Beyond Damage: Does the KL Forgetting Law Predict Recovery After Fine-Tuning?

Yuanning Hui
Department of Computer Science, Princeton University
hh9077@princeton.edu

Abstract. As models become larger and self-updating, post-training on new data can degrade earlier learned capabilities, a phenomenon known as forgetting. Current defenses often require additional compute for replay or retraining, or add step-size gating and parameter-isolation machinery. Yet apparent forgetting may be reversible: a brief intervention could restore the old capability without another full training cycle. We ask whether forward Kullback–Leibler (KL) divergence—the predictor in the KL forgetting law [16]—also predicts recovery, or whether another metric like sub-space geometry is more informative. Across controlled information load, capacity pressure, and full fine-tuning versus low-rank adaptation (LoRA), we apply a task-B-constrained recovery protocol, compare never-knew-A and exposure-matched controls, and evaluate behavioral, representational, and geometric predictors. The study will distinguish when forgetting requires costly retraining from when a cheaper path back to the old capability remains, giving researchers an empirical insight into how to most efficiently mitigate forgetting.

1 Introduction

Modern machine learning has advanced by scaling model size, data, and compute [6, 10]; increasingly, it also seeks models that update continually from new data, feedback, and rewards. It is long known that each post-training cycle creates a stability–plasticity problem: learning task B can degrade capabilities learned on task A [8, 12, 16]. Current remedies either add system complexity through regularization, parameter isolation, gating, and low-rank adaptation (LoRA) [7, 11] or consume additional compute through replay or retraining when they measure a terminal drop in task-A performance. However, in the second case, there is a large defect in the measurement: the same drop in performance could represent a genuine loss of information or a mask or rotation of recoverable skills. This one simple delta gives us no distinguishment between the 2 modes of forgetting, the second of which has old-task labels that remain decodable [3] and rapidly reversible [20].

Specifically, Shenfeld et al. [16] show that forward KL divergence from the base policy on new-task inputs predicts immediate forgetting, a relationship known as the KL forgetting law. Yet predicting *how much* performance is lost does not reveal *what* remains: the same decline may reflect **access loss**, in which a small intervention restores surviving structure, or **content loss**, in which prior exposure provides no measurable recovery advantage within a declared budget. Checkpoints with similar task-A forgetting and new-task KL may therefore recover differently—a distinction relevant to both continual learning and unlearning [5, 19].

Recoverability has direct computational consequences. Treating a recoverable capability as destroyed can mean restoring an earlier checkpoint and rerunning an entire post-training stage, duplicating that stage's accelerator-hours and wall-clock time. Preventing every regression in advance can instead require replay data or increasingly elaborate gates. A reliable recovery signal could identify when a short intervention can restore task A while preserving task B, reserving content-bearing replay or full retraining for cases that actually require it. This turns forgetting from an after-the-fact score into a decision about the cheapest valid intervention.

Thus, we test whether the KL forgetting law extends from immediate forgetting to recovery. After replicating the published relationship, we vary task-B information load and capacity pressure and compare full fine-tuning with rank-8 LoRA. Each checkpoint receives the same task-B-constrained recovery protocol and is compared with never-knew-A and exposure-matched controls. We then benchmark KL against behavioral compression measures, linear probes, representational drift, update-subspace geometry, and sharpness under grouped held-out evaluation.

Research questions.

RQ1: Prediction. Does new-task KL predict recovery, and do information or geometry measures improve prediction beyond KL and immediate

forgetting?

RQ2: Information. Among checkpoints with similar KL and forgetting, does recovery vary with task-B information load, particularly under capacity pressure?

RQ3: Variation. At similar behavioral endpoints, does recovery vary with update parameterization and task-subspace geometry?

Hypotheses. We expect KL to explain recovery variation associated with immediate damage, while information and geometry measures add held-out predictive value (**H1**). Novel associations should reduce recovery advantage primarily under capacity pressure; a random-target main effect without an information-load × capacity-pressure interaction will not count as evidence of capacity displacement (**H2**). Finally, update parameterization should retain predictive value at similar endpoints, with LoRA rank effects attenuating when task-gradient subspaces are well separated [17] (**H3**).

Contributions. The study provides: (i) a direct test of whether the KL forgetting law predicts recovery; (ii) a controlled information-load design separating novelty, conflict, and capacity pressure; (iii) a task-B-constrained recovery assay with never-knew-A and exposure-matched controls; and (iv) a grouped held-out benchmark of KL against information and geometry measures. Together, these contributions provide a principled way to predict when learned knowledge can be recovered rather than retrained.

2 Background and Related Work

From interference to recoverability. Catastrophic forgetting was originally formulated as catastrophic interference [12]: gradient updates for a second mapping can destroy performance on the first because the task-B objective contains no term valuing task A. Classical remedies—regularization [11], rehearsal, and parameter isolation—are often evaluated primarily by the resulting drop in task-A performance. That view became incomplete when Davari et al. [3] showed that a fresh linear classifier can still decode old-task labels after the deployed classifier fails, and Zheng et al. [20] showed that early fine-tuning can produce severe but rapidly reversible behavioral forgetting. Terminal accuracy alone cannot reveal how much information survives in usable form.

The KL law. Shenfeld et al. [16] establish that new-task forward KL from the base policy predicts the immediate magnitude of forgetting, supplying the testbed, metrics, and sweep strategy we extend. Their distillation result—a supervised fine-tuning (SFT) student matching a reinforcement learning (RL) teacher's learning–forgetting trade-off—supports a final-policy account of *immediate* forgetting on measured distributions, but does not establish that recovery dynamics must match.

Reversibility and unlearning. Xu et al. [19] formalize reversibility under bounded relearning and find that representational diagnostics distinguish regimes within unlearning; Fan et al. [5] motivate sharpness as a defense against relearning attacks. Evaluation work shows that static output metrics vary in faithfulness [4] and that adaptive probing recovers behavior missed by fixed prompts [15]. We transfer this evidence cautiously: unlearning objectives suppress behavior deliberately, so predictors validated there may fail under ordinary sequential fine-tuning. Testing that transfer is part of the contribution.

Compression as measurement. For targets $D = \{(x_i, y_i)\}$, define the behavior-write quantity

$$W(D; \theta_0, \theta) = \sum_i \Big[\log_2 p_\theta(y_i \mid x_i) - \log_2 p_{\theta_0}(y_i \mid x_i) \Big]. \tag{1}$$

Here, $W > 0$ means that training increased the targets' log-likelihood, so we call W a *behavioral compression gain*. For independently randomized associations, unseen pairs cannot be predicted above chance without leakage; compression gain on the trained pairs is therefore evidence of memorization [13]. On natural text, however, W mixes memorization, generalization, and convention shifts, so it is not a census of bits in the weights. Following the excess-statistic logic of Tan et al. [18], we define

$$B_{\text{new}} = W(D_B^{\text{train}}; \theta_0, \theta_B) - W(D_B^{\text{held}}; \theta_0, \theta_B), \tag{2}$$

$$L_{\text{old}} = -W(D_A; \theta_0, \theta_B), \tag{3}$$

where D_B^{held} is a size-matched set of untrained associations. Thus, positive B_{new} denotes task-B training-specific compression gain, while positive L_{old} denotes task-A behavioral compression loss. Relatedly, Huang et al. [8] show that self-improvement cannot create information absent from the model (sharpening), which constrains epistemic novelty but does not force $W(D) = 0$ for self-generated data.

Low rank constrains the update, not the mechanism. LoRA [7] restricts updates to $W = W_0 + \frac{\alpha}{r} BA$ with $\text{rank}(\Delta W) \leq r$; SFT specifies a loss and target source. These are orthogonal factors and must be crossed, not conflated. Biderman et al. [2] find that low-rank LoRA often learns less and forgets less

than full fine-tuning—so lower forgetting can be under-adaptation rather than preservation—while Tan et al. [18] show that writable bits depend on placement and need not grow smoothly with rank, and Steele [17] links rank effects on forgetting to principal angles between task-gradient subspaces. None of this establishes that low rank is intrinsically "access loss" or that full fine-tuning is "overwrite"; we use "rotation" only as shorthand for an access/reorientation hypothesis [9, 20].

3 Methodology

3.1 Experimental design principles

Divergent compute-optimal prescriptions [6, 10] and their reanalyses [1, 14] show that an apparent empirical law is defined only on a declared resource surface. We therefore tune learning rates and budgets independently for each parameterization, restrict matched comparisons to common support, report writable capacity at a declared budget, and evaluate predictors out of sample on held-out tasks, mappings, and eventually model scales.

3.2 Phase 0: replicating the KL law

We reproduce the ParityMNIST/FashionMNIST setup of Shenfeld et al. [16]: a three-layer multilayer perceptron (MLP; 785→512→256→10) with a task-indicator input, jointly pretrained on 500 examples per task, then fine-tuned on ParityMNIST under SFT and on-policy (1–0 REINFORCE) objectives across 15 log-spaced learning rates (3×10^{-6} to 10^{-3}), two schedules, one or two epochs, and at least three seeds. We cross objective with update parameterization: full fine-tuning vs. all-linear LoRA with $r = 8$, $\alpha = 8$, zero dropout, and frozen base weights and biases. The $\alpha/r = 1$ scaling and zero-initialized adapter make the initial deployed function exactly θ_0. Learning rates are swept *separately* for each parameterization, since a shared "best" rate would confound update geometry with under-training. At every checkpoint we record task-B accuracy and KL, task-A forgetting and label code length, effective-weight distance, representational similarity, and fixed and fresh task-A linear probes. ParityMNIST serves only as a replication harness; its ten familiar labels cannot create controlled information load, so Phase 0 supports no capacity claims.

3.3 Phase 1: paired information-load experiment

Checkpoints. Using a small decoder model, we (1) start from $\theta_{\text{pre-A}}$, which has never seen synthetic task A; (2) teach task A (synthetic key–value facts) to produce base checkpoint θ_0; (3) fine-tune θ_0 on task B to obtain θ_B; (4) apply the identical task-B procedure to $\theta_{\text{pre-A}}$ to obtain a **never-knew-A control**; and (5) before the same task-B procedure, train a second control on unrelated synthetic facts matched to task A in format, sample count, tokens, and optimization budget. Faster recovery from θ_B than from both controls provides evidence consistent with retained task-A-specific information; it does not prove where or how that information is stored.

Randomized information load. From a base-derived set of prompts and target strings, we select a prespecified fraction $\lambda \in \{0, 0.25, 0.5, 1\}$ of examples and apply a random derangement to their target assignments, preserving the prompt set, exact target strings, length and token-frequency marginals, architecture, optimizer, loss, dataset size, and exposures. Reassignment introduces novelty *and* conflict with the base prior, so a paired **fresh-key cell**—synthetic keys outside all templates, filtered for no systematic base preference among candidate values, and matched on all marginals—provides a contrast designed to separate novelty from override pressure. Fully random strings serve as capacity calibration; a deterministic formatting cell serves as a low-information positive-learning control. Initial loss and gradient statistics are measured in every cell.

Capacity pressure. Per parameterization (and per LoRA placement/rank ablation), we estimate writable capacity via the random-string protocol of Morris et al. [13], Tan et al. [18] and define

$$\rho = \frac{H_B^{\text{novel}}}{C_{\text{write}}}, \tag{4}$$

where H_B^{novel} is the entropy of novel task-B associations and C_{write} is writable capacity at the declared budget, both measured in bits under the same tokenization and scoring convention. We run several ρ bands below and around one and propagate bootstrap uncertainty in C_{write} to ρ. Writing novel facts into a largely empty parameterization does not imply that old facts were displaced; capacity claims require this axis.

Common support. Learning rate and steps are swept independently within every $(\lambda, \rho, \text{parameterization}, \text{conflict})$ cell to retain an overlap region in new-task performance, forward KL, and immediate old-task forgetting. Matched comparisons are made only inside this overlap; extreme runs are shown in descriptive curves but excluded from matched-effect estimates. A secondary algorithm comparison—SFT, 1–0 REINFORCE, and Group Relative Policy Optimization (GRPO)—is restricted to task-B variants with adequate support under θ_0: RL's failure to guess an unguessable code reflects coverage, not "writing no bits."

3.4 Checkpoint measurements

Before any recovery intervention, every checkpoint receives the battery in Table 1: forward/reverse KL on fixed task-B prompts; immediate task-A forgetting; training-specific compression gain B_{new}; base-association conflict (initial loss and probability margins); old-task code-length loss L_{old} across canonical, paraphrased, and content-token variants; fixed and fresh linear probes (readout stability vs. linear decodability); layerwise CKA/CKNNA; realized update rank/spectrum and A/B gradient-subspace principal angles; normalized sharpness; and L_2/Fisher-weighted baselines connecting to the published predictor ablation.

3.5 Recovery protocol

"Irreversible" is not an observable finite-budget property; we use **budget-resistant** and define recovery relative to declared data and compute. Each terminal checkpoint and control receives three arms: **cue-only** (task-format examples with no task-A facts; success beyond both controls provides evidence consistent with access loss), **direct relearning** (a stratified task-A subset, interpreted against both controls), and **ceiling** (the largest prespecified budget, separating slow recovery from an early plateau). No arm explicitly rolls back task B; disabling a LoRA adapter is logged as a sanity bound but excluded because it restores A by discarding B. Recovery counts only while task B stays within tolerance of its post-fine-tuning score:

$$S_B(\theta_{A \leftarrow B}(b)) \geq S_B(\theta_B) - \epsilon_B, \tag{5}$$

with ϵ_B set from evaluation noise and accompanied by sensitivity analyses; violations censor the budget, and the joint (S_A, S_B) Pareto frontier is reported so "recovering A by forgetting B back" cannot count. Each arm is evaluated on a budget ladder (examples, tokens, and steps) with a small prespecified recovery-learning-rate grid.

Primary outcomes are **B-constrained area under the learning curve (AULC)** (area under the admissible task-A curve against $\log(1 + b)$, including the unrecovered checkpoint at $b = 0$); **recovery advantage** (AULC from θ_B minus AULC from each control, with task-A-specific retention requiring an advantage over both); the **joint recovery frontier**; the **recovery ceiling**; and censored C_{90} (the minimum admissible budget that restores 90% of the score lost between θ_0 and θ_B).

3.6 Ablations

The primary full-vs.-r=8 contrast changes both dimension and factorized shape, so staged ablations (Table 2) separate rank ($r \in \{2, 8, 32\}$), placement (all-linear vs. MLP-only vs. attention-only), trainable dimension (matched-dimensional random subspace), learning level (independently tuned grids at matched task-B performance), task-subspace angle (high- vs. low-angle task pairs, testing whether rank stops mattering for recovery when angles are high), and implementation details (bias, α/r). A step-resolved **temporal dissociation** analysis logs task-A performance, code length, probes, and KL at log-spaced early steps: spurious-forgetting theory predicts behavioral damage *preceding* code-length loss; a rewrite account predicts that they change together.

3.7 Analysis plan

Prediction. All predictor families are evaluated against one shared baseline containing immediate forgetting. We prespecify a baseline-plus-KL model, a full model that adds B_{new}, L_{old}, probes, representation measures, parameterization, update spectrum, subspace angle, and sharpness, and leave-one-family-out ablations of that full model. Hyperparameter tuning remains inside grouped training folds. Models are compared by out-of-sample R^2, rank correlation, calibration, and prespecified equivalence margins for incremental predictive value. Entire task templates and mapping seeds are held out (and later, one model scale); checkpoints from one run never straddle folds, and bootstrap intervals accompany every comparison.

Randomized and conditional comparisons. Primary analyses estimate intention-to-treat effects of the assigned information-load, conflict, capacity-pressure, and parameterization conditions without adjusting for post-treatment KL, task-B performance, or forgetting. We then report common-support contrasts at similar endpoints as descriptive conditional effects rather than total causal effects. The fresh-key vs. familiar-permuted contrast is designed to estimate override pressure beyond novelty when the cells overlap on pre-treatment tokenization, loss, and gradient statistics. The information-load × capacity-pressure interaction is the confirmatory capacity test; the parameterization × subspace-angle interaction and a high-angle equivalence test address RQ3. Mediation through B_{new}, L_{old}, or probes is exploratory because these measures are post-treatment. Falsification checks and stop conditions appear in Appendix C.

4 Planned Results

This section will report (i) the Phase-0 replication of the KL–forgetting relationship across the objective × parameterization factorial; (ii) held-out predictive performance of the prespecified predictor families for recovery AULC; (iii) intention-to-treat estimates and descriptive common-support contrasts for information load, capacity pressure, conflict, and parameterization;

and (iv) ablation and temporal-dissociation analyses.

5 Possible Outcomes and Limitations

Each outcome licenses a bounded conclusion (full map in Appendix D). If KL predicts held-out recovery and prespecified equivalence tests rule out practically meaningful gains from other measures, KL is an adequate *operational* predictor of recovery in the tested regimes—not the universal mechanism of forgetting. An information-load × capacity-pressure interaction in the randomized analysis would implicate capacity-constrained deposition in budget-resistant forgetting, without implying that every lost capability was overwritten. A parameterization effect in the randomized analysis, accompanied by descriptive differences at matched endpoints, would show that the update path carries recovery-relevant information beyond behavioral distance; attributing that effect to rank itself still requires the staged ablations. Probe dissociations admit similarly bounded readings: a failed fixed probe with an intact fresh probe and code length indicates readout disruption with linearly usable information remaining; convergent collapse of the fresh probe, code length, and recovery advantage provides evidence consistent with content-level loss *under the declared assays*, never literal zero information in the weights.

Limitations. Toy experiments can test identifiability and measurement behavior; claims about foundation-model post-training require a smaller confirmatory replication using pretrained decoder-only language models, and claims about unlearning remain transfer hypotheses until tested on unlearning objectives. Code-length measures are behavioral, not physical bit counts; probes require A labels and are scientific diagnostics, not deployable signals; and finite-budget recovery resistance is never proof of absence. Checkpoint statistics also cannot reconstruct facts that are genuinely unavailable, and a no-old-data predictor cannot certify post-intervention success without some evaluation of task A. The realistic deployment target is therefore a triage policy—a treatment class, budget range, confidence estimate, or abstention—not an oracle.

Future work: no-old-data triage. A later extension will test whether signals available without task-A examples can triage recovery interventions and estimate their budgets. We will pursue this extension only if Phase 1 predicts recovery on genuinely held-out task families; Appendix E gives the proposed decision setup.

Future work: recovery algorithms. The present study diagnoses recoverability but does not optimize the recovery intervention itself. Future work should develop methods matched to the predicted failure mode: cue- or readout-level repair for access loss, targeted low-rank counter-updates, selective replay, and constrained checkpoint or adapter interpolation that restores task A without sacrificing task B. These algorithms can be evaluated with the same budget ladders and joint recovery frontier used here. The longer-term goal is to route each checkpoint to the cheapest effective recovery method and reserve full retraining for cases in which lighter interventions fail.

References

[1] Besiroglu, T., et al. (2024). *Chinchilla Scaling: A Replication Attempt*. arXiv:2404.10102.

[2] Biderman, D., et al. (2024). *LoRA Learns Less and Forgets Less*. arXiv:2405.09673.

[3] Davari, M., et al. (2022). *Probing Representation Forgetting in Supervised and Unsupervised Continual Learning*. In *CVPR*.

[4] Dorna, V., et al. (2025). *Accelerating LLM Unlearning via Unified Benchmarking of Methods and Metrics*. arXiv:2506.12618.

[5] Fan, C., et al. (2025). *Towards LLM Unlearning Resilient to Relearning Attacks*. arXiv:2502.05374.

[6] Hoffmann, J., et al. (2022). *Training Compute-Optimal Large Language Models*. arXiv:2203.15556.

[7] Hu, E. J., et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv:2106.09685.

[8] Huang, A., et al. (2025). *Self-Improvement in Language Models: The Sharpening Mechanism*. In *ICLR*. arXiv:2412.01951.

[9] Jin, H., et al. (2026). *Rotation-Preserving Supervised Fine-Tuning*. arXiv:2605.10973.

[10] Kaplan, J., et al. (2020). *Scaling Laws for Neural Language Models*. arXiv:2001.08361.

[11] Kirkpatrick, J., et al. (2017). *Overcoming Catastrophic Forgetting in Neural Networks*. *PNAS*.

[12] McCloskey, M., & Cohen, N. J. (1989). *Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem*.

[13] Morris, J. X., et al. (2025). *How Much Do Language Models Memorize?* arXiv:2505.24832.

[14] Porian, T., et al. (2024). *Resolving Discrepancies in Compute-Optimal Scaling of Language Models*. In *NeurIPS*. arXiv:2406.19146.

[15] Rybak, P., et al. (2026). *REBEL: Hidden Knowledge Recovery via Evolutionary-Based Evaluation Loop*. arXiv:2602.06248.

[16] Shenfeld, I., Pari, J., & Agrawal, P. (2025). *RL's Razor: Why Online Reinforcement Learning Forgets Less*. arXiv:2509.04259.

[17] Steele, B. (2026). *Subspace Geometry Governs Catastrophic Forgetting in Low-Rank Adaptation*. arXiv:2603.02224.

[18] Tan, K., Du, H., & Feng, Y. (2026). *How Many Bits Can an Adapter Write?* arXiv:2607.21351.

[19] Xu, X., et al. (2025). *Unlearning Isn't Deletion: Investigating Reversibility of Machine Unlearning in LLMs*. arXiv:2505.16831.

[20] Zheng, J., Cai, X., Qiu, S., & Ma, Q. (2025). *Spurious Forgetting in Continual Learning of Language Models*. In *ICLR*. arXiv:2501.13453.

A Measurement Battery

Construct	Primary measure	Notes
New-task distribution shift	$D_{\mathrm{KL}}(\pi_0 \,\|\, \pi_B)$ on fixed task-B prompts	Match Shenfeld et al. [16]; also report reverse KL.
Immediate forgetting	Task-A drop from θ_0 to θ_B	Baseline predictor and descriptive matching variable; excluded from total-effect adjustment.
Task-B compression gain	B_{new}: training-specific compression gain on B	Interpretation strongest for permuted/random associations.
Base-association conflict	Initial target loss; base probability margin (assigned vs. original value)	Measured covariate; not a substitute for the randomized fresh-key cell.
Task-A compression loss	L_{old}: code-length increase on A targets	Canonical, max-over-paraphrase, and content-token variants.
Old decodability	Fixed and fresh linear probes on frozen A representations	Fixed probe: readout stability; fresh probe: linear decodability. Require A labels; diagnostics only.
Representation change	Layerwise CKA/CKNNA on A, B, neutral prompts	Candidate predictor, not proof of content.
Update geometry	Realized update rank/spectrum; A/B gradient-subspace principal angles	Declared rank is a constraint; realized spectrum is measured.
Local geometry	Normalized perturbed-loss gap; Hessian trace/eigenvalue estimate	Normalization prespecified (parameterization-sensitive).
Update baselines	L_2, Fisher-weighted L_2, cosine alignment	Connect to the published predictor ablation.

Table 1: Pre-recovery measurements at every checkpoint.

B Staged Ablations

Ablation	Comparison	Question isolated
Primary	Full fine-tuning vs. all-linear LoRA $r = 8$	Does parameterization change recovery at matched endpoints?
Rank	LoRA $r \in \{2, 8, 32\}$	Monotonic or thresholded in rank? After the primary contrast only.
Placement	All-linear vs. MLP-only vs. attention-only	Is capacity/forgetting driven by where updates are allowed? Transformer phase.
Parameter budget	LoRA vs. matched-dimensional random subspace	Factorized low rank vs. trainable dimension.
Learning control	Independent LR/step grids; matched task-B performance	Does "LoRA forgets less" survive equal learning?
Geometry	High- vs. low-subspace-angle task pairs	Does rank stop mattering—for recovery too—at high angles?
Temporal dissociation	Dense early task-B checkpoints	Does behavioral damage precede code-length loss?
Bias/scaling	Frozen vs. trainable bias; fixed α/r	Implementation details masquerading as rank effects?

Table 2: Ablations ordered to control scope.

C Falsification Checks and Stop Conditions

Falsification checks. Held-out random associations must show near-zero training-specific compression gain; fresh keys must show no systematic base preference among candidate values; surface-form code-length variants must agree in sign; predictors must transfer to held-out mappings and tasks (checkpoint interpolation is insufficient); results must survive per-run aggregation; full and LoRA cells must overlap in task-B performance, KL, and immediate forgetting, or the comparison is reported as separate Pareto frontiers rather than a conditional matched-recovery claim.

Stop conditions. The study halts or rescopes if: the KL–forgetting relationship fails to replicate (diagnose before any extension claim); information-load cells lack common support (no descriptive matched effects; redesign the sweep); all runs sit at $\rho \ll 1$ (no capacity-displacement conclusion); code-length results reverse across scoring variants (drop the bits predictor as a content measure, retain the randomized treatment); or the known-A checkpoint and both controls recover indistinguishably even before B (the assay lacks power and is repaired first).

Result	Supported conclusion	Not licensed
KL predicts held-out recovery; equivalence tests rule out meaningful incremental signal	KL is an adequate operational recovery predictor in tested regimes.	KL is the universal mechanism of forgetting.
Randomized information-load × capacity interaction	Capacity-constrained deposition contributes to budget-resistant forgetting.	Every lost capability was physically overwritten.
Randomized parameterization effect plus a conditional difference at similar endpoints	Parameterization carries recovery-relevant information beyond the endpoint.	Low rank = rotation; full fine-tuning = overwrite; rank itself is causal without ablations.
Fixed probe fails; fresh probe and code length intact	Readout disrupted; linearly usable A information remains.	All of A intact, or probe-free access by the deployed model.
Fresh probe, code length, recovery advantage all collapse	Convergent evidence consistent with budget-relative content loss under declared assays.	Weights contain literally zero A information.
Cue-only recovery succeeds with little code-length loss	The pattern is consistent with access loss accounting for much of the behavioral drop.	Internal representations unchanged.
Sharpness generalizes after all controls	Local geometry usefully predicts recovery beyond KL.	Unlearning mechanisms transfer unchanged.
No recovery variation after matching damage	This testbed cannot distinguish recovery mechanisms.	Recovery is always determined by KL.

Table 3: What each outcome does and does not license.

D Interpretation Map

E No-Old-Data Triage (Future Work Detail)

At deployment, the triage system knows the target capability only through a task identifier or coarse metadata, and may use the base checkpoint, fine-tuned checkpoint, task-B data/statistics, and training logs—never task-A examples or labels. Without a task identity, "recover this capability" is ill-posed; without A evaluation data, the system can recommend but not certify. The prespecified treatment menu spans cue/format examples and domain-related examples (no A facts) and direct A examples and full replay (A facts), each with an estimated response surface $R_t(b \mid \theta)$. The *deployable* predictor uses forward/reverse KL on B, task-B dynamics and metadata, weight-update and optimizer statistics, parameterization and realized spectrum, sharpness on B, and drift on neutral data; the *scientific upper-bound* predictor additionally receives immediate A forgetting, L_{old}, and probes, quantifying the cost of forbidding old-task access. Outputs are calibrated ranges with fallback policies (e.g., "access-loss probability 0.87; try 32–64 cue-only examples; escalate to 200–300 direct examples; abstain if neither budget is available"), evaluated by held-out task families and scales, interval coverage, selection regret, and abstention quality.