice can also install a procedure that is then run for longer, and the recurrent model shows that too. What has not been established here is the stronger claim that every form of deliberation is retrieval with less work precomputed, and the essay does not rely on it. The vocabulary of two systems should be retained, if at all, as shorthand for the ends of the practice axis, with the understanding that it names a region of a continuum, implies no mechanism of sorting, and cannot be read off from speed.

References

- [1] Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review*, 89(4), 369–406.
- [2] Evans, J. St. B. T., and Stanovich, K. E. (2013). Dual-process theories of higher cognition: Advancing the debate. *Perspectives on Psychological Science*, 8(3), 223–241.
- [3] Heathcote, A., Brown, S., and Mewhort, D. J. K. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, 7(2), 185–207.
- [4] Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus and Giroux.
- [5] Keren, G., and Schul, Y. (2009). Two is not always better than one: A critical evaluation of two-system theories. *Perspectives on Psychological Science*, 4(6), 533–550.
- [6] Li, Q., and Wang, X. (2026). Jeva: System one models from first principles. arXiv:2610.18773.
- [7] Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, 95(4), 492–527.
- [8] Melnikoff, D. E., and Bargh, J. A. (2018). The mythical number two. *Trends in Cognitive Sciences*, 22(4), 280–293.
- [9] Newell, A., and Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive Skills and Their Acquisition*, 1–55. Erlbaum.
- [10] Rickard, T. C. (2004). Strategy execution in cognitive skill learning: An item-level test of candidate models. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(1), 65–82.
- [11] Shenhav, A., Botvinick, M. M., and Cohen, J. D. (2013). The expected value of control: An integrative theory of anterior cingulate cortex function. *Neuron*, 79(2), 217–240.
- [12] Wang, P., Hu, Y., Dong, G., Lin, Y., Wang, H., Liu, Z., Wen, Z., Li, J., Yang, J.-M., Wu, W., and Tong, Y. (2026). PAR: Progress-aligned recurrence for length generalization in state tracking. arXiv:2610.27628.