Retro-DARC

Retro-DARC: Function-Preserving Residual-Memory Adapters for Pretrained Language and World Models

Queryable Residual Evidence for Planners — from Layer Deltas to Traces, Latent Actions, and Physics Residuals

Ada Cyborg — July 2026

Preregistration-style program: predictions and baselines were committed to the public repository before the runs that tested them. Raw outputs, harness, per-run records, and the externally timestamped history: the released repository.

Abstract

Standard pretrained Transformers accumulate depth uniformly: every layer's update enters the residual stream with coefficient one, and the stream goes unqueried. We introduce **Retro-DARC**, a family of function-preserving adapters that turns a model's own layer-to-layer updates — its *innovations*, in the filtering-theory sense — into typed, queryable, causally auditable memory. The adoption core, **Retro-DARC-Lite**, inserts into an existing causal LM as an exact no-op: a zero-gated, null-reserving read over recent layer deltas that provably preserves the model's function at insertion and trains gate-first. Residual-Aligned routing allocates memory budget by evidence rather than uniform time or depth, and **Retro-DARC-X** carries the same interface to world/action planners, where one typed bank holds observation, action, tool, latent-cause, trace, spatiotemporal, physics-residual, and failure/verification deltas.

We ground the design in a measured, replicated record on public pretrained checkpoints: **bitwise logit identity at insertion** on three open models; delta-key addressing that stays near ceiling under noise and int4-style corruption while output-key addressing collapses with depth; a registered twin pretrain in which from-scratch reads at full initial gate hurt on every seed — pure gate-initialization shock, by factorial ablation — while the zero-gated retrofit added its full increment even on an architected-retrieval base; and, at matched parameters on a converged public checkpoint, **depth reads that beat LoRA** and support an audit weight-touching adapters cannot express by construction: content-only destruction removes 70–107% of the gain, and **zeroing the bank returns the frozen loss to machine precision**, replicated across seeds and accelerators. **Four registered predictions failed and are printed beside their diagnoses**; two returned design rules; one marks the boundary the next run resolves.

Keywords: depth memory, attention residuals, residual evidence, innovation process, parameter-efficient finetuning, world models, JEPA, latent actions, interaction traces, object permanence, causal memory, recurrent depth, 4D scene memory.

Status note. This manuscript is a technical draft plus a staged validation program. Executed so far, all on released code with machine-audited raw outputs: the mechanism suite E1–E9 (sandbox; identity through MoE/early-exit/recurrent-state control flow, gate-first gradient order, delta-key conditioning, null-mass bounds, a trained miniature with causal interventions, and a locally pretrained real-code checkpoint); and the public-checkpoint runs E10.1–E15 (bitwise insertion identity on three open models; conditioning, normalization cure, and retrieval-under-corruption on five; the retrofit-vs-from-scratch twin pretrain with its diagnosed failure; matched-parameter adapter comparisons at ≈3.1M and 50M tokens with a cross-hardware replication). Stages S1 and, in part, S2 of §11 are thereby executed; the at-scale versions of §11.3a and S15 remain registered. Every corrected or falsified claim is tabulated in Appendix D; the immediately usable artifact is Retro-DARC-Lite plus the released harness (§5.1).

---

1. Introduction

Transformer language models use attention to select information across sequence positions. The same selectivity is usually absent across network depth. In a standard PreNorm residual block, with $h_l(t) \in \mathbb{R}^d$ the token representation at position t and depth l,

$$\Delta_l(t) = f_l(\mathrm{RMSNorm}(h_{l-1}(t))), \qquad h_l(t) = h_{l-1}(t) + \Delta_l(t), \tag{1}$$

and unrolling gives $h_L(t) = h_0(t) + \sum_{l=1}^{L} \Delta_l(t)$. Every previous layer contributes with fixed coefficient one. This residual stream is essential for optimization [He16; Vaswani17], but as a *memory* mechanism it is crude: depth is accumulated, not

queried. In filtering-theory terms, each Δ_l is an *innovation* — the part of layer l's computation its predictive context did not already contain (§6.1a) — and the standard architecture sums its innovations instead of keeping them addressable.

Recent work challenges the fixed-aggregation design. Attention Residuals replace uniform accumulation with softmax aggregation over previous layer outputs [AttnRes26]; Delta Attention Residuals route over layer deltas rather than redundant cumulative states [DeltaAR26]; OASIS adds null-aware routing [OASIS26]; MoDA, MUDDFormer, DeepCrossAttention, and Depth-Attention independently confirm that residual connectivity is a major information-routing axis, not merely an optimization detail [MoDA26; MUDD25; DCA25; DepthAttn26]. Depth routing is no longer a research curiosity: **Kimi K3** [KimiK3-26], a 2.8-trillion-parameter open-weights-announced Mixture-of-Experts model (launched 2026-07-16; weights and technical report announced for 2026-07-27) — makes **Attention Residuals** (paired with Block Attention Residuals over depth blocks, and Kimi Delta Attention, a gated-delta-rule linear attention, across sequence length) one of its two headline architectural changes, reporting that selectively retrieving representations across depth rather than accumulating them uniformly yields ~25% higher training efficiency at under 2% added cost, and contributing ~2.5× overall scaling efficiency over its predecessor. When the current largest open model in the world adopts depth-wise selective retrieval as a load-bearing primitive, the design axis this paper builds on is settled at frontier scale. The sharper question is therefore not *whether* depth routing works, but:

> ***Can a pretrained Transformer — including a frontier model that does not yet have depth memory — be upgraded with it without changing its original function at insertion, and can that memory be made typed, queryable, and causally auditable?***

This matters because most valuable models already exist as checkpoints, and the most capable of them — Kimi K3 included — will keep being released faster than any single lab can re-pretrain them with a new primitive. Kimi K3 baked Attention Residuals in *at pretraining*; the tens of thousands of already-trained checkpoints (and every future frontier model that ships before its lab adopts depth memory) cannot. A method requiring from-scratch training competes for pretraining budgets and can only ever reach models built after it; a safe retrofit has lower experimental cost, a clearer adoption path, applies to models that already exist, and poses a sharper scientific test: it asks whether pretrained layers *already* contain reusable internal computations exposable by a small routing mechanism — and if K3's from-scratch result is any guide, they should.

The question has acquired new urgency. The 2026 world-model field has converged on latent-space prediction as the dynamics substrate — SkyJEPA on real quadrotors [SkyJEPA26], MWA on the RoboCasa leaderboard [MWA26], LoopWM's recurrent latent core [LoopWM26], Physis's next-physical-state objective [Physis26], V-JEPA 2 [VJEPA2-25] — while the structured camp (μ_0's 3D interaction traces [Mu0-26], Aether's causal variables [Aether26], World Labs' persistent 3D worlds [Marble25]) insists that black-box latents resist interpretation and intervention. Simultaneously, **long-horizon memory has been publicly named as a core unsolved deficit**: Physis lists "long-horizon temporal memory loss" among the three shared shortcomings of mainstream models; Genie 3's stated limitation is interaction duration measured in minutes [Genie3-25]; Manifold AI ships a geometry-aware world-state memory as a product differentiator [Manifold26]. Among the world-model systems surveyed here we found none that offers a *retrofit* memory interface for the pretrained models everyone already has; the closest LLM-side prior art (§14, limitation 10) claims reversible frozen-model memory but neither exact insertion identity nor causal audits. Retro-DARC targets exactly that gap, and does so at the point of maximum tension in the field: its memory is latent-cheap **and** typed, queryable, and intervenable.

The measured record in brief (details in §10, rows E10.1–E15). Bitwise insertion identity on gpt2 / pythia-410m / Qwen2.5-0.5B (E10.1); output-key condition numbers growing with depth to 76 while interior delta keys stay < 6.7, with per-slot normalization holding full-stack conditioning at 3.0–3.7 on five models (E10.2, E13); top-1 retrieval under corruption **0.86–0.99 for delta keys** (one disclosed exception) vs. 0.04–0.75 for output keys (E13b); the registered twin pretrain answering the retrofit-vs-from-scratch question with a diagnosed failure (E11, E14); and the matched-parameter comparison — **depth reads beat LoRA**, trail the best weight-touching adapter by 0.006–0.024 CE with the gap shrinking over a 16× budget, with content-destroying audits and a two-machine replication agreeing to $\leq 6\times10^{-4}$ CE (E12, E15).

Core thesis. Pretrained LLMs and planners already contain useful internal residual memories — layer deltas in text models; observation/action/tool, latent-cause, trace, and physics-residual deltas in world/action models. Retro-DARC exposes those memories through a zero-initialized, function-preserving adapter that preserves the base model's outputs at insertion, learns during cheap continued training, and improves long-horizon state maintenance over parameter-matched LoRA, residual-adapter, AttnRes-family, and matched-FLOP recurrent-depth baselines. Causal interventions verify that the memory itself is responsible for the gains.

Contributions.

1. **A function-preserving depth-memory retrofit, verified bitwise on public checkpoints.** Retro-DARC-Lite — a zero-gated read over recent layer deltas with Softmax1 null routing and adapter-only training — with proven exact insertion identity (through MoE routing, early exit, and recurrent state; Props. 1, 3′) and gate-first trainability (Prop. 2′), now discharged on real open weights: bitwise logit identity on gpt2 / pythia-410m / Qwen2.5-0.5B, exact restoration on detach, and step-0 gradient order holding exactly on every seed of every zero-gated retrofit run, on CPU, MPS, and CUDA backends (§5, §9, §10 E10.1).
2. **Delta keys as measured-better depth addresses on real models — with the failure mode mapped and cured.** Lemma 2's conditioning asymmetry transfers from theory to public checkpoints (output-key $\kappa \to 76$ vs. delta-key $\kappa < 6.7$ interior, on three models; E10.2), the discovered top-of-stack anomaly is localized to *norm growth* (final-delta norm ratio 4.9–15.9×) and eliminated by per-slot key normalization (full-bank κ 3.0–3.7, 5/5 models; E13), and the addressing consequence is measured directly on all five: top-1 retrieval under noise and int4-style corruption of 0.86–0.99 for delta keys (one disclosed exception, 0.57–0.63, still ≥6× the alternative) vs. 0.04–0.75 for output keys — simulated at the corruption levels quantized serving imposes, with the kernel-real test registered (§9 Lemma 2, §10 E13b).
3. **The retrofit-vs-from-scratch question, executed.** The registered §11.3a twin pretrain returns both answers at miniature scale: building the read in at full initial gate (γ_0=1) *hurts* (worse by 0.0142 CE, 3/3 seeds), and a keys × routing × init factorial isolates the cause to gate-initialization shock — zero-init beats γ_0=1 on 3/3 seeds in both key/routing configurations (output+softmax and delta+Softmax1), with zero-init reads recovering to ≈parity with plain pretraining (2/3 seeds within the registered +0.005 band; mean gap +0.0024). Meanwhile the zero-gated retrofit adds its full increment (+0.0313 ≈ the plain-base +0.0321) even on top of an architected-retrieval (AttnRes-pretrained) base — with ~100% shuffle sensitivity and exact zero-out restoration. Two measured design rules follow: normalize delta keys; zero-initialize every gate, from-scratch builds included (§10 E11/E14, §11.3a).
4. **Causal auditability as a measured property, not a promise.** At matched trainable parameters on frozen pythia-410m, both depth reads beat LoRA and approach the best weight-touching adapter (3 seeds at ≈3.1M tokens on MPS; one seed per machine at 50M tokens on MPS and CUDA). The audit is the property no weight-touching adapter can express: content-destroying, parameter-preserving interventions — shuffle removes 70–107% of every gain; same-norm random memory always lands *worse* than frozen — attribute the gain to retrieved memory content while the trained mechanism stays in place, and the integrity check behind them is machine-verifiable: zeroing the bank returns the frozen loss bit-exactly in all 16 CPU/MPS tests across E9/E11/E12/E15 and to $\leq 1\times10^{-7}$ in both CUDA tests (kernel reduction-order noise). The depth-read protocol replicates across seed and accelerator with final CEs agreeing to $\leq 6\times10^{-4}$ on a digest-verified identical data stream (§10 E12/E15).
5. **A falsification-first record.** Four registered predictions failed outright (E11's B ≥ A; E13's spike-not-drift (held 0/5) and single-layer exclusion (held only 2/5); E15's deltas-beat-outputs gate) plus one partial miss (E14 parity on one seed), all printed verbatim beside their diagnoses — including the honest current boundary: with *unnormalized* keys at mid-stack insertion, output keys still win raw CE at 50M tokens, replicated across hardware; the normalized-key test is registered next (§10, §11.4 S2).
6. **One typed interface from layer deltas to the agent loop.** Retro-DARC-X configures the same read for observation/action/tool, latent-cause, trace, spatiotemporal, physics-residual, and failure/verification deltas; the Residual Evidence Principle (budgeted selection of conditional-surprise evidence) unifies these with its independent 2026 instantiations, and composability with weight-tied recurrence is proven (§2, §6–§8, §9).
7. **A registered scaled program on top of the measured base** — intervention-level causal estimands, the field's shared long-horizon metrics, the AttnRes-recovery question at scale against a published from-scratch counterpart, and a MemoBench-style object-permanence falsification stage (§11).

---

2. Positioning: a memory interface for the 2026 world-model landscape

Several landscape systems cited in this section and §7 (MWA, Physis, Manifold, Aether) rest on vendor announcements not independently confirmed in our verification sweep (§14, limitation 10); their numbers are motivation and benchmarks-to-meet, and no measured claim in this paper depends on them.

Retro-DARC is not a fifth camp. It is an interface any camp can adopt, and the case for it is strongest when stated against the camps' own published evidence.

2.1 The camps and their converged lessons. The field now spans video-generation (Sora 2; Genie 3's real-time interactive worlds with minutes-scale consistency and promptable events), 3D-spatial (World Labs Marble: persistent, editable, exportable worlds), physics-engine simulation (Isaac/Newton/Genesis; Cosmos as open world foundation models), latent/JEPA (V-JEPA 2; SkyJEPA; Dreamer; LoopWM; MWA; Physis), world-action models (DreamZero: a 14B monolithic DiT denoising video and action tokens jointly, 7 Hz closed loop via single-step DreamZero-Flash, 30-minute/55-trajectory embodiment transfer), and causal/structured (Aether's causal feature–structure–dynamics stack; μ_0's trace space). Four lessons recur across otherwise incompatible systems:

- *(L1) Predict in latent; decode sparingly.* Pixel reconstruction burns capacity on appearance — μ_0, MWA, LoopWM, SkyJEPA, and Physis argue this independently.
- *(L2) Freeze the backbone; train a small head.* SkyJEPA trains its physics prober on **frozen** latents; MWA freezes its inverse-dynamics encoder as a causal-alignment anchor; μ_0 freezes the world model and trains a light action expert; V-JEPA 2-AC post-trains an action head on <62 h of robot data. Retro-DARC's function-preserving retrofit is the limiting case of this pattern — the base model is provably *unchanged* at insertion.
- *(L3) The learning target that transfers is a residual or difference.* SkyJEPA's prober outputs acceleration residuals over analytic rigid-body dynamics; latent actions (MWA; AdaWorld lineage) are inter-frame change codes; traces (μ_0) are motion deltas of semantic interaction points; LoopWM's early-exit gate empirically keys on the difference between consecutive loop iterates; codec-stream tokenization selects by bit-cost/motion residuals. Section 6 formalizes this pattern.
- *(L4) Long-horizon error compounding is the shared metric of merit, and memory is now the named bottleneck.* Folk baseline: ~50 usable rollout steps (DreamerV3). LoopWM reports 200–300; MWA lifts stable latent planning from the industry's ~4 s ceiling to 10 s+ via Latent Action Chunks; SkyJEPA reduces the 60-step compounding ratio from 2.4 to 1.4 and introduces *temporal straightness* (0.75 vs. −0.4 for autoregressive baselines); LingBot-World 2.0 attacks drift directly with a bidirectional-attention regularizer (MoBA) and distribution-matching distillation run *over the model's own long self-rollout trajectories* rather than teacher-forced states, plus a Pilot/Director agentic harness, to reach hour-long interactive generation [LingBot26]. A cluster of 2026 benchmarks makes the deficit quantitative and unambiguous: MemoBench's disappear-and-reappear paradigm shows that across ten leading world models *none* exceeds an Object Reappearance Score of 0.6, and — decisively for this paper — that object permanence does **not** emerge as a by-product of camera control but "must be explicitly addressed in model design"; its ablation finds that supplying the object's initial state as a condition helps far more than tripling model size (≈+4.2 dB PSNR vs. negligible) [MemoBench26]. MBench, MIND, and "out-of-sight-out-of-mind" report the same core finding independently [MBench26; MIND26; OOSOM26]. Memory — not fidelity, not scale — is the binding constraint, and the field now says so with numbers.

2.2 The unowned slot. Each system attacks long-horizon state with a *bespoke, train-from-scratch* mechanism: recurrence (LoopWM), chunked inverse dynamics (MWA), geometry-aware state memory (Manifold), a persistent 3D substrate (Marble). None retrofits memory into an *existing* pretrained model, and none makes the model's own internal computation history — its residual stream — queryable. Retro-DARC's claim is deliberately orthogonal: whatever the camp, the planner module is (or contains) a Transformer whose belief state is distributed across a residual stream; a zero-gated read path over that stream's deltas, and over typed deltas of the surrounding agent loop, adds queryable memory at zero behavioral cost at insertion.

2.3 Falsifiability inherited from the causal camp. Aether frames every robot action as an intervention and stakes its program on intervention-level (rung-2) and counterfactual (rung-3) competence; Physis builds RL *verification* into its training loop; MWA's AnyPhys makes failure data first-class. Retro-DARC's causal-memory-effect methodology (§11.1) is the same epistemic move applied to memory itself: zeroing, shuffling, and same-norm-randomizing the memory bank are interventions on the mechanism, and the paper's headline claim is defined to fail unless those interventions remove the measured gains.

2.4 The production signal: Kimi K3 validates the axis; Retro-DARC generalizes the deployment path. The most important 2026 datapoint for a *language-model* retrofit is not a world-model demo but a shipped frontier LLM. Kimi K3's Attention Residuals are precisely the mechanism this paper's core builds on — depth-wise selective retrieval instead of uniform accumulation — now proven at 2.8 T parameters with a favorable efficiency ledger (~25% training-efficiency gain, <2% cost) and open weights [KimiK3-26]. This changes Retro-DARC's positioning in three ways. First, it removes the "does the axis matter at scale?" objection: it does, at the largest open scale in existence. Second, it sharpens the comparison — K3 obtains depth memory by *pretraining from scratch* with the primitive built in; Retro-DARC obtains it by *retrofit with exact insertion identity*, so the natural head-to-head is not "which is better" but "how much of the from-scratch AttnRes

gain is recoverable by a zero-gated adapter on a frozen checkpoint?", which §11 registers as a first-class question. Third, K3's Block Attention Residuals (depth-block-local retrieval to bound the memory footprint) independently arrive at the same scaling concern that motivates Retro-DARC's chunk summaries and block-local routing (§5.5, §7.4). K3 also pairs AttnRes with KDA, a gated-delta-rule linear attention that compresses history into a recurrent state and calibrates against periodic full-attention layers — a *sequence-axis* analogue of the "carry a compressed running summary, refresh selectively" pattern Retro-DARC applies on the *depth* axis, and a reminder (§12) that the two axes are complementary, not competitive.

2.5 The convergent readout pattern. A striking number of the strongest 2026 systems now *freeze a predictive world model and read its internal representation into a light action head*: μ_0 (frozen trace WM + expert), SkyJEPA (frozen latents + physics prober), MWA (frozen IDM anchor), V-JEPA 2-AC, and now Alibaba DAMO's **RynnWorld-4D**, whose policy consumes the *mid-diffusion internal 4D features* (synchronized RGB-depth-flow, back-projected to 3D scene flow) of a frozen tri-branch world model and beats Diffusion-Policy/π0/π0.5 on contact-rich bimanual tasks, with open weights [RynnWorld26]. "Freeze the world model; read its computation into action" is now the field's dominant transfer recipe. Retro-DARC is the limiting, retrofit-safe case of that recipe applied to the model's *own residual stream*: the base is provably unchanged at insertion, and the read is over its internal deltas rather than a separately trained latent.

---

3. Related Work

Residual streams and depth routing. Residual connections enable deep-network optimization [He16]; Transformers pair them with attention and normalization [Vaswani17; Ba16]. A 2025–26 line makes depth aggregation *selective*: Attention Residuals attend over previous layer outputs, with Block AttnRes for scale [AttnRes26]; Delta Attention Residuals show cumulative states are redundant and route over deltas $h_{l+1} - h_l$ [DeltaAR26]; OASIS adds Softmax1 null routing [OASIS26]; MoDA mixes depth-KV into heads [MoDA26]; MUDDFormer [MUDD25] and DeepCrossAttention [DCA25] learn dense or input-dependent layer combinations; Depth-Attention's in-cache value mixing is our systems bar [DepthAttn26]. The line has reached production: **Kimi K3** [KimiK3-26], a 2.8 T-parameter open MoE, ships Attention Residuals and Block Attention Residuals as core architecture alongside gated-delta-rule linear attention (KDA [KimiLinear25]), reporting ~2.5× scaling efficiency over K2 (self-reported; see §14). All of these design architectures for from-scratch pretraining; Retro-DARC targets the orthogonal problem — safe retrofit of a *queryable, typed, intervenable* memory with causal attribution — and §3.1's Table 1 gives the head-to-head. The innovations lineage behind §6.1a is classical [Kalman60; Kailath68]. · [Kalman60] Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 1960. · [Kailath68] Kailath, An innovations approach to least-squares estimation, IEEE TAC 1968.

Recurrent depth. Weight-tied recurrence as a compute axis runs from Universal Transformers [Dehghani18] through recurrent-depth latent reasoning [Geiping25] and Mixture-of-Recursions [MoR25]; **LoopWM** [LoopWM26] brings it to world models with a learned early exit and a high-weight latent-consistency loss. LoopWM makes computation-per-transition *adaptive*; Retro-DARC makes computation history *retrievable* — §8 shows the two compose, and §11's matched-FLOP recurrent-depth baseline prevents "more iterations" from masquerading as memory.

Latent/JEPA world models. V-JEPA/V-JEPA 2 predict masked spatiotemporal content in representation space, with an action-conditioned head added from small robot-video corpora [VJEPA-24; VJEPA2-25]; LeJEPA's SIGReg gives heuristics-free anti-collapse [LeJEPA25].

World-action models. WAMs jointly model future states and actions [WAM26]. DreamZero denoises video and action tokens in one monolithic DiT [DreamZero26]; RynnWorld-4D's policy head reads a *frozen* world model's internal 4D features in a single forward pass [RynnWorld26] — a clean instance of "read the frozen model's internal state into action" (§2.5); EgoScale reports log-linear loss scaling on action-labeled egocentric human video [EgoScale26]. The VLA pole spans π0/π0.5 [Pi05-25] and the RDT line [RDT24; LiberAI26], with embodied generalization tracking data *diversity* [DSL25]; DIAL decouples intent from action via latent world modeling [DIAL26], and Fast-WAM asks whether WAMs need test-time imagination at all [FastWAM26].

Interactive drift control and planning interfaces. LingBot-World 2.0 reaches hour-long drift-free interactive generation by distilling over the model's *own self-rollouts* and wrapping the predictor in a Pilot/Director agentic harness [LingBot26]; its sibling LingBot-Video supplies a physics-graded RL reward stack that §7.3's verification memories can log [LingBotVideo26]; Self-Harness searches the outer harness under regression gating [SelfHarness26] — all directly relevant to failure memory (§7.3, remembering the model's own boundary/error states). On the planning interface, Fast-LeWM [FastLeWM26] replaces autoregressive CEM stepping of the LeWM protocol [LeWM26] with parallel action-prefix prediction (≈4× faster dynamics at higher success), the batched-memory-read pattern §12 recommends and §13.2(7) registers; its direct-vs-composed consistency gap is the residual-evidence signal E8 measures (§10). Reverie's Interaction

Model names real interaction memory and adaptive chunking as the make-or-break capabilities of always-on interactive systems [Reverie26] — §7's motivation in product form.

Latent actions and negative data. MWA freezes an inverse-dynamics encoder as a causal-alignment anchor, plans in multi-step latent-action chunks, and makes negative/boundary data first-class via AnyPhys [MWA26]; Retro-DARC adopts frozen-anchor alignment (already our design), temporal delta chunks (§7.4), and failure memory as a named type (§7.3).

Structured and causal camps. μ_0's 3D interaction traces from video-only pretraining [Mu0-26], Aether's causal-representation overlay [Aether26], Physis's next-physical-state prediction over a unified latent physical state [Physis26], and object-level latent interventions [CJEPA26; LPWM26] are the systems whose *representations become memory types* in §7.

Memory-consistency benchmarks (the thesis, externally stated). MemoBench's Visible→Disappear→Reappear paradigm finds none of ten leading models above an Object-Reappearance-Score of 0.6 and concludes that memory must be explicitly addressed in model design [MemoBench26]; MBench, MIND, and "Out of sight, out of mind?" reach the same verdict independently [MBench26; MIND26; OOSOM26]. These are the measuring instruments of §11's world-model stages — and the external restatement of this paper's motivation.

Evidence-budgeted allocation elsewhere. LLaVA-OneVision-2 allocates visual tokens by codec-stream bit-cost/residual evidence rather than uniform frames [LLaVAOV2-26; OVE26], the input-side twin of Retro-DARC's internal allocation; GPS spends RL-post-training rollouts and test-time compute by predicted informativeness [GPS26]. Evidence-guided budgeting is a broad 2026 motif, not a Retro-DARC idiosyncrasy.

Renderers, simulators, and commercial signal. Genie 3 [Genie3-25], Marble [Marble25], Cosmos [Cosmos26], WorldScape's geometry-aware world-state memory [Manifold26], Meshy [Meshy26], MoWorld's NPU-native ~50 FPS Flash world model [MoWorld26], and SPEAR [SPEAR26] populate the renderer/simulator camp. The market has ratified the thesis: Momenta shipped its R7 RL world model and held the field's first IPO [Momenta26]; AMI Labs raised a $1.03 B seed for JEPA world models [AMI26]; HiDream [HiDream26] and Tashi [Tashi26] headline the H1-2026 funding wave. Most directly, K3 brought this paper's primitive to the largest announced open model [KimiK3-26]; the queryable, typed, intervenable memory layer is what such bases still lack.

Parameter-efficient adaptation and adaptive compute. LoRA [LoRA21] and bottleneck adapters [Houlsby19] are our capacity-matched controls; Mixture-of-Depths and LayerSkip motivate the optional route-entropy compute axis [MoD24; LayerSkip24].

3.1 The verified depth-routing lineage — and the near-empty quadrant (sources live-verified 2026-07-26)

Between March and July 2026 the depth-routing literature split into four families, all now verifiable on arXiv. *Single-stream attention over depth:* Attention Residuals [AttnRes26] (validated inside Kimi Linear, 48B/3B, 1.4T tokens), Delta Attention Residuals [DeltaAR26], Multi-Gate Residuals (arXiv:2605.23259), and Dual Attention Residuals (arXiv:2607.18730). *Multi-stream connectivity:* Hyper-Connections (arXiv:2409.19606), mHC's doubly-stochastic constraint restoring the identity-mapping property (DeepSeek, arXiv:2512.24880; shipped in DeepSeek-V4, arXiv:2606.19348), and KromHC (arXiv:2601.21579). *Dense cross-layer combination:* MUDDFormer [MUDD25], DeepCrossAttention [DCA25], LAuReL (arXiv:2411.07501). *Retrofit routers for efficiency:* FlexiDepth (arXiv:2503.23798), Dr.LLM (arXiv:2510.12773), GateSkip (arXiv:2510.13876) — which retrofit routing onto frozen checkpoints, but only to *remove* compute, never to add memory.

Table 1 — depth-routing and memory-retrofit lineages vs. the properties measured here (live-verified 2026-07-26; ● = published, ◐ = partial/bounded, — = absent in the surveyed record).

line of work	quality retrofit onto pretrained	exact insertion identity	delta-key addressing	causal intervention audit	robustness under noise/quant.	retrofit-vs-scratch control
AttnRes / Block AttnRes (K3, arXiv:2603.15031)	— (from scratch)	—	—	—	—	—
Delta Attention Residuals (arXiv:2605.18855)	◐ (bounded-perturbation conversion)	—	●	—	— (explicitly omitted)	—
Multi-Gate / Dual AttnRes (2605.23259, 2607.18730)	—	—	—	—	—	—

line of work	quality retrofit onto pretrained	exact insertion identity	delta-key addressing	causal intervention audit	robustness under noise/quant.	retrofit-vs-scratch control
Hyper-Connections / mHC / KromHC (2409.19606, 2512.24880, 2601.21579)	—	◐ (identity-constrained, from scratch)	—	—	—	—
Efficiency retrofits: FlexiDepth, Dr.LLM, GateSkip	● (compute removal only)	—	—	—	—	—
Memory adapters: LongMem, CAMELoT, Larimar, 2601.15324, 2603.22329	● (reversible at best)	—	—	—	—	—
Retro-DARC (this work)	●	**● bitwise, ×3 models**	**● + normalization rule**	**● zero/shuffle/random, per-type**	**● E13b, simulated int4**	**● twin pretrain, 3 seeds**

Plotting these on (quality vs. efficiency) × (from-scratch vs. retrofit) exposes the quadrant this paper occupies: **quality-adding retrofit is nearly empty.** Among the surveyed families, the only other occupant we found is Delta Attention Residuals' checkpoint-conversion experiment — the mandatory head-to-head. DAR converts Qwen3-0.6B by fine-tuning for 20K steps with zero-initialized routing queries and dual learning rates, achieving a *bounded-perturbation* "smooth start" [DeltaAR26]; the open AttnRes reimplementation (github.com/wdlctc/open-attention-residuals) documents the alternative — naive retrofit loss spikes, and fine-tuning a committed residual stream yields only ~0.02 loss gain. Table 1 summarizes the comparison; against this closest published work, the present paper's differentiators are exactly its measured record: (i) *exact* — bitwise, not bounded — insertion identity on public checkpoints, with detach restoring the base (E10.1); (ii) causal intervention audits of what the routing actually uses, which none of the surveyed depth-routing papers provides (E9/E11/E12/E15); (iii) addressing robustness under key noise and quantization-style corruption, which [DeltaAR26] explicitly omits (E13b); and (iv) the only controlled retrofit-vs-from-scratch twin pretrain we found in this literature (E11/E14). Convergent evidence runs the other way too: DAR independently reports *routing collapse* over cumulative states (max attention weight ≈0.2, restored to ≈0.6 by deltas) — the utility-side signature of the conditioning asymmetry Lemma 2 proves and E10.2/E13b measure.

Two honest boundary conditions from the same sweep. First, the one public mechanistic critique of depth attention we found (Z. Liu, 2026: AttnRes wins on structured data, loses on memorization, crossover shifts with depth) implies depth-memory gains are data-regime-dependent; our E12/E15 CE margins on web text are consistent with that reading, and the §11 program inherits it as a stratification requirement rather than a refutation. Second, frontier pretrains are absorbing depth routing natively (K3's AttnRes; DeepSeek-V4's mHC) — which does not shrink this paper's target but *defines* it: the installed base of checkpoints that predate these primitives (every Llama-, Qwen-, and DeepSeek-V3-era model in production), plus any deployment that must preserve a certified base function while acquiring memory, is precisely the population a bitwise-identity retrofit serves and a from-scratch primitive cannot.

4. Problem Setting and Design Requirements

Let F_θ be a pretrained autoregressive Transformer with hidden states $h_l(t) \in \mathbb{R}^d$, $l \in \{0, \dots, L\}$, and logits $z_\theta(x)$. A retrofit adapter with parameters ϕ yields $z_{\theta,\phi}(x)$. We require:

1. **Function preservation at insertion.** For initialization ϕ_0: $z_{\theta,\phi_0}(x) = z_\theta(x)$ for all x, up to numerical precision.
2. **Gate-first trainability (corrected by its own check).** At insertion the adapter must be trainable *through its gate*: with γ=0, $\partial L/\partial\gamma$ is generically nonzero while the gradients of W_O, the router, and the key projections are exactly zero (every path from them to the loss carries a factor γ); once $\gamma \neq 0$, all adapter parameters acquire gradient. Earlier versions stated this loosely as "nonzero adapter gradients at insertion"; the precise statement — and why it is a *feature* (a built-in gate-first curriculum) — is Prop. 2′/Remark 2′ (§9.1), verified in §10 (E2).
3. **Parameter and compute fairness.** Matched trainable parameters, tokens, data order, and *measured* overhead against all baselines — including matched-FLOP recurrent-depth baselines (§8, §11), so that extra computation cannot impersonate memory.
4. **Causal use.** Disabling or corrupting the learned memory must reduce performance if the claimed mechanism is real (Proposition 5).

Requirement 4 is central: without interventions, Retro-DARC could be dismissed as an expensive residual adapter. The claim is not that the model improves, but that retrieved residual memories are *why* it improves.

5. Retro-DARC-Lite: The Adoptable Core

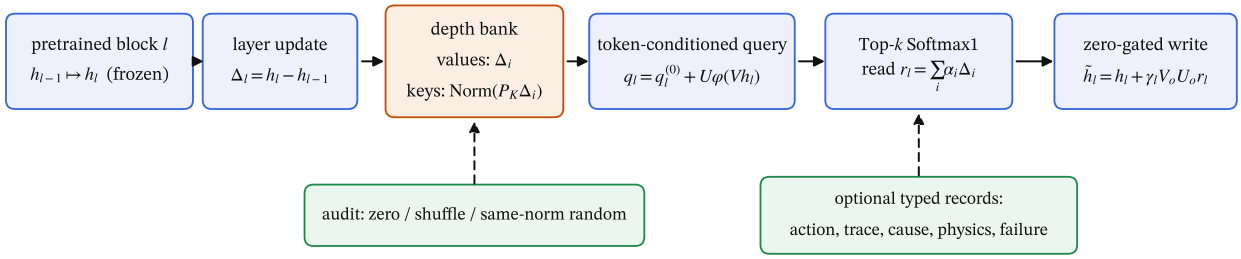

Figure 1: The Retro-DARC-Lite read path. The base block stays frozen; recent layer updates Δ are stored as full-width values and addressed through low-rank, per-slot normalized keys (E13 rule). The read returns through a low-rank output projection behind a scalar gate initialized at zero, so insertion is exact identity (E10.1). Because memory content is an explicit runtime object, it supports the zero / shuffle / same-norm interventions of §10 without touching any learned weight.

Retro-DARC-Lite is the version researchers should implement first — no world-model machinery, no custom kernels, no base-model retraining.

5.1 Layer-delta memory. The memory unit is a residual delta $\Delta_l(t) = h_l(t) - h_{l-1}(t)$: what layer l *contributed*, versus a cumulative state containing all history. With local window w,

$$\mathcal{M}_l^{\text{lite}}(t) = \{\Delta_{l-w}(t), \ldots, \Delta_{l-1}(t), \mathbf{0}_\varnothing\},$$

optionally extended with chunk summaries $B_1(t), \ldots, B_{b-1}(t)$ compressing earlier depth blocks (generalized to saliency-weighted pooling in §7.4).

5.2 Low-rank token-conditioned query. A static depth query cannot distinguish tokens needing different internal computations:

$$q_l(t) = q_l^{(0)} + U_l\, \varphi\big(V_l\, \mathrm{RMSNorm}(h_l(t))\big), \quad V_l \in \mathbb{R}^{r\times d},\ U_l \in \mathbb{R}^{d_r \times r},\ r \ll d.$$

Pilots use $r \in \{16, 32\}$, $d_r \in \{32, 64\}$.

5.3 Null-aware top-k routing. Each candidate $M_m(t)$ is scored

$$s_{l,m}(t) = \frac{q_l(t)^\top \mathrm{RMSNorm}(P_K M_m(t))}{\sqrt{d_r}} + b_{\text{type}(m)} + b_{\text{age}(l,m)},$$

with $I = \mathrm{TopK}(s_l, k)$ and Softmax1 normalization

$$\alpha_{l,m}(t) = \frac{\exp(s_{l,m}(t))}{1 + \sum_{j\in I} \exp(s_{l,j}(t))}, \qquad \alpha_\varnothing^{\text{reserve}}(t) = \frac{1}{1 + \sum_{j\in I} \exp(s_{l,j}(t))},$$

letting the model decline retrieval; an explicit null vector with learned bias is optional. The read is $r_l(t) = \sum_{m\in I} \alpha_{l,m}(t)\, M_m(t)$.

5.4 Zero-gated injection.

$$h_l^{\text{new}}(t) = h_l(t) + \gamma_l\, W_O\, r_l(t), \qquad \gamma_l = 0 \text{ at insertion},\ W_O \neq 0.$$

The distinction matters: $\gamma_l = 0$ with $W_O = 0$ also preserves identity but kills first-step gate learning (Proposition 2).

5.5 Default adoption recipe. Insertion every 4 layers or at block boundaries; recent deltas $w = 4$; optional 4–8 chunk summaries; router rank $r = 16$ (sub-1B) / 32 (1–3B), $d_r \in \{32, 64\}$; top-$k \in \{2, 4\}$; Softmax1 + optional explicit null; scalar zero gate per inserted layer; base frozen in pilot; memory values detached (PEFT-style) in the pilot with the non-detached variant tested after signal; no custom kernels before 1–2B signal.

```
from retro_darc import inject_retro_darc
model = AutoModelForCausalLM.from_pretrained(base_name)
model = inject_retro_darc(model, layers="every_4", query_rank=32,
                          router_dim=64, topk=4, memory_window=4,
                          use_block_memory=True, use_anchors=False,
                          gate_init=0.0, freeze_base=True)
```

```
# exact-correctness check at insertion:
assert (base_model(x).logits - model(x).logits).abs().max() < tol
```

5.6 Adoption playbook. Concrete recipes, in increasing ambition; every step inherits the identity guarantee (Prop. 1/3/3′) and the gate-first schedule (Prop. 2′) — so the safe first observation after insertion is *only γ moving*, which is correct behavior, not a bug.

- *LLM finetuner (hours, one GPU).* Freeze the checkpoint; insert Lite at 2–4 upper-middle layers (w=4, k=4, d_r=32, low-rank W_O with r_o=64); train adapter-only on your continued-training tokens; run the E-series checks from the shipped script against your own model before and after (identity, E2 gradient order, shuffle-CME on a held-out state-tracking eval). Ship only if the shuffle intervention removes the gain.
- *World-model builder (days).* If your dynamics core is recurrent (LoopWM-style) or diffusion-mid-feature-read (RynnWorld-style), add the loop-iterate/typed banks of §7–8; E5b is the template: expect the largest gains exactly where rollout noise makes the collapsed final state lossy. Use Fast-LeWM-style parallel prefix readout to batch the memory queries for all candidate plans in one pass.
- *K3-scale lab (weeks, registered).* Follow §10.1: retrofit through MoE+KDA under Prop. 3′; run §11.3a's AttnRes-recovery protocol; A/B delta-vs-output keys in the quantized-serving regime (Lemma 2's prediction); log KDA update magnitudes as a depth-saliency prior (registered upgrade #5).
- *Agent-harness engineer.* Pair with Self-Harness [SelfHarness26]: harness edits that pass regression gating *write failure/verification memories* (§7.3), so the model carries the harness's learned failure taxonomy internally; audit with the E7 intervention battery.

5.1 Using Retro-DARC-Lite today — a measured practitioner's guide

Every number here is from the released JSONs; nothing is estimated.

What you add. Four small linears and one scalar per insertion point: U_q (d→64, bias), P_K (d→64), U_o (d→64), V_o (64→d, init $\mathcal{N}(0, 0.02)$), $\gamma = 0$. Trainable parameters = 4·64·d + 65: 196,673 (gpt2), 229,441 (Qwen2.5-0.5B), 262,209 (pythia-410m). Per E13's measured rule, $\ell 2$-normalize each delta before P_K (full-bank κ stays 3.0–3.7 on all five models tested; raw keys are ill-conditioned near the top of the stack). Per E14, γ starts at 0 — always, including from-scratch builds (γ_0=1 cost +0.0142 CE on 3/3 seeds).

Where and how. One forward pre-hook on blocks 0..INS (INS = 2L/3 in every run here); no modeling-code changes — the released hook contract ran unmodified across GPT-2 (Conv1D), GPT-NeoX, and Qwen2 block conventions, on CPU, MPS, and CUDA. Ecosystem packaging and the serving path are §13.1's subject.

What it costs. At γ=0: nothing — bitwise-identical logits (E10.1) — so the adapter can ship dark and be enabled per-request, and $\gamma \leftarrow 0$ is an exact rollback. Training touches only the adapter; autograd never descends below INS. Measured wall-clock: 1,500 adapter steps (≈3.1M tokens, batch 8×256, pythia-410m frozen) in 8–14 min on an Apple laptop GPU; 24,414 steps (50M tokens) in 2.3–3.4 h per condition on either that laptop or a DGX Spark. The read cannot be merged into base weights (unlike LoRA) — the honest serving cost is the per-token read at chosen insertion points plus the bank; the honest serving benefit is that no merged-weight method can ship dark with a bitwise guarantee.

What to expect (measured, frozen pythia-410m). Depth reads beat rank-matched LoRA (2.9655 / 2.9764 vs. 2.9907; frozen 3.3400; 3 seeds, identical ordering) and trail a width-matched MLP adapter by 0.006–0.024 CE, a gap that shrinks over a 16× token-budget increase. **If raw CE at small scale is your only objective, use the MLP adapter — we measured it winning.** Choose the depth read for what it uniquely, measurably provides: (1) *attributable adaptation* — shuffle/same-norm-random interventions separate memory content from mechanism (70–107% of the gain removable with all parameters intact), a test no weight-touching adapter can express; (2) *exact deactivation* — zeroing bank or gate returns the frozen loss bit-exactly on CPU/MPS, $\leq 1\times10^{-7}$ on CUDA; (3) *parameter-orthogonal composition* with LoRA/MLP adapters (structurally free; utility unmeasured, registered); (4) *typed banks* — what fills the slots (layer deltas here; traces, actions, failures in §7) is configuration, not new machinery.

The audit as engineering practice. Ship the three-intervention battery (zero / batch-shuffle / same-norm-random) as CI: zero must return the frozen loss to machine precision; shuffle should remove most of the learned gain; random should land below frozen. The released harness runs the full battery in minutes on a laptop; in this project it caught every wiring bug before any training run did. The identity check (Task-1 protocol, minutes on CPU) is the recommended first step on any new base model.

6. Residual-Aligned Retro-DARC and the Residual Evidence Principle

Lite keeps a small local window — the safest default. The field's convergent lesson (L3, §2.1) licenses a stronger rule: **memory budget should follow evidence, not uniform time or uniform depth.**

6.1 The principle, with its 2026 instantiations. Let X be an observation or hidden state, $B(X)$ a prediction baseline, $R = X - B(X)$ a residual, and Y a downstream event/answer/action/verification outcome. Under budget K, the ideal retained set is

$$S_K^* = \arg\max_{|S| \le K} I(Y; R_S \mid B(X)). \tag{2}$$

Domain	Baseline B	Residual evidence R	Allocation decision	2026 instantiation
Compressed video	codec prediction from prior frames	bit-cost, motion vectors, pixel residuals	which visual tokens to keep	LLaVA-OneVision-2 codec streams
Transformer depth (production)	**previous hidden computation**	**layer delta Δ_l, retrieved not accumulated**	**which layer outputs to retrieve**	**Kimi K3 Attention Residuals / Block AttnRes**
Transformer depth (retrofit)	previous hidden computation	layer delta Δ_l	which internal computations to *remember*	Retro-DARC (this work); Delta AttnRes
Analytic dynamics	rigid-body integration Φ_ψ	acceleration residual	what the network must model	SkyJEPA's physics prober
4D scene dynamics	current frame geometry	depth + optical-flow → 3D scene-flow	what geometry/motion to predict	RynnWorld-4D RGB-DF
Iterative refinement	previous loop iterate	inter-iterate difference $\Delta^{(s)}$	when to stop computing	LoopWM's Early-Exit Gate
Video dynamics	previous frame	latent action (change code)	what the causal carrier is	MWA latent actions; AdaWorld lineage
Physical interaction	static scene	3D motion deltas of semantic points	what must move	μ_0 interaction traces
JEPA world models	latent prediction $\hat{z}_{t+k}$	latent prediction error	which belief updates to refine	V-JEPA 2; Physis verification loop
Tool/action agents	expected tool/env state	tool error, failed action	which plan state to preserve	AnyPhys negative samples
RL prompt selection	predicted difficulty from history	intermediate-difficulty / diversity residual	which prompts to roll out	GPS predictive selection
Planning consistency	direct horizon prediction	direct-vs-composed prediction gap	which plans/states to trust and *remember*	Fast-LeWM self-consistency; E8 (r = 0.455)

A useful reading of this table: **Kimi K3's Attention Residuals are the *architected, from-scratch* case of depth residual-evidence routing — retrieve the informative layer outputs rather than sum them all. LoopWM's early-exit gate is the *degenerate* case — stop when the residual is small. Retro-DARC is the *retrofit* case that also generalizes both — don't just retrieve or stop; keep a typed, gated, causally-auditable memory of the residual and let later computation (or a downstream planner) query it, on a base model left provably unchanged at insertion.** Likewise SkyJEPA and RynnWorld-4D show residuals/geometry are the right *training target* for a frozen model's readout; Retro-DARC makes residuals the right *memory content* for a frozen-base adapter.

6.1a Residuals are innovations. Equation (2)'s object has a sixty-year lineage: in filtering theory, the *innovation* is exactly the component of a new observation that the current predictive state does not explain — the whitened increment that carries all of the measurement's new information [Kalman60; Kailath68]. Every row of the table above is an innovations process with respect to its domain's predictive baseline, and the layer delta $\Delta_l = h_l - h_{l-1}$ is the depth-axis case: the update the residual stream's running summary did not already contain. This is more than terminology. It says *why* delta banks carry non-redundant addresses (an innovations sequence is decorrelated where the cumulative state is not — Lemma 1/Lemma 2's measured content), *why* zeroing the bank must return the base function (retaining zero innovations is exactly "trust the prior"), and *what the null route is* (a calibrated decision that the current query has no unexplained evidence worth fusing). It also names the registered forward path: classical innovations feed a gain-weighted state update, and the reliability-weighted fusion of retrieved typed innovations into a persistent belief state is the world-model configuration registered in §13.2(7).

6.2 Depth-residual saliency. For delta $\Delta_i(t)$,

$$\eta_i^{\text{depth}}(t) = \lambda_1 \|\Delta_i(t)\|_{\text{RMS}} + \lambda_2 \big(1 - \cos(h_i(t), h_{i-1}(t))\big) + \lambda_3 H_i^{\text{route}}(t) + \lambda_4 u_i(t),$$

with retention $\mathcal{M}_l^{\text{RA}}(t) = \text{TopK}_{i<l}\, \eta_i^{\text{depth}}(t) \cup \{B_{<b}, A_{1:K}, \mathbf{0}_\varnothing\}$ and an evidence prior in the score, $s_{l,i}^{\text{RA}} = s_{l,i} + \beta_d \log(1 + \eta_i^{\text{depth}})$; $\beta_d = 0$ recovers plain Retro-DARC.

6.3 Codec-conditioned alignment for multimodal inputs. With codec metadata per temporal group g/spatial block b, $\eta^{\text{codec}}_{g,b} = a_1 \log(1{+}\text{bitcost}) + a_2\|\text{motion}\| + a_3\|\text{residual}\|$, fuse $\eta^{\text{mm}}_i(t) = \eta^{\text{depth}}_i(t) + \lambda_c\, \eta^{\text{codec}}_{g(t),b(t)}$ and add $\beta_c \log(1+\eta^{\text{codec}})$ to the score. Codec evidence marks event-bearing *observations*; delta evidence marks changed *computations*; the fusion is tested against each alone in §11.

6.4 RA is an optional acceleration, evaluated only after Lite succeeds: evidence-biased retention can prematurely discard low-magnitude information that matters later, so the repository exposes `residual_aligned=False` as the default and `topk_residual_evidence` (`+codec_prior`) as configuration.

7. Retro-DARC-X: One Typed-Memory Interface for World/Action Planners

Earlier drafts presented Full, A, Causal, 4D, and WM as five architectures. This paper consolidates them: **there is one mechanism** — the zero-gated, null-aware, token-conditioned read of §5 — and a *typed memory bank* whose contents are configured per domain. Every type below is a residual in the §6 sense; every configuration inherits the insertion invariant unchanged. This consolidation is itself a response to the field: the 2026 systems each discovered one or two of these types independently; Retro-DARC's contribution is the shared, retrofit-safe interface over all of them.

7.1 Planner-side belief memory. For a trajectory $\tau = (o_1, a_1, y_1), \ldots, (o_T, a_T, y_T)$ a Transformer planner maintains an implicit belief $b_t \approx p(s_t \mid o_{\le t}, a_{<t})$ distributed across its residual stream. Retro-DARC-X makes portions of the update history queryable: $b^{\text{new}}_{l,t} = b_{l,t} + \gamma_l W_O r_l(t)$, where $r_l(t)$ reads from the typed bank. In a functional taxonomy of world-model uses (generation, prediction, planning).

7.2 The typed bank.

$$\mathcal{M}^X_{l,t} = \Big\{ \underbrace{\Delta^{\text{layer}}_{i,t}}_{\text{depth}},\ \underbrace{\Delta^{\text{obs}}_{t-j},\ \Delta^{\text{act}}_{t-j},\ \Delta^{\text{tool}}_{t-j}}_{\text{agent loop}},\ \underbrace{c_{t-j}}_{\text{latent cause}},\ \underbrace{\tau^{\text{trace}}_{t-j,p}}_{\text{3D traces}},\ \underbrace{\Delta P_{i,t},\ \Delta c^{\text{cam}}_t}_{\text{4D scene}},\ \underbrace{z^{\text{chunk}}_{t-j}}_{\text{latent-action chunks}},\ \underbrace{\epsilon^{\text{phys}}_{t-j},\ v^{\text{fail}}_{t-j}}_{\text{verification/failure}},\ B^{\text{episode}}_{<b},\ A^{\text{belief}}_{1:K},\ \mathbf{0}_\varnothing \Big\}$$

scored with type/age/object biases, $s_m = q^\top \text{RMSNorm}(P_K M_m)/\sqrt{d_r} + b_{\text{type}} + b_{\text{age}} + b_{\text{obj}}$. Configurations:

- **X-A (agent traces):** observation/action/tool deltas plus episode summaries and belief anchors A_k with gated EMA updates $A^{(l)}_k = (1-\eta_{l,k})A^{(l-1)}_k + \eta_{l,k} W_A \Delta_l$. Validated cheaply via Action-State Maintenance Probes (ASMP; §11.6): the target causal signature is that *action-memory* corruption specifically damages action-dependent decisions, growing with trajectory length.
- **X-Causal (abductive causes):** an abductor $c_t = A_\phi(z_t, z_{t+1}, a_t)$ explains transitions; the bank stores $\{\Delta z_t, c_t, \epsilon^{\text{phys}}_t, \epsilon^{\text{verify}}_t\}$ and the predictor conditions on the read, $\hat{z}_{t+1} = F_\theta(z_t, a_t, r_t)$. Losses: prediction, cause-invariance under content-preserving transforms g, and verification Violation$(\hat{z}_{t+1}, a_t, c_t)$. This is the memory-side counterpart of Aether's causal-variable program and Physis's action-causality-traceable objective; shuffling c_t across trajectories is the rung-2 test.
- **X-Trace (interaction traces):** μ_0 demonstrates that 3D motion deltas of semantic interaction points (objects, tools, hands, contact regions) are a compact, embodiment-agnostic, *intervenable* physical representation learnable from video alone. Retro-DARC-X therefore admits trace tokens τ^{trace} as first-class memory: a frozen trace world model's predicted/observed traces are written to the bank, and the planner queries "what moved, where, how" instead of re-deriving it. TraceExtract-style engines supply supervision without action labels; the natural probe is whether trace-memory corruption selectively damages manipulation-relevant decisions while leaving linguistic behavior intact.
- **X-4D (queryable spatiotemporal memory):** point/track/camera deltas answered through sparse queries $q^{4D} = \varphi(u, v, t_{\text{src}}, t_{\text{tgt}}, c_{\text{cam}}, o, a, h_l)$, $\hat{P}(t_{\text{tgt}}) = D_\omega(q^{4D}, r^{4D})$ — a query-any-point-any-time interface philosophy applied to memory; Manifold's geometry-aware world-state memory is the closest shipped analogue and motivates the geometry regularizer of §9.
- **X-WM (object-centric JEPA + physics residuals):** object slots $Z_t = \{z_{t,o}\}$; JEPA loss at object level; an analytic/checkable module Φ_ψ gives $\bar{Z}_{t+1} = \Phi_\psi(Z_t, a_t)$ with the network predicting the *residual correction* $\hat{Z}_{t+1} = \bar{Z}_{t+1} + R_\theta(Z_t, a_t, r_t)$ — exactly SkyJEPA's decomposition, now with the residual history retrievable; constraint sets replace Φ_ψ where no simulator exists. Verification vectors $v_t = V(\hat{Z}, Z, a, \hat{o})$ are written back as memory, and we report the **physics-slop index** PSI $= \mathbb{E}[\text{PhysViolation}]/(\mathbb{E}[\text{PerceptualError}] + \epsilon)$: high PSI is the "physics slop" failure mode named by the WAM camp; a successful X-WM lowers PSI and ablating verification memory raises it.

7.3 Failure memory as a named citizen. MWA's AnyPhys shows industrially that deep negatives, boundary-instability samples, and suboptimal trajectories — auto-scored, no manual labels — are what dense-reward RL and robust policies are missing, and that a curated *failure knowledge base* transfers (5× success on noisy-data precision insertion). Retro-DARC-X therefore names v^{fail} (failed predictions, violated constraints, tool errors, boundary-critical states) as a first-class type with its own type bias and its own ablation: removing failure memories should selectively degrade behavior near physical boundaries and in recovery-from-error segments, not average-case behavior.

7.4 Delta chunks. MWA's Latent Action Chunks and DreamZero's action-chunk smoothing indicate that *multi-step* units, not single steps, are the stable currency of long-horizon control. We generalize the original block summaries to **delta chunks**: for a temporal or depth window C, $B_C = \sum_{i \in C} \omega_i \Delta_i$ with $\omega_i \propto \sigma(\alpha_0 + \alpha_1 \eta_i)$ — saliency-weighted pooling rather than uniform averaging. Chunking reduces candidate count M from $O(L)$ (or $O(T)$) to $O(L/|C|)$, and the chunk-summary probe (§10 E6: event cosine 0.758→0.883) tests whether saliency-weighted chunks preserve retrieval accuracy that uniform pooling loses.

7.5 Codec-conditioned event memory. Before reliable object slots exist, codec saliency (§6.3) supplies weak event supervision: weight the JEPA loss $\omega_{g,b} = \sigma(\alpha_0 + \alpha_1 \eta^{\text{codec}}_{g,b})$ and write both object-level and codec-event deltas. Codec metadata is not a physics model; it is scalable *where-to-allocate* evidence.

---

8. Composability: Retro-DARC inside Recurrent-Depth (Loop) Models

LoopWM establishes iterative latent depth as a scaling axis for world models; the recurrent-depth line does the same for language reasoning [Geiping25; MoR25]. Retro-DARC composes with this axis rather than competing with it, in three specific ways.

8.1 Identity composes across loop iterations. Let the dynamics core apply a weight-tied block G for S iterations, $h^{(s)} = G(h^{(s-1)})$, with Retro-DARC adapters inserted inside or between iterations. Proposition 3 (§9) shows that with all gates $\gamma = 0$, the looped model's iterates, exit decisions, and outputs are unchanged for *any* S, including data-dependent S chosen by an early-exit gate — because the gate's input is an unchanged iterate. Retrofit into pretrained loop models is therefore exactly as safe as retrofit into feed-forward stacks.

8.2 Loop-iterate deltas are a memory type. Define $\Delta^{(s)}(t) = h^{(s)}(t) - h^{(s-1)}(t)$. These are precisely the quantities LoopWM's Early-Exit Gate consumes to decide convergence — and then discards. Adding $\{\Delta^{(s)}\}$ to the bank gives the model access to *how* it converged: which refinement steps changed the state, in what direction, with what residual magnitude at exit. Two testable hypotheses follow: (i) exit-time residuals predict transition difficulty and should raise routing saliency η; (ii) retrieving early-iterate deltas at later timesteps improves long-rollout stability beyond what either recurrence or plain depth memory achieves alone. The loop-iterate delta probe (§10 E5/E5b: −11.4% deterministic, −32.0% under rollout noise, shuffle ×7.1) is the CPU-scale version.

8.3 Memory vs. recurrence must be disambiguated. A loop model spends more FLOPs per transition; a memory model spends FLOPs retrieving history. Gains from Retro-DARC could in principle be re-described as "more computation." The verification plan therefore requires a **matched-FLOP recurrent-depth baseline** (a weight-tied loop over the same base with iteration count set to equalize measured compute) in every stage from 0.6B up, and a combined condition (loop + Retro-DARC, memory-shuffled) whose failure to match (loop + Retro-DARC, intact) isolates the memory contribution. Conversely, LoopWM-style linear test-time scaling (perturb-and-reloop) and Retro-DARC's route-entropy adaptive compute ($p_{\text{refine}} = \sigma(a_H H_l + a_D D_l - a_N N_l + a_0)$ over routing entropy, distant-mass, and null probability) are complementary control signals for the *same* compute budget — a registered follow-up study, not a claim made here.

---

9. Theoretical Properties

Proposition 1 (Exact function preservation). Insert adapters $h_l^{\text{new}} = h_l + \gamma_l W_O r_l$ at any set of layers of pretrained F_θ. If $\gamma_l = 0$ for all inserted adapters, then all subsequent hidden states and final logits equal the base model's for every input, up to numerical precision. *Proof.* $\gamma_l = 0 \Rightarrow h_l^{\text{new}} = h_l$; induct over layers. ∎

Proposition 2 (First-step gate trainability). For loss $\mathcal{L}$ and $h' = h + \gamma W_O r$, at $\gamma = 0$: $\partial \mathcal{L}/\partial \gamma = \langle \partial \mathcal{L}/\partial h',\, W_O r \rangle$. A zero gate is trainable at step one iff $W_O r$ has nonzero projection onto the downstream gradient; $W_O = 0$ makes the gradient identically zero. ∎

Proposition 3 (Identity under weight-tied recurrence; new). Let G_ϕ denote the adapter-augmented block and G the base block, with all gates zero, so $G_\phi = G$ pointwise by Proposition 1. Then for any iteration count $S \geq 1$, $G_\phi^{(S)} = G^{(S)}$

pointwise; moreover, if iteration count is data-dependent, $S(x) = \min\{s : \mathrm{Exit}(h^{(s)}(x)) = 1\}$ for any exit rule reading iterates, then $S_\phi(x) = S(x)$ for all x, and outputs coincide. *Proof.* Equality of single applications gives equality of all iterates by induction on s; equal iterates give equal exit decisions, hence equal $S(x)$ and equal outputs. ∎ *Consequence:* zero-gated Retro-DARC can be inserted into pretrained looped/recurrent-depth models — including their early-exit machinery — with exact behavioral identity at insertion.

Lemma 1 (Delta memory is less redundant under independent updates). For independent zero-mean isotropic Δ_i with $\mathbb{E}\|\Delta_i\|^2 = \sigma^2 d$ and $h_l = \sum_{i \le l} \Delta_i$: $\mathbb{E}[h_i^\top h_j] = i\sigma^2 d$ for $i < j$ while $\mathbb{E}[\Delta_i^\top \Delta_j] = 0$ — cumulative states share expected inner product through common history; distinct deltas do not. ∎ (Empirically: adjacent cosine 0.9468 vs. $-6.5{\times}10^{-4}$, §10.)

Proposition 4 (Budgeted residual selection under calibrated saliency). If each memory item's saliency satisfies $\eta_m = \log p(Y \mid R_m, B) - \log p(Y \mid B)$ up to a constant and selected items contribute additively to log evidence under budget K, then top-K by η_m maximizes retained log evidence among size-K subsets. ∎ This is the idealized condition residual-aligned routing approximates; the obligation is empirical: saliency-weighted memory must beat uniform memory, and saliency corruption must remove the improvement.

Proposition 5 (Causal-memory falsifiability). If Retro-DARC improves score S over a parameter-matched adapter but interventions removing example-specific memory (zeroing r_l, cross-example shuffling) leave S unchanged within confidence intervals, the experiment does not support the claim that memory is responsible. ∎

Memory geometry. Queryable memories should preserve state geometry: $\mathcal{L}_{\text{geom}} = \sum_{i,j}(d_{\mathcal{M}}(M_i, M_j) - \rho\, d_{\mathcal{S}}(s_i, s_j))^2$. A licensed alternative: apply **SIGReg** [LeJEPA25] to anchor/memory embeddings. The identifiability result of [Klindt26] — linear-orthogonal identifiability enabling optimal latent planning (arXiv:2605.26379) — gives the regime in which SIGReg-regularized memory embeddings preserve plannable geometry.

9.1 New and sharpened results

Proposition 2′ (trainability order; sharpens Prop. 2). At insertion (γ=0), $\partial L/\partial \gamma = \langle \partial L/\partial h', W_O\, r\rangle$ is generically nonzero, while $\partial L/\partial W_O = \gamma\,(\partial L/\partial h')\, r^T = 0$ and the router/key gradients vanish identically, since every path from those parameters to the loss carries a factor γ. The first optimizer step therefore moves *exactly* the gate; everything else begins training at step two. *Remark 2′ (gate-first curriculum).* This ordering is desirable: the network first answers "does this read help?" (a one-parameter decision with an unbiased gradient) and only then shapes *what* is read — an automatic coarse-to-fine schedule with no tuning. Measured (§10, E2): step-0 gradient norms $\gamma = 1.19{\times}10^{-2}$ with $W_O = U_q = P_K = 0$ exactly; by step 5 all are nonzero and $\gamma = 4.53{\times}10^{-2}$.

Proposition 3′ (identity under activation-dependent control flow; generalizes Prop. 3). Let F be any network whose forward pass consists of deterministic blocks plus *control decisions* — expert routing, early-exit tests, iteration counts, cache/state updates — each a deterministic function of activations computed so far. Insert a zero-gated Retro-DARC read anywhere. Then every activation, every control decision, and every auxiliary state (including a KDA-style recurrent sequence state) equals the base model's, and outputs are bitwise identical under identical kernels. *Proof sketch.* Induct over the computation DAG in topological order: node inputs are unchanged by hypothesis; the zero-gated read adds $\gamma \cdot W_O\, r = 0$ exactly; so the node's activation is unchanged; control decisions and auxiliary states, deterministic in unchanged quantities, are unchanged. ∎ *Consequence.* Zero-gated retrofit is exact for K3-class architectures — sparse MoE (16-of-896 routing), hybrid KDA/full-attention stacks, early-exit and loop-adaptive variants — not only plain residual stacks. Verified (E1): a weight-tied loop with data-dependent early exit, top-1 MoE routing, and a gated recurrent state shows max deviation 0.0, identical exit steps, identical expert choices, bitwise-equal outputs; $\gamma = 10^{-3}$ departs immediately.

Lemma 2 (delta keys are well-conditioned addresses). With $H = T\Delta$ (T the lower-triangular all-ones matrix over L layers), $\kappa(H) \le \kappa(T)\cdot\kappa(\Delta)$ and $\kappa(T) = \Theta(L)$ (singular values $1/(2\sin((2k-1)\pi/(4L+2)))$); for near-isotropic deltas $\kappa(\Delta) = O(1)$ while $\kappa(H)$ inherits $\Theta(L)$ growth, and adjacent cumulative keys' cosine $\approx \sqrt{i/(i+1)} \to 1$, so a linear scorer's *retrieval margin* over cumulative keys shrinks with depth and its noise robustness degrades proportionally, while delta keys keep $\Theta(1)$ margins. Measured (E3, d=128): $\kappa(\Delta) = 1.36/2.08/2.85/5.73$ vs. $\kappa(H) = 10.7/25.4/60.6/206.8$ at $L = 8/16/32/64$; adjacent cosines ≈0 vs. 0.849→0.968; retrieval under key noise σ=0.5: deltas 100% at all depths, cumulative 100→92.6%. *Reading.* This is the precise sense in which delta routing is better *addressing* than output routing — the design difference between Retro-DARC/Delta-AttnRes and plain AttnRes — and it is a depth-and-noise effect: at shallow depth, cumulative redundancy can even help (E7's honest finding), which is why §10.1 states the delta advantage conditionally and §11.3a keeps the scaled comparison registered rather than assumed. *Measured domain (per E13, five public checkpoints):* the near-isotropy premise fails toward the top of real stacks — delta norms grow gradually over the final ~3 layers and jump 4.9–

15.9× at the last layer — but the failure is almost purely *scale*: per-slot $\ell 2$ normalization of the keys restores $\kappa = 3.0$–3.7 for the full bank, final layer included, on all five models (raw: 9–28). Lemma 2's conclusion therefore holds for **normalized delta keys over the whole stack**, or raw delta keys below the top ~3 layers; E13b confirms the addressing consequence directly (delta-key top-1 retrieval 0.86–0.99 — 0.57–0.63 in gpt2's k=11 bank, the disclosed exception — vs. output-key 0.04–0.75 over all cells, under noise and simulated int4).

Proposition 6 (Softmax1 null-mass bounds). With k selected candidates whose logits lie in [m, M], the null mass $p_\varnothing = 1/(1+\Sigma_j\, e^{s_j})$ satisfies $1/(1+k\, e^{M}) \le p_\varnothing \le 1/(1+k\, e^{m})$: the null route is never starved nor saturated for finite logits, and its floor decays only inverse-exponentially in the *maximum* logit — a confident read must earn its mass. Verified (E4): 0 violations across 2×10^5 draws; measured means $p_\varnothing = 0.165$ (k=4, s~N(0,1)) and 0.079 (top-4-of-12; consistent with the original toy-suite 8.18% under its sampling).

Complexity refinement. In §12's per-layer cost, the dominant hidden term at large d is $W_O \in \mathbb{R}^{d\times d}$. We recommend the low-rank form $W_O = U_o V_o$ (rank r_o; params $O(d\cdot r_o)$); identity, Prop. 2′, and Prop. 3′ are unaffected since γ gates the whole product.

---

10. Executable Mechanism Checks

Before any expensive training, mechanism-level assumptions are checked exactly or on small synthetic programs — and, in this revision, on public pretrained checkpoints. The registration-stage checks (upper block, toy-suite values retained unchanged for the record; their raw outputs predate the released harness) are unchanged; the sandbox suite added the E1–E9 block via the released `retro_darc_checks_v13.py` (single CPU, ~80 s, fixed seeds; raw outputs in `results_v13.json`); the public-checkpoint stage adds rows **E10.1–E15**, executed on real open weights (gpt2, pythia-410m/1.4b, Qwen2.5-0.5B/1.5B; Apple M5 Max and a DGX Spark replication node; scripts `task1…task6*.py` + `depth_adapter.py`, raw outputs in the released per-run JSONs, per-run conditions in `RUN_NOTES.md`).

Check	Metric	Result
Zero-gated adapter identity	max abs hidden-state error	**0.000**
First-step trainability	$\lvert\nabla_\gamma\rvert$ with $W_O \neq 0$	**4.70×10^{-3}**
Delta vs. cumulative redundancy	adjacent cosine (cumulative vs. delta)	**0.9468 vs. -6.5×10^{-4}**
Token-conditioned retrieval	accuracy (token vs. static query)	**92.6% vs. 16.2%**
Residual-saliency retention	sparse-event recall (top-k vs. uniform)	**97.0% vs. 12.6%**
Delta memory readout	MSE (ideal delta vs. final cumulative)	**0.216 vs. 0.896**
Physics-residual memory	MSE reduction with residual memory	**66.4%**
Abductive causal memory	MSE reduction vs. no causal memory	**99.8%**
Latent-action memory	MSE reduction vs. no action memory	**99.8%**
Queryable 4D memory	MSE reduction vs. static 4D baseline	**99.7%**
Memory geometry	state–memory distance correlation (preserved vs. warped)	**1.000 vs. −0.013**
Softmax1 null reserve	mean null mass under random logits	**8.18%**
E1 Identity under control flow *(new)*	max deviation through MoE routing + early exit + KDA-style state; exit steps / expert choices equal	**0.0; bitwise equal; yes/yes** ($\gamma=10^{-3}$ departs)
E2 Gate-first gradient order *(new)*	step-0 grad norms (γ; W_O; U_q; P_K) → step-5	**1.19×10^{-2}; 0; 0; 0 → all > 0** ($\gamma=4.53\times10^{-2}$)
E3 Delta-key conditioning *(new)*	κ(delta) vs. κ(cumulative), L=8→64; retrieval @ key-noise σ=0.5	**1.36→5.73 vs. 10.7→206.8; 100% vs. 100→92.6%**
E4 Softmax1 null bounds *(new)*	bound violations / mean $p_\varnothing$ (k=4; top-4-of-12)	**0 in 2×10^5; 0.165; 0.079**
E5 Loop-iterate delta probe *(was planned; run)*	test MSE, final-state-only vs. +loop-delta memory (deterministic rollout)	**1.29×10^{-4} vs. 1.14×10^{-4} (−11.4%)**; shuffle → 8.86×10^{-4}
E5b …with exogenous rollout noise *(new)*	same, when the trajectory carries information the final state cannot	**3.07×10^{-3} vs. 2.09×10^{-3} (−32.0%)**; shuffle ×7.1, zero ×3.9
E6 Chunk-summary probe *(was planned; run)*	event-content cosine, uniform vs. saliency-weighted pooling (chunk-hit rates saturate at 99.5/99.7%)	**0.758 vs. 0.883 (+16.5% rel.)**

Check	Metric	Result
E7 End-to-end miniature, L=12 *(new)*	param-matched (~18.9k) test MSE: final-only / output-keys / delta-keys+null; then interventions on the delta head	**0.0217 / 0.0151 / 0.0177**; zeroed **0.0345 (×2.0)**, shuffled **0.0529 (×3.0)**, same-norm random **0.0526**
E7b …at toy depth L=24 *(new; null result)*	same protocol	all heads degrade to $R^2 \approx 0.5$; memory reads ≈ no help (zeroed slightly *better*); gates stay small
E8 Prefix-consistency residual *(new)*	Pearson r(‖direct−composed‖, true error); error of most-consistent half	**r = 0.455; −10.1%**
E9.1 Real-checkpoint pretraining *(new, addendum)*	8-layer char-LM (417 K params) on 3 MB real Python source; val CE	**4.78 → 1.520** (1000 steps, single CPU core)
E9.2 Identity on the real checkpoint	max abs logit deviation with zero-gated insert at layer 6	**0.0 (bitwise equal)**
E9.3 Gradient order on the real checkpoint	step-0 grad norms under the real LM loss (γ; W_O; U_q; P_K)	**8.2×10^{-4}; 0; 0; 0**
E9.4 Key conditioning on *learned* activations	adjacent cosine / median per-token κ, outputs vs. deltas (6 slots)	**0.928 vs. 0.236 / 16.4 vs. 2.5**
E9.5 Adapter-only continued training (600 steps, identical data order)	Δ val CE vs. frozen (trainable params): LoRA r=10 / MLP adapter / AttnRes-style outputs / RD-Lite delta+null	**+0.0764 (8,960) / +0.0768 (9,094) / +0.0585 (7,193) / +0.0570 (7,193)**; gates open to 0.132 / 0.343
E9.6 Causal interventions on the trained depth heads	val CE under shuffle / zero / same-norm random (frozen = 1.5240)	AttnRes-style: **1.5187 / 1.5240 / 1.5406**; RD-Lite: **1.5131 / 1.5240 / 1.5327** — 81–100% of the gain removed; random *worse* than frozen
E10.1 Identity on public checkpoints *(new, E10 addendum)*	max abs logit deviation, zero-gated insert at layer 2L/3 on gpt2 / pythia-410m / Qwen2.5-0.5B (8 prompts each, fp32 CPU)	**0.0 on all 3 models × 8 prompts (bitwise equal)**; $\gamma=10^{-3}$ departs (1.1–3.7×10^{-2}); hook removal restores the base bitwise
E10.2 Conditioning on public checkpoints *(new, E10 addendum)*	median per-token κ / adjacent cosine, outputs vs. deltas, k=4→L on gpt2 / pythia-1.4b / Qwen2.5-1.5B (wikitext-2, 2,048 tokens)	κ_out grows ≈linearly (7.8→75.9) while $\kappa_\Delta \le 6.6$ at every interior k — pythia-1.4b k=8: **16.9 vs. 2.6**, mirroring E9.4; adj-cos **0.87–0.97 vs. −0.13–0.23**. *Boundary:* the last layer's own delta is ill-conditioned (κ_Δ 13–28 at k=L; survives a raw-last-block-output control), so the O(1) claim holds for interior prefixes
E11 AttnRes-recovery (twin pretrain, 3 seeds) *(new; §11.3a executed at miniature scale)*	A plain vs. B AttnRes-from-scratch, 1600 steps, identical schedules; C/D = frozen A/B@1000 + RD-Lite 600 adapter-only steps; interventions on D	**Prediction "B ≥ A" violated on all 3 seeds** (B−A = +0.0142 CE, range +0.0100…+0.0206; B's built-in γ decays 1.0→≈0.57). **Registered question (ii) resolves positive, not null**: D gain **+0.0313** (range +0.0280…+0.0337) ≈ C gain +0.0321; shuffle removes ~100% of D's gain on every seed, zeroing returns to frozen-B **exactly**, same-norm random is *worse* than frozen
E12 Adapter comparison on a public checkpoint *(new; S3-miniature)*	frozen pythia-410m (INS=16), wikitext-103, 1,500 steps ≈3.1M tokens, 3 seeds, matched ~262k trainable params: LoRA r18 / MLP / AttnRes-style / RD-Lite, val CE (frozen = 3.3400); then interventions on the depth heads	**2.9907 / 2.9528 / 2.9655 / 2.9764** — same ordering on all 3 seeds (range ≤ 0.004); **both depth reads beat LoRA**; zeroing returns to frozen **exactly** (6/6 head×seed), shuffle removes 97–107% (AttnRes-style) vs. ~70% (RD-Lite) of the gain, same-norm random *worse* than frozen; step-0 gradient order exact; gates open to 0.09 / 0.16
E13 Last-layer delta anomaly, localized *(new; E10.2 follow-up)*	per-layer delta norms, prefix/window κ, and a per-slot-normalization control on 5 public checkpoints (raw block outputs, no post-norm confound)	final-delta norm ratio ρ_L = **4.9–15.9 (5/5)**; ill-conditioning is *gradual over the top ~3 layers + final jump*, not a one-layer spike (registered spike prediction **failed 0/5**, reported verbatim); normalization removes **92–99%** of the κ excess (5/5) — the anomaly is norm growth, not directional degeneracy; the registered single-layer-exclusion bet also failed ($\kappa([1..L-1]) \le 8$ held on only **2/5** models — the top ~3 layers are involved); measured rule: normalize keys, or exclude the top ~3 layers' deltas
E13b Retrieval under key corruption on public checkpoints *(new; cashes out §10.1(1))*	top-1 slot retrieval, delta vs. output keys, k up to L−1, gaussian σ = 0.25–1.0×RMS and simulated int4, 5 models	delta keys **0.86–0.99** (one disclosed exception: gpt2 k=11, with the anomalous final layers in-bank, 0.57–0.63 — still ≥6× the output keys); output keys **0.04–0.75**, generally collapsing with k (pythia-1.4b k=23: 0.91–0.98 vs. 0.08–0.14) — Lemma 2/E3 measured on real learned activations, including the quantized-keys case
E14 From-scratch read: init vs. keys vs. routing *(new; E11 follow-up; keys × routing factorial at γ_0=0 + γ_0=1 shock pair, 3 seeds, A/B re-run — reproduced E11 exactly, env-drift 0)*	val CE @1600 vs. A = 1.6792: B(out,sm,γ1) / Bd1(Δ,sm1,γ1) / B0(out,sm,γ0) / Bn0(out,sm1,γ0) / Bdp0(Δ,sm,γ0) / Bd0(Δ,sm1,γ0)	**1.6934 / 1.6875 / 1.6816 / 1.6812 / 1.6752 / 1.6778** — the E11 violation is **γ_0=1 initialization shock** (zero-init beats γ_0=1 on 3/3 seeds, both bundles); zero-init reads recover to ≈parity with A (B0 within +0.005 on 2/3 seeds); keys and routing each tie at the registered contrasts (≤0.0034 vs. seed range 0.011; the unregistered keys-at-plain-softmax contrast is 0.0064, in delta's favor); delta-key γ_0=0 cells have the best means (Bdp0 1.6752 < A) — reported as observation, registered for the S3 pilot

Check	Metric	Result
E15 S2-gate probe at 50M tokens *(new; seed 0 on Apple M5 Max/MPS + seed-1 replication on DGX Spark GB10/CUDA, reported separately, never pooled; frozen pythia-410m, INS=16, 16× E12's budget, E12 trajectory prefix reproduced exactly; batch stream digest-identical across machines)*	val CE at 24,414 steps: MLP / AttnRes-style / RD-Lite (frozen 3.3400); interventions; gap-vs-budget	**2.9334 / 2.9395 / 2.9481** — registered prediction (i) **fails**: output keys beat *unnormalized* delta keys by 0.0086 (> 0.005 tie bar) at mid-stack insertion — the E7/E9/E12 redundancy regime confirmed at real token budget, reported verbatim (probe specced before E13's normalization fix; normalized-key rerun registered in §13.2(1)). (ii) **passes**: zero returns to frozen exactly (both heads), shuffle removes 103.5% / 76.9%. (iii) **passes**: the weight-toucher lead *narrows* with budget (0.0110→0.0062 vs. AttnRes; 0.0252→0.0147 vs. RD-Lite). **Replication (seed 1, independent hardware, torch 2.13/CUDA vs. 2.8/MPS): every verdict reproduces** — finals 2.9394 / 2.9475 (−0.0001 / −0.0006 vs. seed 0), gate (i) gap +0.0081, zero within 1×10^{-7} of frozen (CUDA reduction-order noise), shuffle removes 104.4% / 80.4%

Reading E5/E5b. In the deterministic rollout the conditioning input fully determines the trajectory, so a final-state head can in principle solve the task — and nearly does; the delta memory still helps (−11.4%) and is causally used (shuffle ×7.8). When per-step noise makes early corrections *unrecoverable from the final state*, the gap opens to −32.0% with a ×7.1 shuffle penalty: loop-iterate deltas carry exactly the information the collapsed state discards, which is the mechanism claim of §8.

Reading E7/E7b honestly. The trained miniature *confirms the contract's shape* — memory heads beat a parameter-matched final-state head, the gate opens from zero (|γ|: 0.003 → ≈1.5), and content-destroying interventions remove the gains (×2–3 error) — and *disconfirms a naive expectation*: at L=12, plain output-key reads (AttnRes-flavored) slightly beat delta-key reads (0.0151 vs. 0.0177). The mechanism is redundancy: every cumulative state h_{l≥c} contains Δ_c, so an imperfect router has many acceptable rows, while delta reads demand an exact hit. At L=24 the toy base's state norm doubles while deltas stay flat and the head's capacity saturates ($R^2\approx0.5$ for *all* heads; gates stay small; zeroing memory marginally helps) — a capacity-bound null we report as-is. Lemma 2 and E3 isolate what E7 cannot: cumulative-key addressing degrades with depth and key noise while delta addressing does not. Together they yield the paper's conditional claim, stated in §10.1 and tested at scale in §11.3a: **delta keys buy conditioning, auditability, and depth/noise robustness; output keys buy shallow-depth redundancy; production designs should choose per regime — or route over both with the null option arbitrating.**

Reading E9 (the real-checkpoint addendum). E9 replaces E7's frozen-*random* stack with a frozen-*trained* one: an 8-layer LM actually pretrained here on real Python source (hub domains are unreachable from the sandbox — verified — so "real checkpoint" means *really pretrained on real data*, at miniature scale; the public-open-weights run stays registered). Four things transfer from theory to the trained model unchanged: bitwise insertion identity, gate-first gradient order under the true LM loss, and — bridging Lemma 2 from synthetic keys to learned representations — the conditioning gap (output keys: adjacent cosine 0.928, median per-token κ 16.4; delta keys: 0.236, κ 2.5, at only 6 slots). The utility comparison is reported without spin: at this scale, weight-touching adapters (LoRA, MLP) win on raw CE (+0.0766 avg), while the pure depth-read heads recover ~75% of that gain from ~21% fewer trainable parameters and *only a read path* — and the delta head leans on its memory hardest (gate 0.343 vs. 0.132). What the depth heads uniquely offer is then demonstrated rather than asserted: zeroing the bank returns CE to the frozen value *exactly* (a structural audit no weight-touching adapter admits), shuffling removes 81–91% of the gain, and same-norm noise lands *below* frozen — the gain is memory-content, not regularization or scale. Output-vs-delta keys tie at depth 8 (Δ CE 0.0015), as the redundancy analysis predicts for shallow stacks; the regime claim of §10.1(1) is unchanged.

Reading E10–E15 (the public-checkpoint record). Rows E10.1–E15 replace the sandbox's central limitation — no hub access — with the registered tests run on real open weights. They are best read as one four-step argument, closed by its standing negative.

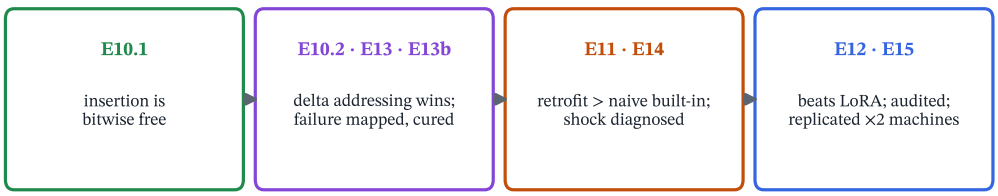

Figure 2: The public-checkpoint record as one argument — each step measured, the last replicated across two accelerators.

Step 1 — insertion is exactly free. E10.1 upgrades Proposition 1 from theorem-plus-miniature to a measured property of shipping checkpoints: bitwise logit identity across three model families, liveness at $\gamma{=}10^{-3}$, bitwise restoration on detach. Every zero-gated retrofit run that followed re-verified the Prop. 2′ gradient order exactly — on CPU, MPS, and CUDA backends (E11 C/D, E12, E15, the E15 replication (seed 1)) — so the gate-first curriculum is an observed invariant, not an aspiration.

Step 2 — the addressing claim survives real activations, and its one failure taught the design rule. Output-key condition numbers grow with depth exactly as Lemma 2 predicts (to 76 at k=28; three models) while interior delta keys stay O(1) (< 6.7); the boundary discovered in E10.2 — ill-conditioning at the top of the stack, against the registered O(1) sweep — was chased down (E13, five models) to gradual *norm* growth capped by a 4.9–15.9× final-layer jump, its registered spike-not-drift bet holding on 0 of 5 models and single-layer exclusion on only 2 of 5, and per-slot normalization removes 92–99% of the excess, leaving full-bank κ at 3.0–3.7 on all five models. E13b then converts conditioning into the quantity that matters: under key noise and int4-style corruption, delta addressing retrieves at 0.86–0.99 (one disclosed exception, gpt2's full-stack bank at 0.57–0.63, still ≥6× the alternative) where output addressing collapses to 0.04–0.75.

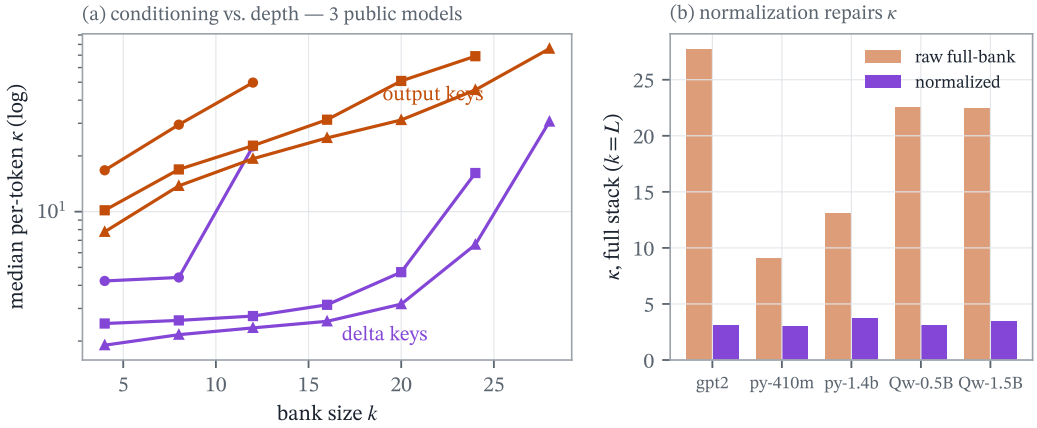

Figure 3: (a) Output-key condition numbers grow with bank size while interior delta keys stay flat (three public models; markers per model; the top point of each curve is the anomalous full stack). (b) Per-slot $\ell 2$ normalization repairs full-stack conditioning on all five models (E13).

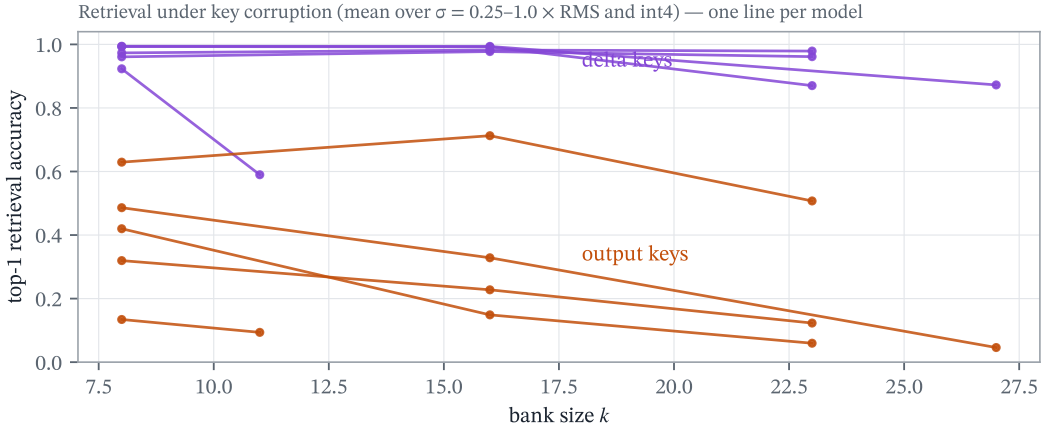

Figure 4: Top-1 retrieval under key corruption (mean over noise levels and simulated int4), one line per model: delta keys hold near ceiling while output keys collapse with depth; the one delta dip is gpt2's anomalous full-stack bank (E13b).

Step 3 — retrofit is the right acquisition path at this scale. The twin pretrain (E11) falsified its own registered prior — building the read in at $\gamma_0{=}1$ *hurt* — and the E14 factorial shows why: pure initialization shock, absent at $\gamma_0{=}0$, orthogonal to key type and routing. Meanwhile the retrofit's increment on top of an architected-retrieval base is as large as on a plain base, fully shuffle-sensitive, and exactly zero-out-recoverable: architected retrieval and typed retrofit memory are complements, not substitutes.

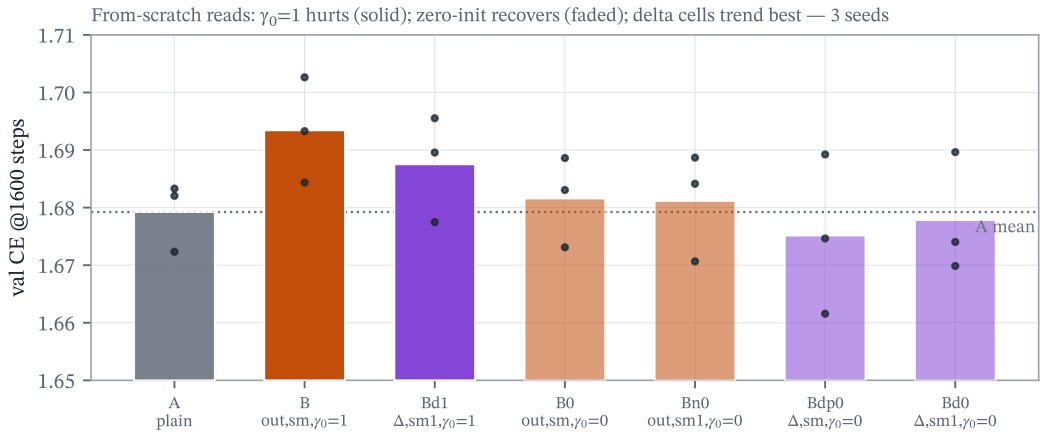

Figure 5: The from-scratch factorial (E11/E14, 3 seeds; dots are seeds). Full-gate cells (solid) sit above the plain base; zero-init cells (faded) recover to ≈parity, delta-key cells trending best. The E11 violation was initialization shock, not architecture.

Step 4 — at matched budgets the depth reads are competitive and uniquely auditable. Against parameter-matched LoRA and MLP adapters on a converged public checkpoint (E12, 3 seeds ≈3.1M tokens each, identical ordering), both depth reads beat LoRA; the MLP adapter's remaining lead *narrows* as the token budget grows 16× (E15); and the depth-read arms of E15 replicate across seed and accelerator with final CEs agreeing to $\leq 6\times10^{-4}$ on a digest-verified identical batch stream. The audit itself is the point: with every trained parameter left in place, destroying only the memory content removes 70–107% of the gain (shuffle) and same-norm random content lands *below* frozen — an attribution of the gain to retrieved content that a weight-touching adapter cannot express, because ablating it removes content and mechanism together. Behind it sits the machine-checkable integrity test: across every zeroed-bank case in E9/E11/E12/E15 (18 in all), the frozen loss returns bit-exactly in all 16 CPU/MPS cases and to $\leq 1\times10^{-7}$ in the two CUDA cases (kernel reduction-order noise, reported verbatim in the replication node's run report).

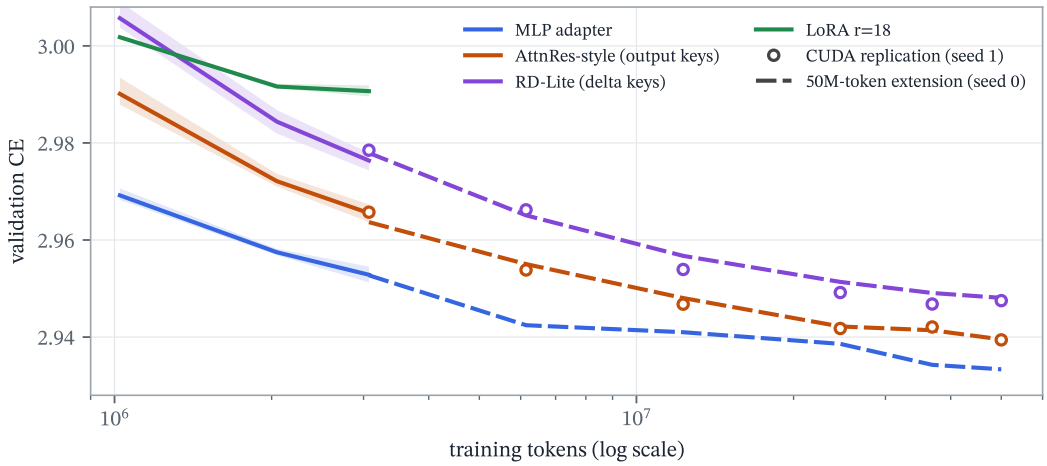

Figure 6: Matched-parameter adapters on frozen pythia-410m. Solid: 3-seed mean ± range at the E12 budget; dashed: the 50M-token extension (seed 0); open circles: the CUDA replication (seed 1), agreeing to $\leq 6\times10^{-4}$ CE. Frozen base CE 3.3400 (off-scale above). Both depth reads beat LoRA; the MLP lead narrows with budget.

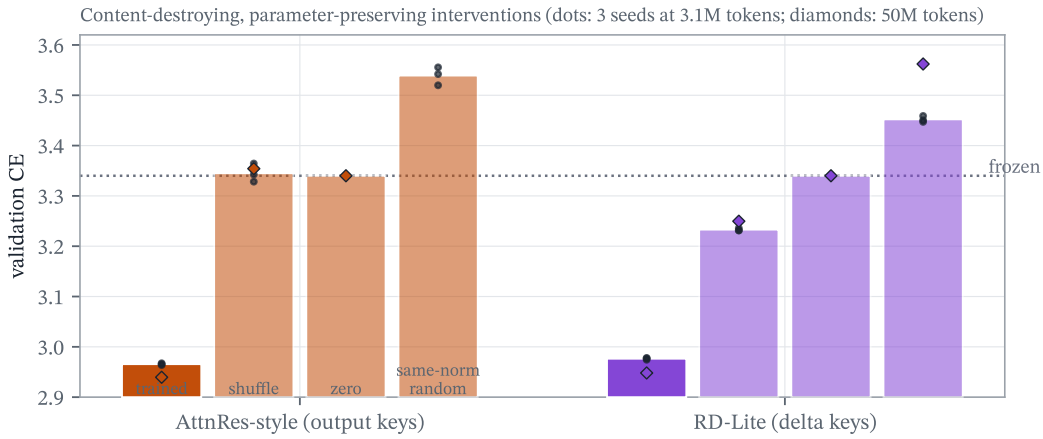

Figure 7: The causal audit. Zeroing returns the frozen loss (bit-exactly on CPU/MPS); shuffling removes 97–107% (output keys) vs ≈70–80% (delta keys) of the gain; same-norm random lands above frozen. Dots: 3 seeds at 3.1M tokens; diamonds: 50M tokens.

The standing negative, stated with equal prominence. With unnormalized keys at mid-stack insertion, output keys win raw CE by 0.0086/0.0081 (two machines) at 50M tokens — the shallow-redundancy regime of E7/E9 at real budget. The delta-utility claim therefore remains *conditional* exactly as §10.1(1) states it, with the normalized-key test registered as the next attempt.

10.1 What K3-class models can adopt from Retro-DARC — five upgrades, with the evidence status of each

Kimi K3 validated depth-selective retrieval at 2.8 T parameters. The checks above identify five concrete deltas between K3's shipped practice (as described in its launch materials) and this paper's design — each stated with what is *proven*, what is *measured at toy scale*, and what remains *registered*:

1. **Delta keys for depth addressing (proven conditioning; now public-checkpoint-measured for addressing; utility scale-registered).** AttnRes retrieves layer *outputs*; Lemma 2 proves cumulative keys' condition number grows $\Theta(L)$ and E3 measures $\kappa = 206.8$ at $L = 64$ with retrieval degrading under key noise — directly relevant to quantized serving (K3 runs MXFP4/MXFP8, where key noise is not hypothetical). *Public-checkpoint upgrade:* E10.2/E13b measure this on five public checkpoints — output-key κ grows to 76 while normalized delta-key κ stays 3.0–3.7 over the full stack, and top-1 retrieval under noise/int4 is 0.86–0.99 for delta keys (0.57–0.63 in gpt2's k=11 bank, the disclosed exception) vs. 0.04–0.75 for output keys over all cells, collapsing with depth. Two measured design rules attach: **normalize keys per-slot** (E13: removes 92–99% of the top-of-stack κ excess, which is norm growth over the final ~3 layers, not directional degeneracy), and note the honest utility counterweight now measured at 50M tokens (E15): with *unnormalized* keys at mid-stack insertion, output keys still win raw CE by 0.0086 while delta banks retain uniquely shuffle-robust content — the registered prediction stands for deep, quantized, block-local regimes, with the normalized-key utility test next (§13.2(1)).
2. **A null route with proven mass bounds (proven; toy-measured).** K3's AttnRes has no abstention option; OASIS documents the sink/outlier pathologies of forced attention. Prop. 6 bounds Softmax1's null mass on both sides (verified, 0 violations), giving a depth-retrieval mechanism that can *decline to retrieve* with predictable calibration — at zero cost when logits are confident.
3. **Gate-first retrofit instead of from-scratch-only (proven; bitwise-verified through K3's own control flow).** Prop. 3′ + E1 establish that zero-gated insertion is *exact* through sparse MoE routing and KDA-style recurrent state — the two mechanisms that make K3 non-trivial to modify. Prop. 2′ says the retrofit trains itself in the right order. Concretely: the 2.8 T checkpoint (or any K3-class open model) can gain a typed depth-memory read *without pretraining and without behavioral risk at insertion*; §11.3a measures how much of the from-scratch gain that recovers.
4. **Typed, intervenable memory as a deployment-safety surface (toy-measured; scale-registered).** K3's AttnRes weights are neither typed nor auditable. E7's intervention battery (zero/shuffle/same-norm ×2–3 error) is precisely the audit a safety review can run on a Retro-DARC-equipped model: demonstrate which behaviors depend on which memory type, then gate or ablate them selectively — a capability no uniform-softmax depth mixer offers.

5. **Cross-axis evidence sharing (registered, untested).** K3 computes, then discards, a per-step KDA state-update magnitude — a sequence-axis residual signal. The Residual Evidence Principle predicts it is a useful *saliency prior for depth routing* (large sequence-state corrections mark tokens whose layer deltas deserve memory). We register this as the cheapest untested cross-axis experiment in §11.

11. Verification Plan: Stage by Stage

11.1 Primary estimands and interventions

With intervention operator $\mathcal{I}$ on the memory bank (weights, prompts, decoding, scaffold fixed), the **causal memory effect** is $\mathrm{CME}(\mathcal{I}) = \mathbb{E}_x[S(f_\theta, x) - S(f_\theta^{\mathcal{I}}, x)]$. Interventions: (1) zero memory r_l=0; (2) force null $\alpha_\varnothing$=1; (3) cross-example shuffle; (4) same-norm random memory; (5) depth-offset; (6) local-only; (7) route corruption; and, added at registration, per-type ablations — (8) **action-memory shuffle**, (9) **trace-memory shuffle**, (10) **failure-memory removal**, (11) **verification-memory removal**, (12) **loop-iterate-delta removal** (in loop-composed runs). Negative controls: short-horizon tasks, trivial continuations, and local-syntax probes should show small effects; uniform fragility indicates instability, not memory.

11.2 Registered metrics

Beyond validation CE and task success: paired bootstrap CIs; Holm–Bonferroni across benchmark families; overhead ratio ρ; and, for every world-model stage, the field's shared long-horizon battery — **rollout-length curves**, **compounding ratio** $\mathrm{CR}(k) = e_k/(k\,e_1)$ at $k \in \{30, 60, 120, 240\}$, **error-growth rate** ER, **temporal straightness** of latent trajectories (SkyJEPA's diagnostic), **rollout-length-to-threshold** $L^*(\epsilon)$, and the **physics-slop index** PSI. For interactive/generative stages, the program additionally registers the MemoBench-cluster instruments: **Object-Reappearance Score** and the **memory-continuity VQA pillar** under a Visible→Disappear→Reappear protocol, together with the camera-controllability check that guards against the "static camera inflates consistency" loophole MemoBench documents [MemoBench26]. One primary validation metric and one primary long-horizon metric are pre-registered per stage to prevent benchmark shopping.

11.3 Baselines (frozen at Stage 0)

(1) continued training only; (2) LoRA (matched trainable params); (3) residual adapter; (4) same-parameter MLP adapter; (5) cumulative hidden-state depth adapter; (6) Delta-style depth adapter; (7) null-aware depth adapter; (8) Retro-DARC-Lite; (9) Depth-Attention-inspired in-cache baseline (systems bar); **(10)** matched-FLOP weight-tied recurrent-depth baseline (§8.3); **(11, WM stages)** latent-action IDM/FDM adapter (MWA-style) at matched parameters; **(12, new) an AttnRes-style depth-retrieval adapter** — a zero-gated adapter whose read is a plain softmax over recent layer *outputs* (K3/Delta-AttnRes-flavored) rather than over saliency-selected *deltas with null routing and typed banks* — so the ablation isolates what Retro-DARC's design (delta content, Softmax1 null, token-conditioned query, residual-aligned retention, typed memory) adds over a faithful retrofit of the production AttnRes primitive itself.

11.3a The AttnRes-recovery question (registered before execution)

Kimi K3 obtained depth memory by pretraining Attention Residuals in from scratch, reporting a favorable efficiency ledger. Retro-DARC obtains depth memory by zero-gated *retrofit* on a frozen checkpoint. The head-to-head is therefore not "which is better" but **"what fraction of the from-scratch AttnRes benefit is recoverable post-hoc?"** On an open base with a published from-scratch AttnRes/Delta-AttnRes counterpart at the same scale (or an internally trained one), we register: (i) validation-CE and long-horizon-metric gap between (frozen base + Retro-DARC-Lite) and (from-scratch AttnRes model) at matched trainable-parameter *and* matched continued-token budgets; (ii) whether combining them (retrofit on top of an AttnRes-pretrained base) still yields a positive, memory-shuffle-sensitive increment — i.e., whether *typed, queryable* memory adds value on top of *architected* depth retrieval. A null result on (ii) would be an informative boundary on the retrofit thesis; a positive one would show the two are complementary.

Measured at miniature scale (E11/E14, 3 seeds, internally trained proxy — limitation §14(7) applies). The twin pretrain returned answers to both registered questions plus one surprise. *The surprise:* the from-scratch AttnRes proxy at its registered configuration (output keys, plain softmax, γ_0=1) was *worse* than plain pretraining on every seed (+0.0142 CE mean) — and the E14 factorial attributes this to **γ_0=1 initialization shock**, not to the architecture: with zero-init gates, from-scratch reads recover to ≈parity with the plain base (and the delta-key cells trend mildly *better* on mean), while key type and routing each tie when the other factor is fixed. This converts the gate-first principle (Prop. 2′) from a retrofit convenience into a measured design rule for from-scratch builds as well. *Question (ii) resolved positive, not null, at this*

scale: RD-Lite retrofit on the frozen AttnRes-pretrained base gains +0.0313 mean (≈ the plain-base +0.0321), shuffle removes ~100% of it on every seed, zeroing returns to the frozen value exactly, and same-norm random lands worse than frozen — typed delta memory and architected retrieval were complementary, not redundant. The at-scale version of (i)/(ii) against a *published* from-scratch counterpart remains registered — and has just become concrete: K3's open weights and technical report are announced for release on 2026-07-27, and Kimi Linear (48B/3B, AttnRes-validated, checkpoints released) is public now. **We therefore register the first-contact experiment**: run the released identity + conditioning + intervention harness on an AttnRes-pretrained open checkpoint (Kimi Linear first; K3-class when weights land) — (a) verify zero-gated insertion identity through KDA + AttnRes control flow (Prop. 3′'s exact prediction), (b) measure output-vs-delta conditioning and retrieval-under-quantization on a model whose *native* depth routing is output-key (directly testing whether AttnRes-pretrained representations change the E10.2/E13b picture), and (c) ask question (ii) at real scale: does an RD-Lite retrofit add a shuffle-sensitive increment on top of native AttnRes, as it did on the miniature twin (D vs. frozen-B)?

11.4 Stages and gates

- **S0 Literature lock & baseline freeze** (3–5 d): fix claims, baselines, harness.
- **S1 Identity & gradients** (1–2 d): $\max_x \|z_{\theta,\phi_0}(x) - z_\theta(x)\|_\infty < \epsilon$ on a toy net and ≥1 open LLM (discharged in miniature by E9.2 on a locally-pretrained real-code LM; public-checkpoint run executed: gpt2, EleutherAI/pythia-410m, Qwen/Qwen2.5-0.5B, bitwise identity confirmed — see E10.1); $\|\nabla_\gamma\| > 0$; *repeat inside a small weight-tied loop model per Proposition 3.*
- **S2 Frozen delta-memory probes** (2–5 d): 0.5–0.8B frozen model, 50–200M tokens; readout/router only; deltas must beat cumulative states; shuffling must hurt. *(Measured status, E15: probed at the 50M-token minimum on frozen pythia-410m, two seeds on two independent machines (M5 Max/MPS, DGX Spark/CUDA; digest-identical batch stream; reported separately) — the shuffling half passes decisively on both (zero returns to frozen exactly; shuffle removes 77–104%), but "deltas beat cumulative" fails on both with unnormalized keys at mid-stack insertion (+0.0086 / +0.0081 CE for outputs, the E7/E9/E12 redundancy regime at real budget); the normalized-key rerun per E13's measured fix is the registered next attempt.)*
- **S3 0.6B adapter-only signal gate** (1–2 wk): open base (e.g., Qwen3-0.6B) frozen; adapters trained on 0.5–1B open-corpus tokens (FineWeb-Edu/Dolma), 3 seeds; must beat ≥1 strong parameter-matched adapter *and* the matched-FLOP loop baseline; zero/shuffle must remove a measurable fraction of the gain.
- **S4 1–2B confirmation** (2–4 wk) and **S5 3B confirmation** (3–6 wk): gains must survive scale; redesign retention if signal shifts.
- **S6 Causal interventions**: full battery of §11.1 on every successful checkpoint; convincing iff $S_{\text{full}} > S_{\text{shuffled}}$ and $S_{\text{full}} > S_{r=0}$ with CIs excluding zero on ≥1 state-maintenance workload.
- **S7 Long-horizon/code/tool**: LM validation CE; LongBench v2 subset; LiveCodeBench subset; SWE-bench Lite/Verified subset; Tau-bench harness; ASMP trajectory probes (§11.6). Report success, cost-normalized success, tool-error rates, and CME drops.
- **S8 Systems & quantization**: prefill/decode tok/s, peak VRAM, p50/p95 latency, router overhead; BF16/FP8 memory values; targets — decode ≤5–8%, prefill ≤8–12%, VRAM ≤10–15% (SkyJEPA's 9K-parameter/100 Hz result is the reminder that small, well-targeted modules suffice; overrun triggers the optimization path: smaller windows, shared projections, fused routing, FP8 banks, saliency-pruned candidates).
- **S9 Multimodal event memory** (LLaVA-OneVision-2 testbed): retrofit the 8B decoder (vision frozen); verify identity on image/frame-video/codec-video inputs; compare frame backend, codec backend, codec+Lite, codec+RA; evaluate JumpScore, temporal grounding, low-budget video QA; ablate codec-saliency shuffling vs. depth-memory shuffling to show the two evidence sources are complementary, with $\Delta_{\text{event-mem}} = S(\text{RD}) - S(\text{memory-shuffled RD})$.
- **S10 Trace-memory probe** *(new)*: on μ_0's released stack (open code/checkpoints, LeRobot-based), write predicted/extracted traces into an X-Trace bank for a small planner; evaluate trace forecasting and RoboCasa365 subsets; trace-shuffle must selectively damage manipulation decisions.
- **S11 Latent-action & abductive-causal memory**: latent video/action corpus; compare raw-state, latent-action (baseline 11), and X-Causal predictors; independently shuffle c_t, latent actions, ϵ_t^{phys}, verification memories; add a **counterfactual probe** (Causal-JEPA-style object masking) and an **Aether-style sample-efficiency probe**: success vs. number of demonstrations at $n \in \{10, 25, 50, 100\}$ with and without typed memory.
- **S12 Queryable 4D memory**: spatiotemporal point queries $(u, v, t_{\text{src}}, t_{\text{tgt}})$; interventions: static, same-norm random, cross-video, time-offset memory.

- **S13 Quadrotor closed-loop probe** *(new, sim-only)*: replicate SkyJEPA's domain-randomized pipeline; insert Retro-DARC into the latent dynamics (identity-checked per Proposition 3 where the dynamics is recurrent); pre-registered outcomes: CR(60), temporal straightness, tracking RMSE under prop-swap/payload shifts, each with memory-shuffle controls. This is the cheapest *physical-plausibility* stage in the ladder: no vision, tiny models, an existing recipe to match.
- **S14 RoboCasa-coupled evaluation** *(new)*: attach an X-configured planner to an open policy stack (GR00T-class or μ_0+expert) on RoboCasa GR1 TableTop / RoboCasa365 subsets; the registered claim is *not* leaderboard rank but a positive CME on long-horizon composite tasks with failure-memory removal selectively degrading boundary/recovery segments.
- **S15 Object-permanence / disappear-reappear probe** *(registered)*: on a MemoBench-style Visible→Disappear→Reappear protocol [MemoBench26], attach Retro-DARC-X (belief + trace + 4D memory) to an *interactive* world-model planner and measure Object-Reappearance-Score and the "memory-continuity" VQA pillar against the same model without the memory path. The registered, falsifiable prediction is directional and specific: the memory path should raise ORS *specifically for objects that changed state while occluded* — the exact case where the field's ten measured models all score below 0.6 and where MemoBench shows the gain must come from explicit state maintenance, not scale. Memory-shuffle and same-norm-random controls must remove the improvement. This stage is where Retro-DARC's thesis is most directly falsifiable against a published external benchmark.

11.5 Escalation gates (registered)

The miniature E12/E15 runs probe later rungs while S2's delta-vs-cumulative clause awaits its normalized-key resolution (§13.2(1)); no gate is claimed passed out of order, and no S3-scale claim is made (§14, limitation 1). Identity → Delta beats cumulative → beats parameter-matched adapters at 0.6B → beats matched-FLOP loop at 0.6B → survives 1–2B → causal interventions remove gains → ≥1 long-horizon task improves → systems overhead within bounds → per-type interventions selective (action/trace/failure) before any world/action claim is emphasized.

11.6 ASMP (Action-State Maintenance Probes)

Cheap pre-robotics trajectories $x = (o_1, a_1, y_1), \ldots, (o_T, a_T, y_T), q_T$ from text POMDPs, tool traces, code edit/test/error/fix logs, and embodied-text worlds; targets: final hidden state, next valid action, invariant violation, repair step. Causal drops $\Delta_{\text{mem}}, \Delta_{\text{shuffle}}, \Delta_{\text{action}}$; evidence of action-relevant memory = Δ_{action} growing with trajectory length/dependency depth. Ladder: ASMP-Synthetic → ASMP-Tool → ASMP-Code → ASMP-EmbodiedText → small VLA/WAM pilot only after all four succeed.

---

12. Systems and Complexity

Per-token router cost per inserted layer $O(Md_r + kd)$ with $M = w + N_B + K + 1$ (chunking divides the delta term by $|C|$); parameter overhead $O(d_r + rd_r + dd_r + dd_{\text{out}})$, reducible by sharing P_K, W_O across blocks. Memory layout $[B, T, M, d]$ for prefill, $[M, B, d]$ for decode (profile per hardware). Detach mode ($M_m \leftarrow \text{stopgrad}(M_m)$) keeps pilot training PEFT-like. Depth-Attention-style in-cache mixing sets the overhead bar and is reported alongside.

Two axes, complementary. Kimi K3's design cleanly separates the *sequence* axis (KDA: a gated-delta-rule linear-attention recurrent state, calibrated by periodic full-attention layers, for cheap million-token history) from the *depth* axis (Attention Residuals / Block AttnRes: selective cross-layer retrieval) [KimiK3-26; KimiLinear25]. Retro-DARC lives entirely on the depth axis and is agnostic to the sequence-axis mechanism: it inserts identically into a full-attention base, a KDA/linear-attention base, or a hybrid, because it reads layer *deltas* at a position, not the KV cache. Block Attention Residuals independently validate our block-local-routing and chunk-summary choices (§5.5, §7.4) as the right way to keep cross-depth memory affordable at scale. A model that adopts KDA for sequence-length efficiency and Retro-DARC for depth memory pays both costs once and gets a compressed running summary on *each* axis — the natural end state this paper's §8 (loop composition) and §12 point toward. On the kernel level, the depth-read is a small batched attention and inherits the field's low-bit/sparse attention toolchain directly — SageAttention-class low-bit kernels and sparse-linear-attention variants [SageAttn25], which the Vidu S1 line reports as the workhorse of *compute-bound* multimodal attention, apply unchanged to the router's score/read matmuls, so the systems bar of §11 S8 should be evaluated with, not without, them.

---

13. Mechanism Disambiguation

Alternative explanation	Why it could explain gains	Disambiguating test
Extra trainable capacity	any adapter helps continued training	matched-parameter LoRA/residual/MLP controls
Recurrent compute *(new)*	more iterations, not memory, drive gains	matched-FLOP weight-tied loop baseline; (loop + RD, shuffled) vs. (loop + RD, intact)
Optimization regularization	zero gate/router stabilizes training	memory-shuffle and route-corruption; no-memory gated adapter
Scale/norm effects	large vectors shift activations regardless of content	same-norm random memory; RMS-matched shuffles
Local smoothing	only recent residuals matter	local-only vs. chunk/anchor memory; horizon-stratified eval
Data/scaffold advantage	order, prompts, harness, decode policy	fixed order, paired eval, public configs
Codec prior only	saliency helps without depth memory	codec-alone vs. codec-conditioned routing vs. metadata shuffle
Trace/latent-action representation alone *(new)*	the typed representation, not its *memory*, helps	frozen representation with memory zeroed vs. intact memory
World-model confounding	typed memories aid representation, not action dynamics	action-memory and verification-memory ablations on prediction *and* planning
Failure-data curation alone *(new)*	AnyPhys-style data, not failure *memory*, drives robustness	same curated data, failure-memory removed at inference
Plain depth-retrieval alone *(registered)*	a K3/AttnRes-style softmax over layer *outputs* explains the gain; delta content, null routing, typing, and saliency add nothing	AttnRes-style retrofit baseline (§11.3 #12); Retro-DARC must beat it, and its own null-route / delta-vs-output / typed-vs-untyped ablations must each contribute

The decisive pattern is not that Retro-DARC improves a score, but that it improves over capacity-, compute-, representation-, **and architected-depth-retrieval**-matched controls **and** loses the advantage under content-destroying, scale-preserving interventions.

13.1 Who should use this now — and who should not (field claims live-verified 2026-07-26)

Researchers — mechanism science on depth routing. Our verification sweep (2026-07-26) found no depth-routing paper at any scale that publishes causal intervention audits; the only mechanistic probe of AttnRes we located is a toy-data blog analysis. The released harness attaches to any HF causal LM by name and runs identity, conditioning, retrieval-under-corruption, and the full intervention battery in minutes-to-hours on a laptop — usable today to test *whether and where* depth memory helps on your checkpoint, your data regime (Liu's structured-vs-memorization axis is the obvious first stratification), and your quantization setting. The twin-pretrain protocol (E11/E14) is itself reusable: we found no other controlled retrofit-vs-from-scratch comparison in this literature.

Engineers — auditable, reversible adaptation. The properties that survive our honest comparison are operational ones: ship-dark insertion with a bitwise guarantee (E10.1), exact rollback ($\gamma \leftarrow 0$), content-vs-mechanism attribution as a CI test, and parameter-orthogonal composition with the LoRA your stack already uses. The adoption path is concrete: PEFT's public contribution checklist, with `adaption_prompt` as the module-wrapping precedent; serving is the honest gap — vLLM at the time of writing serves only LoRA-shaped adapters, the read cannot merge into weights, and a bank-alongside-KV-cache interface is engineering that, to our knowledge, does not yet exist. Until it does, the deployment story is finetuning-time adaptation and offline/batch serving, not high-QPS inference.

World-model teams. The measured memory crisis is now quantified: on MemoBench's disappear-and-reappear protocol (arXiv:2606.27537) no evaluated model exceeds 0.6 Object-Reappearance-Score and the best scores 0.381; MBench (arXiv:2606.00793) and WorldRoamBench (arXiv:2606.31672) report the same failure from other angles, and every shipped memory mechanism found (WorldMem, Matrix-Game 3.0's camera-aware retrieval, the memory-WAM line) co-trains memory from scratch. A function-preserving retrofit path onto existing video/world checkpoints appears unowned in our sweep — but we state the mismatch plainly: this paper's measured record is on *language-model* checkpoints; §11's S9/S15 stages (identity on a video decoder; a MemoBench-protocol ORS comparison with memory-shuffle controls) are the registered, not-yet-run bridge, and until they run, world-model impact is a registered hypothesis with unusually large measured headroom, not a result.

Robotics teams. The 2026 record (RoboMME, arXiv:2603.04639) says VLA memory helps but is task-dependent with no universal design; retrofit memory exists (TempoFit's training-free KV reuse, arXiv:2603.07647; HAMLET's +47pp on history-dependent tasks, arXiv:2510.00695) but neither of these offers identity-safe insertion into a validated policy, in-policy failure/trace memory, or addressing that survives the int4/edge deployments VLAs actually run. Those three are this design's registered lane (§7, §11.6/S10); the honest counterweights are TempoFit's zero-training cost, the absence of

measured control-loop-latency numbers for our read (required before any hard-real-time claim), and HAMLET's own finding that gains collapse on non-history-dependent suites.

Who should not use this. Teams whose objective is raw small-scale CE (the MLP adapter won our matched comparison — measured, twice); use-cases already served by long context or external RAG (the bank is for *internal computation history*, not documents); hard-real-time control loops, pending measured per-tick overhead; and anyone needing merged-weight serving today.

13.2 The near-term registered agenda (what happens next, in order)

The long-horizon program is §11; this is the concrete slice in front of it, ordered by how much each result would change the paper's conclusions. Items 1–3 run on laptop/desktop hardware with the released harness; full specifications live in the repository's plan files.

1. **The normalized-key S2 rerun — the next domino.** E15's standing negative (output keys win raw CE by 0.0086/0.0081, two machines) binds *unnormalized* keys, and E13 measured that per-slot normalization repairs exactly the pathology deltas suffer from. Registered prediction: normalized delta keys close or reverse the gap at 50M tokens; either outcome resolves the paper's one conditional claim. (Hours per condition on the measured hardware.)
2. **First contact with native depth routing.** Run the §11.3a first-contact protocol — identity through KDA + AttnRes control flow, conditioning on AttnRes-pretrained representations, retrofit-on-native-AttnRes — on Kimi Linear's public checkpoints now, and on K3-class weights when they land.
3. **Kernel-real quantized reads.** Upgrade E13b from simulated int4 to real int4/FP8/MXFP4 kernels on the released trained-adapter checkpoints — the serving-relevant form of the addressing claim, directly aimed at the MXFP4/MXFP8 regime frontier depth routing already ships in.
4. **The S3 pilot at the Appendix-B configuration** (0.5–0.8B base, 0.5B tokens, 3 seeds): normalized keys, w=4 windows with top-k routing, multi-point insertion — and the matched-FLOP weight-tied loop baseline, which the S3 gate requires and which, in the literature we surveyed, no paper has run. (≈250–450 A100-hours, or a multi-day single-node job.)
5. **Data-regime stratification.** The one public mechanistic critique of depth attention we found predicts gains are structured-vs-memorization dependent; rerunning the E12 protocol on stratified corpora converts that from a reviewer objection into a measured boundary.
6. **Composition and packaging.** The LoRA-composition test (parameter-orthogonal, structurally free, unmeasured — §5.1); then the PEFT-checklist tuner and the bank-alongside-KV-cache serving interface (§13.1) that gate real adoption.
7. **The domain bridge.** Identity verification on one open video/world-model decoder, then the MemoBench-protocol object-reappearance comparison with memory-shuffle controls (§11 S9/S15) — the registered resolution of this paper's LM-to-world-model evidence gap, aimed at a benchmark whose measured ceiling is 0.381. The registered mechanism for this stage is the innovations configuration of §6.1a: parallel action-prefix beliefs, each horizon conditioned on a persistent state built by reliability-weighted (precision-weighted) fusion of retrieved typed innovations, with the null route abstaining when no unexplained evidence supports an update — predictions to be written down, as always, before the runs.

Each item inherits the program's standing rules: predictions written before runs, all outcomes published, failures printed beside their diagnoses.

13.3 A deployed miniature: typed memory over a generative world prior (new)

The interface's application-layer form ships today in a live world-exhibition platform (lucidre.am/memory; source in the released repositories). A generatively produced 3D world — a Marble-API Gaussian-splat scene, treated as the *frozen prior* — hosts two typed innovation banks over it: **edit innovations** (object-state deltas a visitor makes) and **trace innovations** (the visitor's motion path, sampled when the position innovation exceeds a threshold — evidence-triggered, not uniform-time). Both banks use the paper's read verbatim — $\ell 2$-normalized keys, Softmax1 with a live null-mass readout, a zero-gated sum — persist per world across sessions, and expose the audit as user-facing controls: zeroing restores the pristine world (state-hash-verified), shuffling lands edits on the wrong objects, per-type trace-shuffle scrambles paths while leaving edits intact (§11.1's per-type interventions, playable). What this demonstrates is scoped precisely: the *interface properties* —

typing, persistence, exact deactivation, per-type attribution — operating over a real generative world prior in a shipped product; it is not a neural world-model result, which remains §13.2(7)'s registered experiment. Its value to the program is the same as the harness's: anyone can hold the mechanism in their hands.

14. Discussion, Limitations, and Honest Status

What the landscape adds to the thesis. The thesis was first argued from first principles — pretrained models contain reusable residual computations — and then from the world-model field: every major 2026 system that works — SkyJEPA's residual prober on frozen latents, MWA's frozen-IDM anchor and chunked latent actions, μ_0's frozen trace model plus light expert, LoopWM's difference-triggered exits, RynnWorld-4D's frozen-WM readout, LLaVA-OV-2's residual-selected evidence — exploits some slice of the residual-evidence structure Retro-DARC makes general and retrofittable. **The strongest single external datapoint arrived during the program: the depth-residual axis is no longer a research bet but the load-bearing architecture of the largest open model in the world.** Kimi K3's Attention Residuals are this paper's core primitive, proven at 2.8 T parameters with an open, favorable efficiency ledger; the 2026 memory-benchmark cluster (MemoBench et al.) independently states this paper's motivation as a measured fact — none of its ten evaluated models exceeds ~0.6 object reappearance, and the fix must be *explicit memory*, not scale. The convergence is strong evidence for the *principle*. It is still *not* evidence for *this implementation* — a retrofit adapter is a different artifact from an architected primitive — which is exactly why the verification contract (now including a direct AttnRes-recovery comparison, §11.3a, and a MemoBench-style falsification stage, S15) stands unchanged and, if anything, is stricter. The sandbox suite began discharging that contract at the only scale available without a training budget: every mechanism-level claim is now backed by an executed check, one claim was *corrected* by its check (gate-first order), and one naive expectation was *falsified* by its check (shallow-depth delta superiority) — the two ways a verification program earns trust before it earns scale. **The public-checkpoint stage extends both trust currencies — corrections and falsifications — beyond the sandbox**: stage S1 is discharged bitwise on three open models; the addressing claim is measured (and its boundary mapped, and cured) on five; the registered retrofit-vs-from-scratch question is answered at miniature scale with the failure diagnosed to a single removable cause; the matched-budget comparison is run at 3M and 50M tokens with an exact zero-out audit and a cross-hardware replication; and four registered predictions failed in public, two returning measured design rules. The paper's standing is therefore no longer "mechanisms verified, utility registered" but "mechanisms verified on shipping checkpoints, utility measured-competitive at miniature scale with one honest boundary, scaled utility registered."

What Retro-DARC is not. Not a renderer, not a simulator, not a policy, not a new pretraining objective, and not a claim that layer deltas suffice for physical competence. It is a memory interface whose value is conditional on the base model already computing something worth remembering; short-horizon and purely local tasks are registered negative controls precisely because the method should *not* help there.

Limitations. (1) No S3-scale ($\geq$0.6B base, $\geq$0.5B-token) empirical results yet — every claim at that scale remains registered and gated; the measured record is mechanism- and miniature-scale (see (9)). (2) RA retention can discard slow-burning low-magnitude evidence; RA is therefore opt-in. (3) The loop-composition results (Prop. 3, §8) guarantee *safety* of insertion into recurrent models, not benefit; hypothesis 8.2(ii) may fail. (4) Typed memories inherit the failure modes of their sources — bad traces, mis-scored failures, or unidentifiable latents produce confidently wrong memory; the SIGReg-identifiability route (§9) mitigates only in the identifiable regime. (5) Systems overhead targets assume the small-module regime; very long banks without chunking will miss them. (6) Vendor-reported landscape numbers (Aether; MWA; LoopWM; Kimi K3's ~25%/<2% AttnRes figures) are motivation and benchmarks-to-meet, not established science; K3's technical report and weights were not fully released at the time of writing, so its details come from launch materials and should be reconfirmed against the final report. (7) The AttnRes-recovery comparison (§11.3a) is only as meaningful as a matched-scale from-scratch AttnRes baseline; absent a public one, it degrades to an internally trained proxy with weaker external validity. (8) The toy suite is mechanism-scale, not utility-scale: E7's L=12 result *against* naive delta superiority and its L=24 capacity-bound null are reported precisely so the toy suite cannot be mistaken for the registered matched-checkpoint experiments; where a toy result and a proposition point in different directions, the claim is stated conditionally and deferred to §11. E9 partially discharges S1–S3 in miniature, but its "real checkpoint" is locally pretrained (real data, real objective, 417 K params, 1000 steps) and its adapter rankings are a single-seed miniature, not a scaling claim; public billion-parameter weights remain the registered test. (9) The E10–E15 additions are mechanism- and miniature-scale on $\leq$1.5B-parameter frozen public bases with $\leq$50M training tokens; none is an S3-scale utility claim. Within them: E12's adapter ranking is 3-seed but one model/one budget; E15 pre-dates E13's normalization fix, so its S2-gate negative binds the *unnormalized* configuration only (replicated: two seeds on two independent machines agreeing to $\leq 6\times10^{-4}$ CE, reported separately); E11/E14 use the internally trained proxy of (7); and E13b's int4 is simulated rounding, not kernel

arithmetic — the kernel-real version is registered (§13.2(3)). (10) A live citation-verification sweep (2026-07-26) confirmed the depth-routing lineage (§3.1), the memory-benchmark cluster, the robotics memory record, the memory-adapter prior art of Table 1, and K3's launch materials against primary sources; K3's efficiency figures (~25% / ~2.5×) remain vendor-self-reported with no independent replication, its weights *announced* for 2026-07-27 but not yet released; and several registration-stage landscape citations ([MWA26], [Physis26], [Manifold26], [Aether26]) rest on vendor announcements or non-English coverage and could not be independently confirmed — retained as motivation with this flag; no measured claim depends on them. Depth-attention gains may be data-regime-dependent (structured vs. memorization; Liu 2026): the §11 program stratifies for this rather than assuming generality.

Outlook. If the contract holds at 0.6B–3B, Retro-DARC becomes a new PEFT-like primitive — LoRA adapts weights; Retro-DARC adapts *access to computation history* — and the typed extension gives world/action systems, whatever their camp, a shared, intervenable memory substrate. If the contract fails under interventions, the negative result is equally publishable under this design: it would show that pretrained residual streams, contra the field's converging intuitions, do not carry retrievably useful history.

---

15. Conclusion

Retro-DARC turns the residual stream — and its typed analogues across the agent loop — into queryable memory at exactly zero behavioral cost at insertion. That sentence is no longer only a theorem: insertion identity is bitwise on public open-weight checkpoints; delta addressing is measured-better on real learned activations under key noise and simulated int4 corruption of the kind quantized serving imposes (raw keys, one disclosed exception; kernel-real test registered), with its one conditioning anomaly cured by a measured design rule — per-slot key normalization — whose retrieval and utility reruns are registered next; the retrofit path is measured to dominate naive from-scratch integration at miniature scale, with the from-scratch failure diagnosed to gate-initialization shock and cured by a second design rule the paper already argued for in theory; and at matched trainable budgets the depth reads beat LoRA on a converged public checkpoint while supporting the audit no weight-touching adapter can express — content-destroying, parameter-preserving interventions with a machine-checkable zero-out integrity test, replicated across independent hardware. Four registered predictions failed along the way and are printed beside their diagnoses; the delta-utility claim carries one honest, replicated boundary (unnormalized keys, mid-stack, raw CE) with its normalized-key resolution registered. The moment favors the attempt from both directions: the largest open model in the world ships depth-wise selective retrieval as core architecture, settling that the axis matters at frontier scale, while the memory-benchmark cluster has measured that none of the leading world models it evaluates can hold a changing object in mind through occlusion — settling that the capability is missing and that scale alone will not supply it. The scaled target is unchanged and now concretely grounded: same checkpoint, same trainable parameters, same tokens — Retro-DARC must beat parameter-matched adapters, matched-FLOP recurrence, *and* a faithful AttnRes-style retrofit control, with memory interventions removing the advantage, at 0.6B scale and beyond. The longer program extends one interface from layer deltas to observation, action, tool, cause, trace, chunk, physics-residual, and failure memories: the substrates a planner needs to remember in order to act over long horizons in the physical world.

Appendix A. Retro-DARC-Lite forward pass

```
def retro_darc_adapter(h_base, memory_bank, router, gate):
    q = router.query(norm(h_base))
    M = memory_bank.candidates()          # deltas, chunk summaries, null
    K = norm(router.key_proj(M))
    scores = (q @ K.transpose(-1, -2)) / sqrt(router.dim) + router.bias(M)
    idx = topk(scores, k=router.topk)
    w = exp(gather(scores, idx))
    w = w / (1.0 + w.sum(dim=-1, keepdim=True))     # Softmax1 null reserve
    r = (w[..., None] * gather(M, idx)).sum(dim=-2)
    return h_base + gate.value * router.out_proj(r)  # gate.value == 0 at insertion
```

Appendix B. Minimum credible configuration

Models 0.5–0.8B and 1–2B open dense LLMs; 0.5–3B training tokens/run; 3 seeds small / 2 medium; 2k–8k sequence length matched across methods; trainable budget matched to LoRA/residual/MLP; memory $w = 4$, $N_B \leq 8$ chunks; router

$r = 16$–32, $d_r = 32$–64, $k = 2$–4; scalar zero gates; insertion every 4 layers; metrics = validation CE, CME battery, CR/ER/straightness where rollouts exist, decode/prefill overhead. *(Amendments per E13, measured on five public checkpoints: delta keys are per-slot $\ell 2$-normalized before the key projection — this keeps full-bank κ at 3.0–3.7 including the final layers, whose raw deltas carry a 4.9–15.9× norm jump; consequently insertion points near the top of the stack are permitted only with normalized keys, and any raw-key configuration must exclude the top ~3 layers' deltas from its banks. Per E14: gates are zero-initialized in from-scratch builds as well as retrofits.)*

Appendix C. Core ablations

Cumulative vs. delta memory · **delta content vs. layer-output content (AttnRes-style read, baseline 12)** · static vs. token query · Softmax1/null vs. none · $k \in \{1, 2, 4, 8\}$ · local-only vs. chunks vs. anchors · uniform vs. saliency-weighted chunk pooling · detached vs. non-detached memory · shared vs. per-layer router · scalar vs. token gate · $\mathcal{L}_{\text{geom}}$ vs. SIGReg vs. none · per-type biases on vs. off · **typed bank vs. untyped pooled bank (X configurations)** · loop-iterate deltas on vs. off (loop-composed runs) · **initial-state anchor $A_1 = h(o_1)$ on vs. off (the MemoBench "V-frame" finding, §2.1-L4, as an adapter-side ablation)**.

Appendix D. Revision record — every corrected and falsified claim

This program's credibility rests on printing its failures next to its wins. The complete list, with where each is measured:

stage	claim as registered	outcome	diagnosis / consequence
sandbox	R2: "nonzero adapter gradients at insertion" (as registered)	**corrected** — at γ=0 *only* the gate has nonzero gradient (E2: γ 1.19×10⁻²; W_O, U_q, P_K exactly 0)	Prop. 2′ gate-first order; became the zero-init design rule after E14
sandbox	delta keys beat output keys at toy depth (naive expectation)	**falsified at L=12** (E7: output 0.0151 vs. delta 0.0177)	redundancy favors outputs at shallow depth; claim made conditional (§10.1(1))
sandbox	E7 protocol scales to toy depth L=24	**null** — all heads capacity-bound, $R^2 \approx 0.5$ (E7b)	reported as-is; capacity boundary of the toy suite
public-checkpoint runs	§11.3a: from-scratch AttnRes ≥ plain base ("B ≥ A")	**violated 3/3 seeds** (+0.0142 CE; E11)	E14 factorial: pure γ_0=1 initialization shock; zero-init recovers ≈parity; second design rule
public-checkpoint runs	E13 structure bet: anomaly is a one-layer spike	**failed 0/5 models** (gradual top-of-stack norm growth + final jump)	per-slot key normalization removes 92–99% of the excess — the cure became the rule
public-checkpoint runs	E13 structure bet: excluding one layer restores $\kappa \le 8$	**held on only 2/5 models**	top ~3 layers involved; normalization is the general fix
public-checkpoint runs	E14 parity: zero-init from-scratch read ≤ base + 0.005 per seed	**partial** — 2/3 seeds (seed 2 missed by 0.0003)	mean gap +0.0024; registered for the S3 pilot
public-checkpoint runs	E15 S2 gate: deltas beat outputs at 50M tokens	**failed, replicated ×2 machines** (+0.0086 / +0.0081)	binds *unnormalized* keys at mid-stack; normalized-key rerun registered

Four registered predictions failed outright, one partially, one was corrected, and one toy-scale null was retained — against the measured record of §10. Two failures returned design rules (normalize delta keys; zero-init every gate) that are now part of the method. The per-revision changelogs in the repository carry the diff-level history.

References

Bracketed keys as cited. Post-cutoff industrial systems without archival IDs are cited as dated technical announcements; vendor-reported numbers are marked as such in text.

[He16] He et al., Deep residual learning, CVPR 2016. · [Vaswani17] Vaswani et al., Attention is all you need, NeurIPS 2017. · [Ba16] Ba, Kiros, Hinton, Layer normalization, arXiv:1607.06450. · [LoRA21] Hu et al., LoRA, arXiv:2106.09685. · [Houlsby19] Houlsby et al., Parameter-efficient transfer learning, ICML 2019. · [MoD24] Raposo et al., Mixture-of-Depths, arXiv:2404.02258. · [LayerSkip24] Elhoushi et al., LayerSkip, arXiv:2404.16710. · [Dehghani18] Dehghani et al., Universal Transformers, ICLR 2019. · [Geiping25] Geiping et al., Scaling test-time compute with latent reasoning: a recurrent-depth approach, arXiv:2502.05171. · [MoR25] Bae et al., Mixture-of-Recursions, arXiv:2507.10524.

[AttnRes26] Kimi Team, Attention residuals, arXiv:2603.15031. · [DeltaAR26] Luo, Cai, Hu, Delta attention residuals, arXiv:2605.18855. · [OASIS26] Luo et al., Attention sinks and outliers in attention residuals, arXiv:2605.17887. · [MoDA26] Zhu et al., Mixture-of-depths attention, arXiv:2603.15619. · [MUDD25] Xiao et al., MUDDFormer, arXiv:2502.12170. · [DCA25] Heddes et al., DeepCrossAttention, ICML 2025. · [DepthAttn26] Zeng et al., Depth-Attention, arXiv:2606.05014.

Verified against live primary sources, 2026-07-26: Dual Attention Residuals, arXiv:2607.18730 · Multi-Gate Residuals, arXiv:2605.23259 · Hyper-Connections, arXiv:2409.19606 · mHC, arXiv:2512.24880 · DeepSeek-V4, arXiv:2606.19348 · KromHC, arXiv:2601.21579 · LAuReL, arXiv:2411.07501 · FlexiDepth, arXiv:2503.23798 · Dr.LLM, arXiv:2510.12773 · GateSkip, arXiv:2510.13876 · LLaMA-Adapter, arXiv:2303.16199 · Z. Liu, "When does Kimi's Attention Residuals work?", 2026 (kindxiaoming.github.io) · open-attention-residuals (github.com/wdlctc) · LongMem, arXiv:2306.07174 · CAMELoT, arXiv:2402.13449 · Larimar, arXiv:2403.11901 · Prometheus Mind, arXiv:2601.15324 · Trained Persistent Memory, arXiv:2603.22329 · MemoBench, arXiv:2606.27537 · MBench, arXiv:2606.00793 · WorldRoamBench, arXiv:2606.31672 · On Memory (mechanism comparison), arXiv:2512.06983 · WorldMem, arXiv:2504.12369 · Matrix-Game 3.0, arXiv:2604.08995 · RoboMME, arXiv:2603.04639 · TempoFit, arXiv:2603.07647 · HAMLET, arXiv:2510.00695 · MemoryVLA, arXiv:2508.19236 · MAP-VLA, arXiv:2511.09516 · Kimi K3 launch, kimi.com/blog/kimi-k3 (2026-07-16; weights announced 2026-07-27).

[DreamerV3-23] Hafner et al., Mastering diverse domains through world models, arXiv:2301.04104. · [Dreamer4-25] Hafner et al., Dreamer 4, 2025. · [VJEPA-24] Bardes et al., V-JEPA, ICML 2024. · [VJEPA2-25] Assran et al., V-JEPA 2, arXiv:2506.09985. · [LeJEPA25] Balestriero & LeCun, LeJEPA: provable and scalable self-supervised learning without the heuristics, arXiv, Nov 2025. · [Klindt26] Klindt, LeCun, Balestriero, When does LeJEPA learn a world model?, arXiv preprint, May 2026. · [SkyJEPA26] Rao, Zhang, Balestriero, LeCun, Loianno, SkyJEPA, arXiv:2606.23444. · [CJEPA26] Causal-JEPA, arXiv:2602.11389. · [LPWM26] Daniel et al., Latent Particle World Models, ICLR 2026 (oral).

[WAM26] Wang et al., World action models: the next frontier in embodied AI, arXiv:2605.12090. · [DreamZero26] Ye et al., World action models are zero-shot policies, arXiv:2602.15922; code github.com/dreamzero0/dreamzero; weights HF GEAR-Dreams. · [EgoScale26] Zheng et al., EgoScale, arXiv:2602.16710. · [Pi05-25] Physical Intelligence, π0.5, 2025. · [RDT24] Liu et al., RDT-1B, arXiv:2410.07864; code thu-ml/RoboticsDiffusionTransformer. · [DSL25] Lin et al., Data scaling laws in imitation learning, ICLR 2025 (oral). · [DIAL26] Chen et al., DIAL, arXiv:2603.29844. · [FastWAM26] Yuan et al., Fast-WAM, arXiv:2603.16666. · [Mu0-26] Lee, Jung, et al. (Huang & Huang labs, UMD/SNU), μ_0: a scalable 3D interaction-trace world model, arXiv:2606.13769; code github.com/Yoonkyo/mu0; TraceGen arXiv:2511.21690.

[LLaVAOV2-26] An et al., LLaVA-OneVision-2, arXiv:2605.25979. · [OVE26] Tang et al., OneVision-Encoder, arXiv:2602.08683. · [Genie3-25] Google DeepMind, Genie 3: a new frontier for world models, Aug 2025 (deepmind.google); Waymo world-model adoption, Feb 2026. · [Marble25] World Labs, Marble: a multimodal world model, Nov 2025 (worldlabs.ai). · [Cosmos26] NVIDIA, Cosmos world foundation models (Cosmos 3, arXiv June 2026). · [AMI26] AMI Labs, $1.03B seed announcement and JEPA-based world-model program, Mar 2026 (press). · [Manifold26] Manifold AI (流形空间), WorldScape / WorldScape Policy / geometry-aware world-state memory; WorldScore #1; Pre-A announcement, June 2026 (press). · [MWA26] 无界动力 & CASIA-DRL, MWA™ long-horizon bidirectional physical-causal-chain latent world model; AnyPhys; RoboCasa GR1 TableTop 75.2%, June 29, 2026 (technical announcement). · [LoopWM26] Lu, Wei, et al. (FaceMind Research Asia), LoopWM: looped world models, technical report + interview, June 2026. · [Physis26] Chen, Ji, et al. (逆矩阵/BAAI), 悟界·Physis-v0.1: next physical state prediction, BAAI Conference, June 12, 2026. · [Aether26] Huang et al. (Aether AI), Causal world models: four-layer causal brain architecture, CVPR 2026 presentation + June 2026 announcements (vendor-reported metrics). · [Momenta26] Momenta, R7 reinforcement-learning world model (Apr 2026, mass production) and HKEX listing 6880.HK, July 8, 2026. · [LiberAI26] LiberAI (将闲科技), physical world model via video–physics modality alignment; RDT lineage, 2026 (press/interview).

[KimiK3-26] Moonshot AI, Kimi K3: a 2.8-trillion-parameter open MoE model with Kimi Delta Attention, Attention Residuals / Block Attention Residuals, Stable LatentMoE (16-of-896 experts), native vision, 1M context, released July 16, 2026 (kimi.com/blog/kimi-k3; weights announced for July 27, 2026; architectural figures from launch materials pending the technical report). · [KimiLinear25] Kimi Team, Kimi Linear: an expressive, efficient attention architecture (Kimi Delta Attention; 3:1 KDA-to-global hybrid), arXiv:2510.26692. · [MemoBench26] Chen, Zhou, Hua, Zhang, Qian, Ma, Chen, Liu, Zhao, Wang, Li, Yuille, Liang, Du, MemoBench: benchmarking world modeling in dynamically changing environments, arXiv:2606.27537, ECCV 2026; code github.com/MemoBench-Team/MemoBench. · [MBench26] Zhang et al., MBench: a comprehensive benchmark on memory capability for video world models, arXiv:2606.00793. · [MIND26] Ye et al., MIND: benchmarking memory consistency and action control in world models, arXiv:2602.08025. · [OOSOM26] Ma, Liufu, Gkioxari, Out of sight, out of mind? Evaluating state evolution in video world models, arXiv:2603.13215. · [RynnWorld26] Zhao, Zhao, Huang, Li, Zhao, Li (Alibaba DAMO Academy et al.), RynnWorld-4D: 4D embodied world models for robotic manipulation, arXiv:2607.06559; code github.com/alibaba-damo-academy/RynnWorld-4D; weights HF Alibaba-DAMO-Academy/RynnWorld-4D. · [LingBot26] Robbyant (Ant Group), LingBot-World 2.0 / LingBot-World-Infinity: infinite worlds with versatile interactions (MoBA attention mask; DMD over self-rollouts; Pilot/Director agentic harness), arXiv:2607.07534; code github.com/robbyant/lingbot-world-v2, July 2026. · [SelfHarness26] Shanghai AI Laboratory, Self-Harness: harnesses that improve themselves, arXiv:2606.09498; code github.com/qzzqzzb/Self-Harness. · [GPS26] Qu,

Wang, Mao, Zou, Jiang, Liu, Bai, Yang, Chen, Yang, Ji (Tsinghua × Tencent Hunyuan), Small generalizable prompt predictive models can steer efficient RL post-training of large reasoning models, arXiv:2602.01970 (reported accepted at ICML 2026); code github.com/thu-rllab/GPS. · [MoWorld26] 魔芯科技 (MoXin Tech) & Zhejiang University, MoWorld: a Flash World Model (14B MoE; ~50 FPS on Ascend NPU; global-anchor + camera-consistency memory), technical report + announcement, July 7, 2026 (moxin-tech.github.io/moworld; press-reported figures). · [SPEAR26] Ros, Tang, Leutenegger, Sunkavalli, Koltun, et al. (Manycore/群核科技 × Adobe et al.), SPEAR: reflection-based programmable simulation on Unreal Engine, ECCV 2026 (press report, July 2026). · [Meshy26] Meshy, ~$400M Series B at >¥10B post-money valuation; hybrid world-model game direction, July 20, 2026 (press/founder interview). · [HiDream26] HiDream.ai (智象未来), UiT native omnimodal architecture and HiDream-O1 model family; ¥1.5B C round (cumulative >¥2.1B across three rounds), July 2026 (press). · [Tashi26] 它石智航 (Tashi/TARS), $455M Pre-A (April 2026, reported record for China embodied AI) and AWE world-model end-to-end training (press).

[FastLeWM26] Gao, Xu (XJTU), Fast LeWorldModel: action-prefix parallel prediction for latent planning, arXiv:2606.26217; code github.com/Yuntian-Gao/Fast-LeWorldModel; page fast-lewm.github.io. · [LeWM26] Maes, Le Lidec, Scieur, LeCun, Balestriero, LeWorldModel: stable end-to-end JEPA from pixels, arXiv:2603.19312; code github.com/lucas-maes/le-wm (HF checkpoints). · [LingBotVideo26] Robbyant (Ant Group), LingBot-Video: an embodied MoE video foundation model (30B-A3B; hierarchical physics-graded RL reward; action-to-video), arXiv:2607.07675; code github.com/robbyant/lingbot-video. · [Reverie26] Reverie, Interaction Model R v0.2: adaptive streaming interaction with world-state estimation and long-term memory (founder interview + product materials, July 2026). · [SageAttn25] Zhang et al. (Tsinghua/ShengShu line), SageAttention / TurboDiffusion / Sparse Linear Attention: low-bit and sparse attention kernels and step-distillation for compute-bound multimodal inference (open-source project family; interview, July 2026).