Spaced Repetition Algorithm Optimization: A Technical Analysis

Modern spaced repetition scheduling has undergone a fundamental transformation. **FSRS-6 now outperforms the classic SM-2 algorithm in 99.6% of user collections**, delivering 20-30% fewer daily reviews (GitHub) while maintaining equivalent retention. This shift represents decades of evolution from Piotr Wozniak's 1987 SM-2 algorithm through machine learning-optimized systems that model memory as three interdependent variables: stability, difficulty, and retrievability. The algorithmic arms race continues—neural networks like RWKV achieve state-of-the-art prediction accuracy, though at computational costs that remain impractical for personal use. For practitioners, the choice is increasingly clear: FSRS provides near-optimal scheduling with 21 trainable parameters that can be personalized from 1,000 reviews, while SM-2's simplicity comes at the cost of the notorious "ease factor hell" that traps difficult cards in endless review cycles.

SM-2's mathematical elegance conceals fundamental design flaws

The SM-2 algorithm tracks three properties per card: repetition count, easiness factor (EF), and inter-repetition interval. Its core scheduling formula appears deceptively simple:

Interval calculations:

- I(1) = 1 day
- I(2) = 6 days
- For n > 2: I(n) = I(n-1) × EF (Super-memory)

The easiness factor adjusts after each review using the formula: **EF' = EF + (0.1 - (5-q) × (0.08 + (5-q) × 0.02))**, where q represents the user's grade on a 0-5 scale. (Super-memory) This yields asymmetric adjustments: a perfect score adds +0.10, while grade 3 subtracts -0.14, and a complete blackout removes -0.80. The minimum EF is capped at 1.3. (Super-memory)

This asymmetry creates the "ease factor death spiral." Since only perfect responses increase EF, while any difficulty decreases it, cards inevitably accumulate at the 1.3 floor. A card stuck at minimum ease requires **22 successful reviews to reach a one-year interval**—a punishing schedule regardless of eventual mastery. Additionally, SM-2's fixed initial intervals (1 day, then 6 days) (Gitbook) ignore research showing the optimal first interval at 90% retention is approximately 3.96 days. The algorithm also lacks any personalization mechanism—identical formulas apply to all users regardless of their individual forgetting rates.

Wozniak's subsequent algorithms addressed these limitations incrementally. SM-5 (1989) introduced the Optimal Factor matrix, replacing fixed intervals with adaptable values. (Super-memory) SM-8 (1995) brought data-driven determination through retention factor matrices derived from actual forgetting curves. (Supermemo) The pivotal breakthrough came with SM-17 (2016), which introduced (Wikipedia) the **three-component model of memory**: stability (time for retrievability to drop to 90%), retrievability (recall probability at any moment), and difficulty (inherent item complexity). (GitHub) SM-18 (2019) refined the stabilization function, (Supermemo) though these algorithms remain proprietary to SuperMemo.

FSRS implements the three-component model through machine learning optimization

Jarrett Ye created FSRS while working at MaiMemo, (lesswrong) publishing the foundational work at ACM KDD 2022. The algorithm achieved official Anki integration in November 2023, (lesswrong) bringing optimized scheduling to millions of users. FSRS models memory using the same DSR framework as SM-17/SM-18 (GitHub) but with open-source, machine-learning-trained parameters.

The forgetting curve in FSRS-6 uses a power function with a trainable decay parameter: **$R(t,S) = (1 + factor \times t/S)^{(-w_{20})}$**

where factor = $0.9^{(-1/w_{20})} - 1$, ensuring retrievability equals 90% when elapsed time equals stability. This formulation outperforms the exponential curves used in earlier algorithms because power functions better match population-aggregated forgetting data—a mathematical consequence of averaging exponential curves with varying individual rates.

FSRS-6 employs **21 trainable parameters** organized into functional groups. Parameters w_0-w_3 set initial stability for each first rating (Again/Hard/Good/Easy), with defaults of approximately 0.21, 1.29, 2.31, and 8.30 days respectively. Parameters w_4-w_7 govern difficulty dynamics, including mean reversion that prevents the ease hell phenomenon. The stability increase formula for successful reviews incorporates three key insights:

$S'(D,S,R,G) = S \times (e^{w_8} \times (11-D) \times S^{(-w_9)} \times (e^{(w_{10} \times (1-R))}-1) \times modifier + 1)$

This encodes that higher difficulty yields smaller stability increases, higher existing stability yields smaller increases (saturation effect), and lower retrievability yields larger increases (the spacing effect). Parameters w_{15} and w_{16} apply penalties for "Hard" ratings and bonuses for "Easy" ratings.

FSRS avoids ease hell through difficulty mean reversion: $D'' = w_7 \times D_0(4) + (1 - w_7) \times D'$, which continuously pulls difficulty toward the default value. Combined with linear damping that asymptotically approaches the maximum difficulty of 10, this prevents cards from becoming permanently trapped.

Benchmark results on 350 million reviews from 10,000 users demonstrate FSRS-6's advantages quantitatively. Log loss drops from approximately 0.37 for SM-2 to 0.32 for optimized FSRS-6. RMSE on binned predictions improves from ~0.065 to ~0.043. Most critically, **97.4% of users achieve better predictions with FSRS** compared to SM-2 when both use default parameters.

Machine learning approaches achieve higher accuracy at substantial computational cost

Deep Knowledge Tracing (DKT), introduced by Piech et al. at NeurIPS 2015, first demonstrated that recurrent neural networks could outperform traditional knowledge tracing. DKT uses LSTM networks to maintain a continuous latent vector representing knowledge state, achieving AUC of 0.85 on Khan Academy data versus 0.68 for Bayesian Knowledge Tracing—a **25% improvement**.

Duolingo's Half-Life Regression (HLR) takes a more interpretable approach, modeling recall probability as $p = 2^{(-\Delta/h)}$ where Δ is time since last review and h is the memory half-life. The half-life is estimated as $h = 2^{(\theta \cdot x)}$, incorporating features like right/wrong counts, lexeme difficulty (word frequency, morphological

complexity), and historical spacing. HLR reduced prediction error by 45% compared to Leitner and Pimsleur baselines, and increased Duolingo daily engagement by 12%.

Transformer-based models have evolved rapidly since 2019. Self-Attentive Knowledge Tracing (SAKT) first applied attention mechanisms to learning histories. Attentive Knowledge Tracing (AKT) integrated psychometric theory by embedding Rasch model parameters for question difficulty, achieving interpretability without sacrificing performance. Recent work includes simpleKT (ICLR 2023), which demonstrates that straightforward dot-product attention with Rasch embeddings achieves top-3 performance across seven datasets —57 wins against 12 deep learning baselines.

The current benchmark leader is **RWKV**, a recurrent neural architecture achieving log loss of 0.278 versus FSRS-6's 0.307. However, RWKV requires 2.76 million parameters compared to FSRS's 21, and training occurs across users rather than per-user optimization. For practical spaced repetition, the compute-accuracy tradeoff strongly favors FSRS until neural network training becomes dramatically cheaper.

Personalization requires sufficient review data and appropriate cold-start strategies

User-specific parameter optimization in FSRS uses gradient descent with binary cross-entropy loss, treating each review as a binary classification problem. The system employs two-phase optimization: initial stability parameters are estimated via curve-fitting from first/second review data, then remaining parameters are optimized through gradient descent (lesswrong) with recency weighting (FSRS-5+).

The cold-start problem presents significant challenges. FSRS recommends **1,000-2,000 reviews** before optimization provides meaningful benefits over default parameters. These defaults were trained on several hundred million reviews from approximately 10,000 users, providing robust population priors. Re-optimization is recommended every two months, more frequently for newer collections.

Individual differences in memory are both substantial and orthogonal to general intelligence. Research by Sense et al. (2016) demonstrated that **speed of forgetting is a stable individual trait**—consistent within individuals over time but varying across materials. Critically, this forgetting rate is the strongest predictor of delayed recall performance, outperforming working memory capacity or IQ measures.

The mathematical form of individual versus population forgetting curves differs fundamentally. Heathcote et al. (2000) proved that individual learning curves follow exponential functions, but population-averaged curves appear as power functions due to aggregation of exponentials with gamma-distributed rates. This explains why FSRS transitioned from exponential (v3) to power-law (v4+) forgetting curves for population-level modeling, (lesswrong) while FSRS-6 introduced the trainable w_{20} parameter allowing personalization of curve shape.

Response time provides an underexploited signal for memory strength estimation. Research consistently shows retrieval fluency correlates inversely with memory strength—fast responses indicate confident, strong memories. Lingvist's system schedules repetitions at 80% predicted recall based partly on response patterns, though FSRS does not yet incorporate timing data beyond the grade.

Curriculum integration demands prerequisite-aware scheduling and repetition compression

Traditional spaced repetition treats items independently, but educational content involves prerequisite relationships that fundamentally change optimal scheduling. Math Academy's Fractional Implicit Repetition (FIRe) algorithm represents the most sophisticated approach to this challenge. FIRe models content as a directed acyclic graph where edges represent both prerequisites and "encompassing" relationships—when practicing an advanced topic implicitly exercises simpler topics. justinmath

The key insight behind repetition compression is that reviewing advanced material can "knock out" multiple simpler reviews simultaneously. Without compression, hierarchical subjects generate exponential review burdens: a course with 300 topics requiring 3-5 questions per review would quickly overwhelm students. FIRe's bidirectional credit flow awards implicit repetitions to encompassed topics while propagating failure penalties forward to dependent topics. justinmath

Multi-objective optimization in curriculum-integrated SRS balances competing goals. The SSP-MMC (Stochastic Shortest Path - Minimize Memorization Cost) formulation frames scheduling as optimal control, using dynamic programming to balance forgetting probability against review cost. The MEMORIZE algorithm (Settles et al., PNAS) provides provable guarantees for review scheduling and achieved 12% increases in Duolingo user retention.

Interleaving and spacing represent complementary but distinct phenomena. Spacing effect research shows **10-30% retention improvements** from distributed versus massed practice. Interleaving—mixing problem types rather than blocking by type—improves discrimination between similar concepts, with meta-analyses showing effect sizes (Hedges' g) up to 0.65-0.66. The discriminative-contrast hypothesis suggests interleaving highlights subtle differences that blocking obscures.

PSI-KT (Prerequisite-Structure Informed Knowledge Tracing), published in 2024, advances the theoretical foundation by modeling knowledge states as latent variables evolving stochastically with temporal decay while incorporating structural influences from prerequisite knowledge components. It achieves superior multi-step prediction accuracy compared to baselines by jointly estimating learner traits and prerequisite graph structure through Bayesian inference.

Implementation requires appropriate complexity-accuracy tradeoffs

SM-2 operations are O(1) per card for both time and space, storing only easiness factor, interval, and repetition count. FSRS maintains the same O(1) scheduling complexity but adds O(n×m) overhead for parameter optimization where n equals reviews and m equals training iterations. Neural approaches scale as O(sequence_length × hidden_dim²) per forward pass—acceptable for large platforms but impractical for personal use.

The open-spaced-repetition GitHub organization maintains reference implementations across languages. Hugging Face The primary implementation is **fsrs-rs** (Rust), used by Anki with optimizer and scheduler crates.io in 6,400 lines of code. **py-fsrs** provides Python bindings with JSON serialization. **ts-fsrs** offers TypeScript

compatibility for web applications. Scheduler-only implementations exist in Go, Swift, Dart, Clojure, and Ruby. GitHub

Benchmarking methodology uses TimeSeriesSplit with 5-fold cross-validation, training on older reviews and testing on newer ones to simulate real-world prediction. Primary metrics include log loss (binary cross-entropy measuring prediction accuracy), RMSE on binned predictions (custom metric grouping by interval/repetitions/lapses), and AUC for discrimination between recall and lapse.

Two major datasets support algorithm development. **FSRS-Anki-20k** contains 1.7 billion reviews from 20,000 users—the largest spaced repetition dataset in existence. **anki-revlogs-10k** provides 727 million reviews from 10,000 users with deck and preset metadata. lesswrong Duolingo's HLR dataset offers 13 million user-word pairs with lexeme features.

Production implementations should store cards with stability, difficulty, state, due date, and review count fields. Review logs require timestamp, card ID, rating, scheduled interval, and elapsed days. Timezone handling should store UTC internally while displaying local time. Fuzz factors prevent review clustering on specific days. The maximum interval cap defaults to 36,500 days (100 years) to prevent overflow issues.

Conclusion

The spaced repetition algorithm landscape has consolidated around the three-component DSR model, Supermemo with FSRS representing the practical optimum for individual users. Its 21 parameters GitHub capture the essential dynamics of human memory—initial stability varying by first response, difficulty mean-reverting to prevent card stagnation, and stability increases governed by spacing, saturation, and difficulty effects. The 99.6% superiority rate over SM-2 reflects not incremental improvement but fundamental architectural advantages: modeling memory strength separately from item difficulty, incorporating the spacing effect mathematically, and enabling per-user optimization.

For curriculum-integrated applications, FIRe's repetition compression approach dramatically reduces review burden through implicit practice credit. justinmath awesome-fsrs Machine learning approaches like RWKV achieve marginally better predictions but require computational resources inconsistent with personal use cases. The practical frontier lies in extending FSRS with response time signals, prerequisite awareness, and improved cold-start handling through transfer learning from similar users.

Future development will likely focus on FSRS-7 as the terminal hand-crafted algorithm, with neural approaches dominating only when training costs decrease substantially. For current implementations, FSRS-6 with personalized parameters represents the evidence-based choice, requiring only 1,000+ reviews for optimization and delivering measurable efficiency improvements that compound over years of learning.