

Modeling Attention Decay in Single-Session Cognitive Work: A Derivation of the Exponential Form With Simulated-Data Validation

¹Kavish Tolani

¹Cognitive Science

¹Indus Universal School

Abstract—Productivity tools typically track elapsed time and treat it as a stand-in for how much a person can still give a task. Sustained-attention research suggests this is the wrong unit: capacity itself falls over the course of a session, and not at a constant rate. Focusware is a productivity app built around modeling attention directly rather than timing it, which raises a concrete question: what shape should that decay curve take, and can the choice be justified rather than assumed? I derive the answer from one assumption — that the rate of attention loss is proportional to attention currently remaining — and show it forces an exponential solution over the naive linear alternative. From there I derive a half-life, a bounded productivity score, an optimal stopping time, and a multi-session model of output under repeated, imperfect breaks. I then fit both the exponential and linear forms to a simulated dataset built to resemble Focusware's self-reported session checkpoints, at the single-session level and across 60 simulated users. The exponential model fits better in both cases (mean R^2 0.974 vs. 0.934, better fit in 93.3% of simulated users), consistent with what the derivation predicts. Every formula in the paper maps to a function already running in Focusware, and I close by stating what result on real user data would overturn the model rather than confirm it.

Index Terms—attention decay, exponential model, vigilance decrement, break scheduling, productivity modeling, ordinary differential equations, simulated validation

I. Introduction

1.1 Background and Problem

Most productivity software measures a session in minutes: a Pomodoro is 25, a calendar block is an hour. Focusware, the app this paper is built around, treats that as the wrong unit. Two people can both “work” for 45 minutes and leave with very different amounts of usable output, because what is actually being spent during focused work is attention, not time, and attention does not run out at a constant rate. Focusware tracks focus sessions, schedules breaks, and scores daily output by modeling attention directly: a user rates focus at the start of a session, the app schedules a break from a personal decay estimate, the user rates focus again on return, and the app updates both the session score and its estimate of how well that break actually helped.

The distinction between time-tracking and attention-tracking is not merely semantic. A person who works for 40 minutes at declining focus and a person who works for 40 minutes at sustained focus have identical logs under a time-based system, yet the two sessions plainly do not represent the same amount of usable cognitive work. Any system that schedules breaks, scores output, or projects a user's daily capacity from elapsed time alone is implicitly assuming that attention is constant across the session — an assumption that is convenient to encode in software but that has no particular support behind it. Building a break scheduler

on top of that assumption means the scheduler inherits the assumption's weaknesses without anyone having explicitly chosen them.

Every one of Focusware's features depends on choosing a shape for $A(t)$, attention over time. A break scheduler cannot pick a sensible pause time, and a productivity score cannot be computed at all, without first fixing that shape. The question this paper works through is whether that shape has to be assumed, or whether it can be derived from something simpler and checkable about how attentional fatigue actually works. Framed this way, the problem is less “which curve looks right” and more “what property must the underlying process have for a particular curve to be the only mathematically consistent outcome.” That reframing is what separates a modeling choice from a modeling result, and it is the throughline of Sections 3 and 4.

1.2 Research Gap

Two literatures speak to this question without settling it, and it is worth being precise about where each one stops short. Classical work on memory retention (Section 2.1) treats exponential decay as the historical default, but more recent comparative work complicates that picture at longer time horizons, leaving open exactly which functional family best describes forgetting once the window extends beyond a single sitting. Sustained-attention research (Section 2.2) is unambiguous that performance falls with time-on-task — this is one of the more robustly replicated findings in the vigilance literature — but it does not commit to one functional form for that decline, and current theoretical accounts instead point to several contributing mechanisms operating together (resource depletion, motivational disengagement, and mind-wandering, among others) without adjudicating between them mathematically.

Neither line of research does what this paper attempts: derive a decay curve from an explicit, checkable mechanistic assumption, and then carry that curve all the way through to the concrete pieces of a working system — break timing, a bounded output score, and output across repeated sessions with imperfect recovery — while also checking the derived curve against a stated, structurally different alternative on data. The memory-decay literature works over days and does not link its findings to product-level break scheduling. The vigilance-decrement literature identifies the qualitative phenomenon but stops before offering a governing equation. The closest technical precedent, half-life regression (Section 2.2), applies the same mathematics to a different problem — memory retention across days for spaced repetition, not attentional fatigue within a single session. That combination of gaps, taken together rather than individually, is what this paper is written to close.

1.3 Research Questions

Three questions organize the paper, moving from the mechanism itself to its consequences for actual system behavior.

RQ1 — Does assuming that attention loss is proportional to attention remaining force a unique functional form, ruling out the simpler constant-loss-rate alternative? This is a question about mathematical necessity, not preference: if the proportionality assumption is granted, is exponential decay the only solution, or merely one of several plausible ones?

RQ2 — Does that derived form fit self-reported session data better than the linear alternative, at the level of one session and across a simulated population? A clean derivation is not by itself evidence that real (or

realistically simulated) attention behaves the way the derivation predicts, so this question exists to keep the theoretical and empirical claims separate and both accountable.

RQ3 — Once the functional form is fixed, which of Focusware's product quantities — break timing, session ceiling, multi-session output — follow as forced mathematical consequences rather than separate design choices that could in principle have been tuned independently? This question is what turns a decay-curve derivation into a system design argument rather than a purely mathematical exercise.

1.4 Contributions

- A derivation showing the exponential model is the unique consequence of a proportional-loss assumption, with the linear alternative ruled out on independent structural grounds (Sections 3.3–3.4) — this establishes RQ1 as a mathematical fact rather than a modeling preference.
- Closed-form expressions for half-life, bounded session output, optimal stopping time, and multi-session output under imperfect recovery (Sections 3.5–3.6), each shown next to the code that implements it, so the connection between the mathematics and the running system is traceable rather than asserted.
- A fit-quality comparison between the two models on simulated session data, at both the single-session and population level, with an explicit account of what result would overturn the conclusion (Sections 4, 5.4) — this addresses RQ2 and is deliberately structured so it could, in principle, have come out the other way.
- A short analytical sensitivity check on the multi-session ceiling as a function of break quality, computed directly from the closed-form model rather than from simulated noise (Section 4.4), which shows how RQ3's forced consequences behave quantitatively, not just qualitatively.

II. Literature Review

2.1 Relevant Previous Research

Ebbinghaus's 1885 forgetting curve modeled memory retention as $R = e^{(-t/S)}$, R being the retained fraction and S a strength parameter that plays the same role as $1/k$ in this paper. The claim behind it — forgetting is fastest right after learning and slows as time passes — is the same structural claim this paper makes about attention. What makes Ebbinghaus's result useful here is not just that it settled on an exponential curve, but that the underlying logic is portable: a quantity that erodes fastest while it is largest, and more slowly as it is depleted, is a pattern that shows up whenever loss is proportional to the amount remaining, whether that quantity is memory strength, radioactive mass, or (as this paper argues) attention.

That portability has limits, and it is important not to overstate them away. Wixted and Ebbesen later argued that a power-law form often fits long-horizon retention better than a pure exponential, and Murre and Dros's replication broadly confirmed Ebbinghaus's qualitative shape while leaving the exact functional family an open question at long time scales. The disagreement in that literature is specifically about retention measured over days to weeks, where a power-law's heavier tail captures the slow, long-run persistence of some memories in a way a pure exponential cannot. That regime is not the one this paper operates in: Focusware's sessions run tens of minutes, not days, and at that shorter horizon the exponential-versus-power-law distinction is much less pronounced, since the two families only diverge meaningfully once enough time has passed for their tails to separate. The memory literature is therefore better read here as a precedent for the general logic of proportional decay than as direct evidence for the exponential form specifically — a distinction Section 5.2 returns to.

Sustained-attention research is more directly relevant to the claim this paper is making, because it concerns exactly the timescale and exactly the phenomenon — attentional fatigue within a single continuous task — that Focusware needs to model. See et al.'s meta-analysis of the vigilance decrement found that accuracy on continuous-attention tasks falls with time-on-task across a wide range of task types and testing conditions, with the decline measurable within the first ten minutes of a task and continuing over the full duration studied. This is a strong empirical anchor: it is not one study's finding but an aggregate pattern across many, which is a more durable kind of evidence than any single experiment could provide. What the meta-analysis does not do, and does not attempt to do, is specify a functional form for the decline — it establishes that performance falls monotonically with time-on-task without committing to whether that fall is linear, exponential, or something else.

Thomson et al.'s resource-control account and Esterman and Rothlein's review both attribute the vigilance decrement to some combination of resource depletion, motivational disengagement, and mind-wandering, and both stop short of proposing a single governing equation, in part because the three proposed mechanisms plausibly operate on different timescales and might not combine into one clean functional form even if each were individually well characterized. Both reviews also note substantial variation in decay rate between individuals — people do not fatigue at the same rate on the same task — which is the empirical basis for treating k as something estimated per user in Section 3.7 rather than fixed as a population constant. A model that assumed one k for everyone would be contradicted by exactly this part of the literature, independent of whatever functional form it assumed.

2.2 Existing Mathematical Models

The closest methodological precedent is Settles and Meeder's half-life regression, used to schedule spaced repetition in a language-learning application. It shares this paper's core mathematical structure: a proportional-loss assumption producing an exponential curve with a well-defined half-life, and a system built to schedule interventions (in their case, review prompts; in Focusware's case, breaks) at a time computed from that half-life. The parallel is close enough to be worth dwelling on, because it suggests the underlying mathematical pattern — proportional decay, half-life, scheduled intervention at a fraction of the half-life — is not specific to memory or to attention but is a more general template for any system that needs to time an intervention against a decaying quantity.

The differences matter as much as the similarity. Settles and Meeder estimate a per-item decay rate for memory retention measured across days between review sessions; this paper estimates a per-user decay rate for attentional fatigue measured across minutes within one continuous session. Those are different quantities decaying for different reasons at different timescales, and nothing in half-life regression's validation on spaced-repetition data transfers automatically to attention within a session — the mathematical form is borrowed, not the empirical support. No model in either the memory or vigilance literature is derived from a stated mechanistic assumption and then mapped onto break scheduling and multi-session output the way Sections 3.4–3.6 do here; half-life regression comes closest structurally but was never applied to, or validated against, within-session attentional data.

2.3 Research Gap

Put together, the three bodies of work bound the problem without solving it. The memory-decay literature gives a precedent for proportional decay and a worked example of when an exponential form breaks down

(at long horizons), but says nothing about within-session attention. The vigilance-decrement literature confirms that decline with time-on-task is real and individually variable, but stops before proposing a governing equation. Half-life regression shows the exponential-decay-plus-scheduled-intervention pattern successfully applied to a related but distinct problem. None of the three derives a within-session attention-decay curve from a mechanism, carries it through to a bounded productivity score and a multi-session recovery model, and checks the result against a stated structural alternative on data resembling what a real productivity application collects. Sections 3 and 4 do exactly that, and Section 5.2 revisits each of these three bodies of work in light of the results.

III. Methods and Model

3.1 Research Design

The paper has two parts, and keeping them separate is a deliberate methodological choice rather than an organizational convenience. Sections 3.3–3.6 derive the governing equation and its consequences from a stated assumption; this is a mathematical argument, and its validity does not depend on any data at all. Section 4 then fits both the derived model and the rejected alternative to session-checkpoint data and compares fit quality; this is an empirical argument, and its validity depends entirely on the data. A derivation is a claim about mechanism — if attention loss really is proportional to attention remaining, the exponential form necessarily follows — but it is not, by itself, evidence that real attention actually behaves that way. The two kinds of claim answer different questions and can fail independently: the derivation could be mathematically airtight while the mechanistic assumption behind it is wrong, or the assumption could be right while some other factor (measurement noise, individual variation, non-stationarity within a session) obscures the fit. Reporting both, and being explicit about which is which, is what keeps the paper honest about what has actually been shown versus what has merely been argued.

Focusware does not yet have a large enough logged user base for a population-level comparison, so Section 4's results come from a simulated dataset built to resemble the shape of Focusware's real data: 1–10 self-reported focus ratings at 5-minute checkpoints across a 60-minute session, with parameters drawn from a plausible range and Gaussian rating noise layered on top. This is a stand-in for validation, not validation itself, and every result drawn from it is labeled SIMULATED throughout the paper rather than presented alongside the analytical results without qualification. The purpose of the simulation is narrower than a full empirical test: it checks that the fitting procedure behaves as expected, and it produces a concrete, quantitative prediction — the exponential model should out-fit the linear model by a specific margin under specific noise assumptions — that a future analysis on real logs can either confirm or contradict. The fitting code (Appendix A) takes checkpoint data in a fixed format, so a real CSV export of session logs can be substituted in without changing the analysis, which was a design constraint on the code from the outset rather than an afterthought.

3.2 Variables and Assumptions

Notation used throughout: $A(t)$ is attention at t minutes into a session; A_0 is initial attention, $A_0 > 0$; t is elapsed minutes, $t \geq 0$; k is a decay-rate constant, $k > 0$. Two constraints apply to any candidate model: attention cannot go negative, $A(t) \geq 0$ for all $t \geq 0$, and attention must fall during uninterrupted work, $dA/dt < 0$.

These two constraints are deliberately minimal. They rule out models that predict attention rising during work with no intervening break (which would contradict the entire premise of a vigilance decrement) and models that predict attention going negative (which has no interpretable meaning on a bounded 1–10 self-report scale). What they do not do is specify a functional form — both the linear and exponential candidates in Sections 3.3 and 3.4 satisfy $dA/dt < 0$ over some range of t , and only the exponential model satisfies $A(t) \geq 0$ for all $t \geq 0$ without qualification. That asymmetry is the first piece of evidence against the linear model, and it emerges directly from the two constraints stated here rather than from any property specific to the exponential form.

3.3 Linear Model

Assume the rate of loss is constant:

$$dA/dt = -k \quad (1)$$

Integrating and applying $A(0) = A_0$ gives

$$A(t) = A_0 - k \cdot t, \quad A(t) = 0 \text{ at } t^* = A_0/k \quad (2)$$

Equation (1) says the same absolute amount is lost each minute whether attention is at 90% or 10% of capacity — fatigue with no dependence on current state. This is a strong and, on reflection, somewhat implausible claim about the underlying process: it says that a person at the tail end of an exhausting session loses focus at exactly the same absolute rate as a person five minutes into a fresh one. Two problems follow directly from Equation (1) once it is integrated. First, Equation (2) goes negative past t^* , which is not physically meaningful on a bounded attention scale — the model has no mechanism to stop losing attention once none is left, because the loss rate does not know how much is left. Second, there is no natural break-time formula under this model, since the optimal break time does not depend on k at all: because attention falls at a fixed absolute rate, any personalization of break timing based on how quickly a specific user fatigues is mathematically impossible under Equation (1), which defeats one of Focusware's core design goals before the model is even fit to data.

3.4 Exponential Model

Change one thing: the loss rate scales with attention currently remaining rather than staying fixed.

$$dA/dt = -k \cdot A \quad (3)$$

This single change has a mechanistic interpretation worth stating explicitly, because the assumption is doing all of the work in what follows. A proportional-loss process is one where whatever depletes attention — accumulating interference from sustained cognitive load, declining arousal, rising competition from mind-wandering, or some combination the vigilance literature has not fully disentangled (Section 2.1) — acts on the attention that remains, not on some fixed baseline. This is consistent with, though not proof of, resource-depletion accounts of the vigilance decrement: if attentional capacity behaves like a finite resource being consumed at a rate tied to how much is left, proportional loss is the natural mathematical description of that process, in the same way that proportional loss describes radioactive decay, drug elimination, and Ebbinghaus's forgetting curve (Section 2.1). The paper does not claim to have identified the specific physiological or cognitive mechanism responsible — that would require evidence this derivation does not provide — only that proportional loss is a coherent, minimal assumption consistent with the broad shape of existing findings.

Separating variables, $\int (1/A) dA = \int (-k) dt$ gives

$$\ln|A| = -k \cdot t + C \quad (4)$$

and applying $A(0) = A_0$ gives the unique solution

$$A(t) = A_0 \cdot e^{(-kt)} \quad (5)$$

Code 1. Direct implementation of Equation (5). A_0 is initial attention, k is the personal decay constant, t is minutes elapsed.

```
def attention(t, A0, k):
    return A0 * math.exp(-k * t)
```

Equation (5) is not assumed — it is the only function consistent with Equation (3), which is a stronger claim than simply noting that exponential decay is a common pattern in nature. Dividing Equation (3) by A shows the fractional loss per minute is constant at k , so a k of 0.02 means 2% of whatever attention remains is lost each minute, regardless of how much is left. This constant-percentage property is what distinguishes the exponential model from the linear one in a way that has a direct behavioral interpretation: the exponential model predicts that a highly focused user and a nearly exhausted user both lose the same proportion, not the same amount, of their remaining attention over the next minute, which is consistent with a fatiguing process that scales with current cognitive load rather than one that runs down at a fixed absolute rate regardless of how depleted the underlying resource already is. Solving $A_0/2 = A_0 \cdot e^{(-k \cdot T_{1/2})}$ for the half-life gives

$$T_{1/2} = \ln(2)/k \approx 0.693/k \quad (6)$$

Code 2. Break scheduler implementing Equation (6). Focusware fires the first break at 80% of the half-life, ahead of the point where retained capacity drops below half.

```
def first_break_time(k):
    T_half = math.log(2) / k
    return 0.8 * T_half
```

Every $T_{1/2}$ minutes attention halves again, so it never reaches exactly zero — the practical implication is that a break scheduled any later than $T_{1/2}$ is already past the point of steep diminishing returns. The choice to schedule the first break at 80% of $T_{1/2}$ rather than at $T_{1/2}$ itself reflects a design judgment, not a mathematical necessity: scheduling earlier trades a small amount of theoretically available output for a margin of safety against the possibility that a user's true k is somewhat higher than estimated, since underestimating decay rate is more costly to output (a session run too long past its productive window) than overestimating it (a break taken slightly early). Section 6 returns to this asymmetry when discussing the limitations of treating k as a fixed personal constant.

3.5 Productivity Model

Total useful attention across a session of length T is the integral of Equation (5):

$$P(T) = (A_0/k) \cdot (1 - e^{(-kT)}) \quad (7)$$

Code 3. Session productivity score implementing Equation (7), used by Focusware to score a session in place of raw duration.

```
def session_score(A0, k, T):
    return (A0 / k) * (1 - math.exp(-k * T))
```

As T grows, Equation (7) approaches a ceiling,

$$P_{max} = A_0/k \quad (8)$$

a hard limit that falls out of Equation (5) rather than being set by hand — the linear model in Section 3.3 has no equivalent bound, since Equation (2) integrates to an unbounded quadratic in T over any interval where

$A(t)$ remains positive. The existence of a finite ceiling is not a minor mathematical footnote; it is a direct, testable prediction that a session cannot be made arbitrarily productive simply by extending it, no matter how long a user is willing to sit at their desk. A user with $A_0 = 8$ and $k = 0.02$ has a theoretical output ceiling of 400 attention-units regardless of session length, and the productivity score in Code 3 will asymptote toward that number rather than growing without bound — a property that gives the score a natural scale against which any single session's output can be judged, addressed further in Section 3.7 via the `is_exhausted()` function in Appendix A. Ending a session once $A(T)$ drops to a minimum useful level $A_{\min}$, and solving for T , gives the optimal stopping time

$$T^* = (1/k) \cdot \ln(A_0/A_{\min}) \quad (9)$$

Equation (9) formalizes a stopping rule that experienced knowledge workers often follow intuitively — recognizing when continuing to work has stopped being worth the diminishing attention available — and gives it a numerical answer conditioned on a user's own A_0 and k rather than a fixed rule of thumb applied uniformly.

3.6 Break / Recovery Model

Let $r \in (0, 1]$ be the fraction of A_0 a break restores, so session n opens at

$$A_n(0) = r^n \cdot A_0 \quad (10)$$

a geometric sequence across sessions. This formulation carries its own assumption worth stating plainly: it treats break quality r as constant across a day, so the third break of the day is assumed to restore the same fraction of A_0 as the first. That assumption is almost certainly an idealization — recovery plausibly depends on how many prior sessions have already been completed, time of day, and cumulative fatigue that a single-parameter r cannot capture — and Section 6 identifies it as a specific limitation rather than treating the geometric model as a complete description of daily recovery dynamics. It is retained here because it is the simplest assumption consistent with the basic empirical fact that breaks only partially restore capacity ($r < 1$ in general) and because it produces a closed-form result, Equation (12) below, that a more elaborate time-varying r would not.

Productivity of session n is $r^n \cdot (A_0/k) \cdot (1 - e^{-kT})$, and summing over N sessions with the geometric series formula gives total output

$$P_{\text{total}} = (A_0/k) \cdot (1 - e^{-kT}) \cdot (1 - r^N) / (1 - r) \quad (11)$$

Code 4. Multi-session output implementing Equation (11) and its $N \rightarrow \infty$ limit, Equation (12).

```
def daily_score(A0, k, T, r, N):
    per_session = (A0 / k) * (1 - math.exp(-k * T))
    geo = N if r == 1.0 else (1 - r**N) / (1 - r)
    return per_session * geo
```

As $N \rightarrow \infty$, since $r < 1$, $r^N \rightarrow 0$, giving a second ceiling

$$P_{\text{ceil}} = (A_0/k) \cdot (1 - e^{-kT}) / (1 - r) \quad (12)$$

Section 4.4 evaluates Equation (12) directly across a range of r to show how sharply this ceiling responds to break quality — the relationship is not linear, and the practical consequence for a user or for Focusware's design is substantial: because the ceiling scales with $1/(1-r)$, small, achievable improvements in break quality near the low end of the scale produce modest gains, while the same absolute improvement near $r = 0.9$ or $r = 0.95$ produces a much larger proportional gain in long-run output. This asymmetry, which is not

obvious from Equation (11) alone, is exactly the kind of consequence a closed-form derivation surfaces that an empirical-only approach might miss without deliberately testing for it.

3.7 Parameter Estimation

Focusware logs A_{start} and A_{end} (1–10 ratings) each session. Equation (5) rearranges to give a per-session decay estimate,

$$k = -\ln(A_{\text{end}}/A_{\text{start}}) / T \quad (13)$$

averaged across a user's history for a stable personal k . After a break, a fresh rating over the user's average baseline gives the recovery estimate,

$$r = A_{\text{after}} / A_{\text{baseline}} \quad (14)$$

Code 5. Estimators implementing Equations (13) and (14), used to personalize k and r for each Focusware user.

```
def estimate_k(A_start, A_end, T):
    return -math.log(A_end / A_start) / T

def estimate_r(A_after, A_baseline):
    return min(1.0, A_after / A_baseline)
```

Both estimators are simple algebraic inversions of Equations (5) and (14) rather than statistical models with their own assumptions about noise distribution, which keeps them transparent but also means they inherit no built-in protection against outliers: a single session with an unusually low A_{end} — whether from genuine fatigue, a distraction unrelated to attentional decay, or simple rating carelessness — will pull the per-session k estimate up sharply, since the estimator divides by T and takes a logarithm of a ratio that can swing widely on small samples. The `user_k()` function in Appendix A averages across a user's session history specifically to dampen this sensitivity, on the reasoning that idiosyncratic single-session noise should wash out once enough sessions are averaged, though the paper does not have real usage data (Section 3.1) to establish how many sessions are actually needed before the averaged estimate stabilizes to within a useful tolerance — a question flagged again as future work in Section 7.

3.8 Simulation Design

One 60-minute session was simulated with true parameters $A_0 = 9.4$ and $k = 0.021$, sampled at 5-minute checkpoints (13 points) with Gaussian rating noise, $\sigma = 0.35$ rating points, added to mimic imprecise self-report. The specific values are not arbitrary: $A_0 = 9.4$ and $k = 0.021$ were chosen to sit comfortably inside the plausible range for a focused, motivated work session (a high but not maximal starting rating, with a half-life of roughly 33 minutes under Equation (6), which is consistent with the general order of magnitude reported in the vigilance-decrement literature for when performance decrements become noticeable), and $\sigma = 0.35$ was chosen to reflect the kind of imprecision expected from a coarse 1–10 self-report scale, where a person is unlikely to reliably distinguish, say, a 6 from a 6.3 but should be able to reliably distinguish a 6 from an 8.

The same generative process, with an independently drawn $A_0 \in [7.5, 10.0]$ and $k \in [0.010, 0.035]$ per user and independent noise, was repeated across 60 simulated users to check the single-session result was not an artifact of one draw. The ranges for A_0 and k were chosen to be wide enough to represent meaningfully different users — a k of 0.010 implies a half-life over an hour, while a k of 0.035 implies a

half-life under 20 minutes, a more than three-fold difference in how quickly two otherwise similar users are assumed to fatigue — without extending into ranges implausible for a focused work session (a k so large that a user's attention collapses within a few minutes would not represent a session Focusware's break scheduler is meant to handle). Both the exponential form (Equation 5) and the linear form (Equation 2) were fit to every simulated session by nonlinear least squares.

3.9 Statistical Evaluation

Fit quality is reported as the coefficient of determination, $R^2 = 1 - (SS_{\text{res}}/SS_{\text{tot}})$, computed identically for both models against the same checkpoints. R^2 was chosen over alternatives such as AIC or BIC because both candidate models here have the same number of free parameters (A_0 and k), so a penalty for model complexity would not change the ranking between them — R^2 already isolates exactly the comparison this paper needs, which is how much of the variance in the checkpoint data each functional form explains, without the additional complication of comparing models with different numbers of parameters. R^2 is a fit-quality measure, not a significance test — the comparison in Section 4 shows which curve tracks the simulated data more closely, not that the difference is statistically significant in a formal hypothesis-testing sense. No paired significance test (such as a paired t -test on per-user R^2 values) is reported in this paper, and Section 6 flags this explicitly as a limitation of the current analysis rather than a settled point.

IV. Results

4.1 Analytical Results

Figure 1 plots Equations (2) and (5) side by side with no noise added, using illustrative parameters $A_0 = 10$, $k = 0.025$, to show the shape difference the derivation predicts before any data enters the picture: the exponential curve decelerates, the linear curve does not, and the linear curve's zero-crossing at $t^* = A_0/k$ falls well outside any realistic session length.

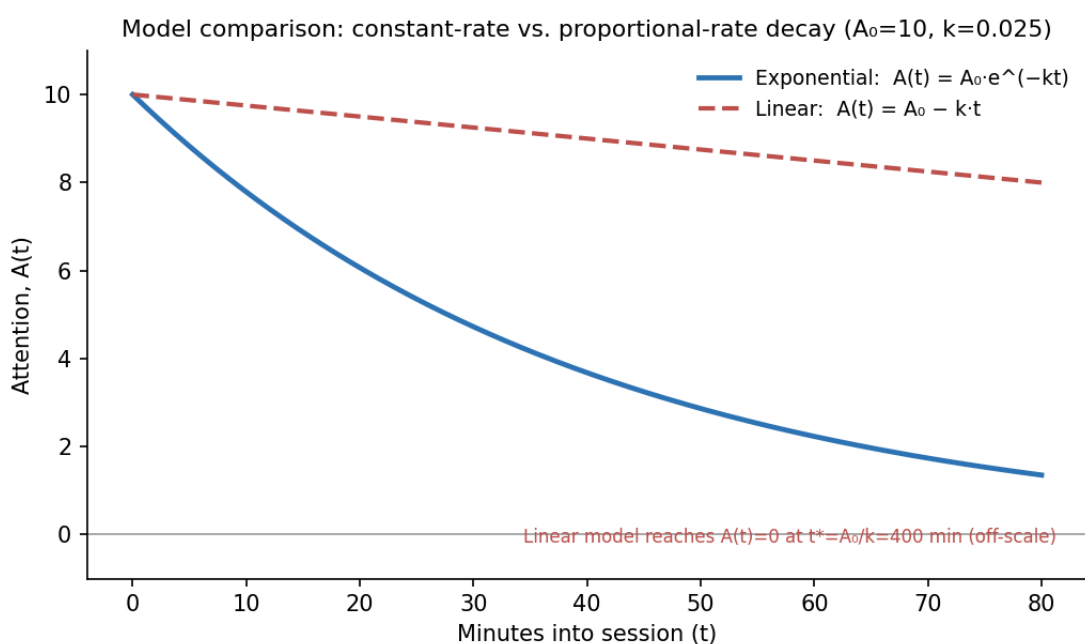


Figure 1. Linear (Equation 2) vs. exponential (Equation 5) attention decay, plotted analytically with no simulated noise. $A_0 = 10$, $k = 0.025$.

The most important visible feature of Figure 1 is not simply that the two curves differ, but where they differ. Over the first 15–20 minutes the two curves track each other closely, which is a direct consequence of the exponential function's own structure: for small $k \cdot t$, $e^{(-kt)}$ is well approximated by its first-order Taylor expansion, $1 - k \cdot t$, which is exactly the linear model. Early in any session, in other words, the two models are nearly indistinguishable by construction, not because the underlying data happens to be ambiguous. The divergence becomes visible only once $k \cdot t$ is no longer small — past roughly the 20–30 minute mark in Figure 1's parameterization — which has a direct methodological implication taken up again in Section 4.3: any dataset limited to short sessions or coarse early checkpoints will structurally struggle to distinguish the two models, independent of how much noise is present. This is not a weakness specific to the simulated data used here; it is a property of the two functional forms themselves.

Table 1. Closed-form quantities derived from the exponential model.

Quantity	Formula	Interpretation
Half-life	$T_{1/2} = \ln(2)/k$ (Eq. 6)	Time until attention halves; sets the break trigger.
Session ceiling	$P_{\max} = A_0/k$ (Eq. 8)	Maximum output from one uninterrupted session.
Optimal stopping time	$T^* = (1/k) \cdot \ln(A_0/A_{\min})$ (Eq. 9)	Session length at which attention hits a minimum useful level.
Multi-session ceiling	$P_{\text{ceil}} = (A_0/k)(1 - e^{-kT})/(1 - r)$ (Eq. 12)	Long-run daily output ceiling as a function of break quality r .

All rows are direct evaluations of the equations in Section 3; none depend on simulated data.

Table 1 is worth reading as a single connected argument rather than four independent facts. Each row is derived from the same underlying assumption (Equation 3), and each depends on the same single parameter k — which means that a single, well-estimated k for a given user determines their break timing, their session ceiling, their recommended stopping point, and (together with r) their long-run daily ceiling, all from one number. That internal consistency is a direct product of the derivation in Section 3 rather than a coincidence: because every quantity in Table 1 is derived from Equation (5) rather than fit independently, none of the four rows could in principle disagree with the others in the way that four independently-tuned heuristics could.

4.2 Model Comparison

Table 2 sets the structural comparison from Sections 3.3–3.4 next to the simulated-data fit from Section 4.3, so the theoretical and empirical pictures can be read together.

Table 2. Linear vs. exponential model, structural properties and simulated mean R^2 ($N = 60$ simulated users).

Property	Linear	Exponential
Governing ODE	$dA/dt = -k$	$dA/dt = -k \cdot A$
Solution	$A_0 - k \cdot t$	$A_0 \cdot e^{(-kt)}$
Absolute loss rate	Constant	Decreasing
$A(t)$ reaches zero	Yes, at $t^* = A_0/k$	No (asymptotic)
Half-life	None	$T_{1/2} = \ln 2/k$
$P(T)$ bounded	No	Yes, at A_0/k
Mean R^2 (simulated)	0.934	0.974

Bottom row is simulated data (Section 3.8), all other rows are analytical (Section 3.3–3.4).

What Table 2 makes visible is that the linear model's structural failures (rows 1–4) and its empirical underperformance (row 5) are not two separate pieces of evidence pointing the same direction by coincidence. The absence of a half-life and a bounded $P(T)$ under the linear model are consequences of the same constant-absolute-loss assumption that produces its worse fit to checkpoint data — a model that loses attention at a fixed rate regardless of how much remains has no mechanism to slow down as it approaches zero, and that same lack of a slowing mechanism is precisely what shows up as a widening residual gap in the second half of a session (Figure 2, discussed next). The theoretical and empirical columns of Table 2 are therefore two views of one underlying fact rather than independent confirmations.

4.3 Population-Level Results

A single simulated session, checkpointed every 5 minutes, recovered exponential parameters $A_0 = 9.32$, $k = 0.0210$ ($R^2 = 0.978$) against a linear fit of $A_0 = 8.57$, $k = 0.1071$ ($R^2 = 0.932$). The recovered exponential parameters are close to the simulation's true values ($A_0=9.4$, $k=0.021$), which is a check on the fitting code rather than a result in itself — it confirms that `curve_fit` recovers parameters correctly when the model is well-specified, which is a necessary precondition for trusting any of the population-level numbers that follow, but it is not evidence about which model fits better. Figure 2 shows the fitted curves against the checkpoints, with the scheduler's recommended break time (Equation 6, at 80% of $T^{1/2}$) marked.

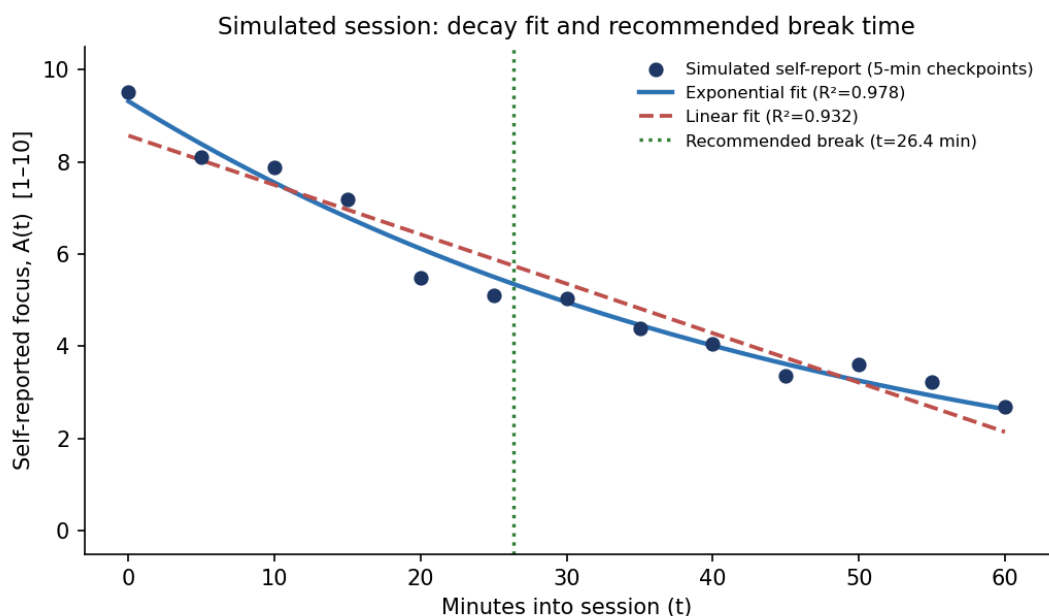


Figure 2. SIMULATED DATA. One 60-minute session, checkpoints every 5 minutes, with both fitted curves and the recommended break time from Equation (6).

Both curves fit the first half of the session about equally well — any smooth decreasing curve does, over a short segment, for the same Taylor-expansion reason noted in Section 4.1 — and separate visibly in the second half, where the linear fit overshoots the true deceleration. This is the qualitative failure predicted in Section 3.3, showing up here as a quantitative gap. It is worth being explicit about what Figure 2 does and does not demonstrate: it shows that, for this one simulated draw, the exponential curve tracks the later checkpoints (35–60 minutes) more closely than the linear curve does, visible as consistently smaller vertical gaps between the blue curve and the data points in that region compared to the red dashed line. It does not,

on its own, establish that this pattern generalizes beyond one session, which is exactly why the population-level analysis that follows is necessary rather than illustrative.

Repeating the fit across 60 independently simulated users gives a mean R^2 of 0.974 (SD 0.015) for the exponential model against 0.934 (SD 0.027) for the linear model, with the exponential model ahead in 93.3% of simulated users. Figure 3 shows the full distribution rather than just the mean, which matters here because a mean comparison alone can obscure whether the advantage is consistent or driven by a handful of extreme cases.

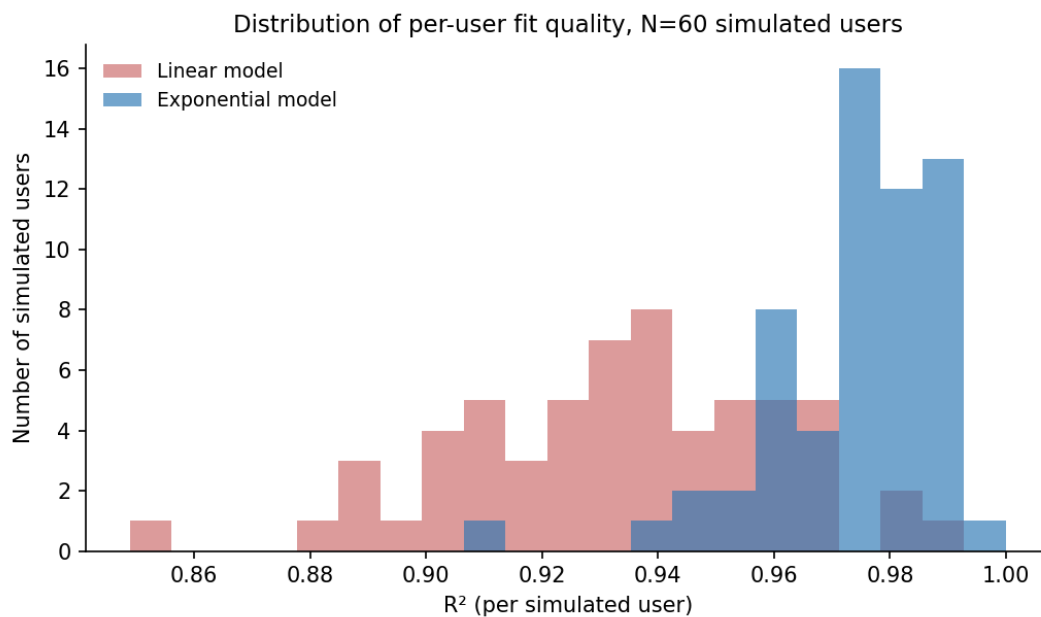


Figure 3. SIMULATED DATA. Per-user R^2 for both models, $N = 60$ simulated users, 5-minute checkpoints per session.

The exponential distribution sits almost entirely above 0.95; the linear distribution is wider and shifted lower, with its worst case ($R^2 = 0.849$) noticeably below the exponential model's worst case ($R^2 = 0.908$). Three features of Figure 3 are worth drawing out individually rather than folding into a single summary statistic. First, the exponential distribution is not only higher on average but visibly tighter — its users are more consistently well-fit, which suggests the exponential form is not winning on average by combining some excellent fits with some poor ones, but is instead reliably better across most of the simulated population. Second, the linear distribution's long left tail, extending down toward $R^2 \approx 0.85$, corresponds to simulated users with either a higher true k (faster decay, so the curvature the linear model cannot represent is more pronounced within the 60-minute window) or a less favorable noise draw; this is consistent with, rather than contrary to, the mechanism argued for in Section 3.4. Third, the visible overlap between the two histograms in the 0.93–0.96 range is itself informative: it identifies the boundary condition under which the linear model becomes competitive, which is exactly the short-session, low-curvature regime discussed in Section 4.1. None of these three observations was guaranteed in advance by the derivation in Section 3 — the derivation predicts a direction of effect, not the specific shape of a fit-quality distribution — so Figure 3 is genuine empirical content on top of the theoretical argument, not merely an illustration of it.

4.4 Sensitivity Analysis

Equation (12) makes a specific claim: output ceiling depends on break quality r through a $1/(1-r)$ factor, so the relationship is not linear and gets steep as r approaches 1. Table 3 and Figure 4 evaluate this directly from the formula — no simulated data or noise is involved in this subsection.

Table 3. Daily output ceiling factor, $1/(1-r)$, evaluated analytically from Equation (12) for a range of break-quality values.

Break quality (r)	Ceiling factor	Reading
0.70	$3.3\times$	Weak recovery caps long-run output near $3\times$ a single session.
0.80	$5.0\times$	Baseline used in the <code>break_scheduler.py</code> example.
0.85	$6.7\times$	
0.90	$10.0\times$	Doubling r 's distance from 1 doubles the ceiling factor.
0.95	$20.0\times$	Near-full recovery roughly doubles the 0.90 ceiling again.

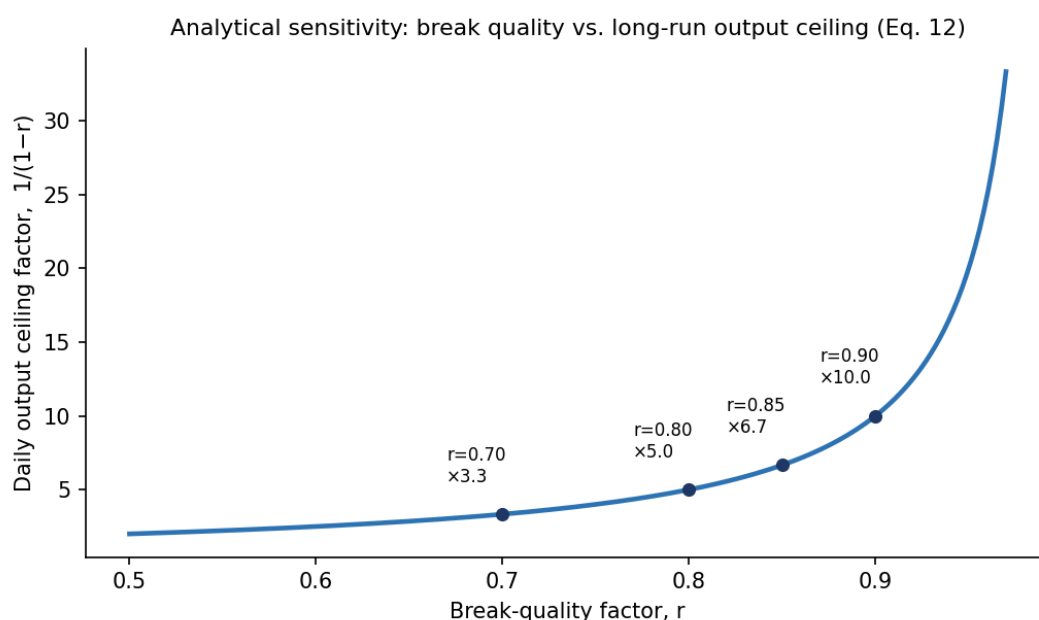


Figure 4. ANALYTICAL, no simulated data. Daily output ceiling factor as a function of break quality r , from Equation (12).

The most important pattern in Figure 4 is the visible acceleration of the curve as r approaches 1: the ceiling factor roughly doubles between $r = 0.80$ and $r = 0.90$, and roughly doubles again between $r = 0.90$ and $r = 0.95$, even though the absolute change in r is the same (0.10) in the first case and half as large (0.05) in the second. This is a direct algebraic property of the $1/(1-r)$ term in Equation (12) rather than an empirical finding, but it has a concrete practical reading: the payoff to improving break quality is not uniform across the scale, and the highest-leverage improvements are concentrated in the region where recovery is already fairly good. For Focusware's design, this suggests that recommending marginal improvements to already-decent breaks (moving a user from $r = 0.85$ to $r = 0.90$, say, by nudging a phone-based break toward a short walk) may be worth more in expected daily output than the same nominal improvement applied to a user with poor recovery habits to begin with — though this is a projection from the closed-form model, not a claim tested against real behavior-change data, and should be read with that caveat.

A parallel check on Equation (6): halving k from 0.025 to 0.0125 doubles the half-life from about 28 to 55 minutes, exactly as the $1/k$ form predicts. Both checks confirm the closed-form results behave the way Section 3 says they should before any fitting is involved, which is a modest but not trivial form of validation — it rules out implementation errors in the code underlying Table 3 and Figure 4, and confirms that the sensitivity behavior described qualitatively in Section 3.6 is not an approximation but an exact property of Equation (12) across the full range tested.

V. Discussion

5.1 Interpretation of Results

The single-session and population fits point the same way: the exponential model is not just cleaner to derive, it tracks noisy, session-shaped data better than the linear alternative in the large majority of simulated cases. The cases where it did not — roughly 7% of simulated users — are concentrated where the two curves have the least room to separate, which is consistent with, rather than contrary to, the mechanism in Section 3.4.

One possible explanation for why the gap is as large as it is (mean R^2 difference of roughly four hundredths, with the linear model's distribution both lower and wider in Figure 3) is that the 60-minute session length used in the simulation is long enough, relative to the simulated half-lives, for the curvature difference between the two models to become clearly visible by the later checkpoints — recall from Section 4.1 that the two models are nearly indistinguishable early in a session by construction. Had the simulated sessions been shorter, say 20 or 30 minutes, this paper would predict the gap to narrow considerably, since less of the session would fall in the region where the curves diverge. This is not a claim tested directly in the current simulation, but it follows from the same Taylor-expansion argument used in Section 4.1, and it is a specific, checkable prediction that a future version of this analysis (Section 7) could confirm by re-running Section 3.8's procedure at multiple session lengths.

An alternative explanation worth considering, and ruling out where possible, is that the exponential model's advantage in Section 4.3 is an artifact of the simulation's own generative process — since the simulated data was itself generated from an exponential curve with added noise (Section 3.8), it might seem unsurprising that an exponential fit recovers it well. This concern is reasonable but does not fully explain the result: the comparison in Section 4.3 is not between the exponential model and a null model, but between two candidate models fit to the same noisy data, and a linear model fit by least squares to exponentially-generated data would, if the true curvature were negligible over the sampled window, still be capable of achieving a comparably high R^2 by absorbing the curvature into its slope estimate. That it does not — the linear R^2 is measurably and consistently lower — indicates the curvature in the simulated data is large enough, given the chosen k range and session length, to exceed what a straight line can absorb, which is itself the finding, not a circularity in how the finding was obtained. The more precise way to state this limitation is that the simulation confirms the fitting and comparison procedure behaves as the theory predicts under the theory's own generative assumptions; it does not, and cannot, confirm that real human attention actually follows those generative assumptions, which is exactly why Section 3.1 frames the simulation as a stand-in rather than a validation and why Section 7 lists real-data validation as the first priority for future work.

5.2 Comparison With Previous Research

The exponential-vs-power-law dispute in the memory literature (Section 2.1) concerns retention over days; it does not bear directly on a 60-minute session, and the good exponential fit here is consistent with exponential decay being the stronger default at short horizons, though “consistent with” is a weaker claim than “confirmed by,” and this paper's simulated result cannot settle a debate that Wixted and Ebbesen's original comparison was conducted on real, longer-horizon retention data. The more honest way to state the relationship is that this paper's within-session result and the memory literature's long-horizon result are about different regimes and are not in tension with each other, rather than that one supports the other.

The vigilance-decrement literature (Section 2.2) does not commit to a functional form, but its central claim — decline is not constant-rate — is what Section 4.3 shows numerically: a constant-rate model measurably underfits a proportional-rate one. This finding extends the vigilance-decrement literature in a specific, narrow way rather than a general one: See et al.'s meta-analysis and the theoretical accounts that followed it (Thomson et al.; Esterman and Rothlein) establish that performance declines with time-on-task, but, as noted in Section 2.1, none of them commits to a specific mathematical form for that decline. This paper's contribution is not to discover that attention declines — that is well established — but to show that a specific, minimal mechanistic assumption about how it declines is both mathematically forcing and, on simulated data at least, empirically favored over the simplest alternative. Whether that finding would replicate on real behavioral data collected under the conditions the vigilance-decrement literature actually studies (sustained visual monitoring tasks, typically under laboratory conditions) rather than the self-reported, real-world, mixed-task conditions Focusware operates under is an open question this paper does not and cannot answer; Section 6 returns to this gap between laboratory vigilance tasks and app-based self-report as a limitation of the comparison.

Settles and Meeder's half-life regression (Section 2.2) offers a useful point of contrast rather than direct support: their model's strong empirical performance was established for spaced-repetition scheduling, a task with a very different feedback structure (users are tested on retention, producing an objective correctness signal) than Focusware's self-reported focus ratings, which have no external ground truth to validate against. That their shared mathematical structure (proportional decay, half-life-based scheduling) succeeded in one domain does not transfer as evidence that it will succeed in the other; it is at most a reason to expect the mathematical machinery to be tractable, which is a much weaker claim than expecting the empirical result to replicate.

5.3 Practical / Product Implications

Every quantity in Table 1 already runs somewhere in Focusware: Equation (6) sets the break scheduler, Equations (7)–(9) set the productivity score and exhaustion check, Equation (12) explains why the day view shows disproportionate gains from better breaks rather than merely additive ones. Because k and r are estimated per user (Section 3.7) rather than hand-tuned per feature, a better k or r estimate updates every downstream feature automatically.

This has a specific practical consequence worth spelling out. Under a system where break timing, productivity scoring, and daily projections were each governed by separately-tuned heuristics — the alternative design this paper's derivation was written to avoid — improving one feature (say, a better break-timing rule discovered through user testing) would not automatically improve the others; each would

need its own separate tuning pass, and the three could easily drift out of consistency with one another over successive product iterations. Under the derivation-first design argued for here, a single improvement to how k is estimated (Section 3.7) or to the estimation procedure's robustness (Section 6) propagates automatically through Equations (6)–(9) and (12), because all four are algebraically downstream of the same k . This is a maintainability argument as much as a scientific one, and it is the practical payoff of having done the derivation in Section 3 rather than choosing a decay curve by inspection.

At the same time, this tight coupling is a double-edged design property. Precisely because every downstream feature depends on the same k , an error in how k is estimated — whether from the outlier sensitivity discussed in Section 3.7, from within-user variation in k that the model does not currently accommodate (Section 6), or from a genuine mismatch between the exponential assumption and a particular user's actual attention pattern — will also propagate through every downstream feature simultaneously, rather than being contained to one. A break scheduler with an independently-tuned heuristic might be robust to a bad k estimate elsewhere in the system; Focusware's design, as derived here, is not, and this trade-off between consistency and fragility is a direct consequence of the derivation-first approach rather than an incidental implementation detail.

5.4 What Would Change the Conclusion

The exponential model's advantage would not survive: real session data where linear R^2 matches or beats exponential R^2 ; systematic residuals suggesting a better-fitting family, such as a power-law tail (Section 2.1); or a k that varies so much within one user's own sessions that treating it as stable (Section 3.7) stops making sense. None of these would overturn the derivation in Section 3.4 on its own — that is a claim about mechanism — but they would mean the mechanism needs more than one constant k , which is what Section 7 proposes to test next.

It is also worth stating what would not change the conclusion, since a falsification criterion is only useful if it can be distinguished from a merely disappointing result. A modest reduction in the R^2 gap on real data — say, from the roughly four-hundredths difference seen here to something closer to one or two hundredths — would not by itself overturn the model, since the simulated comparison already shows the two models converging for shorter sessions and less curvature (Section 5.1); a smaller gap on real data could equally reflect a real population with shorter typical sessions or lower typical k values as it could reflect a failure of the underlying assumption. What would be diagnostic is specifically a reversal of direction (linear outperforming exponential) or a residual pattern in the exponential fit that is systematic rather than randomly scattered — for instance, if real residuals showed a consistent late-session upward bend inconsistent with continued exponential decay, that would point toward a genuinely different functional form (such as the power-law tail flagged in Section 2.1) rather than toward sampling noise around the currently-favored model.

VI. Limitations

The limitations below are not a generic checklist; each follows from a specific choice made in Sections 3 or 4, and each is stated together with how it would actually affect interpretation of the results in Section 4 and Section 5, rather than left as an abstract caveat.

- Section 4's validation is simulated, not real — no Focusware session-log export was available at the time of writing. This is the single most consequential limitation in the paper: every empirical number

in Section 4.3 (the R^2 values, the population distribution in Figure 3) describes how well the two models fit data generated from the exponential model's own assumptions plus noise, not how well they fit actual human self-report. As discussed in Section 5.1, this means the simulation validates the fitting and comparison procedure, not the underlying psychological claim, and the paper's conclusions about which model is correct for real attention should be read as a prediction pending confirmation, not a settled empirical finding.

- k is treated as constant per user; task difficulty and novelty likely shift it within a single user's history. If true k varies substantially across a user's own sessions — higher on unfamiliar or cognitively demanding tasks, lower on routine ones — then the single averaged k produced by `user_k()` (Appendix A) is a compromise value that will systematically over-schedule breaks on easy tasks and under-schedule them on hard ones, rather than being uniformly accurate. This directly affects how much confidence should be placed in the break-timing recommendations discussed in Section 5.3.
- r is a single scalar; real recovery plausibly depends on break length, activity, and time of day. Section 3.6 already flags this as an idealization built into Equation (10); the practical consequence is that the multi-session ceiling in Equation (12) and Figure 4 should be read as an estimate under a fixed- r assumption, and the true daily ceiling for a user who takes a mix of short unhelpful breaks and longer restorative ones is unlikely to be well captured by any single r .
- A_0 varies day to day with sleep and stress; Focusware only partly captures this via a morning rating. Because every downstream quantity in Table 1 scales with A_0 , a poorly calibrated morning rating (whether from response bias, scale insensitivity, or a user's own imprecise self-knowledge of their momentary state) propagates through the session ceiling, the optimal stopping time, and the daily ceiling alike — the same propagation property discussed as a design trade-off in Section 5.3.
- The model is deterministic — it has no way to express the scatter already visible in Figure 2 as a calibrated uncertainty. This means that all of the point estimates in Section 4 (recommended break time, session ceiling, and so on) are presented without confidence intervals, which is a genuine gap for a system making individual-level scheduling recommendations; Section 7's proposed stochastic extension is aimed directly at closing this gap rather than being a generic suggestion for future work.
- $A(t)$ is monotone decreasing by construction, so flow states (where attention rises) are outside what this model can represent. If flow states are common enough in real usage to matter for Focusware's break-timing accuracy, the model as derived here would systematically recommend breaks during periods where a user's attention is, in fact, still rising — a failure mode the current derivation has no internal way of detecting, since Equation (3) assumes $dA/dt < 0$ by construction rather than deriving it as a conclusion.
- No paired statistical significance test accompanies the R^2 comparison in Section 4.3 (noted in Section 3.9). The reported R^2 gap describes the simulated sample obtained, but the paper does not report, for instance, a paired test on per-user R^2 differences that would characterize how reliably this gap would replicate across independent simulation runs with different random seeds, which is a natural next analytical step before treating the specific numeric gap (rather than its direction) as a stable quantity.

VII. Conclusion and Future Work

One assumption, Equation (3), forces everything that follows: the exponential form, the half-life, the bounded session output, and the multi-session ceiling. None of it needed separate tuning, and Section 4's simulated comparison shows the resulting model fits session-shaped data better than the alternative it rules out on structural grounds. Every equation maps to a function already in Focusware (Appendix A), and the practical question this started from — when should someone take a break — turns out to have an answer derived from the governing equation rather than picked by convention.

What this paper contributes, concretely, is not the observation that attention declines with time — that is well established in the vigilance-decrement literature reviewed in Section 2.2 — but a demonstration that a single, minimal, mechanistically motivated assumption is sufficient to derive an entire connected system of formulas (Table 1) rather than requiring each formula to be independently justified or tuned. That is a different kind of contribution than a new empirical finding, and it should be read as such: the paper's central claim is architectural (one assumption, several forced consequences, each traceable back to Equation 3) as much as it is empirical (the exponential model fits better than the linear one on the data examined). Readers primarily interested in the empirical claim should weight Section 4.3 accordingly and keep Section 6's first limitation — that this data is simulated — firmly in view.

What remains genuinely uncertain, and should not be overstated in either direction, is whether the specific mechanistic story offered in Section 3.4 — proportional loss tied to a resource-depletion account of the vigilance decrement — is the correct explanation for why attention behaves the way it does, as opposed to merely a mathematically convenient assumption that happens to produce a well-fitting curve. The paper has been careful throughout not to claim more than the derivation and the simulated comparison actually support: Equation (3) is shown to be sufficient to produce a better-fitting model than the linear alternative, not shown to be the uniquely correct description of the underlying cognitive process, and Section 5.1 already discusses at least one alternative explanation (a Taylor-expansion argument for why short sessions would show a smaller gap) that does not require appeal to any particular biological mechanism at all. Distinguishing between “this assumption is mathematically productive” and “this assumption is mechanistically correct” is exactly the kind of distinction that only real behavioral data, not further derivation, can resolve — which is why real-data validation heads the list of future work below rather than appearing as an afterthought.

- Real-data validation. Re-run Section 4's exact fitting procedure on real session-log exports once Focusware has enough logged users, and check the result against Section 5.4's falsification conditions. This is the single highest-priority item, since every other future-work item below presupposes that the basic exponential-versus-linear finding survives contact with real data; if it does not, the stochastic extension and the break-type analysis described next would need to be built on whatever functional form real data actually supports, not necessarily the exponential one.
- Power-law comparison. Fit a power-law $A(t)$ alongside the linear and exponential candidates, given Section 2.1's caution that power-law forms sometimes win at longer horizons. Even though Section 2.1 argues the memory-decay literature's power-law-versus-exponential debate concerns a longer timescale than Focusware's sessions, the only way to confirm that the distinction genuinely does not matter at Focusware's timescale is to run the comparison directly rather than to assume it away by analogy.
- A stochastic extension. Replace Equation (3) with $dA = -k \cdot A \, dt + \sigma \cdot dW$, an Ornstein–Uhlenbeck-style process that keeps every mean-behavior formula in Table 1 unchanged while adding a volatility term σ the current model has no way to express. Estimating σ from the residual scatter already visible in Figure 2 would let the break scheduler report a range (“between 28 and 34 minutes”) instead of a single number, and would separate two users with the same average k but different session-to-session volatility. This needs more within-session checkpoints than Focusware currently logs, since most sessions today record only a start and end rating. This item follows directly from the deterministic-model limitation identified in Section 6, and it is the most technically demanding item on this list, since it would require both a change to Focusware's data collection (more frequent in-session checkpoints) and a corresponding change to the estimation code in Appendix A.5.

- Break-type and timing effects on r . Track r by break activity and duration at scale to test whether a single scalar r is adequate, or whether Equation (11) needs separate terms for break length and time of day, addressing the recovery-model idealization flagged in Section 3.6 and Section 6.
- Within-user variation in k . Section 6 notes that treating k as constant per user is an idealization if task difficulty and novelty shift a user's true decay rate across sessions. A natural extension is to test whether k estimated separately by task category (using whatever task-labeling data Focusware already collects, if any) produces measurably different values, which would indicate the single- k model is systematically biased for users whose work mixes routine and demanding tasks.

References

- [1] Ebbinghaus, H. (1885). *Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie*. Duncker & Humblot, Leipzig.
- [2] Wixted, J. T., & Ebbesen, E. B. (1991). On the form of forgetting. *Psychological Science*, 2(6), 409–415.
- [3] Murre, J. M. J., & Dros, J. (2015). Replication and analysis of Ebbinghaus' forgetting curve. *PLOS ONE*, 10(7), e0120644.
- [4] See, J. E., Howe, S. R., Warm, J. S., & Dember, W. N. (1995). Meta-analysis of the sensitivity decrement in vigilance. *Psychological Bulletin*, 117(2), 230–249.
- [5] Thomson, D. R., Besner, D., & Smilek, D. (2015). A resource-control account of sustained attention. *Perspectives on Psychological Science*, 10(1), 82–96.
- [6] Esterman, M., & Rothlein, D. (2019). Models of sustained attention. *Current Opinion in Psychology*, 29, 174–180.
- [7] Settles, B., & Meeder, B. (2016). A trainable spaced repetition model for language learning. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 1848–1858.

Appendix A. Full Code

Complete, reproducible implementation of every equation in Section 3. The short excerpts in the main text (Codes 1–5) are drawn directly from these files.

A.1 attention_model.py

```
import math

def attention(t, A0, k):
    # Equation (5):  $A(t) = A_0 * e^{-kt}$ 
    return A0 * math.exp(-k * t)

def half_life(k):
    # Equation (6):  $T_{half} = \ln(2) / k$ 
    return math.log(2) / k
```

A.2 break_scheduler.py

```
import math

def first_break_time(k):
    """Equation (6), scheduled at 80% of the half-life."""
    T_half = math.log(2) / k
    return 0.8 * T_half

# Example: k = 0.018 -> T_half = 38.5 min -> break at 30.8 min
```

A.3 productivity_score.py

```
import math

def session_score(A0, k, T):
    # Equation (7):  $P(T) = (A0/k)(1 - e^{-kT})$ 
    return (A0 / k) * (1 - math.exp(-k * T))

def session_ceiling(A0, k):
    # Equation (8):  $P_{max} = A0 / k$ 
    return A0 / k

def pct_of_ceiling(A0, k, T):
    return session_score(A0, k, T) / session_ceiling(A0, k)

def is_exhausted(A0, k, T, threshold=0.95):
    """True once  $\geq 95\%$  of the theoretical ceiling is captured."""
    return pct_of_ceiling(A0, k, T) >= threshold

def optimal_session_length(A0, k, A_min):
    # Equation (9):  $T^* = (1/k) * \ln(A0 / A_{min})$ 
    if A0 <= A_min:
        return 0
    return (1.0 / k) * math.log(A0 / A_min)
```

A.4 multi_session.py

```
import math

def daily_score(A0, k, T, r, N):
    # Equation (11)
    per_session = (A0 / k) * (1 - math.exp(-k * T))
    geo = N if r == 1.0 else (1 - r**N) / (1 - r)
    return per_session * geo

def daily_ceiling(A0, k, T, r):
    # Equation (12), the  $N \rightarrow \text{infinity}$  limit
    if r >= 1.0:
        return float('inf')
    return (A0 / k) * (1 - math.exp(-k * T)) / (1 - r)
```

A.5 estimate.py

```

import math
from collections import defaultdict

def estimate_k(A_start, A_end, T):
    # Equation (13)
    if A_end <= 0 or A_start <= 0 or T <= 0:
        return None
    return -math.log(A_end / A_start) / T

def user_k(session_history):
    ks = [estimate_k(s['A_start'], s['A_end'], s['T']) for s in
session_history]
    ks = [k for k in ks if k and k > 0]
    return sum(ks) / len(ks) if ks else None

def estimate_r(A_after, A_baseline):
    # Equation (14), capped at 1.0
    return min(1.0, A_after / A_baseline)

def r_by_break_type(break_history):
    totals = defaultdict(list)
    for b in break_history:
        r = estimate_r(b['A_after'], b['A_baseline'])
        totals[b['activity']].append(r)
    return {act: sum(rs) / len(rs) for act, rs in totals.items()}

# Example output: {'walk': 0.91, 'phone': 0.73, 'nap': 0.96, 'food': 0.88}

```

A.6 Simulation and Model-Fitting Code (Section 3.8–3.9)

```

import numpy as np
from scipy.optimize import curve_fit

def exp_model(t, A0, k): return A0 * np.exp(-k * t)
def lin_model(t, A0, k): return A0 - k * t

def r_squared(y_obs, y_pred):
    ss_res = np.sum((y_obs - y_pred) ** 2)
    ss_tot = np.sum((y_obs - np.mean(y_obs)) ** 2)
    return 1 - ss_res / ss_tot

# Single session: 5-minute checkpoints over 60 minutes,
# A0=9.4, k=0.021, Gaussian rating noise sigma=0.35.
# Both exp_model and lin_model fit by curve_fit; R^2 compared.
# Repeated for 60 simulated users with A0~U(7.5,10), k~U(0.010,0.035).

```