Retro-DARC: Function-Preserving Depth-Memory Adapters for Pretrained LLMs

Residual Evidence Across Time, Depth, and Dynamics

Ada Cyborg

June 2026

Abstract

Pretrained language models compute rich intermediate layer updates, but existing checkpoints expose mainly the cumulative residual stream. We introduce **Retro-DARC**, a family of function-preserving depth-memory adapters that makes those updates queryable without changing the base model at insertion. The immediately usable variant, **Retro-DARC-Lite**, inserts a zero-gated read over recent layer deltas, optional block summaries, and an explicit null route into a pretrained causal LM. Because the gate is initialized to zero, hidden states and logits are exactly preserved, while adapter-only training can update only the router, gate, and output projection. We then extend Lite to **Retro-DARC-Full** for long-horizon anchor memories, **Retro-DARC-A** for typed text/observation/action/tool memories, and **Retro-DARC-WM** for object-centric JEPA-style world models with physics-residual verification.

The paper develops a **Residual Evidence Principle**: residuals relative to prediction baselines identify information worth allocating budget to. In video, codec bit-cost, motion, and residual cues indicate event-bearing observations; in Transformer depth, layer deltas indicate useful internal computations; in world/action models, JEPA latent prediction errors and physics residuals indicate belief-state updates to store and verify. We prove exact function preservation and first-step trainability, derive a budgeted residual-selection objective, define causal-memory estimands, and provide CPU-executable mechanism checks for identity preservation, nonzero gate gradients, delta non-redundancy, token-conditioned retrieval, residual-saliency retention, and typed action/physics residual memory. The main empirical contract is direct: same checkpoint, same trainable parameter budget, same continued-training tokens, same evaluation harness; Retro-DARC should outperform parameter-matched adapters and AttnRes-family retrofits, and memory interventions should remove the gains.

Keywords: depth memory, residual streams, parameter-efficient finetuning, world models, JEPA, multimodal event memory, causal interventions.

Status note. This manuscript is a full technical draft, reference architecture, and staged experimental protocol. The immediately usable output is **Retro-DARC-Lite**: an adapter-only retrofit that can be inserted into a pretrained causal LM, verified to preserve logits at insertion, and trained without custom CUDA or from-scratch pretraining. The algebraic properties are proven. Current executable checks verify exact identity, nonzero first-step gate gradients, lower delta redundancy than cumulative hidden states, token-conditioned routing advantage, residual-saliency selection, physics-residual memory, and typed action-memory interventions. The model-scale claims are specified as matched-compute retrofit experiments with causal memory pass/fail tests rather than assumed.

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, if $h_l(t) \in \mathbb{R}^d$ denotes the token representation at position t and depth l, a sublayer contributes

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

Unrolling gives

$$h_L(t) = h_0(t) + \sum_{l=1}^{L} \Delta_l(t). \tag{2}$$

Every previous layer contributes with fixed coefficient one. This residual stream is essential for optimization [3, 4], but as a memory mechanism it is crude: depth is accumulated, not queried.

Recent work challenges this design. Attention Residuals replace uniform residual accumulation with softmax aggregation over previous layer outputs and introduce Block AttnRes for scalability [6]. Delta Attention Residuals argue that cumulative hidden states are redundant and route over layer deltas instead [7]. OASIS adds null-aware routing to address sinks, outliers, and quantization brittleness in AttnResidual-style architectures [8]. MoDA, MUDDFormer, DeepCrossAttention, and Depth-Attention independently reinforce the same conclusion: residual connectivity is not merely an optimization detail; it is a major information-routing axis [9–12].

The strategic question is therefore no longer whether depth routing can be useful. That space is now active. The sharper question is:

> *Can a pretrained Transformer be upgraded with internal depth memory without changing its original function at insertion?*

This question matters because most valuable models already exist as pretrained checkpoints. A method that requires from-scratch architecture training competes for large pretraining budgets. A method that safely retrofits existing checkpoints has a lower experimental cost, a clearer adoption path, and a sharper scientific test: it asks whether pretrained layers already contain reusable internal computations that can be exposed by a small routing mechanism.

We propose **Retro-DARC**, a *retrofittable* Delta-Attention Residual Compute adapter. The method inserts a trainable depth-memory read path into a frozen or lightly tuned pretrained model. The read path is initialized as an exact no-op, so the base model's hidden states and logits are preserved at insertion. During continued training, the adapter learns to retrieve layer-delta memories, optional block summaries, and optional latent anchors through a low-rank token-conditioned query and null-aware top-k routing. The immediate adoption path is **Retro-DARC-Lite**: insert adapters every few layers, freeze the base model, train only the router/gate/projection, and verify causal memory effects by zeroing or shuffling the memory bank.

Core thesis.

> **Pretrained LLMs already contain useful internal layer-delta memories. Retro-DARC exposes those memories through a zero-initialized, function-preserving adapter that preserves the base model's logits at insertion, learns during cheap continued training, and improves long-horizon state maintenance over parameter-matched LoRA, residual-adapter, and AttnRes-family retrofit baselines. Causal interventions verify that the depth memory itself is responsible for the gains.**

This thesis is deliberately narrower than "Retro-DARC is better than Attention Residuals." The contribution is not merely another cross-layer mixer. It is a function-preserving retrofit mechanism and evaluation framework for internal computation memory.

It is now also part of a broader residual-evidence program: in multimodal systems such as LLaVA-OneVision-2, codec residuals allocate observation budget to event-bearing video regions; in Retro-DARC, layer residuals allocate internal memory to computation-bearing depth regions; in object-centric JEPA/world models, latent and physics residuals allocate verification budget to belief-state updates.

Broader implication. World-model training is increasingly important in robotics, video generation, and embodied agents. In that setting, the same mechanism suggests a planner-side belief-memory interface: text, observation, action, and tool deltas can be made queryable by later computation without requiring a new planner from scratch. We treat this implication seriously, but we keep the first empirical burden on the cheaper LLM-retrofit claim.

Contributions.

1. We define **Retro-DARC-Lite**, an adoption-oriented LLM retrofit with recent layer-delta memories, optional block summaries, a null state, low-rank token-conditioned routing, and zero-gated residual injection.
2. We define **Retro-DARC-Full**, which adds anchor memories and richer diagnostics for long-horizon state maintenance.
3. We define **Retro-DARC-A**, a typed extension for action-observation and world/action models, positioning depth memory as planner-side belief memory rather than a renderer or simulator.
4. We prove exact function preservation and identify the necessary nonzero-output-projection condition for first-step gate gradients.
5. We specify a staged, low-cost verification program: identity tests, frozen probes, 0.6B adapter signal gates, 1B–2B confirmation, causal memory interventions, long-horizon/code/tool evaluation, and systems profiling.
6. We give an adoption recipe that allows a researcher to inject Retro-DARC into a Hugging Face style causal LM without custom CUDA or from-scratch pretraining.
7. We introduce the **Residual Evidence Principle**, connecting codec residuals in video perception, layer deltas in Transformer depth, JEPA latent prediction residuals, and physics/verification residuals in world models.
8. We specify a concrete multimodal extension using LLaVA-OneVision-2 as an open testbed for event-memory and codec-conditioned depth-memory evaluation.

2 Retro-DARC Family Overview

Retro-DARC is a family rather than one monolithic mechanism. The first implementation should be minimal enough for ordinary LLM researchers to adopt, while the full research program extends the same memory interface to long-horizon agents and world/action models. Table 1 summarizes the design stack.

Table 1: Retro-DARC family. The variants share one invariant: insertion is function-preserving because the memory read is gated by $\gamma = 0$ at initialization.

Variant	Target use	Memory bank	First validation
Retro-DARC-Lite	Pretrained LLM retrofit and PEFT-like continued training	recent layer deltas, optional block summaries, null route	identity, gradients, adapter-only LM loss, memory interventions
Retro-DARC-Full	long-horizon state maintenance	Lite + anchor memories + route/entropy diagnostics	code, tool, long-context, ASMP state probes
Retro-DARC-A	action-observation planners	typed text, observation, action, tool, episode and belief memories	action-memory corruption and trajectory-state tasks
Retro-DARC-WM	object-centric latent world models	object/action deltas, JEPA residuals, physics residuals, verification memories	latent prediction, physics-slop index, planning success
Residual-Aligned Retro-DARC	event-heavy multimodal systems	saliency-selected deltas using layer, codec, and physics residual evidence	LLaVA-OneVision-2 event-memory tests

The family structure is important. Lite gives the adoption path; Full tests long-horizon computation memory; A makes text/observation/action/tool transitions first-class; WM connects the same mechanism to object-centric JEPA and physics verification. This lets the paper claim a coherent research program without forcing every variant to be validated in one experiment.

3 Contribution Structure and Validation Contract

Retro-DARC is organized around a compact contribution stack: an insertion invariant, an adoption-ready retrofit mechanism, an intervenable memory attribution test, and a residual-evidence extension to multimodal and world/action settings. These layers are mutually reinforcing. The invariant makes insertion safe; the retrofit mechanism makes the method cheap to test; the causal tests turn "memory" into an intervenable object; the world/action extension explains why residual memory matters beyond text.

Primary evaluation contract. The central empirical test is precise:

> *Same pretrained checkpoint, same trainable parameter budget, same continued-training tokens, same evaluation harness: Retro-DARC improves over non-memory adapters, and intervention on the memory bank removes the gain.*

This contract lets the paper claim more than improved loss. It establishes whether pretrained models contain reusable computation traces and whether a small adapter can expose them.

4 Why Now: From Depth Routing to Planner Memory

The practical paper should be LLM-first, but the world-model motivation is not decoration. It explains why a depth-memory adapter may matter beyond perplexity.

Table 2: Contribution structure and validation targets for Retro-DARC. Each layer has a concrete artifact in the method or release.

Layer	Contribution	Validation target
Invariant	A zero-gated Retro-DARC adapter preserves the pretrained model's hidden states and logits at insertion.	Algebraic proof plus numerical identity tests on real checkpoints.
Retrofit utility	After cheap continued training, Retro-DARC improves over parameter-matched LoRA, residual-adapter, MLP-adapter, and AttnRes-family retrofit baselines.	Same checkpoint, data, trainable parameters, tokens, optimizer, and compute accounting.
Causal memory	The improvement comes from example-specific depth memory rather than generic adapter capacity.	Zero, force-null, shuffle, route-corrupt, local-only, depth-offset, and random-same-norm interventions.
Residual evidence	Residuals across time, depth, and dynamics indicate what to observe, remember, and verify.	LLaVA-OV-2 event-memory tests; JEPA/world-model residual-memory tests.
World/action memory	Typed text, observation, action, tool, and physics-residual memories improve planner-side belief maintenance.	Object-centric JEPA or world/action experiments with action-memory and verification-memory ablations.

4.1 Continuous Latent Computation

ELF formulates language generation as a diffusion/flow process in continuous embedding space and discretizes only at the final step [25]. Retro-DARC is not a diffusion language model, but the philosophical alignment is useful: language computation need not be understood only as a sequence of discrete tokens. Pretrained LLMs also contain continuous trajectories across depth. A layer delta Δ_l is a local velocity-like update in representation space. If such updates are useful, later computation should be able to retrieve them rather than rely only on their cumulative sum.

The analogy is domain-specific: ELF models a generative trajectory over sampling time, while Retro-DARC reads a pretrained Transformer's computational trajectory over network depth. The shared principle is that useful structure can live in continuous intermediate states before final token emission.

4.2 World Models: Renderers, Simulators, and Planners

A functional taxonomy of world models separates the agent loop into actions, hidden world state, observations, and policy/planning functions, and classifies world models by what they output: observations for renderers, states for simulators, and actions for planners [26]. In that taxonomy, Retro-DARC is not a renderer and not a simulator. It does not generate pixels, nor does it claim to predict the full physical state. Its natural role is inside a planner: it exposes a reusable internal belief/computation memory over prior latent updates.

For a partially observable process with hidden state s_t, observation o_t, and action a_t, a planner maintains a belief-like internal variable

$$b_t \approx p(s_t \mid o_{\leq t}, a_{<t}). \tag{3}$$

In a Transformer planner, b_t is not usually an explicit object. It is distributed across the residual stream and updated through depth. Retro-DARC makes some of this internal update history queryable:

$$b_{l,t}^{\text{new}} = b_{l,t} + \gamma_l W_o r_l(t), \tag{4}$$

where $r_l(t)$ is a read from prior deltas, block memories, anchors, or null. This is why the method is a plausible planner-side belief-memory interface.

4.3 World Action Models and Action-First Learning

World Action Models (WAMs) are an emerging embodied-AI paradigm that jointly models future world states and actions rather than mapping observations to actions alone [27]. DreamZero instantiates this direction with a 14B autoregressive video diffusion model that jointly predicts future video and actions, reports over 2× generalization improvements over VLA baselines in real-robot experiments, and reaches 7Hz closed-loop control after system optimizations [28]. NVIDIA's WAM glossary similarly defines WAMs as models that jointly predict future states and robot actions, using video as a dense learning signal for how the world evolves [30]. EgoScale shows a complementary scaling trend: large-scale egocentric human video can serve as a predictable supervision source for dexterous manipulation, with over 20k hours of action-labeled video and a reported log-linear relationship between data scale and validation loss [29].

These developments motivate typed memory. In text-only LLMs, the relevant deltas are layer-delta memories over tokens. In world/action planners, there are also observation deltas, action deltas, tool-result deltas, and episode-level belief deltas. A depth-memory adapter that can separate and retrieve these types could improve action-state maintenance without requiring the planner itself to be rebuilt from scratch.

4.4 The Tension: Explicit Checkability vs. Scaled Learned Dynamics

A useful way to understand the current world-model debate is as a tension between two strategies. The explicit strategy emphasizes object identity, geometry, contact, collision, dynamics, and checkable structure. In this view, a world model should not merely generate plausible futures; it should produce states that can be inspected, constrained, edited, and verified. Fei-Fei Li's functional taxonomy makes this distinction crisp by separating renderers, simulators, and planners inside an agent–state–observation loop [26]. The implicit strategy emphasizes scale: large video and action models learn dynamics from massive observation streams, then align those dynamics to action. DreamZero is the clean example: a pretrained video diffusion backbone becomes useful for robot control by jointly predicting future video and robot actions [28]. EgoScale strengthens the scaling thesis by showing that large-scale egocentric human video provides a predictable supervision source for dexterous manipulation [29].

These strategies disagree about what to do with visually plausible but physically wrong futures. Public summaries of Jim Fan's AI Ascent remarks popularized the phrase "physics slop" for cases where a video model appears to learn physical regularities but can also exploit unobserved geometry or produce physically invalid futures. In a scaled-dynamics view, such behavior may be an early, imperfect phase of emergent physical understanding that action fine-tuning can align. In an explicit-structure view, it is a hazard: a planner that acts on physically invalid imagination can fail catastrophically.

Retro-DARC does not choose one camp. It addresses a lower-level bottleneck shared by both. Whether a model uses explicit object states or implicit video/action latents, it must maintain a history of internal computations: object bindings, inferred contact states, action hypotheses,

failed predictions, verification residuals, and uncertainty. If these computations are collapsed into a residual stream or discarded after each layer, the planner loses information before the renderer, simulator, or action head has a chance to use it. Retro-DARC makes that internal computation history queryable.

4.5 JEPA, Object-Centric Prediction, and Why Pixels Are Not the Right Target

LeCun's JEPA program argues that world models should predict missing information in an abstract representation space rather than reconstruct every pixel. V-JEPA predicts masked spatio-temporal regions in learned latent space and explicitly avoids pixel-level reconstruction as the training target [33, 34]. V-JEPA 2 scales this recipe to over one million hours of internet video, then post-trains an action-conditioned predictor with less than 62 hours of robot videos for zero-shot robot control [35]. This is closely aligned with the Retro-DARC thesis: the useful state of a model is not necessarily the visible output, but the intermediate latent representation used for prediction and planning.

However, patch-level latent prediction still risks learning correlations that are not object-causal. Recent object-centric world-model work pushes the JEPA principle toward entities and interactions. Causal-JEPA applies object-level masking so an object's state must be inferred from other objects, inducing counterfactual-like latent interventions and improving counterfactual reasoning [36]. Latent Particle World Models learn keypoints, masks, and object-like particles directly from video, supporting action, language, and image-goal conditioning [37]. These works suggest a stronger version of Retro-DARC for world models: read and write memory not only over layers, but over object-centric latent deltas and action-conditioned residuals.

Implication for this paper. The first Retro-DARC paper should remain practical and LLM-first. But the world-model extension should be formulated concretely enough to be testable: use JEPA-style latent prediction as the target, object slots as the state representation, differentiable physics as a residual prior, and a verification signal to write constraint violations back into memory. This makes the world-model section an executable research plan rather than speculative motivation.

4.6 Codec-Stream Perception and the Residual Evidence Principle

LLaVA-OneVision-2 adds an important input-side analogue of Retro-DARC. Its key method, codec-stream tokenization, treats compressed video as a continuous bit-cost stream: P/B-frame bit-cost dynamics determine adaptive temporal groups, while motion and residual cues select salient spatial evidence into compact visual canvases [38]. The project release describes LLaVA-OneVision-2 as a fully open 8B multimodal model that unifies images, long-form video, and spatial understanding, with code, data, checkpoints, and evaluation pipelines released end-to-end [40]. The Hugging Face model card states that the 8B instruct model uses a Qwen3-8B language backbone with a OneVision-style vision encoder and ships both a frame-sampling backend and an optional codec video backend driven by motion vectors and bit-cost [41]. The dataset release includes large-scale video and spatial-reasoning corpora for the LLaVA-OneVision-2 multimodal family [42].

This is not merely a video preprocessing trick. It is a statement about evidence allocation. Uniform frame sampling allocates visual tokens by elapsed time. Codec-stream tokenization allocates visual tokens by prediction difficulty: high bit-cost, motion, and residual content indicate where the compressed stream was hard to predict and where event-bearing evidence is concentrated. LLaVA-OneVision-2 reports large gains on temporal localization and event-heavy video tasks under matched visual-token budgets, including JumpScore, a benchmark designed for fine-grained localization in repeated high-frequency motion [38].

Table 3: Residual evidence across domains. Retro-DARC uses the same evidence-allocation logic internally that codec-stream tokenization uses for video observation.

Domain	Prediction baseline	Residual evidence	Allocation decision
Compressed video	Codec prediction from prior frames	Bit-cost, motion vectors, pixel residuals	Which visual tokens to preserve
Transformer depth	Previous hidden computation	Layer delta $\Delta_l = h_l - h_{l-1}$	Which internal computations to remember
JEPA world models	Latent prediction $\hat{z}_{t+k}$	Latent prediction error	Which belief updates to refine
Physics-grounded planners	Differentiable/checkable dynamics Φ_ψ	Physics residual and constraint violation	Which imagined futures to reject or repair
Tool/action agents	Expected tool or environment state	Tool error, state transition, failed action	Which plan state to preserve

Retro-DARC applies the same principle internally. Standard residual accumulation allocates memory uniformly over depth by adding every layer contribution into one cumulative stream. Retro-DARC treats layer deltas as computational residuals and learns which deltas should remain queryable. In a JEPA/world-model setting, the analogous residuals are latent prediction errors, physics residuals, and verification failures. These residuals identify where the planner's belief state changed or where an imagined future violated a checkable constraint.

We therefore introduce the **Residual Evidence Principle**:

> *Useful information is often sparse in residuals relative to a prediction baseline. Video codec residuals indicate what to observe; Transformer layer deltas indicate what to remember internally; JEPA and physics residuals indicate what belief-state updates must be verified and preserved.*

This principle also identifies the operating regime. Codec residual sampling is strongest when decisive evidence is sparse in time or space; dense frame paths can remain important for smooth trajectory estimation or fine visual detail. Likewise, Retro-DARC should help most when decisive information appears as state-changing computations: variable binding updates, failed-tool corrections, action-state transitions, proof subgoals, physics-residual violations, or event-localization memories. It should not replace dense local computation.

4.7 Formalizing Residual Evidence

The residual-evidence principle can be made precise enough to guide experiments. Let X denote the current observation or hidden state, $B(X)$ a prediction baseline, and $R = X - B(X)$ a residual. Let Y be a downstream event, answer, action, or verification outcome. A saliency score $\eta(R)$ is useful under a budget K if the retained residual set $\mathcal{S}_K$ maximizes downstream information:

$$\mathcal{S}_K^* = \arg\max_{|\mathcal{S}| \le K} I(Y; R_{\mathcal{S}} \mid B(X)). \tag{5}$$

In video, R may be codec residual and motion evidence; in depth memory, R is a layer delta; in JEPA/world models, R is a latent prediction error or physics residual. Retro-DARC does not assume

that residual magnitude is always sufficient. Instead, it treats residual magnitude, directional change, router uncertainty, loss probes, and codec/physics metadata as candidate estimators of conditional surprise.

Proposition 1 (Budgeted residual selection under calibrated saliency)**.** *Suppose each memory item m has a saliency score η_m satisfying $\eta_m = \log p(Y \mid R_m, B) - \log p(Y \mid B)$ up to an additive constant, and suppose selected items contribute additively to the log evidence for Y under a budget K. Then selecting the top-K items by η_m maximizes the retained log evidence among all size-K subsets.*

Proof. Under the additive assumption, the retained evidence of a subset $\mathcal{S}$ is $\sum_{m \in \mathcal{S}} \eta_m$ plus a constant independent of $\mathcal{S}$. The maximizing subset of fixed size is therefore the K largest scores. The result states the idealized condition that residual-aligned routing attempts to approximate in real models. □

This proposition gives a mathematical interpretation of Residual-Aligned Retro-DARC. The saliency term is not a heuristic decoration; it is an estimator of where residual information is most likely to matter. The empirical obligation is to test whether saliency-weighted memory improves over uniform memory and whether saliency corruption removes the improvement.

4.8 Scope Boundary

The execution order is staged so that each level contributes evidence before the next level scales:

1. **Core LLM retrofit:** exact logit preservation, adapter-efficiency gains, and causal layer-delta memory use.
2. **Planner-state tasks:** text/code/tool trajectories where hidden state must be maintained through observations and actions.
3. **World/action extension:** typed memory on action-observation models or WAM-like planners after the first two levels succeed.

This structure keeps the paper practical while taking the world-model implication seriously.

5 Related Work

Residual streams and normalization. Residual connections enable optimization of deep networks by providing additive skip paths [4]. Transformers combine residual connections with self-attention and normalization [3, 5]. In decoder-only LLMs, PreNorm residuals are common because they improve stability, but they also imply fixed depth aggregation.

Attention over depth. Attention Residuals replace fixed residual accumulation with attention over previous layer outputs and use Block AttnRes to compress history [6]. DeepCrossAttention dynamically combines previous layer outputs through input-dependent weights and extends this idea into attention queries, keys, and values [12]. MUDDFormer dynamically generates dense cross-layer connection weights for query, key, value, and residual streams [11]. These methods show that depth is an informative routing axis, but they primarily target architectural design rather than safe pretrained retrofit.

Delta and null routing. Delta Attention Residuals identify redundancy in routing over cumulative hidden states and propose routing over layer deltas $h_{l+1} - h_l$ [7]. OASIS identifies attention sinks and outliers in AttnResidual architectures and introduces null-aware Softmax1 routing and token-to-depth null coupling [8]. Retro-DARC uses delta memory and null-aware routing, but the novelty is the function-preserving adapter formulation and causal retrofit evaluation.

Depth mixing inside attention. MoDA allows attention heads to attend to sequence KV pairs at the current layer and depth KV pairs from preceding layers [10]. Depth-Attention performs cross-layer value mixing inside the attention module and emphasizes no additional persistent inference state beyond the standard KV cache [9]. These are strong systems baselines. Retro-DARC competes through safe retrofit and causal depth-memory testing; overhead is measured directly against these systems baselines.

Codec-aligned perceptual sparsity. LLaVA-OneVision-2 and OneVision-Encoder make a complementary claim on the input side: visual-token allocation should follow codec bit-cost, motion, and residual evidence rather than uniform frame sampling [38, 39]. LLaVA-OneVision-2 is especially useful for Retro-DARC because it releases an 8B multimodal checkpoint, codec and frame video backends, training/evaluation code, and data artifacts, making it an open testbed for asking whether internal depth memory improves multimodal event-state maintenance [40–42]. We treat LLaVA-OneVision-2 as a perception and grounding substrate, not as a full world model: it decides what evidence enters a multimodal model, while Retro-DARC asks what internal computations should be preserved after evidence enters.

Parameter-efficient adaptation. LoRA freezes pretrained weights and injects trainable low-rank matrices into Transformer layers, reducing trainable parameters and memory relative to full finetuning [15]. Classical adapters insert small bottleneck modules into pretrained networks [16]. Retro-DARC must be compared against these methods because its first adoption path is adapter-style training.

Adaptive depth compute. Mixture-of-Depths dynamically allocates FLOPs to token positions across layers [13]; LayerSkip trains early-exit/self-speculative models for faster inference [14]. Retro-DARC does not initially skip layers. Instead, route entropy, null probability, and distant-depth mass can later drive adaptive internal compute.

World models and embodied agents. World-model reinforcement learning trains latent dynamics models and improves behavior by imagination, as in DreamerV3 [31]. JEPA-style methods predict in latent representation space rather than raw pixels or tokens [32, 33]. World/action models extend this trend by jointly modeling future observations and actions [27, 28]. Retro-DARC does not replace these objectives; it proposes an internal memory interface for planner modules trained under them.

6 Problem Setting and Design Requirements

We consider a pretrained autoregressive Transformer F_θ with hidden states $h_l(t) \in \mathbb{R}^d$ for token position t and layer or sublayer index $l \in \{0, \dots, L\}$. Let $z_\theta(x)$ denote the final logits for input x.

A retrofit adapter introduces trainable parameters ϕ and yields logits $z_{\theta,\phi}(x)$. We require four properties:

1. **Function preservation at insertion.** For initialization ϕ_0, $z_{\theta,\phi_0}(x) = z_\theta(x)$ for all inputs x up to numerical precision.
2. **Trainability.** The adapter must receive nonzero gradients at insertion under the language-modeling loss.
3. **Parameter and compute fairness.** Comparisons must match trainable parameter count, training tokens, data order, and measured overhead.
4. **Causal use.** Disabling or corrupting the learned depth memory should reduce performance if the method's claimed mechanism is correct.

The fourth requirement is central. Without causal memory interventions, Retro-DARC could be dismissed as an expensive residual adapter. The paper's key scientific claim is not just that the model improves, but that the improvement is due to retrieved layer-delta memories.

7 Retro-DARC-Lite: The Adoptable Core

Retro-DARC-Lite is the version researchers should implement first. It has no world-model machinery, no custom kernels, and no requirement to retrain the base model from scratch.

7.1 Layer-Delta Memory

The basic memory unit is a residual delta:

$$\Delta_l(t) = h_l(t) - h_{l-1}(t). \tag{6}$$

A delta represents what layer l contributed, whereas a cumulative state contains all previous contributions. For a layer l with local memory window w, the Lite memory bank is

$$\mathcal{M}_l^{\text{lite}}(t) = \{\Delta_{l-w}(t), \ldots, \Delta_{l-1}(t), 0_\varnothing\}. \tag{7}$$

A practical implementation may also include block summaries:

$$\mathcal{M}_l^{\text{lite+block}}(t) = \mathcal{M}_l^{\text{lite}}(t) \cup \{B_1(t), \ldots, B_{b-1}(t)\}, \tag{8}$$

where b is the current block index and B_j is a compressed summary of deltas in block j.

7.2 Low-Rank Token-Conditioned Query

A static depth query cannot distinguish tokens that need different internal computations. Retro-DARC uses a cheap token-conditioned query:

$$q_l(t) = q_l^{(0)} + U_l \phi\left(V_l \,\text{RMSNorm}(h_l(t))\right), \tag{9}$$

where $V_l \in \mathbb{R}^{r \times d}$, $U_l \in \mathbb{R}^{d_r \times r}$, $r \ll d$, and d_r is the router dimension. In pilots, $r \in \{16, 32\}$ and $d_r \in \{32, 64\}$ are sufficient.

7.3 Null-Aware Top-k Routing

Each memory candidate $M_m(t) \in \mathcal{M}_l(t)$ is scored by

$$s_{l,m}(t) = \frac{q_l(t)^\top \,\mathrm{RMSNorm}(P_K M_m(t))}{\sqrt{d_r}} + b_{\mathrm{type(m)}} + b_{\mathrm{age(l,m)}}. \tag{10}$$

Let $I = \mathrm{TopK}(s_l, k)$ be the selected memory indices. Retro-DARC uses Softmax1 normalization:

$$\alpha_{l,m}(t) = \frac{\exp(s_{l,m}(t))}{1 + \sum_{j \in I} \exp(s_{l,j}(t))}, \qquad m \in I. \tag{11}$$

The implicit reserve mass

$$\alpha_{\varnothing}^{\mathrm{reserve}}(t) = \frac{1}{1 + \sum_{j \in I} \exp(s_{l,j}(t))} \tag{12}$$

lets the model decline retrieval. One may also include an explicit null vector $0_\varnothing$ with a learned bias. The depth read is

$$r_l(t) = \sum_{m \in I} \alpha_{l,m}(t) M_m(t). \tag{13}$$

7.4 Zero-Gated Injection

The adapter update is

$$h_l^{\mathrm{new}}(t) = h_l(t) + \gamma_l W_O r_l(t). \tag{14}$$

The gate is initialized to zero:

$$\gamma_l = 0, \tag{15}$$

while W_O is initialized nonzero with small variance. This distinction is important: setting both $\gamma_l = 0$ and $W_O = 0$ preserves identity but can kill first-step gate learning.

7.5 Default Adoption Recipe

The default version for a public GitHub release should use:

Table 4: Recommended Retro-DARC-Lite defaults for first adoption.

Component	Default
Insertion	every 4 layers or at block boundaries
Memory	recent deltas, $w = 4$; optional 4–8 block summaries
Router rank	$r = 16$ for sub-1B, $r = 32$ for 1B–3B
Router dim	$d_r = 32$ or 64
Top-k	$k = 2$ or 4
Null route	Softmax1 plus optional explicit null vector
Gate	scalar per inserted layer, exactly zero at insertion
Base model	frozen in pilot; partial unfreeze only after adapter signal
Memory gradients	detach values in adapter-only pilot; test non-detached variant later
Custom kernels	none until 1B–2B signal is positive

Figure 1 summarizes the Lite architecture.

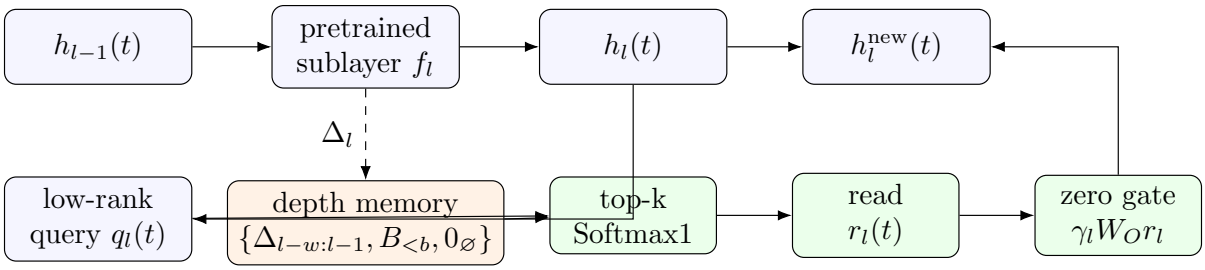

Figure 1: Retro-DARC-Lite adds a zero-gated depth-memory read to a pretrained residual stream. At insertion $\gamma_l = 0$, so the model is unchanged. During training, the adapter learns to retrieve recent layer deltas, block summaries, or null.

8 Residual-Aligned Retro-DARC: Allocating Memory to Evidence

Retro-DARC-Lite keeps a small local window of recent deltas. This is the safest adoption default. However, the LLaVA-OneVision-2 codec result suggests a stronger design rule: memory budget should follow evidence, not uniform time or uniform depth. We therefore define **Residual-Aligned Retro-DARC** (Retro-DARC-RA), a variant that retains and routes to deltas according to computational saliency.

8.1 Depth Residual Saliency

For a layer delta $\Delta_i(t)$, define a depth-residual evidence score

$$\eta_i^{\text{depth}}(t) = \lambda_1 \|\Delta_i(t)\|_{\text{RMS}} + \lambda_2\big(1 - \cos(h_i(t), h_{i-1}(t))\big) + \lambda_3 H_i^{\text{route}}(t) + \lambda_4 u_i(t), \tag{16}$$

where H_i^{route} is a router-entropy or uncertainty diagnostic when available, and u_i is an optional learned or probe-based utility estimate. The memory bank becomes

$$\mathcal{M}_l^{\text{RA}}(t) = \text{TopK}_{i<l}\left[\eta_i^{\text{depth}}(t)\right] \cup \{B_{<b}(t), A_{1:K}(t), 0_\varnothing\}. \tag{17}$$

The retrieval score receives an evidence prior:

$$s_{l,i}^{\text{RA}}(t) = \frac{q_l(t)^\top k_i(t)}{\sqrt{d_r}} + \beta_d \log(1 + \eta_i^{\text{depth}}(t)) + b_{\text{type(i)}} + b_{\text{age(l,i)}}. \tag{18}$$

Setting $\beta_d = 0$ recovers ordinary Retro-DARC. A positive β_d biases retrieval toward layers where the representation changed substantially or where uncertainty suggests unresolved computation.

8.2 Codec-Conditioned Residual Alignment for Multimodal Models

For video inputs, codec-stream metadata can provide an external saliency prior. Let $c_{g,b}$ denote metadata for temporal group g and spatial block b: bit-cost, motion magnitude, and codec residual magnitude. Define

$$\eta_{g,b}^{\text{codec}} = a_1 \log(1 + \text{bitcost}_{g,b}) + a_2 \| \text{motion}_{g,b} \| + a_3 \| \text{residual}_{g,b} \|. \tag{19}$$

If a visual token or multimodal hidden state is associated with group/block (g, b), its depth-memory score can use a fused prior:

$$\eta_i^{\text{mm}}(t) = \eta_i^{\text{depth}}(t) + \lambda_c \eta_{g(t),b(t)}^{\text{codec}}. \tag{20}$$

Then Eq. (18) becomes

$$s_{l,i}^{\mathrm{MM}}(t) = \frac{q_l(t)^\top k_i(t)}{\sqrt{d_r}} + \beta_d \log(1 + \eta_i^{\mathrm{depth}}(t)) + \beta_c \log(1 + \eta_{g(t),b(t)}^{\mathrm{codec}}) + b_{\mathrm{type}} + b_{\mathrm{age}}. \tag{21}$$

This gives a concrete bridge between codec-aligned perception and internal depth memory. Codec evidence tells the model which observations were event-bearing; layer-delta evidence tells the model which internal computations changed after observing them.

8.3 Residual-Aligned Routing as an Optional Acceleration

Retro-DARC-RA should be evaluated after Retro-DARC-Lite succeeds. Evidence-biased retention can improve efficiency, but it can also prematurely discard low-magnitude information that becomes important later. The default repository should therefore expose RA as an optional configuration:

```
model = inject_retro_darc(
    model,
    layers="every_4",
    memory="recent_deltas",
    residual_aligned=False,      # default Lite path
)

model_ra = inject_retro_darc(
    model,
    layers="every_4",
    memory="topk_residual_evidence",
    residual_aligned=True,
    codec_prior=True,            # only for codec-video inputs
)
```

The scientific question is whether residual evidence improves memory selection beyond uniform local windows while preserving the function-preserving retrofit property.

9 Retro-DARC-Full: Long-Horizon State Maintenance

Retro-DARC-Full adds two mechanisms after Lite shows signal: anchor memories and compute diagnostics.

9.1 Anchor Memories

Anchor memories are small persistent latent slots:

$$\mathcal{M}_l^{\mathrm{full}}(t) = \mathcal{M}_l^{\mathrm{lite+block}}(t) \cup \{A_1(t), \ldots, A_K(t)\}. \tag{22}$$

A lightweight update is

$$A_k^{(l)}(t) = (1 - \eta_{l,k}(t)) A_k^{(l-1)}(t) + \eta_{l,k}(t) W_A \Delta_l(t), \tag{23}$$

$$\eta_{l,k}(t) = \sigma\left(u_k^\top \mathrm{RMSNorm}([h_l(t); \Delta_l(t)])\right). \tag{24}$$

Anchors are introduced after the local/block memory path establishes signal. They are designed for longer horizons. In long-horizon tasks, anchors may specialize into entity bindings, plan states, verifier states, tool-result interpretations, or action-state beliefs.

9.2 Routing Diagnostics and Optional Adaptive Compute

Depth routing gives signals that may predict computational difficulty:

$$H_l(t) = -\sum_{m \in I} \alpha_{l,m}(t) \log \alpha_{l,m}(t), \tag{25}$$

$$D_l(t) = \sum_{m \in I \cap \text{distant}} \alpha_{l,m}(t), \qquad N_l(t) = \alpha_{\varnothing}(t). \tag{26}$$

The first paper should log these diagnostics and use them to characterize when memory reads are local, distant, confident, or null. A later adaptive-compute version can define

$$p_{\text{refine}}(l,t) = \sigma(a_H H_l(t) + a_D D_l(t) - a_N N_l(t) + a_0) \tag{27}$$

for extra refinement, larger expert budgets, or verifier passes. This connects Retro-DARC to frontier effort-control systems and creates a natural follow-up path from memory retrieval to compute allocation.

10 Retro-DARC-A: Typed Depth Memory for World/Action Planners

The world/action version is a serious extension with a natural path into world-model training.

10.1 Planner-Side Belief Memory

Consider a trajectory

$$\tau = (o_1, a_1, y_1), \ldots, (o_T, a_T, y_T), \tag{28}$$

where o_t is an observation, a_t is an action, and y_t may be reward, tool output, environment feedback, or proprioceptive state. A planner is asked to output a next action, repair step, future latent, or goal-conditioned plan. A Transformer planner processes a serialized or multimodal representation of τ and maintains implicit belief through hidden states.

Retro-DARC-A extends the memory bank with typed deltas:

$$\mathcal{M}_l^{\text{typed}}(t) = \{\Delta_i^{\text{text}}, \Delta_i^{\text{obs}}, \Delta_i^{\text{act}}, \Delta_i^{\text{tool}}, B_j^{\text{episode}}, A_k^{\text{belief}}, 0_{\varnothing}\}. \tag{29}$$

The scoring function adds type and temporal biases:

$$s_{l,m}(t) = \frac{q_l(t)^\top \,\text{RMSNorm}(P_K M_m(t))}{\sqrt{d_r}} + b_{\text{type(m)}} + b_{\text{age(l,m)}} + b_{\text{episode(m)}}. \tag{30}$$

10.2 Why This Is Profound but Not Confusing

The LLM core and the world/action extension answer the same abstract question: how can a pretrained or prelearned planner reuse its own internal computation history? The difference is the memory type. In an LLM, the relevant history is layer-delta computation over text tokens. In a WAM/VLA-style planner, the relevant history includes observation/action/tool deltas. Figure 2 shows the relationship.

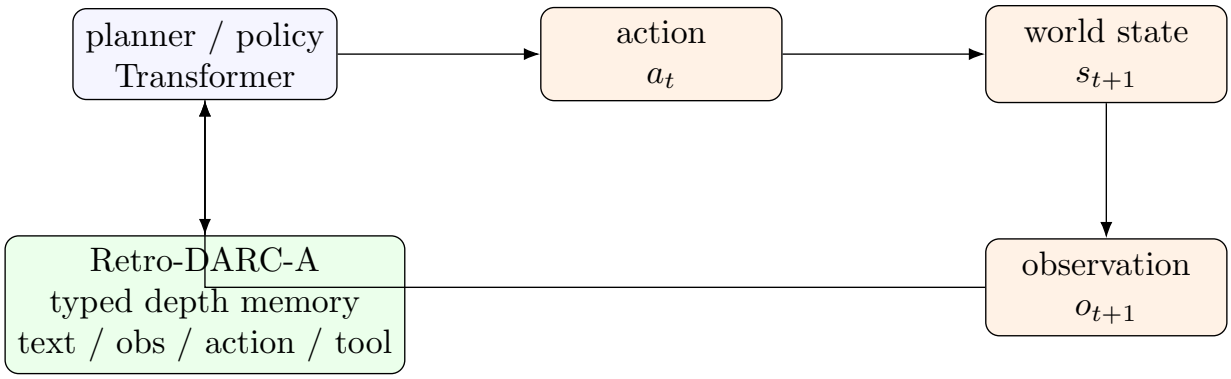

Figure 2: In a world-model taxonomy, Retro-DARC-A is a planner-side belief-memory interface. It does not render observations or simulate full states; it lets a pretrained planner retrieve prior internal deltas associated with text, observations, actions, tools, and episode-level beliefs.

10.3 Action-State Maintenance Probes

Before expensive robotics, we propose cheap **Action-State Maintenance Probes** (ASMP). Each example is a trajectory plus a query:

$$x = (o_1, a_1, y_1), \ldots, (o_T, a_T, y_T), q_T. \tag{31}$$

The target asks for a final hidden state, next valid action, invariant violation, repair step, or explanation of a failed action. ASMP can be generated from text games, browser traces, code-execution logs, spreadsheet workflows, and simulated POMDPs.

For a trained model, evaluate causal drops:

$$\Delta_{\text{mem}} = S(\mathcal{M}) - S(r = 0), \tag{32}$$

$$\Delta_{\text{shuffle}} = S(\mathcal{M}) - S(\text{shuffle}_{\text{example}}(\mathcal{M})), \tag{33}$$

$$\Delta_{\text{action}} = S(\mathcal{M}) - S(\text{shuffle}_{\text{action}}(\mathcal{M})). \tag{34}$$

If Δ_{action} grows with trajectory length or action-dependency depth, Retro-DARC-A has evidence of action-relevant belief-memory use.

11 Retro-DARC-WM: Object-Centric Latent Memory with Physics Residuals

The world-model extension builds on the LLM retrofit core and can be stated as a precise follow-up architecture. We call it **Retro-DARC-WM**. Its purpose is to combine the best parts of the explicit and implicit approaches: predict in a JEPA-style latent space, organize that latent space around objects or particles when possible, and add a checkable residual channel grounded by physics or environment verification.

11.1 Object-Centric Latent State

Let an encoder E map an observation o_t to object-centric latent slots

$$Z_t = E(o_t) = \{z_{t,1}, \ldots, z_{t,O}\}, \qquad z_{t,o} \in \mathbb{R}^{d_z}. \tag{35}$$

The slots may come from supervised object tracks, self-supervised slots, latent particles, or object masks. For text-only LLMs, the analogous units are token representations; for world models, the units are object or particle representations.

Given context latents $Z_{\leq t}$ and actions $a_{t:t+k-1}$, a JEPA-style predictor estimates future latent slots without reconstructing all pixels:

$$\widehat{Z}_{t+k} = P_\theta(Z_{\leq t}, a_{t:t+k-1}, r_{\leq t}), \tag{36}$$

where $r_{\leq t}$ denotes Retro-DARC memory reads. The latent prediction loss is

$$\mathcal{L}_{\text{JEPA}} = \sum_{o \in \mathcal{O}_{\text{mask}}} \|\widehat{z}_{t+k,o} - \text{stopgrad}(z_{t+k,o})\|_2^2, \tag{37}$$

with masking performed at the object or particle level rather than only at the patch level.

11.2 Codec-Conditioned Object and Event Memory

LLaVA-OneVision-2 suggests an intermediate route between raw frame prediction and full object-centric simulation. Before a system has reliable object slots, codec metadata can identify event-bearing regions that deserve latent capacity. Given codec saliency $\eta_{g,b}^{\text{codec}}$ from Eq. (19), a JEPA-style latent loss can be weighted by event evidence:

$$\mathcal{L}_{\text{codec-JEPA}} = \sum_{g,b} \omega_{g,b} \|\hat{z}_{g,b} - \text{stopgrad}(z_{g,b})\|_2^2, \qquad \omega_{g,b} = \sigma(\alpha_0 + \alpha_1 \eta_{g,b}^{\text{codec}}). \tag{38}$$

This does not replace object-centric prediction. It supplies cheap event supervision: high codec-residual regions are plausible candidates for object transitions, contacts, interactions, or state changes. A world/action Retro-DARC model can therefore write both object-level and codec-event memories:

$$\mathcal{M}_{l,t}^{\text{event}} = \{\Delta_{t-j,o}^{\text{obj}}, \Delta_{g,b}^{\text{codec}}, \Delta_{t-j}^{\text{act}}, B_{<b}^{\text{episode}}, A_{1:K}^{\text{belief}}, 0_\varnothing\}. \tag{39}$$

The purpose is not to make codec metadata a physics model. It is to use codec residuals as scalable weak supervision for where the latent planner should allocate memory and verification.

11.3 Differentiable-Physics-Grounded Residual

An explicit physical prior should not be asked to model everything. Instead, we propose a residual form. Let Φ_ψ be a differentiable or checkable physics module that predicts a coarse next state from object state and action:

$$\bar{Z}_{t+1} = \Phi_\psi(Z_t, a_t). \tag{40}$$

The neural world model predicts a residual correction

$$\widehat{Z}_{t+1} = \bar{Z}_{t+1} + R_\theta(Z_t, a_t, r_t). \tag{41}$$

Here R_θ is not free hallucination; it is conditioned on retrieved depth memory r_t, including previous contact hypotheses, action deltas, tool outcomes, and verification residuals.

For environments where an explicit differentiable simulator is unavailable, Φ_ψ can be replaced by a set of checkable constraints C_j over predicted states:

$$\mathcal{L}_{\text{phys}} = \sum_j \lambda_j C_j(\widehat{Z}_{t:t+k}, a_{t:t+k-1}). \tag{42}$$

Examples include object permanence, contact consistency, non-penetration, approximate energy or velocity bounds, gripper-object attachment consistency, and action-coordinate consistency. These constraints should be treated as verification signals, not as complete physics.

11.4 Typed Depth-Time Memory

Retro-DARC-WM extends the LLM memory bank from layer deltas to typed depth-time memory:

$$\begin{aligned}\mathcal{M}^{\mathrm{WM}}_{l,t} = \{&\Delta^{\mathrm{layer}}_{i,t,o},\ \Delta^{\mathrm{obj}}_{t-j,o},\ \Delta^{\mathrm{act}}_{t-j},\ \epsilon^{\mathrm{phys}}_{t-j,o},\\ &\Delta^{\mathrm{tool}}_{t-j},\ B^{\mathrm{episode}}_{<b},\ A^{\mathrm{belief}}_{1:K},\ 0_{\varnothing}\}.\end{aligned} \tag{43}$$

Here $\Delta^{\mathrm{layer}}_{i,t,o}$ is a model-depth delta for object o, $\Delta^{\mathrm{obj}}_{t-j,o} = z_{t-j+1,o} - z_{t-j,o}$ is a temporal object delta, $\Delta^{\mathrm{act}}_{t-j}$ is an action delta or action embedding, and

$$\epsilon^{\mathrm{phys}}_{t,o} = z_{t+1,o} - \Phi_{\psi}(z_{t,o}, a_t) \tag{44}$$

is a physics residual or constraint-violation embedding. A query can attend over memory by type:

$$s_m = \frac{q^{\top}\,\mathrm{RMSNorm}(P_K M_m)}{\sqrt{d_r}} + b_{\mathrm{type(m)}} + b_{\mathrm{age(m)}} + b_{\mathrm{object(m)}}. \tag{45}$$

The read r_t is then injected into the predictor or planner through the same zero-gated form as the LLM adapter.

11.5 Verification Memory and Physics-Slop Index

To distinguish useful learned dynamics from visually plausible physical error, the model should write verification outcomes back into memory. Define a verification vector

$$v_t = V(\widehat{Z}_{t:t+k}, Z_{t:t+k}, a_{t:t+k-1}, \widehat{o}_{t:t+k}), \tag{46}$$

where V may include latent prediction error, contact violation, collision violation, action success, or simulator check results. The verification memory stores

$$M^{\mathrm{verify}}_t = W_v v_t. \tag{47}$$

We propose reporting a *physics-slop index* (PSI) whenever decoded rollouts are evaluated:

$$\mathrm{PSI} = \frac{\mathbb{E}[\mathrm{PhysViolation}(\widehat{Z}, \widehat{o}, a)]}{\mathbb{E}[\mathrm{PerceptualError}(\widehat{o}, o)] + \epsilon}. \tag{48}$$

High PSI means predictions look acceptable under perceptual loss but violate physical or action constraints. A successful Retro-DARC-WM should reduce PSI, and ablating verification memory should increase it. This directly tests whether memory helps with the "physics slop" failure mode rather than simply improving visual prediction.

11.6 Training Objective

A world-model version can be trained with

$$\mathcal{L} = \mathcal{L}_{\mathrm{JEPA}} + \lambda_{\mathrm{phys}}\mathcal{L}_{\mathrm{phys}} + \lambda_{\mathrm{act}}\mathcal{L}_{\mathrm{act}} + \lambda_{\mathrm{verify}}\mathcal{L}_{\mathrm{verify}} + \lambda_{\mathrm{mem}}\mathcal{L}_{\mathrm{mem}}. \tag{49}$$

The memory term includes causal regularization: performance should degrade when the memory is shuffled, forced to null, or replaced with random same-norm memories. This is the same scientific standard as the LLM paper, transferred to world models.

Staged execution. Retro-DARC-WM is the world/action branch of the same residual-memory program. The efficient execution path validates the retrofit mechanism first, then applies the same causal-memory tests to object-centric latent prediction and physics-residual verification.

12 Theoretical Properties

Proposition 2 (Exact function preservation)**.** *Let F_θ be a pretrained Transformer. Insert Retro-DARC adapters of the form*

$$h_l^{\text{new}}(t) = h_l(t) + \gamma_l W_O r_l(t) \tag{50}$$

for any set of layers. If $\gamma_l = 0$ for all inserted adapters, then for every input x, all subsequent hidden states and final logits are identical to the original model up to numerical precision.

Proof. For each inserted layer, $\gamma_l = 0$ implies $h_l^{\text{new}}(t) = h_l(t)$. Thus the input to every following original sublayer is unchanged. By induction over layers, all hidden states and logits equal those of the base model. □

Proposition 3 (First-step gate trainability)**.** *Consider a loss $\mathcal{L}$ and a single adapter output $h' = h + \gamma W_O r$. At $\gamma = 0$,*

$$\frac{\partial \mathcal{L}}{\partial \gamma} = \left\langle \frac{\partial \mathcal{L}}{\partial h'}, W_O r \right\rangle. \tag{51}$$

Therefore, a zero gate is trainable at first step whenever $W_O r$ has nonzero projection onto the downstream gradient. If $W_O = 0$, this gradient is identically zero.

Proof. Differentiate $h' = h + \gamma W_O r$ with respect to γ and apply the chain rule. If $W_O = 0$, then $\partial h'/\partial \gamma = 0$. □

Lemma 1 (Delta memory is less redundant in a simple independent-update model)**.** *Assume $\Delta_i \in \mathbb{R}^d$ are independent, zero-mean, isotropic random updates with $\mathbb{E}\|\Delta_i\|^2 = \sigma^2 d$, and $h_l = \sum_{i=1}^{l} \Delta_i$. Then for $i < j$,*

$$\mathbb{E}[h_i^\top h_j] = i\sigma^2 d, \tag{52}$$

whereas $\mathbb{E}[\Delta_i^\top \Delta_j] = 0$ for $i \neq j$. Thus cumulative states share expected inner product due to common history, while distinct deltas do not.

Proof. Since $h_i = \sum_{a=1}^{i} \Delta_a$ and $h_j = \sum_{b=1}^{j} \Delta_b$, independence and zero mean eliminate cross terms except $a = b \leq i$, giving $i\sigma^2 d$. The delta result follows directly from independence and zero mean. □

Proposition 4 (Causal-memory falsifiability)**.** *Suppose Retro-DARC improves a score S over a parameter-matched adapter. If interventions that remove example-specific memory, such as $r_l = 0$ or cross-example memory shuffling, do not change S within confidence intervals, then the experiment does not support the claim that depth memory is responsible for the gain.*

This proposition turns mechanism attribution into an empirical estimand: the memory read is meaningful when performance depends on preserving example-specific memory content.

13 Executable Mechanism Checks

Before expensive LLM training, several claims can be checked exactly or through small synthetic programs. We include these checks in the release as executable scripts. They verify the mechanism-level assumptions that make larger pretrained-model runs well-founded.

Table 5: CPU-runnable mechanism checks currently included in the release. These tests validate the insertion invariant, zero-gate trainability, delta non-redundancy, residual-saliency retention, and typed action/physics residual memory in controlled settings.

Check	Metric	Current result
Zero-gated adapter identity	max absolute hidden-state error	0.000
First-step trainability	$\lvert\nabla_\gamma\rvert$ with $W_O \neq 0$	3.07×10^{-2}
Delta vs cumulative redundancy	adjacent cosine, cumulative vs delta	0.9353 vs 0.00065
Residual-saliency retention	sparse-event recall, top-k vs random	94.3% vs 25.2%
Delta memory readout	MSE, ideal delta vs final cumulative	0.216 vs 0.896
Physics-residual memory	MSE reduction with residual memory	66.4%
Typed action memory	final-state accuracy, full vs no-verification	100.0% vs 79.6%
Softmax1 null reserve	mean null mass under random logits	8.18%

The checks support the design choices needed before launching pretrained-model runs. First, exact identity is achieved by setting $\gamma = 0$. Second, zero-gate learning is active when W_O is nonzero. Third, cumulative hidden states are highly redundant under independent-update dynamics, while deltas are nearly orthogonal. Fourth, when useful evidence is sparse and correlated with residual saliency, top-k residual selection recovers event evidence far more efficiently than random retention. Fifth, delta memory can preserve information that is diluted in the final cumulative state. Sixth, typed verification and physics-residual memories are causal in controlled action-state and residual-dynamics settings. These checks are not substitutes for matched-compute pretrained-LLM training; they are the executable local validations that make those runs worth funding.

14 Verification Plan: Stage by Stage

The project is designed to move quickly while preserving clean attribution. Each stage produces a reusable artifact and unlocks the next scale.

14.1 Primary Estimands and Intervention Design

The core scientific question is causal: does the retrieved depth memory cause the performance gain? We use the language of interventions and potential outcomes in the standard causal-inference sense [1, 2]. We define an intervention operator $\mathcal{I}$ that changes the memory bank during evaluation while keeping prompts, decoding, weights, and scaffold fixed. Let $S(f, x)$ be a task score for model f on example x and let $f_\theta^{\mathcal{I}}$ be the same trained Retro-DARC model evaluated under intervention $\mathcal{I}$. The

primary causal-memory effect is

$$\mathrm{CME}(\mathcal{I}) = \mathbb{E}_{x\sim\mathcal{D}}\left[S(f_\theta, x) - S(f_\theta^{\mathcal{I}}, x)\right]. \tag{53}$$

The strongest interventions are:

1. **Zero memory:** set $r_l(t) = 0$ while keeping the trained adapter parameters fixed.
2. **Force null:** set $\alpha_\varnothing = 1$, testing whether the explicit no-memory route explains behavior.
3. **Cross-example shuffle:** replace $\mathcal{M}(x_i)$ with $\mathcal{M}(x_j)$ from another example in the same batch.
4. **Same-norm random memory:** replace each memory value by random vectors with matched RMS norms, testing whether scale alone explains gains.
5. **Depth-offset memory:** shift memory slots by a fixed depth offset, preserving statistics but destroying semantic alignment.
6. **Local-only memory:** retain only recent deltas, removing long-depth and block memories.
7. **Route corruption:** keep values fixed but randomly permute top-k routing assignments.

A validation-loss gain becomes a depth-memory result when it is accompanied by positive CME across interventions. A modest task gain with robust CME is especially valuable because it indicates that the example-specific computation trace is being used.

Negative controls. We also require interventions that should *not* strongly hurt. Shuffling memory on tasks with very short horizon, forcing null on trivial next-token continuations, or removing distant memory on local syntax tasks should have smaller effects than on long-horizon code, tool, event, or action-state tasks. This selectivity is important: a method that is equally fragile under every perturbation may be unstable rather than memory-dependent.

14.2 Statistical Reporting Standard

All empirical results should report uncertainty and cost. For validation loss, use document-level bootstrap intervals and report the loss difference against each baseline:

$$\Delta L_A = L_{\text{Retro-DARC}} - L_A. \tag{54}$$

For downstream tasks, use paired bootstrap over examples with the same prompts and decoding seeds. For multi-trial agent tasks, report both average success and reliability pass^k where applicable. For compute, report both nominal tokens and measured wall-clock/FLOP overhead:

$$\rho = \frac{\mathrm{cost}_{\text{Retro-DARC}} - \mathrm{cost}_{\text{base}}}{\mathrm{cost}_{\text{base}}}. \tag{55}$$

A claim of superiority must survive parameter matching, compute-overhead correction, and a memory-causal intervention. We recommend Holm-Bonferroni correction across benchmark families and pre-registering one primary validation metric plus one primary long-horizon metric to avoid benchmark shopping.

14.3 Stage 0: Literature Lock and Baseline Freeze

Duration: 3–5 days. **Cost:** negligible.

Freeze the exact claims, baselines, and evaluation harness. The method is positioned as function-preserving retrofit of depth memory, not as the first depth-routing mechanism.

Required baselines:

1. continued training only,
2. LoRA with matched trainable parameters,
3. standard 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 or value-mixing baseline if feasible.

14.4 Stage 1: Identity and Gradient Tests

Duration: 1–2 days. **Cost:** negligible.

Run a tiny random Transformer and at least one pretrained open LLM. Verify:

$$\max_x \|z_{\theta,\phi_0}(x) - z_\theta(x)\|_\infty < \epsilon, \tag{56}$$

with $\epsilon \approx 10^{-5}$ for fp32 and an appropriate bf16 tolerance. Verify $\|\nabla_\gamma\| > 0$ on a standard language-modeling batch.

Gate: train after exact preservation and nonzero gradients are verified.

14.5 Stage 2: Frozen Delta-Memory Probes

Duration: 2–5 days. **Cost:** low.

Use a frozen 0.5B–0.8B model to collect layer deltas on 50M–200M tokens. Train only a small readout/router to predict next-token residual correction or validation-loss improvement. Compare random memory, cumulative states, local deltas, block summaries, and delta+null memory.

Gate: deltas should beat cumulative states, and memory shuffling should hurt.

14.6 Stage 3: 0.6B Adapter-Only Signal Gate

Duration: 1–2 weeks. **Cost:** moderate.

Use a public base model such as Qwen3-0.6B or a similar small dense LLM [17]. Freeze the base model and train only adapters on 0.5B–1B tokens from an open corpus such as FineWeb/FineWeb-Edu or Dolma [23, 24]. Use three seeds.

Compare Retro-DARC-Lite to LoRA, residual adapters, MLP adapters, and delta/null retrofit baselines at matched trainable parameter count.

Gate: Retro-DARC must beat at least one strong parameter-matched adapter and cumulative-state depth adapter. Zeroing or shuffling the memory must remove a measurable fraction of the gain.

14.7 Stage 4: 1B–2B Confirmation

Duration: 2–4 weeks. **Cost:** moderate.

Repeat with Qwen3-1.7B or comparable open models. Use 1B–3B training tokens and two or three seeds. Include the best baselines from Stage 3.

Gate: the gain must survive scale. If the 0.6B signal changes at 1B–2B, use the comparison to redesign memory retention before larger runs.

14.8 Stage 5: 3B Confirmation

Duration: 3–6 weeks. **Cost:** higher.

Use a 3B open model such as SmolLM3-3B or a Qwen3-family model if available and licensing fits the release [18]. Run 1B–5B adapter-training tokens with one or two seeds. This stage makes the paper credible beyond toy scale.

Gate: Retro-DARC must improve validation loss or a long-horizon task, and memory ablations must causally reduce the gain.

14.9 Stage 6: Causal Memory Interventions

For every successful checkpoint, evaluate:

1. $r_l = 0$ for all adapters,
2. force null route,
3. shuffle memory across examples,
4. shuffle memory across positions,
5. replace deltas with cumulative hidden states,
6. local-only memory,
7. remove block summaries,
8. remove anchors if using Full.

A convincing result is:

$$S_{\text{full}} > S_{\text{shuffled}} \quad \text{and} \quad S_{\text{full}} > S_{r=0} \tag{57}$$

with confidence intervals excluding zero on at least one state-maintenance workload.

14.10 Stage 7: Long-Horizon and Code Evaluation

Use a small but meaningful suite:

1. language-model validation cross-entropy,
2. LongBench v2 subset for long-context reasoning [19],
3. LiveCodeBench subset for code generation/self-repair [21],
4. SWE-bench Lite or a small SWE-bench Verified subset [20],
5. Tau-bench or a small fixed tool-use harness [22],

Table 6: Registered escalation gates.

Gate	Pass condition
Identity	logits match base at insertion within tolerance
Delta	delta memory beats cumulative memory in probe and pilot
Adapter	Retro-DARC beats parameter-matched LoRA/residual adapter at 0.6B
Scale	gain survives at 1B–2B
Causal	zeroing/shuffling memory removes meaningful fraction of gain
Long-horizon	at least one state-maintenance/code/tool task improves
Systems	decode/prefill overhead remains within predefined bounds

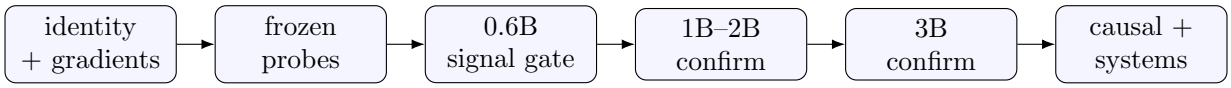

Figure 3: Retro-DARC should be verified through staged gates. 3B and world/action experiments start after the cheaper retrofit signals are positive.

6. ASMP trajectory probes for state maintenance.

Report task success, cost-normalized success, tokens used, tool-call errors, and causal ablation drops.

14.11 Stage 8: Systems and Quantization

Measure prefill tokens/sec, decode tokens/sec, peak VRAM, memory-bank size, p50/p95 latency, and router overhead. Test BF16, FP8 memory values, and W8A8 if available. The first paper should report systems overhead with the same precision as quality metrics.

Figure 3 summarizes the staged plan.

14.12 Stage 9: LLaVA-OneVision-2 Multimodal Event-Memory Testbed

After LLM-only Retro-DARC passes the 0.6B–3B gates, the most efficient multimodal extension is to retrofit an open perception model such as LLaVA-OneVision-2. The recommended setup is:

1. Use the released LLaVA-OneVision-2-8B-Instruct checkpoint as the base multimodal model.
2. Insert Retro-DARC adapters into the Qwen3-8B language decoder first; keep the vision encoder frozen.
3. Verify exact logit preservation on image, frame-video, and codec-video inputs.
4. Train adapter-only on a small subset of LLaVA-OneVision-2 video-caption and spatial data.
5. Compare frame backend, codec backend, codec backend plus Retro-DARC-Lite, and codec backend plus Residual-Aligned Retro-DARC.

The evaluation should focus on tasks where event memory matters:

$$\Delta_{\text{event-mem}} = S(\text{Retro-DARC}) - S(\text{memory-shuffled Retro-DARC}). \tag{58}$$

Suitable tasks include JumpScore, temporal grounding, video QA under low token budgets, referring video tracking, spatial grounding, and manipulation-trace reasoning. The core ablations are:

1. zero all depth-memory reads;
2. shuffle depth memory across videos;
3. shuffle codec saliency metadata while keeping video tokens fixed;
4. force null route;
5. remove visual-token-associated memories;
6. compare frame-only and codec-conditioned routing.

A positive result would show that codec residuals and layer deltas are complementary: codec metadata identifies event-bearing observations, while Retro-DARC preserves the internal computations produced by those observations.

15 Systems and Complexity

For a memory bank with M candidates, router dimension d_r, and top-k k, the per-token router cost per inserted layer is roughly

$$O(Md_r + kd). \tag{59}$$

With local window w, N_B block summaries, and K anchors,

$$M = w + N_B + K + 1. \tag{60}$$

The parameter overhead per inserted layer is approximately

$$O(dr + rd_r + dd_r + dd_{\text{out}}), \tag{61}$$

corresponding to low-rank query maps, key projection, and output projection. Sharing P_K and W_O across blocks can reduce overhead.

Memory layout. The first implementation can store memory as

$$\texttt{memory\_bank} : [B, T, M, d] \tag{62}$$

for prefill. For single-token decode, a layout like $[M, B, d]$ may be faster. The correct layout is hardware-dependent and must be profiled.

Detach mode. In adapter-only training, memory values can be detached:

$$M_m \leftarrow \text{stopgrad}(M_m), \tag{63}$$

which reduces activation memory and makes the method closer to PEFT. A non-detached variant should be evaluated after signal is positive.

Overhead targets. A practical first paper should target:

$$\text{decode overhead} \leq 5\%\text{–}8\%, \quad \text{prefill overhead} \leq 8\%\text{–}12\%, \quad \text{peak VRAM overhead} \leq 10\%\text{–}15\%. \tag{64}$$

If overhead exceeds these targets, the optimization path becomes part of the release: smaller windows, shared projections, fused routing, and FP8 memory values.

Systems competitor. Depth-Attention claims cross-layer value mixing with no extra persistent inference state beyond the standard KV cache [9]. Retro-DARC should report this comparison directly: its advantage is safe retrofit and causal memory attribution, while Depth-Attention-style methods set the overhead bar.

16 Implementation and Reproducibility

A public release should expose one simple entry point:

```
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,
)
```

Then the correctness check is exact:

```
with torch.no_grad():
    base_logits = base_model(input_ids).logits
    darc_logits = model(input_ids).logits
err = (base_logits - darc_logits).abs().max().item()
assert err < tolerance
```

The repository should include:

1. adapter injection utilities for common decoder-only architectures,
2. identity-preservation tests,
3. gradient tests for zero gate and nonzero W_O,
4. LoRA/residual/MLP adapter baselines,
5. frozen-probe scripts,
6. causal memory intervention hooks,

7. validation-loss and long-horizon evaluation harnesses,
8. latency profiling scripts.

17 What Would Count as a Significant Result?

The result tiers are clear. A mechanism result establishes identity preservation and synthetic routing behavior. An adapter result improves over continued training or generic adapters. A strong depth-memory result shows:

1. lower validation loss than parameter-matched LoRA/residual/MLP adapters,
2. improvement over delta/null depth-retrofit baselines,
3. causal memory ablation drops,
4. survival from 0.6B to 1B–2B and ideally 3B,
5. improvement on at least one long-horizon/code/tool-state task,
6. acceptable overhead.

The most important finding would be:

> *Same pretrained model, same trainable parameter budget, same continued-training tokens: Retro-DARC improves performance, and memory shuffling removes the gain.*

That would show that the improvement comes from reusable internal computation history rather than generic adapter capacity.

For world/action models, the analogous strong result would be:

> *Typed observation/action/tool depth memories improve action-state maintenance, and action-memory corruption specifically damages action-dependent decisions.*

This would make Retro-DARC a plausible planner-side memory primitive for WAM/VLA systems.

18 Mechanism Disambiguation

The Retro-DARC memory thesis is strongest when simpler explanations are measured directly. Table 7 lists the main competing explanations and the interventions that separate them from example-specific depth-memory use.

The decisive pattern is not merely that Retro-DARC improves a score. It is that Retro-DARC improves over parameter-matched non-memory adapters and loses its advantage under interventions that destroy the content of the memory while preserving scale, shape, and compute.

19 Design Stress Tests

Adapter capacity masquerading as memory. Retro-DARC is compared against parameter-matched LoRA, residual, and MLP adapters. The memory interventions then test whether the gain is content-specific.

Table 7: Alternative explanations and disambiguating tests.

Alternative	Why it could explain gains	Disambiguating test
Extra trainable capacity	Any adapter can improve continued training.	LoRA, residual, and MLP adapters with matched trainable parameters and training tokens.
Optimization regularization	The zero gate or router may stabilize training without using memory.	Memory-shuffle and route-corruption interventions; compare to no-memory gated adapters.
Scale or norm effects	Large memory vectors may change activations regardless of content.	Same-norm random memory and RMS-matched shuffled memory.
Local smoothing	Only recent residuals matter, not long-depth memory.	Local-only vs block/anchor memory and horizon-stratified evaluation.
Data or scaffold advantage	Gains may come from different data order, prompt, tool harness, or decode policy.	Fixed data order, fixed agent harness, paired evaluation, and public configs.
Codec prior only	In multimodal tests, codec saliency may help without depth memory.	Compare codec backend alone, codec-conditioned routing, and codec metadata shuffling.
World-model confounding	Typed memories may improve representation learning without affecting action dynamics.	Action-memory and verification-memory ablations on object/action prediction and planning.

Zero-gate learning speed. Exact identity preservation uses $\gamma = 0$ and nonzero W_O. The first-step gradient proposition shows why this is trainable. A small nonzero gate warmup can be explored as an ablation, but the primary method preserves exact insertion identity.

Memory bandwidth. Lite keeps the deployment path practical through small local windows, top-k routing, detach mode, optional block summaries, and delayed kernel work. The optimization path is explicit: shared projections, FP8 memory banks, fused RMSNorm/dot/top-k routing, and codec/depth saliency to reduce candidate count.

Competition from in-cache methods. Depth-Attention-style baselines are treated as systems competitors. Retro-DARC's distinct advantage is retrofit into existing checkpoints plus causal memory attribution.

Long-horizon specificity. The method is expected to be most valuable when useful evidence appears as state-changing computation: code repair, proof subgoals, tool errors, action transitions, event localization, or physics-residual violations. Short local tasks are useful negative controls.

20 Discussion

Retro-DARC reframes residual-stream improvement as a retrofit problem. The key distinction is between building a new model with better depth routing and giving an existing model a trainable interface to its own intermediate computations. The latter is cheaper to test, easier to adopt, and scientifically sharper: if causal ablations show that layer-delta memories matter, then the method reveals an underused resource inside pretrained models.

The world-model perspective strengthens the program. Renderers output observations; simulators output states; planners output actions. Retro-DARC belongs inside the planner as a memory interface over internal belief updates. For explicit, checkable world models, the memory bank can store object, contact, constraint, and verification deltas. For scaled WAM-style learned dynamics, it can store action-conditioned rollout deltas and failed-prediction residuals. For JEPA-style models, it stores latent prediction deltas instead of pixel reconstructions. The same mechanism therefore connects LLM depth memory, multimodal event memory, and planner-side belief memory.

The adoption path is simple. A researcher can inject Retro-DARC-Lite into an open pretrained LLM, verify exact logit preservation, train only adapter parameters, and run causal memory ablations. That makes Retro-DARC closer to a new PEFT primitive than to a conventional architecture replacement: LoRA adapts weights, while Retro-DARC adapts access to internal computation history.

21 Research Boundaries

The claims are organized by execution stage. The algebraic properties and mechanism checks establish the insertion invariant and the basic behavior of residual memory. The pretrained-LLM experiments establish adapter utility and causal depth-memory use. The multimodal and world/action sections define concrete extensions using the same mathematical interface: codec-conditioned residual saliency, typed action/observation/tool memories, object-centric JEPA targets, and physics-residual verification. This structure lets the paper claim a broad research program while keeping each empirical conclusion tied to a measurable test.

22 Conclusion

We introduced Retro-DARC, a function-preserving depth-memory adapter family for pretrained LLMs and world/action planners. The core method exposes layer-delta memories through a zero-initialized read path that preserves the model's logits at insertion, then learns to retrieve useful internal computation traces during continued training. The architecture combines delta memory, low-rank token-conditioned queries, null-aware top-k routing, and zero-gated residual injection. The broader Residual Evidence Principle unifies codec residuals in video perception, layer deltas in Transformer depth, JEPA latent residuals, and physics-verification residuals in world/action models.

The immediate research target is decisive and practical: same pretrained model, same trainable parameter budget, same continued-training tokens, Retro-DARC should improve over parameter-matched adapters, and memory interventions should remove the advantage. If this pattern holds, Retro-DARC establishes a new PEFT-like primitive: function-preserving access to reusable internal computation history. The longer-term program extends the same memory interface to multimodal event perception and planner-side belief memory, where typed text, observation, action, tool, object, and physics-residual memories become first-class substrates for long-horizon intelligence.

A Algorithm

Listing 1: Retro-DARC-Lite forward pass for one inserted adapter.

```
def retro_darc_adapter(h_base, memory_bank, router, gate):
    # h_base is the output of the original pretrained layer.
    q = router.query(norm(h_base))
    M = memory_bank.candidates()  # deltas, block summaries, null
    K = norm(router.key_proj(M))
    scores = (q @ K.transpose(-1, -2)) / sqrt(router.dim)
    scores = scores + router.bias(M)
    idx = topk(scores, k=router.topk)
    selected_scores = gather(scores, idx)
    selected_values = gather(M, idx)

    # Softmax1: denominator includes 1.0 reserve for null/no-op.
    weights = exp(selected_scores)
    weights = weights / (1.0 + weights.sum(dim=-1, keepdim=True))
    r = (weights[..., None] * selected_values).sum(dim=-2)

    # gate.value is exactly zero at insertion.
    return h_base + gate.value * router.out_proj(r)
```

B Minimal Experiment Configuration

Table 8: Minimum credible configuration.

Component	Setting
Models	0.5B–0.8B and 1B–2B pretrained dense LLMs
Training tokens	0.5B–3B per run
Seeds	3 small, 2 medium
Sequence length	2k–8k, matched across methods
Trainable budget	matched to LoRA/residual/MLP controls
Memory	local window $w = 4$, optional $N_B \leq 8$ block summaries
Router	rank $r = 16$–32, $d_r = 32$–64, top-k $k = 2$–4
Gate	scalar per inserted layer, initialized to exactly zero
Insertion	every 4 layers or block boundaries
Metrics	validation CE, causal ablation drop, decode/prefill overhead

C Registered Escalation Criteria

Before scaling beyond the pilot, use the following criteria to decide the next experiment and keep the claim aligned with the evidence:

1. Retro-DARC should beat LoRA or same-parameter residual adapters at 0.5B–0.8B before larger runs.

2. The 1B–2B run should preserve or clarify the 0.5B–0.8B signal.
3. Memory shuffling or zeroing should reduce the Retro-DARC advantage on state-maintenance tasks.
4. Delta memory should outperform cumulative hidden-state memory in at least the probe or pilot.
5. Decode overhead should be reduced through smaller windows, shared projections, or fused routing when it exceeds the target range.
6. Depth-Attention-inspired retrofit baselines should be reported as the systems comparison point.
7. ASMP/action-state probes should show a selective effect from action-memory corruption before the world/action claim is emphasized.

D Statistical Reporting

For validation loss, report paired checkpoint comparisons and mean with 95% confidence intervals across seeds where available. For downstream benchmarks, use paired bootstrap over examples. For agent/tool tasks, fix the harness, decoding settings, max turns, compaction policy, and retry policy before evaluation. A result should be called significant only if the confidence interval excludes zero and the effect survives parameter-matched controls.

E Retrofitting Variants to Ablate

Table 9: Core ablations.

Ablation	Question
Cumulative states vs. deltas	Are layer contributions better memories than hidden states?
Static query vs. token query	Is token-conditioned retrieval necessary?
No null vs. Softmax1/null	Does null routing prevent forced mixing?
Top-k $k = 1, 2, 4, 8$	What quality/latency trade-off is best?
Local-only vs. block memory	Is distant depth useful?
No anchors vs. anchors	Do persistent slots help long-horizon tasks?
Detached vs. non-detached memory	Is PEFT-style training sufficient?
Shared vs. per-layer router	Can overhead be reduced without losing quality?
Scalar vs. token gate	Is token-specific injection worth cost?

F World/Action Experimental Ladder

The world/action ladder should explicitly test the LeCun–Li–Fan synthesis rather than merely cite it.

Ladder A: JEPA latent-memory probe. Freeze a video encoder, train a small action-conditioned predictor with and without Retro-DARC memory, and predict future latent states under object or patch masking. Compare MLP predictor, recurrent predictor, Transformer predictor, and Retro-DARC predictor.

Ladder B: object-centric intervention. Use object slots or latent particles and mask whole objects. Test whether retrieved memory improves counterfactual object reasoning. Ablate by shuffling object memories across objects and across videos.

Ladder C: physics-residual verification. Add differentiable or checkable physics constraints. Report latent prediction loss, physical-constraint violation, and the physics-slop index in Eq. (48). The key test is whether verification memory reduces physically invalid but visually plausible rollouts.

Ladder D: action-conditioned planning. Train a latent action-conditioned predictor on small robot trajectories. Evaluate model-predictive control or goal-conditioned planning in simulation before any real robot. Compare action success with normal memory, null-forced memory, shuffled action memory, and shuffled verification memory.

Pass condition. The world/action claim is established when Retro-DARC-WM improves latent prediction or action success and causal ablations show that typed memory, especially physics-residual or action memory, is responsible for the gain.

The world/action extension should follow this ladder:

1. **ASMP-Synthetic:** text POMDPs with hidden state and distractor observations.
2. **ASMP-Tool:** tool-call traces where validity depends on earlier constraints.
3. **ASMP-Code:** edit/test/error/fix traces where repairs depend on failed-test memory.
4. **ASMP-EmbodiedText:** textual household/gridworld trajectories with partial observability.
5. **VLA/WAM pilot:** typed memory on a small action-observation Transformer if the first four succeed.

Only the first four are needed to make the world-model implication credible in an LLM paper.

References

[1] Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology.*

[2] Pearl, J. (2009). Causality: Models, reasoning, and inference. *Cambridge University Press.*

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems.*

[4] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[5] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

[6] Kimi Team. (2026). Attention residuals. *arXiv preprint arXiv:2603.15031*.

[7] Luo, C., Cai, Z., and Hu, J. (2026a). Delta attention residuals. *arXiv preprint arXiv:2605.18855*.

[8] Luo, H., Dai, H., Zhang, S., Chen, X., Jiang, E. H., Li, Y., Huang, J., Qiu, C., Xu, C., Pan, Z., Zhang, H., Wang, B., and Chen, Y. (2026b). Attention sinks and outliers in attention residuals. *arXiv preprint arXiv:2605.17887*.

[9] Zeng, B., Hao, Y., Wang, Z., Song, S., Li, H., Song, F., Liu, Y., He, Z., Wang, X., and Lin, Z. (2026). Depth-attention: Cross-layer value mixing for language models. *arXiv preprint arXiv:2606.05014*.

[10] Zhu, L., Fang, Y., Liao, B., Wang, S., Cheng, T., Huang, Z., Chen, C., Wei, L., Zeng, Y., Wang, Y., Lin, Y., Li, Y., and Wang, X. (2026). Mixture-of-depths attention. *arXiv preprint arXiv:2603.15619*.

[11] Xiao, D., Meng, Q., Li, S., and Yuan, X. (2025). MUDDFormer: Breaking residual bottlenecks in Transformers via multiway dynamic dense connections. *arXiv preprint arXiv:2502.12170*.

[12] Heddes, M., Javanmard, A., Axiotis, K., Fu, G., Bateni, M., and Mirrokni, V. (2025). Deep-CrossAttention: Supercharging Transformer residual connections. *Proceedings of the 42nd International Conference on Machine Learning*.

[13] Raposo, D., Ritter, S., Richards, B., Lillicrap, T., Humphreys, P. C., and Santoro, A. (2024). Mixture-of-depths: Dynamically allocating compute in Transformer-based language models. *arXiv preprint arXiv:2404.02258*.

[14] Elhoushi, M., Shrivastava, A., Liskovich, D., Hosmer, B., Wasti, B., Lai, L., Mahmoud, A., Acun, B., Agarwal, S., Roman, A., Aly, A. A., Chen, B., and Wu, C.-J. (2024). LayerSkip: Enabling early exit inference and self-speculative decoding. *arXiv preprint arXiv:2404.16710*.

[15] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

[16] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for NLP. *International Conference on Machine Learning*.

[17] Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., and others. (2025). Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

[18] Hugging Face. (2025). SmolLM3: smol, multilingual, long-context reasoner. *Hugging Face Blog*.

[19] Bai, Y. and others. (2025). LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *Annual Meeting of the Association for Computational Linguistics*.

[20] Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. (2024). SWE-bench: Can language models resolve real-world GitHub issues? *International Conference on Learning Representations*.

[21] Jain, N., Han, K., Gu, A., Li, W.-D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., and Stoica, I. (2024). LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

[22] Yao, S., Shinn, N., Razavi, P., and Narasimhan, K. (2025). Tau-bench: A benchmark for tool-agent-user interaction in real-world domains. *International Conference on Learning Representations*.

[23] Penedo, G., Kydlicek, H., Allal, L. B., Lozhkov, A., Mitchell, M., Raffel, C., von Werra, L., and Wolf, T. (2024). The FineWeb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*.

[24] Soldaini, L., Kinney, R., Bhagia, A., Schwenk, D., Atkinson, D., Authur, R., Bogin, B., Chandu, K., Dumas, J., Elazar, Y., and others. (2024). Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*.

[25] Hu, K., Qiu, L., Lu, Y., Zhao, H., Li, T., Kim, Y., Andreas, J., and He, K. (2026). ELF: Embedded language flows. *arXiv preprint arXiv:2605.10938*.

[26] Li, F.-F. (2026). A functional taxonomy of world models: Renderers, simulators, planners, and the loop that connects them. *Substack post*, June 3, 2026.

[27] Wang, S., Shi, J., Fu, Z., He, X., Liu, F., Yang, C., Zhou, Y., Fei, Z., Gong, J., Fu, J., Shou, M. Z., Huang, X., Qiu, X., and Jiang, Y.-G. (2026). World action models: The next frontier in embodied AI. *arXiv preprint arXiv:2605.12090*.

[28] Ye, S., Ge, Y., Zheng, K., Gao, S., Yu, S., Kurian, G., Indupuru, S., Tan, Y. L., Zhu, C., Xiang, J., Malik, A., Lee, K., Liang, W., Ranawaka, N., Gu, J., Xu, Y., Wang, G., Hu, F., Narayan, A., Bjorck, J., Wang, J., Kim, G., Niu, D., Zheng, R., Xie, Y., Wu, J., Wang, Q., Julian, R., Xu, D., Du, Y., Chebotar, Y., Reed, S., Kautz, J., Zhu, Y., Fan, L., and Jang, J. (2026). World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*.

[29] Zheng, R., Niu, D., Xie, Y., Wang, J., Xu, M., Jiang, Y., Castaneda, F., Hu, F., Tan, Y. L., Fu, L., Darrell, T., Huang, F., Zhu, Y., Xu, D., and Fan, L. (2026). EgoScale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*.

[30] NVIDIA. (2026). What is a world action model (WAM)? *NVIDIA Glossary*.

[31] Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. (2023). Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*.

[32] Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., and Ballas, N. (2023). Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*.

[33] Meta AI. (2024). V-JEPA: The next step toward advanced machine intelligence. *Meta AI Blog*.

[34] Bardes, A., Garrido, Q., Ponce, J., Chen, X., Rabbat, M., LeCun, Y., Assran, M., and Ballas, N. (2024). V-JEPA: Latent video prediction for visual representation learning. *International Conference on Machine Learning*.

[35] Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Komeili, M., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., and others. (2025). V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*.

[36] Nam, H. and others. (2026). Causal-JEPA: Learning world models through object-level latent interventions. *arXiv preprint arXiv:2602.11389*.

[37] Daniel, T., Qi, C., Haramati, D., Zadeh, A., Li, C., Tamar, A., Pathak, D., and Held, D. (2026). Latent Particle World Models: Self-supervised object-centric stochastic dynamics modeling. *ICLR 2026 Oral*.

[38] An, X., Xie, Y., Tang, F., Yan, Y., Tan, H., Zhu, D., Chen, C., Zhao, X., Qin, B., Yang, K., Shen, Y., Zhang, Y., Zhang, K., Zhang, W., Cheng, Z., Zhang, N., Wu, C., Ge, C., Ran, Z., Song, D., Li, C., Feng, S., Hu, M., Chen, Z., Niu, J., Li, B., Feng, Z., Liu, Z., Ge, Z., and Deng, J. (2026). LLaVA-OneVision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*.

[39] Tang, F., An, X., Yan, Y., Xie, Y., Qin, B., Yang, K., Shen, Y., Zhang, Y., Li, C., Feng, S., Chen, C., Tan, H., Hu, M., Zhang, M., Li, B., Feng, Z., Liu, Z., Ge, Z., and Deng, J. (2026). OneVision-Encoder: Codec-aligned sparsity as a foundational principle for multimodal intelligence. *arXiv preprint arXiv:2602.08683*.

[40] EvolvingLMMs Lab. (2026). LLaVA-OneVision-2: Fully open framework for democratized multimodal training. *GitHub repository*.

[41] LMMS Lab Encoder. (2026). LLaVA-OneVision-2-8B-Instruct model card. *Hugging Face model card*.

[42] MVP Lab. (2026). LLaVA-OneVision-2-Data dataset card. *Hugging Face dataset card*.