Policy-Governed Knowledge Retention and Suppression

by Nick Clark | Published March 27, 2026 | PDF

Not all learned knowledge should persist equally. Some knowledge should be reinforced over training. Some should be maintained at current levels. Some should be actively suppressed as new information supersedes it. Policy-governed knowledge retention provides structured control over the lifecycle of learned patterns within model parameters.

What It Is

Policy-governed knowledge retention defines rules for how learned patterns in model parameters evolve over continued training. Reinforcement policies increase the strength of specified knowledge patterns. Maintenance policies preserve patterns at their current influence level. Suppression policies actively reduce the influence of outdated, incorrect, or rights-expired knowledge.

Why It Matters

Without retention governance, continued training produces catastrophic forgetting of previously learned knowledge, or equally problematic, inability to unlearn incorrect or outdated information. Retention policies provide explicit control over what the model remembers, forgets, and updates, rather than leaving these outcomes to the optimizer's implicit dynamics.

How It Works

Retention policies are expressed as constraints on the training process. Reinforcement policies periodically present examples that strengthen target knowledge. Maintenance policies apply regularization that preserves target parameter regions. Suppression policies apply targeted gradient updates that reduce the influence of specified knowledge patterns.

All retention operations are governed by the same admissibility evaluation and lineage recording as initial training. The model's knowledge lifecycle is fully auditable.

What It Enables

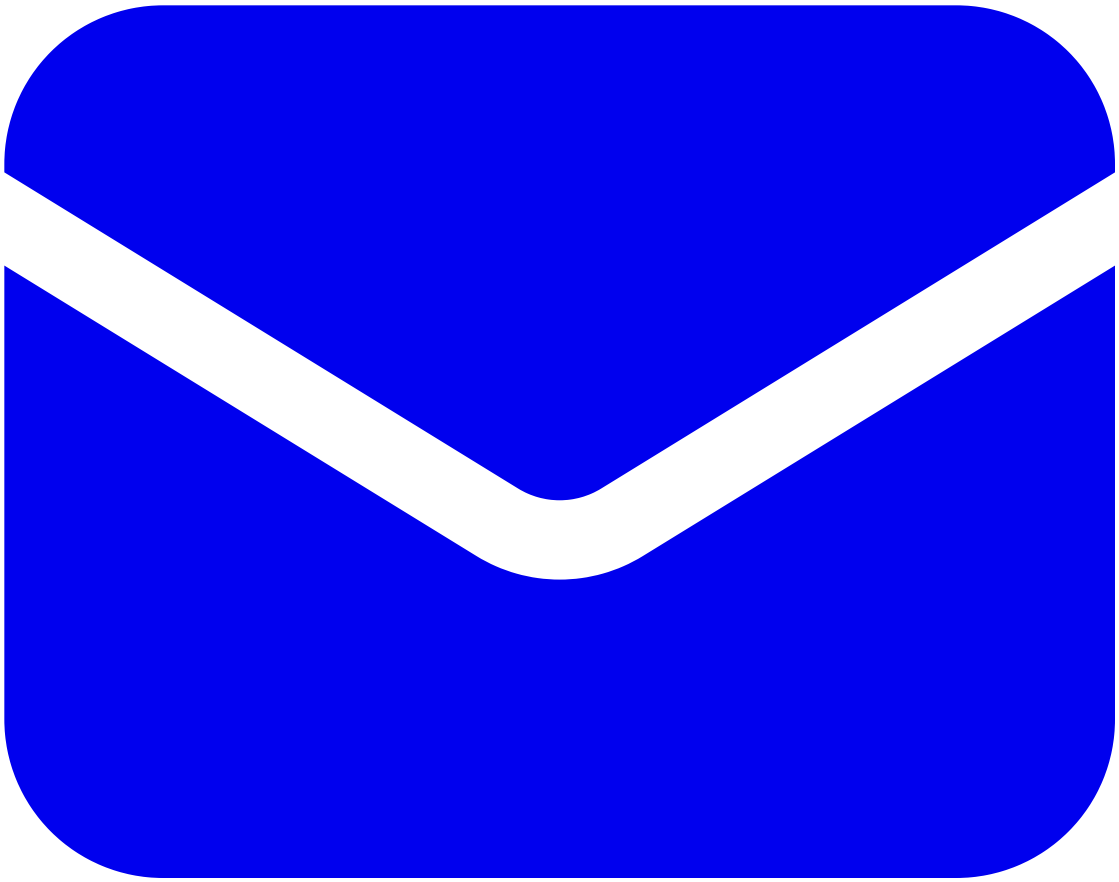
Knowledge retention governance enables models that evolve responsibly over time. Outdated medical knowledge can be suppressed when new evidence emerges. Rights-expired content knowledge can be suppressed when licenses end. Core safety knowledge can be reinforced to ensure it is never degraded by continued training. The model's knowledge becomes a governed, evolving resource rather than an opaque accumulation.

Training Governance All 21 steps →

Govern what the model learns, at what depth, with what provenance.

Primary Technical Disclosure
○ Depth-Selective Training Governance for Machine Learning Systems
Secondary Technical
○ Training Examples as Proposed Semantic Mutations○ Entropy-Band-Indexed Training Depth Profiles○ Depth-Selective Gradient Routing for Governed Training○ Training-Level Memorization Detection○ Differential Privacy Through Depth-Selective Routing○ Governed Fine-Tuning With Verifiable Provenance○ The Training Loop as a Governed Execution Environment● Policy-Governed Knowledge Retention and Suppression○ Provenance-Traceable Training Dynamics○ Curriculum-Integrated Depth Scheduling○ Affect-Modulated Training Depth○ Training-Inference Governance Integration○ Training Governance for Human-Relatable Agents
Applications (General)
○ Rights-Compliant Model Training Through Depth-Selective Routing○ Regulated Industry Model Governance With Provenance○ Training Governance for Medical AI○ Training Governance for Legal AI○ Training Governance for Financial Model Training○ Training Governance for Defense AI○ Training Governance for Educational AI Models○ Training Governance for Creative AI
Applications (Specific)
○ OpenAI's Training Pipeline Has No Depth-Selective Governance○ Constitutional AI Training Lacks Depth-Selective Control○ Stable Diffusion's Training Has No Provenance Layer○ Midjourney Trains Aesthetics Without Governed Depth○ Scale AI Labels Data Without Governing What Models Learn○ Labelbox Manages Annotation Workflows, Not Learning Dynamics○ Snorkel AI Programs Labels but Does Not Govern Gradient Depth○ Weights & Biases Tracks Experiments, Not Learning Governance○ Determined AI Orchestrates Compute, Not Learning Depth○ MosaicML Optimizes Training Efficiency, Not Learning Governance
Training Governance overview →
AQ
deterministic
autonomy

Last updated: 2026-03-03



-
- nick@qu3ry.net
- 72 28 14 36 01

Invented by Nick Clark | Founding Investors: Devin Wilkie