EWC protects what its Fisher sees: a literature synthesis on elastic weight consolidation, scale and coherence

W. Schulz

Nov 10, 2026

Abstract

A literature synthesis reading EWC as a diagonal approximation to a KL anchor on old inputs. At 1–8B it protects what its Fisher data exercises and little else, and published rankings depend on the estimator. It proposes a per-fact Fisher ledger, since tested on a toy and found null. Many sources are unreviewed 2025–26 preprints.

W. Schulz, NeuroGenesys, November 2026. A literature synthesis, not an experiment: the numbers are the cited authors' own and were not reproduced here. Many sources are 2025–26 preprints, several unreviewed, and some were read only from abstracts, search snippets or secondary summaries; those are flagged where they appear. Derivations marked as this report's algebra or arithmetic are not cited results. Research notes: `research_notes/Elastic weight consolidation deep dive/` *in the NeuroGenesys repository.*

Since this review. Its experiments E2 (the Fisher ledger) and E3 (does stiffness decouple edits from consequences?) were run together on the lab's toy knowledge-base transformer and reported in EWC stiffness freezes stale facts. Two of the predictions below failed there. Stiffness did not widen the gap between direct edits and their consequences; it blocked both together. And the Fisher ledger was a null against matched random release, because a superseded fact supplied only about 0.4% of the stiffness its correction faced. The "written beside" bypass pattern appeared at moderate stiffness but was secondary to blockage. These are toy-scale results; the text below is unchanged from the review.

Elastic Weight Consolidation is a one-step Laplace approximation to Bayesian updating. After learning old data, it replaces the posterior over weights with a Gaussian centred on the old solution, with a diagonal Fisher as precision, and it charges every new update $(\lambda/2)\cdot\Sigma_i\, F_i(\theta_i - \theta^*_i)^2$. That one equation explains both its strengths and its failures. At 1–8B parameters, EWC is a strong protector of exactly the capabilities its Fisher data exercises. In 2026 machine-translation fine-tuning of Llama 3.1 8B, it held the general-benchmark drop to 1.7 points against 11.0 for plain SFT, and it beat data mixing, merging and KL-to-base. But it lost task-specific instruction following that its Fisher set did not cover (35.3 vs a base of 71.1). On multi-task continual benchmarks it is a mid-pack baseline whose ranking depends on implementation more than on the method. "EWC" scored below plain sequential LoRA on TRACE (42.19 vs 52.81 on Llama3.1-8B), and a properly built EWC-LoRA beat O-LoRA at 1B. For logical coherence, the theory predicts something worse than "EWC blocks contradicting updates." The Fisher is the local curvature of the KL divergence from the old model, so when a stale fact is in the Fisher data, EWC is a cheap version of the output anchor that NeuroGenesys already found opposes contradicting updates. The update then routes through low-Fisher parameters, which leaves the old trace intact beside the new one and freezes the edit's consequences. Stiffness is also allocated over parameters by frequency, not over propositions by support, so it imposes an implicit entrenchment ordering that nobody has checked against belief-revision rules. The constructive answer is to make stiffness selective.

The Fisher is additive over examples, so a superseded fact's own stiffness can be subtracted exactly ("Fisher ledger"). Release can be gated on prediction error, as in biological reconsolidation, and new stiffness can be granted only to beliefs that pass an independent check. None of these has been tested anywhere. NeuroGenesys has also never run EWC itself, so all six experiments proposed at the end are new to the lab and fit an M2 Max plus a few rented GPU-days.

A Laplace prior whose approximations break in predictable places

Kirkpatrick et al. start from Bayes' rule, $\log p(\theta|D) = \log p(D_B|\theta) + \log p(\theta|D_A) - \log p(D_B)$. They approximate the intractable old-task posterior $p(\theta|D_A)$ as a Gaussian with mean θ^*_A and diagonal precision from the Fisher, which gives the familiar penalty $L_B(\theta) + \Sigma_i (\lambda/2) F_i (\theta_i - \theta^*_{A,i})^2$ (Kirkpatrick et al.). The lab's earlier survey already covers the basic mechanics, online EWC and the empirical-Fisher critique (Fortifying knowledge in continual learning). Three further points matter for what follows.

First, the multi-task form is not the Bayesian recursion. Done exactly, sequential Laplace keeps *one* quadratic, anchored at the latest mode θ^*_B, with accumulated precision $F_A + F_B$. Summed EWC instead keeps one quadratic per task, each at its own anchor. θ^*_B was itself trained under the F_A penalty, so summed EWC counts F_A's information twice. Huszár argued that "the penalties for multiple tasks are applied inconsistently" (Huszár). The authors' reply conceded the point formally. They defended the summed penalty only empirically, on the grounds that it "forces the network to remember older tasks more vividly" (PNAS reply, from search snippets). That concession matters later, because it admits that summed EWC is systematically stiffer than the posterior it claims to approximate. Online EWC (Progress & Compress) implements the correct recursion with a running Fisher $\tilde{F}_t = \gamma \tilde{F}_{t-1} + F_t$ anchored at the latest weights (Schwarz et al.). $\gamma = 1$ is exact Laplace, and $\gamma < 1$ is a deliberate "forgetting prior." EWC++ estimates the Fisher as a moving average during training instead of in a separate pass (Chaudhry et al.).

Second, λ hides a dataset size. The Laplace precision is the Hessian of the *summed* log-likelihood, about $N_A \cdot \bar{F}$ for per-example Fisher $\bar{F}$. Nearly all code computes the per-example mean, so λ silently absorbs N_A. A principled λ is therefore about N_A/N_B relative to the new-task mean loss, before any correction for misspecification. Almost nobody sets it that way, and this is one root of EWC's notorious λ-sensitivity. This is a standard derivation, not an empirical claim.

Third, every step assumes something deep networks violate. Ordered by likely severity, the quadratic is only a local Taylor expansion, but fine-tuning moves weights far relative to the curvature scale. θ^* is usually early-stopped rather than a true mode, so the dropped first-order term is not zero. Fisher equals the Hessian only for a well-specified model at its optimum; Martens shows the Fisher equals the generalized Gauss-Newton matrix "in many important cases," and the GGN in turn equals the Hessian only when output-layer residual terms vanish (Martens). The diagonal ignores strong within-layer correlations. A unimodal Gaussian is a poor posterior over a landscape full of permutation symmetries and flat directions. Two 2025–2026 results put numbers on the diagonal's cost. CSQN, a sampled quasi-Newton extension, reduces EWC's forgetting by 50% and improves its accuracy by 8% on average (Vander Eeckt & Van hamme, IEEE Access 2025). In diffusion models the empirical Fisher is effectively rank-1, aligned with the mean gradient. A rank-1 EWC that costs the same as the diagonal nearly eliminates class-incremental forgetting on MNIST and more than halves it on ImageNet-1k (Wang et al., ICLR 2026). Nobody has measured how far the true old-task loss departs from EWC's quadratic prediction as a function of distance travelled, which is a cheap diagnostic the lab can run.

One identity ties EWC to its neighbours. For softmax outputs, $F = E_x[J_x^\top \Lambda_x J_x]$, and $\tfrac{1}{2}\Delta^\top F \Delta$ is the second-order expansion of $E_x\, KL(p_{\theta^*} \| p_{\theta+\Delta})$ *over the Fisher's inputs. The first-order term vanishes*

for the true* Fisher even when θ^* is not a minimum, because it is an expectation under the model. EWC is therefore the local quadratic of function-space KL anchoring on old inputs. LwF and "KL-to-base" regularizers are the exact KL, but on new inputs. RWalk reads EWC's penalty the same way, as a Fisher-metric distance between output distributions (Chaudhry et al.). K-priors unify weight-space and function-space priors and reach "performance similar to full retraining" from a handful of past examples (Khan & Swaroop). Natural Continual Learning uses the same Kronecker-factored Fisher both as the penalty and as a gradient preconditioner, which makes EWC a trust region in the Fisher metric, and it beats both pure regularization and pure projection (Kao et al., NeurIPS 2021). Knowledge editing hides a further link. MEMIT minimizes $\|WK_0 - M_0\|^2 + \|WK_1 - M_1\|^2$, with $C_0 = \lambda \cdot E[kk^\intercal]$ estimated from Wikitext keys and $\lambda \approx 1.5\times10^4$ (Meng et al., ICLR 2023). Written as $\mathrm{vec}(\Delta)^\intercal(C_0 \otimes I)\mathrm{vec}(\Delta)$, that preservation term is exactly a Kronecker-factored Laplace penalty: it has the input-covariance factor but an identity output factor, and it is solved in closed form. MEMIT's λ sweep, in which specificity rises and efficacy falls monotonically with the best aggregate near 10^4, is EWC's stability-plasticity curve in editing form. AlphaEdit's null-space projection is the $\lambda \to \infty$ limit on the top eigenspace of C_0. Both readings are this report's algebra, not cited results.

The Fisher-as-Gram identity also matters for coherence. Stack whitened per-example Jacobians into G. Then $F = G^\intercal G/N$ over parameters, and $K = GG^\intercal/N$ over examples is an NTK-type kernel with the same non-zero spectrum. The *empirical* Fisher's dual is exactly the per-example loss-gradient kernel $k(x, x') = g_x{}^\intercal g_{x'}$, the quantity behind GradSim's ripple predictions. Diagonal EWC keeps only diag(F), which discards precisely the cross-example structure that predicts which facts an edit drags along.

The estimator, not the idea, decides most published results

"EWC" is a family indexed by how the Fisher is computed, and the choice is rarely reported. van de Ven's controlled comparison (ICLR 2025 blog track, 30 seeds) is the clearest evidence.

Fisher estimator	How it is computed	Split CIFAR-10 Task-IL acc.	Train time
EXACT	expectation over all classes, all old samples	84.91 ± 0.65	972 s
EXACT (n = 500)	same, 500 random samples	84.57	668 s
SAMPLE	one label sampled from the model per input (unbiased MC)	83.77	817 s
EMPIRICAL	squared gradient of the gold label	83.28 ± 0.77	817 s
BATCHED (b = 128)	squares mini-batch-summed gradients (Avalanche, PyCIL default)	83.38	667 s
No EWC	—	69.53	627 s

Source: (van de Ven 2025). The differences look small at each estimator's best λ, but the optimal λ for BATCHED is orders of magnitude larger than for EXACT. Reusing another paper's λ is therefore invalid, and "many currently reported results for EWC could likely be improved by changing the way

the Fisher Information is computed" (van de Ven). The scale gap follows from simple arithmetic. Near an optimum the mean gradient is about zero, so squaring a batch-mean gradient shrinks the estimate by roughly the batch size. Away from the optimum it injects a rank-1 mean-gradient direction, the same structure the diffusion paper exploits on purpose. The practical rule is to cut samples rather than label marginalization: exact on 500 samples nearly matches the full exact Fisher.

For LLMs the exact Fisher is infeasible (about 10^5 vocabulary entries per token). The semantically correct default is SAMPLE: draw next tokens from the model's own distribution. That choice matters for coherence work in particular. The sampled Fisher measures how sensitive the model's *own beliefs* are, which is what "preserve what the model currently believes" requires, and only the true Fisher has the KL interpretation above. The empirical Fisher also weights the directions where the model is *wrong*, and those are exactly the directions an edit may need to move. Kunstner et al. show the empirical Fisher "does not generally capture second-order information" (Kunstner et al.). No transformer-scale version of van de Ven's estimator comparison exists.

Structured estimators buy off-diagonal curvature. Kronecker-factored online Laplace stays above 90% accuracy across 50 permuted-MNIST tasks (Ritter, Botev & Barber, NeurIPS 2018). R-EWC rotates parameter space until the Fisher is nearly diagonal (Liu et al., ICPR 2018). EK-FAC is proven at scale for influence functions on LLMs up to 52B parameters (Grosse et al. 2023), and later work found that MLP parameters dominate influence (Bao et al. 2025). Nobody has used K-FAC for LLM EWC. Among the relatives, Benzing showed that SI and MAS both approximate the square root of the Fisher, so the importance family is closer to one method with different estimators than to distinct ideas (Benzing, AISTATS 2022). Bayesian cousins add value mostly in task-incremental settings: VCL propagates a variational posterior (Nguyen et al.) and UCL ties uncertainty node-wise (Ahn et al., NeurIPS 2019).

The scenario still decides more than the estimator. In the 2019 three-scenarios benchmark, EWC lifted Split MNIST task-incremental accuracy from 87.19% to 98.64%. It was mediocre domain-incrementally (63.95% against 98.42% offline) and at chance class-incrementally (20.01% against a 19.90% no-protection baseline, while generative replay reached 90.79% and iCaRL 94.57%). On permuted MNIST it reached 94.31% domain-incrementally and beat LwF by more than 20 points, because the tasks' inputs are disjoint and distillation on new inputs protects little (van de Ven & Tolias). These are the 2019-protocol figures. The 2022 Nature Machine Intelligence version cited in the lab's earlier report (EWC 20.6%) uses a revised protocol, and both put EWC at chance. The reason is structural. Class-incremental failure is a *between-task discrimination* failure. No quadratic around θ^*_A can encode logits for classes that did not yet exist, so no better estimator will fix it. Domain-incremental results split on input overlap. EWC works when tasks use disjoint inputs and fails when inputs are shared and the mapping changes. LLM fact updates are the bad case: same inputs, new mapping.

Biology supplies one algorithmically sharp idea, multi-timescale stiffness. In Benna and Fusi's model each synapse is a chain of fast-to-slow variables, which raises memory capacity from square-root to almost linear in synapse count (Columbia summary of Benna & Fusi 2016; full text not checked). Kaplanis et al. used such synapses for continual RL (ICML 2018). A "cascade EWC" with geometrically spaced decays would turn online EWC's exponential fading of old constraints into power-law fading. The lab should temper its expectations for this idea, though. Its own toy multi-timescale blocks were no better than an equal-parameter single-rate network ($+4.6 \pm 7.6$, n.s.) (findings ledger 4.3). The more useful biological parallel, reconsolidation, comes up below.

At 1–8B, EWC holds general skills and little else

The lab's earlier report quoted the SLoRA paper (ACL 2026) from search snippets as saying that on Llama3-8B "regularization methods remain weak at this scale, often falling below Seq-LoRA." That quotation is wrong. The paper's text reads: "On Llama3.1-8B and Llama3-8B, methods such as EWC and GEM often fall below the simple Seq-LoRA baseline." The figures quoted earlier, EWC 44.22 against O-LoRA 46.47 on Qwen2.5-7B, are correct. But on Qwen2.5-7B both of them fall below Seq-LoRA (51.54), and EWC finishes below the untuned base model (48.62) (SLoRA, ACL 2026). The full tables, now checked against the PDF, show how much the verdict depends on the benchmark.

Benchmark (SLoRA, LoRA r = 64)	Model	EWC	LwF	GEM	O-LoRA	Seq-LoRA	Base	MTL
TRACE, 8 tasks	Llama3.1-8B	42.19	30.22	46.93	49.31	52.81	40.82	64.98
TRACE, 8 tasks	Qwen2.5-7B	44.22	48.44	47.63	46.47	51.54	48.62	64.28
TRACE, 8 tasks	Llama3-8B	46.62	32.23	46.71	53.53	55.50	40.43	—
Standard CL, 5 tasks (avg. 3 orders)	Llama3-8B	70.01	67.41	—	75.68	62.48	69.29	83.02

On the first four TRACE tasks at Llama3-8B, EWC forgot slightly *more* than unregularized Seq-LoRA (AFR 10.31 vs 9.31) and learned less: new-task accuracy averaged 4.03 points lower. These baselines come from the official TRACE codebase (SLoRA). Older tables are harsher still. On T5-large, EWC averaged 50.3 against O-LoRA's 75.8 on the 5-task benchmark (O-LoRA). On TRACE with LLaMA-2-7B-Chat, EWC's backward transfer (−5.97) was actually better than O-LoRA's (−7.16) at near-equal average accuracy (42.36 vs 42.78) (arXiv 2605.25525).

Implementation is the confound. EWC-LoRA (van de Ven's group, Feb 2026) computes the Fisher over the full $\Delta W = AB$ rather than the A/B factors, because the gradient with respect to W and ΔW is identical and the factor-space Fisher is not invariant to $A \to AS$, $B \to S^{-1}B$. It accumulates the Fisher online ($F \leftarrow 0.9F + F$[?]) and uses $\lambda = 10^7$. It scored 76.39 on T5-large against O-LoRA's 75.01, and 63.08 against 60.01 on LLaMA-3.2-1B-Instruct (Zheng et al.). The same model family and benchmark thus give EWC a 26-point range depending on how it is built. Its overhead was small: peak memory rose from 18 GB to 24 GB and per-task time by about 1–3%. An exact Fisher on 500 samples beat the empirical one. Any 2024–2026 baseline row labelled "EWC" should be read as a verdict on one codebase, not on the method.

EWC's clearest LLM win is retaining general capability during single-domain fine-tuning. Scholz et al. fine-tuned Llama models for translation with a diagonal empirical Fisher on a Tülu-3 subset (75k examples at 1B, 10k at 8B). At 1B, standard SFT cost 6.3–8.1 points of general-benchmark average, while EWC held the drop to ≤2.5. On Amharic, EWC scored 42.4 against a base of 44.9, ahead of KL-to-base (39.7), LoRA (39.6) and four other methods, at a COMET cost of only 0.6. At 8B Spanish, SFT

dropped 11.0 points and EWC only 1.7, against 60.0 for data mixing and 61.5 for 0.5/0.5 merging, with equal translation quality (Scholz et al., arXiv 2609.28395). The same paper shows the limit precisely. MT-specific instruction following fell from 71.1 to 32.3 under SFT, and EWC recovered only to 35.3. Computing the Fisher on that instruction-following data itself (1,600 examples) did not help (34.8), and it lowered the general average to 58.1. The authors conclude that "retention on general benchmarks does not establish that MT-specific instruction following is preserved." The Gemma2-2B Lithuanian study shows the same dependence on what the Fisher measures from another angle. Its Fisher was computed on MMLU. GSM8K was retained only at $\lambda > 10^6$, and λ was swept from 10^2 to 10^{12} at about 4 hours per value on 8×H100 (Šliogeris et al.). EWC is a selection mechanism whose selection is set by the Fisher dataset. It protects what that set exercises, and choosing that set is the whole problem.

Two failures are especially relevant to fact updates. In a key-overlap study on Qwen2.5-1.5B, EWC, LoRA, weight decay and L2 all left held-out accuracy at about 0.78–0.84, roughly the same as SFT. Self-distillation to a frozen snapshot cut the drop from about 15 pp to about 2 pp (arXiv 2604.15574). And penalties fight Adam. Penalizing a direction also shrinks Adam's second-moment estimate there, which *inflates* the effective step in the very directions being protected, by about $1/(1-\alpha)$, an effect also seen at 7B under LoRA (arXiv 2604.22407). Every EWC arm should therefore be run under both AdamW and SGD-momentum, or report effective step sizes.

The RL and editing evidence is indirect but points one way. No paper compares a weight-space EWC penalty with the PPO/GRPO KL-to-reference term. In RLHF, weight averaging beat the KL reward penalty on the alignment–forgetting trade-off (Lin et al., EMNLP 2024). In continual GRPO on Qwen3-VL-4B, the current-task KL term lowered plasticity (MFT 61.77 vs 67.29) without a forgetting gain. CPO's old-policy Fisher proxy, an L1 penalty on the most-moved coordinates, reached MFN 71.56, 14.73 points above the best RL baseline, though EWC itself was not a baseline (CPO, arXiv 2607.04364). The pattern across SFT, RLHF and RL is that weight-space anchors weighted by *old* data protect general retention better than output anchors computed on *new* data. That is this report's reading across papers, not a controlled result. In editing, FisherEdit (Oct 2026) argues that AlphaEdit's robustness "stems primarily from update regularization rather than projection" and replaces projection with a closed-form Fisher-weighted penalty. Its numbers have been seen only in secondary summaries (FisherEdit listing). WISE places FT-EWC, along with ROME, among editors that fail the reliability–generalization–locality triangle (Wang et al.). WikiBigEdit's 502k-edit stream ran no Fisher regularizer at all (Thede et al., ICML 2025).

Costs are manageable at the lab's scale. A full-parameter diagonal EWC stores the anchor (bf16) plus the Fisher (fp32). That is about 3 GB at Qwen2.5-0.5B, 9 GB at 1.5B and 48 GB at 8B, on top of weights, gradients and Adam states. This is the report's arithmetic, since no paper reports measured memory. On a 64 GB M2 Max, full-parameter EWC at 1.5B fits and 8B does not. The ΔW-space EWC-LoRA route costs one Fisher value per adapted-matrix element, about 6 GB fp32 at 1.5B with every linear layer adapted, and less if only MLPs are adapted. A diagonal Fisher over a few hundred to a few thousand sequences takes minutes on the M2 Max. Published λ values range from 10^2 to 10^7 depending on per-token versus per-sequence normalization, so normalizing F per layer by its mean or trace would make λ unit-free. That is folklore, not a tested result. One trap applies to the lab's pipelines: merging an adapter into the base resets ΔW, so a merge-then-new-LoRA loop must carry the Fisher over in full-weight space. In Compose-CL, SI importances interfered with merged LoRA "because importances live in coordinates that merging resets" (the lab's seven-methods study; arXiv 2609.06986).

Stiffness over parameters fights revision and freezes consequences

No study measures EWC on contradicting fact updates with a stale-answer metric. That is the central gap, and the indirect evidence shows why it matters. Clemente et al. trained GPT-2 on CounterFact. Learning 1,000 *non-contradicting* facts kept about 93% of old knowledge, while just 10–100 *contradicting* facts destroyed "up to 80% of unrelated knowledge." Routing updates to low-importance "plastic" neurons, a hard-mask analogue of EWC, gave >98% retention for non-contradicting updates. For contradicting ones every targeting strategy was catastrophic, and plastic routing sometimes did *worse* than plain fine-tuning (Clemente et al., "In Praise of Stubbornness"). The 2022 continual-knowledge-learning benchmark, which defines outdated knowledge as information "conflicting between D0 and D1," found that RecAdam (an *annealed, isotropic* quadratic pull toward pretrained weights) slightly *helped* updated facts (12.55 vs 10.17 EM). Rehearsal of the initial corpus nearly halved them (6.49) (Jang et al., ICLR 2022). On 114 monthly Common Crawl dumps, loss- and optimizer-only methods "insufficiently prevent forgetting and plateau" (TiC-LM, ACL 2025). Importance protection thus works for consistent additions, fails for contradictions, and the one regularizer that did not hurt updated facts was the one whose stiffness was removed by the end of training.

The mechanism follows from the KL identity. If the stale triple (s, r, o_old), or text mentioning s, is in the Fisher data, EWC's penalty locally approximates "keep p(o_old | s, r) where it was." That is the output anchor NeuroGenesys has already measured. In the toy, KL to the frozen model on the new-fact inputs blocked learning (5% immediate accuracy) (findings ledger 4.5), and rehearsal entrenched stale answers (catalog A2, A4). This report calls the effect intransigence by inheritance. Averaging over D_old and the diagonal approximation dilute it, but they do not change its direction. The cost of a contradicting update scales with the stale fact's share of Fisher mass on the parameters the update needs.

The more damaging prediction is bypass rather than blockage. When high-Fisher coordinates are expensive, gradient descent finds the new answer through cheap low-Fisher ones, and LLMs have plenty of those. The new fact then wins on the trained phrasing while the old fact's high-Fisher circuit stays intact, still reachable through paraphrases, multi-hop paths or persona shifts. That is the "belief inertia" pattern AGM-Bench found in context, reproduced in weights. It also fits the 2026 finding that forgotten facts still sit in the log-probabilities (Logical coherence of LLM world models). And it fits unlearning, where TOFU-1B extraction strength rebounded by 0.09–0.64 after one retain-only epoch for gradient methods, and RMU, the method that forgot most deeply, recovered about 10× as much as forget-specific pruning (FRAG, EMNLP 2026). Overwritten old facts may be just as recoverable. A cheap relearning probe belongs in every update experiment.

Ripple effects get worse for the same reason. Across editors, direct edits typically exceed 90% accuracy while ripple success rarely exceeds 50% (RippleCoT; Cohen et al., TACL 2024). GradSim, the cosine between the edited fact's gradient and a related fact's gradient, correlates strongly with ripple success, and its failure cases (negation, over-ripple, multilingual) sit at very low GradSim (Qin et al.). Under an EWC penalty one step is $\Delta\theta \approx -(H + \lambda F)^{-1} g_e$, so the change in a consequence r's log-odds is about $-g_r{}^{\top}(H + \lambda F)^{-1} g_e$. At large λ this becomes a Fisher-preconditioned GradSim. Facts about one entity share the entity-representation directions, and those are high-Fisher precisely because the entity is common in the old data. EWC suppresses that shared component and keeps the edit-specific one. The prediction is that as λ rises, direct efficacy falls slowly and consequence accuracy falls fast, so the incoherence gap widens with stiffness. The opposite outcome is possible if the relation-specific path is the stiff one, because then EWC would force the edit through the shared subject representation and increase propagation. Which outcome holds is empirical and probably layer-dependent. No paper reports ripple, MQuAKE or portability metrics as a function of any stiffness parameter.

Entrenchment exposes the deepest mismatch. In AGM theory, epistemic entrenchment is a total preorder over *sentences*. When revision forces inconsistency, the least entrenched beliefs yield first, and every revision function corresponds to such an ordering (Coniglio et al.; Williams, IJCAI 1997). EWC orders *parameters*, each storing many facts in superposition. A rank-one update's effect on another fact depends on key overlap ⟨k_new, k_old⟩ (arXiv 2604.15574). EWC still induces an ordering over propositions without saying so. The cheapest penalized change that flips belief p, with margin m_p and gradient g_p, costs about $E_F(p) \approx m_p^2 / (g_p^\intercal F^{-1} g_p)$, the Fisher distance to the decision boundary. That score mixes confidence with entanglement: a fact becomes entrenched because it shares heavily used circuits, not because it is well supported. Entity hubs would be over-entrenched, which connects to FACTPROP's finding that hub facts are the most corrupted by updates (Zhang et al. 2026). The induced ordering probably breaks AGM's *dominance* condition, which requires that if A entails B, B is at least as entrenched as A. Two-hop consequences are rarely stated in training data and are computed by composition, so they plausibly sit on lower-Fisher paths than their premises. Keeping A then does not protect B, and contracting A does not withdraw B. Stiffness is also set by frequency, so a frequently repeated falsehood is maximally entrenched. "Belief entrenchment" in the LLM literature means something else, a behavioural confirmation bias measured with a martingale score (arXiv 2512.02914). No KR or ML paper derives proposition-level entrenchment from the Fisher or tests it against AGM postulates.

Release stiffness selectively, gate it on evidence, and re-stiffen after revision

Stiffness has an exploitable property: the empirical Fisher is additive over examples, $F = \Sigma_x \nabla \log p(x)^2$. If a lab stores per-fact (or per-cluster) Fisher contributions F_f, or a sparse top-k sketch of them, then superseding fact f permits exact de-consolidation: $F \leftarrow F - F_f - \Sigma_\{d \in \text{derived}(f)\} F_d$. This removes exactly the stiffness the stale belief and its consequences contributed, and nothing else. It is the stiffness-space twin of NeuroGenesys's rehearsal-pool pruning, which removed stale facts from the *data* and gained +22.8 pp on updates at 1.5B (A4). Existing tools release stiffness either globally or destructively. Online EWC decays everything by γ. RecAdam anneals everything to zero. SSD computes Fisher on the forget and retain sets and dampens parameters whose forget-importance exceeds retain-importance (Foster et al., AAAI 2024). Fisher Forgetting injects importance-scaled noise (Golatkar et al., CVPR 2020), and an LLM Fisher-removal variant reportedly erases more than gradient ascent at lower utility cost (arXiv 2403.10557). CoME unlearns the outdated fact while inserting the new one and concludes that targeted removal of the old fact is crucial for conflicting edits (CoME, NAACL 2025). The search found no paper that subtracts per-example Fisher contributions to release superseded knowledge, and no use of the term "unconsolidation."

Release and erasure are separate dials, and conflicts probably need both. Ledger subtraction only stops *protecting* the old fact, and the new-fact gradient does the overwriting. SSD and CoME actively *erase*. The Stubbornness result argues that erasure must be targeted, because untargeted dissonant updates wreck unrelated knowledge. Two cautions come from the evidence. The lab found that toy directional suppression did no better than a random control (A7), so any targeted release must beat random release at matched Fisher mass. And the FRAG ablation found that Fisher importance has the wrong sign for predicting which unlearning resists relearning (Spearman +0.57 across 15 checkpoints, where negative is correct). Activation importance $|W| \cdot |X_f|$ scored −0.92 and Hessian$\cdot W^2$ −0.79 (FRAG). The squared-gradient Fisher measures loss sensitivity, not where a fact lives in a form that resists re-learning. It is therefore a weak fact locator, and a Fisher ledger should be checked against an activation-importance ledger.

Neuroscience supplies the *timing* rule that a ledger lacks. A consolidated memory becomes labile again when reactivated and must restabilize (Nader, Schafe & LeDoux, Nature 2000). Destabilization requires

a prediction-error signal, positive or negative (Exton-McGuinness, Lee & Reichelt 2015), and a 2022 study found that signal necessary but not sufficient (Sci. Rep. 2022). Attractor-network models have the retrieved trace go labile and absorb new information (Nowicki, Verga & Siegelmann). In EWC terms, retrieval is running the incoming fact's query, and prediction error is the model confidently predicting o_old when the input says o_new. That is the dissonance signal, which a linear classifier on internals detects at 94.7–99.5% accuracy (Clemente et al.). Lability is temporarily zeroing that fact's ledger Fisher, and restabilization is recomputing it on the updated belief and its consequences. Non-dissonant facts would be learned under full stiffness, which is exactly where importance protection already works. Metaplastic binarized networks implement the state variable this needs, a per-weight stiffness that rises with consolidation (Laborieux et al.). No deep-learning method found implements prediction-error-gated release followed by re-stiffening.

The last lever is *which* beliefs earn stiffness in the first place. Write-time gates exist for agent memory: ConsistencyGate admits a memory only if five samples support it (aggregator page), and FGSE commits only falsification-tested lessons (arXiv 2611.14326). Gates also exist for self-training data (CREST, NAACL 2025). Nobody gates *stiffness*. Computing the Fisher only over beliefs that pass a check would leave unverified beliefs cheap to revise, which is the AGM property that weakly supported beliefs yield first. Gating on whole clusters (premise plus consequences) would stiffen premises and consequences together and partly repair the dominance violation. The gate's *type* is the real variable. TrueTeacher found that self-verification filtering raised label accuracy without helping downstream (arXiv 2305.11171). The lab's coherence report showed that unanchored consistency converges to self-consistent error. A self-consistency gate therefore stiffens coherent errors, and only an external anchor (KB or provenance) avoids that.

NeuroGenesys's own results already predict how EWC behaves

The lab's selection work explains EWC's best-documented failure. On Split-CIFAR-100, frozen per-task experts were right 96% of the time within a task but routed inputs to the right task only 36% of the time. Selection carried $S = 0.91$ of the between-method variance, and as tasks grew from 5 to 20, selection error rose from 0.534 to 0.788 while within-task accuracy *rose*, "so the whole decline is selection" (findings ledger 8.15, 8.18). EWC is the purest *within-task* protector in the literature. It keeps each old function near its anchor and contains no term for discriminating between tasks, which is why it sits at chance class-incrementally. The lab's result and the three-scenarios table describe the same fact from two sides: protecting old functions does nothing for the selection problem, and the selection problem carries most of the loss. Any EWC experiment the lab runs should report it as a within-task baseline, not a competitor to selection.

Forgetting-as-selection maps onto stiffness directly. Choosing the Fisher dataset is choosing what survives, and Scholz et al. show the cost of choosing badly. The Fisher ledger turns retirement into a stiffness operation, and the consistency gate turns admission into one. The lab's reliability findings say which gates can work. A reject threshold worked only on a score *independent* of the expert being judged (+0.545 at 71% coverage on a density margin, against +0.114 on the expert's own confidence), and set-level decisions over related inputs cut selection error from 0.095 to 0.002 (findings ledger 9.3, 9.9). Stiffness gated on the model's own confidence is the stiffness analogue of the failed threshold. Stiffness gated on KB verification or provenance is the analogue of the working one. Cluster-level gating is the set-level decision. If the induced-entrenchment score E_F proves predictive, it is a natural per-fact "revision cost" signal for the reliability product's `selection/v1` events. A fact's revision cost belongs next to its selection reliability, since both price how much a decision will cost to undo.

For consequence-consistent updates, the analysis predicts that adding EWC to CCU's in-weights arms would *widen* the gap to in-context editing, by freezing consequences more than direct edits. CCU's MEMIT and AlphaEdit arms already contain an EWC-like Kronecker prior ($C_0 \otimes I$). That suggests one untested variant, "Fisher-MEMIT," which replaces the identity output factor with the output-gradient covariance so that preserved keys are weighted by how much predictions depend on them, not by how often they occur. CCU's pre-registration (A13, H1–H15) must not be contaminated, so none of the experiments below uses CCU test splits. Any run on CCU streams is logged as a §17 exploratory deviation. Finally, the lab's measurements show that stiffening is real only for methods that consume subspace (O-LoRA and history-aware projection) and that anchoring alone collapsed from 92.1% to 20.4% retention over 100 stages (findings ledger 4.9, 6.1). Summed EWC predicts that plasticity in shared directions decays like 1/T, and online EWC predicts that constraints older than about $1/(1-\gamma)$ updates fade exponentially. Neither prediction has been measured in an LLM.

Six experiments a two-machine lab can run without repeating itself

Each experiment targets a mechanism specific to per-parameter stiffness. None re-tests rehearsal entrenchment, pool pruning, KL anchoring alone, consequence expansion, low-confidence targeting or any CCU hypothesis. They are also distinct from the coherence report's five tests. Its entrenchment ablation chooses which belief yields during *explicit* repair, while E4 below measures the ordering that weights impose on *implicit* revision. Its kernel × KB logging is passive, while E3 intervenes on stiffness. Its chewing test gates *data*, while E6 gates *stiffness*. All arms use the SAMPLE Fisher by default with per-layer normalization, ≥3 seeds (5 in the toy) and thresholds above the roughly 6 pp MPS noise floor. EWC arms run under AdamW and SGD-momentum, because of the Adam interaction above. Every update experiment adds a relearning probe (one retain-only epoch) to separate overwriting from suppression.

#	Question	Setup and compute	Primary metric and kill rule
E1. EWC vs KL anchor, with estimator diagnostics	Does weight-space anchoring protect old facts that neither the Fisher set nor the KL sample contains, and what does it do to contradicting updates?	Pythia-2.8b (on disk) now; Qwen2.5-1.5B LoRA once weights are re-downloaded. Arms: KL-to-base; EWC with Fisher on old-fact probes; EWC with Fisher on generic text; λ log-swept 10^2–10^9. Includes SAMPLE vs EMPIRICAL vs BATCHED and a quadratic-validity curve (true old loss vs EWC prediction against $\|\theta - \theta^*\|$). M2 Max, about a day per model.	Retention on *held-out* old facts; the efficacy–stale-rate frontier on contradicting updates; a bypass index (old-object log-prob on held-out paraphrases ÷ on the trained phrasing). Kill if the EWC frontier sits within 6 pp of KL's everywhere.

#	Question	Setup and compute	Primary metric and kill rule
E2. The Fisher ledger	Does subtracting a superseded fact's own Fisher (plus its KB-derived consequences') restore updates and cut stale answers better than global or random release?	Toy KB transformer (CPU) with an exact ledger, then 1.5B with a top-k sparse ledger. Arms: SFT, static EWC, online EWC (γ swept), RecAdam annealing, ledger, ledger + derived, random release at matched mass, ledger + SSD dampening, and an activation-importance ledger.	Stale-direct, stale-paraphrase and stale-2-hop rates at matched retention. Kill if ledger + derived beats matched-mass random release by < 5 pp on stale 2-hop in the toy.
E3. Does stiffness decouple edits from consequences?	As λ rises, does consequence accuracy fall faster than efficacy? Is $g_r^\top F^{-1} g_e$ a better per-pair predictor than GradSim?	Pythia-160m/2.8b and the toy; RippleEdits and MQuAKE edits from `~/.cache/neurogenesys-ccu/ccu/raw` (not CCU splits); single and 50-edit batches. Causal arm: zero the Fisher on the top-k entity-shared gradient subspace. CPU/MPS.	Slope of the incoherence gap (efficacy − consequence accuracy) against log λ; per-pair AUROC of both coupling scores. Kill if the slope is < 3 pp per decade in both models.
E4. Induced-entrenchment audit	Does $E_F(p) \approx m_p^2/(g_p^\top F^{-1} g_p)$ respect AGM dominance, and does it predict which side of a linked pair yields under a contradiction?	Gradients only for the audit; then train a small contradiction set. Pythia-160m/2.8b; KB triples with `ccu_expansion`-style rules. Compare with margin alone, entity degree and BM25 support. A Hessian-vector-product check on a subset guards against empirical-Fisher error.	Dominance-violation rate; Spearman between each score and observed yield order. Kill if E_F beats margin alone by $\Delta\text{Spearman} < 0.05$.

#	Question	Setup and compute	Primary metric and kill rule
E5. Prediction-error-gated reconsolidation	Does opening a local lability window only for dissonant facts, then re-stiffening on the revised belief, beat uniform EWC, no EWC and always-labile training?	Toy, then 1.5B; 70:30 novel:dissonant stream. Gate on old-answer confidence or a linear dissonance probe. Controls: random gate at matched rate, gate without re-stiffening, lability scaled by E_F. Reuses the E2 ledger.	Collateral loss on unrelated facts per dissonant update; stale rates; the revised fact's own later revisability. Kill if the PE gate beats the matched-rate random gate by < 5 pp on collateral loss.
E6. Consistency-gated consolidation	If the Fisher is accumulated only over beliefs passing a check, are false beliefs still correctable later? Which gate type minimizes entrenched errors?	Toy KB, then 1.5B. Phase 1 stream is 25% false; Fisher over all facts, KB-verified facts, self-consistent clusters (five samples), or random at matched size. Phase 2 delivers gold corrections plus unrelated facts.	Entrenched-error rate (false beliefs surviving correction) and retention of true facts. Kill if the KB gate beats all-facts by < 5 pp; the secondary prediction is that the self-consistency gate does worse than KB whenever the false facts are internally coherent.

The sequencing follows the hardware. E3 and E4 need only inference or small training runs on CPU and Pythia, which is already on disk, so they can run while CCU Phase 1 and the CIFAR queue hold the MPS lock. E1 needs Qwen2.5-1.5B re-downloaded (into `~/.cache`, never `/private/tmp`) and should come next, because it gives the lab its first EWC baseline and the λ and estimator settings the others reuse. E2, E5 and E6 share the ledger code, so it should be built and validated against the exact toy ledger first. The per-fact sketch's memory at 1.5B is unmeasured: a LoRA-r16 diagonal is only megabytes per fact, but derived-fact clusters multiply it. Rented GPUs earn their cost in one place: replicating whichever of E1, E3 or E5 succeeds at Llama-3.1-8B. Full-parameter EWC there needs about 48 GB of extra state, so it requires an 80 GB A100/H100, or the ΔW-space EWC-LoRA route on a 48 GB card. A λ sweep at 8B with three seeds is a few GPU-days on that route. The Gemma2 study's full-parameter sweep cost about 24 hours on 8×H100, which is the budget to avoid.

Conclusion

The useful way to read EWC is as a function-space KL anchor on old data, compressed into a diagonal of per-parameter prices, rather than as a forgetting fix that underperforms at scale. That reading explains the 2023–2026 record without invoking scale. EWC protects whatever its Fisher set exercises, fails where the problem is selection rather than retention, and inherits every weakness of output anchoring when old and new beliefs conflict. It adds one weakness of its own: it prices *parameters*, so revision takes the cheapest path rather than the logically correct one. The consequence for coherence is a specific, testable failure, bypass. The new fact lands beside the old trace, the consequences stay with the old

trace, and the incoherence gap grows with stiffness. Static EWC is therefore not a coherence tool, and adding it to update pipelines unexamined is likely to raise stale-paraphrase and stale-multi-hop rates while looking good on direct-edit and retention metrics.

The same structure makes EWC unusually programmable. Because the Fisher is additive over examples, stiffness can be bookkept per proposition. It can be released exactly when a belief is superseded, opened only when a prediction error signals real dissonance, and granted only to beliefs that pass an independent check. That turns a blunt regularizer into the stiffness-space version of the selection loop the lab's coherence report described: admission, retirement and entrenchment become operations on one ledger. Whether this works is entirely open. Its most valuable first output is diagnostic rather than a better method: the first measurement of the ordering over propositions that weight-space stiffness actually imposes, and of whether it respects the logical dependencies a coherent world model needs. If E_F violates dominance as predicted, any method that consolidates knowledge into shared weights has to correct the entrenchment ordering explicitly. Uniform consolidation will not keep a model coherent.

Published by NeuroGenesys (AW3 Technology, Inc.). Canonical version: https://neurogenesys.si/research/elastic-weight-consolidation-review