Memory Inception: Latent-Space KV Cache Manipulation for Steering LLMs

Andy Zeyi Liu[*] Yale University

Michael Zhang Princeton University

Ilana Greenberg Yale University

Adam Alnasser Jump Trading

Lucas Baker Jump Trading

John Sous[*] Yale University

Abstract

Prompting and activation steering are commonly used to guide the behavior and personality of LLMs, yet each has its drawbacks: prompting caches repeatedly and is diluted over long context, while activation steering is weaker and struggles to convey nuance. This work introduces *Memory Inception* (MI), which reformulates steering as a selective KV cache allocation problem and thus steers in latent attention space via text-derived key-value banks applied to specific layers. On personality-steering tasks, its effectiveness is similar to prompting and consistently higher than contrastive addition, while offering greater flexibility in mid-conversation shifts without rewriting transcripts. Additionally, it scores higher than both on a post-shift alignment test with Qwen3. MI outperforms prompting on HARDMath and PHYSICS (10/12 subject$\times$mode cells) while cutting added KV storage by up to 118$\times$ on matched content. For steering that requires persistent application and structure, or when transcript manipulation is expensive, MI offers a powerful yet flexible new method. Our code is available here.

1 Introduction

The standard method of steering language model behavior, whether for crafting personality, altering tone, or persisting facts and preferences for later use, is prompting: placing instructions verbally in context. Activation steering, often used in alignment research, is another commonly seen method that works by directly updating hidden states through learned directions or contrastive residuals. Prompting is powerful and flexible, but often heavy-handed and exhibits diminishing effects over time. Activation steering, in contrast, is compact but rigid, and cannot easily express multi-part or structured behavioral modifications.

In this paper, we study latent attention state as a means of steering, introducing *memory inception* (MI) to balance the power of prompting with the exactitude of activation steering. As illustrated in Figure 1, MI is a training-free steering method that uses latent key-value slots to encode text-based descriptors, facts, and heuristics. The model can then attend to these reminder banks at selected attention sites while keeping steering outside the visible transcript. Rather than impacting all L layers as with prompting, MI targets only the small subset of layers L_{ctrl} chosen to maximize behavioral impact, dramatically shrinking its KV cache footprint.

We evaluate MI across three regimes. Firstly, we evaluate the tradeoff between control and drift on matched personality tasks in which all three methods receive the same control signal. Secondly, we measure the effectiveness of each method on persistent activation and mid-dialogue updates in long conversations. Finally, we test whether the same mechanism can also propagate reasoning

[*]Correspondence to: andy.liu@yale.edu and john.sous@yale.edu.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

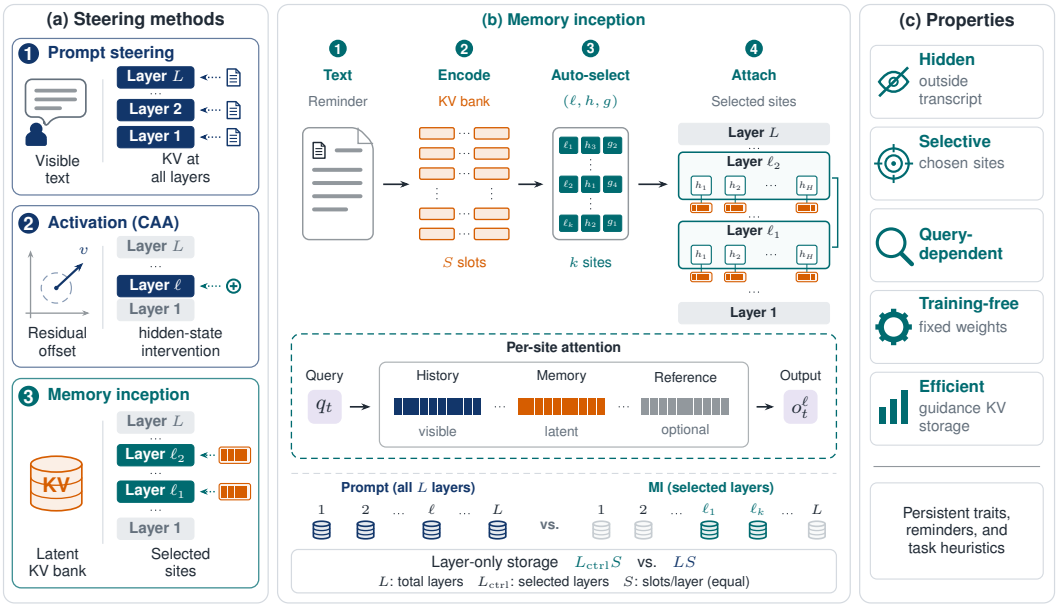

Figure 1: **Memory inception (MI).** Reminder text is encoded into latent memory bank slots in the KV cache, then appended to a subset of automatically chosen layers, heads, or key-value groups. MI improves on prompting by keeping the reminder outside the visible transcript. Unlike contrastive activation steering, which acts consistently on specific hidden states, MI can act differently per query based on the attention directed to the reminder slots.

techniques in the HARDMath [Fan et al., 2025] and PHYSICS [Feng et al., 2025] benchmarks. Our main contributions are as follows:

- **Method.** We introduce *memory inception* (MI), a training-free method that encodes reminder text into latent KV banks and makes them available at selected attention sites.
- **Implementation.** We formulate MI as bank-level attention routing. We give a canonical-key RoPE implementation with query-head selection for Llama and KV-group selection for Qwen3.
- **Evaluation.** On the primary personality-steering tasks, MI remains competitive with prompting while outperforming CAA on average drift-adjusted scores. On PHYSICS, MI outperforms prompting in 10/12 subject×mode cells. On HARDMath, MI improves over the plain model but trails prompting overall.
- **Estimated KV storage.** Under content-matched accounting, we estimate reductions of 6–118× in guidance-derived KV storage relative to visible prompting.

2 Related Work

The list below covers the methods most directly comparable to MI, with additional related work in Appendix A.

Prompting and latent prompts. Prompting is still the most prevalent means of steering language models, from context instructions to chain-of-thought [Brown et al., 2020, Wei et al., 2022, Kojima et al., 2022]. A closely related family of methods learns continuous prefixes without changing model weights [Li and Liang, 2021, Lester et al., 2021]. Like prefix-tuning, MI provides latent information that subsequent tokens can attend to.

Activation and attention steering. Activation Addition and CAA steer models by adding residual vectors derived from context pairs of positive and negative examples [Turner et al., 2023, Rimsky et al., 2024], but usually this is a few words with one precise meaning. Representation-engineering and function-vector work likewise shows that reusable latent directions can capture high-level behaviors or in-context functions [Zou et al., 2023, Todd et al., 2024]. Related attention-based methods, such as AutoPASTA and Spotlight, steer by allocating more attention toward instruction tokens [Zhang et al., 2024, Venkateswaran and Contractor, 2026]. In contrast, MI adds text-derived KV slots for attention

to read. Belitsky et al. [2025] steer frozen models using contrastive KV vectors from prompts with and without reasoning traces. They shift existing cache entries once after prefill at a designed prompt position across layers.

3 Memory Inception: Formalism & Implementation

Memory inception (MI) is a training-free steering method that encodes reminder content into latent KV banks and attaches those banks only at a small set of selected layers and attention units (we use unit because under different attention mechanism this has different meanings), where the model attends to them alongside ordinary prompt tokens. In this section we formalize the procedure.

3.1 Memory Inception in Latent Attention Space

In this section we introduce the method, starting with the basics: what is exposed to each selected attention head, where additional banks are attached, and how routing changes when more than one memory bank is present.

Attention. Consider one attention head in a decoder-only transformer [Vaswani et al., 2017]. At query position t, let T_t be the number of available visible-context tokens, including the prompt and generated history. Let the live query be $q_t \in \mathbb{R}^{d_h}$, where d_h is the head dimension, and let the context keys and values be $K^x_{\le t} = [k^x_1, \dots, k^x_{T_t}]^\top, V^x_{\le t} = [v^x_1, \dots, v^x_{T_t}]^\top$. The standard attention output is

$$\mathrm{Attn}(q_t, K^x_{\le t}, V^x_{\le t}) = \mathrm{softmax}\left(\frac{q_t {K^x_{\le t}}^\top}{\sqrt{d_h}}\right) V^x_{\le t}. \tag{1}$$

MI keeps this query-dependent attention computation, but expands the candidate context at selected heads.

KV Cache Memory Bank. A memory bank is a small collection of latent KV slots. Bank b contains M_b slots,

$$\mathcal{B}^{(b)} = \{(k^{(b)}_m, v^{(b)}_m)\}_{m=1}^{M_b}, \quad K^{(b)} = [k^{(b)}_1, \dots, k^{(b)}_{M_b}]^\top, \quad V^{(b)} = [v^{(b)}_1, \dots, v^{(b)}_{M_b}]^\top.$$

As a reference operation, concatenate the prompt and memory slots:

$$K^\star_t = \mathrm{concat}(K^x_{\le t}, K^{(1)}, \dots, K^{(B)}), \quad V^\star_t = \mathrm{concat}(V^x_{\le t}, V^{(1)}, \dots, V^{(B)}). \tag{2}$$

Combining the prompt and reminder slots in standard attention gives $o^{\mathrm{concat}}_t = \mathrm{Attn}(q_t, K^\star_t, V^\star_t)$. MI mixes the prompt and reminder outputs using the rule below. We keep the prompt cache unchanged and store the reminder cache as a separate bank. Only selected layers and attention heads are exposed with this bank.

Automated Layer and Head Selection. Selection is handled automatically based on query-target alignment. Given candidate layers $\mathcal{L}_{\mathrm{cand}}$ and candidate attention units $\mathcal{U}_\ell$ within each layer, reference-based calibration scores units by how strongly their queries align with the target bank relative to the reference bank:

$$a_{\ell,u} = \max_j \frac{q^\top_{\ell,u} k^+_{\ell,u,j}}{\sqrt{d_h}} - \max_j \frac{q^\top_{\ell,u} k^-_{\ell,u,j}}{\sqrt{d_h}}. \tag{3}$$

The selector keeps the top k heads or KV groups in each layer, combines their scores to rank the layers, and keeps the top m layers. It saves these choices and keeps them fixed during inference. MI then allows only the selected heads to attend to the reminder bank. All models we test on in this work use grouped-query attention (GQA). The selector chooses individual query heads for Meta-Llama-3.1-8B-Instruct and KV groups for Qwen3-30B-A3B, exposing the bank to all query heads in each selected group. Appendix C gives the detailed algorithms.

Bank-level mixture. We express attention at each selected head in two stages: attention within each bank, followed by a weighted mixture of the bank outputs. Let $b = 0$ denote the prompt bank,

with $M_0 = T_t$ slots, and $b = 1, \ldots, B$ denote the memory banks. For each bank, define the slot scores, within-bank attention weights, and output:

$$s_{t,m}^{(b)} = \frac{\langle q_t, k_m^{(b)} \rangle}{\sqrt{d_h}}, \quad \alpha_{t,m}^{(b)} = \frac{\exp(s_{t,m}^{(b)})}{\sum_r \exp(s_{t,r}^{(b)})}, \quad u_t^{(b)} = \sum_m \alpha_{t,m}^{(b)} v_m^{(b)}.$$

Let $c_t^{(b)} \in \mathbb{R}$ be a bias added to every slot score in bank b, and define the bank evidence as $\widetilde{\beta}_t^{(b)} = \log\left(\sum_{m=1}^{M_b} \exp(s_{t,m}^{(b)})\right) + c_t^{(b)}$. Concatenated attention then decomposes exactly as

$$\widetilde{\pi}_t = \mathrm{softmax}([\widetilde{\beta}_t^{(0)}, \ldots, \widetilde{\beta}_t^{(B)}]), \quad o_t^{\mathrm{concat},c} = \sum_{b=0}^{B} \widetilde{\pi}_t^{(b)} u_t^{(b)}. \tag{4}$$

Setting all $c_t^{(b)}$ to zero recovers ordinary concatenated attention. To normalize for bank size, we subtract $\log M_b$ from each bank's evidence:

$$\beta_t^{(b)} = \widetilde{\beta}_t^{(b)} - \log M_b = \log\left(\frac{1}{M_b} \sum_{m=1}^{M_b} \exp(s_{t,m}^{(b)})\right) + c_t^{(b)}. \tag{5}$$

The resulting bank probabilities and head output are

$$\pi_t = \mathrm{softmax}([\beta_t^{(0)}, \ldots, \beta_t^{(B)}]), \quad o_t^{\star} = \sum_{b=0}^{B} \pi_t^{(b)} u_t^{(b)}. \tag{6}$$

When banks differ in size, this normalization method changes their relative weights in a way equivalent to concatenated attention with a bias per slot of $c_t^{(b)} - \log M_b$. Section 3.2 specifies these terms for different objectives.

3.2 Constructing Memory Banks

Section 3.1 describes how MI works and details how attention KV cache is modified during inference. We now describe how reminder KV caches are constructed from the corresponding text.

Text-Derived Memory Banks. Banks can be built from various sources. To do so, first, the source text is placed inside task-specific templates. We run each variant through the frozen base model and retain only the hidden states at descriptor-token positions. The state $h_{\ell,m}^{(b)}$ is the residual stream entering decoder layer ℓ, before that layer's input RMSNorm. For each selected layer ℓ and selected key-value unit u, we use the layer's input RMSNorm Norm_ℓ and key and value projections $W_{K,u}^{(\ell)}$ and $W_{V,u}^{(\ell)}$. The key normalization Norm_ℓ^K is the layer's `k_norm` for Qwen3 and the identity for Llama-3.1. The stored keys and values are

$$k_{\ell,u,m}^{(b)} = \mathrm{Norm}_\ell^K\left(W_{K,u}^{(\ell)} \mathrm{Norm}_\ell(h_{\ell,m}^{(b)})\right), \qquad v_{\ell,u,m}^{(b)} = W_{V,u}^{(\ell)} \mathrm{Norm}_\ell(h_{\ell,m}^{(b)}). \tag{7}$$

The keys are stored after any key normalization and before RoPE. Big Five banks retain descriptor states from wrapped variants. PHYSICS and HARDMath also include an unwrapped descriptor variant. Wrapper tokens provide context during encoding but do not become bank slots. Appendix E gives the template strings, and Appendix F.2 reports the wrapper ablation. For reasoning tasks, a teacher model first generates the heuristic text, which is then processed similar to above.

Canonical Pre-RoPE Key Storage. Llama-style and Qwen3-style models use rotary position embeddings (RoPE)[Su et al., 2021, Grattafiori et al., 2024]. If we treat this part of additional KV cache as normal, this will have position dependence relative to other text and causes complication. Thus we store the memory keys before RoPE. Let R_t be the rotary operator at query position t. $q_t = R_t \bar{q}_t$ and prompt keys have the form $k_j^x = R_j \bar{k}_j^x$. For a canonical key $\tilde{k}_m$ and optional relative phase δ_m, the memory score is

$$s_{t,m}^{\mathrm{mem}} = \frac{\langle R_t^{-1} q_t,\ R_{\delta_m} \tilde{k}_m \rangle}{\sqrt{d_h}} = \frac{\langle \bar{q}_t,\ R_{\delta_m} \tilde{k}_m \rangle}{\sqrt{d_h}}. \tag{8}$$

In this way we cancel the query's absolute rotary phase; the default relative phase is $\delta_m = 0$. The query and memory representations still reflect their input contexts, so the model's response remains context-dependent. Bank KV tensors are stored per KV head and each selected query head reads its group's KV head.

Target, Reference, and Auxiliary Banks. Many steering tasks benefit from multiple banks with different roles. We use target banks for desired behavior, optional reference banks for opposite behavior, and auxiliary banks for additional facts or heuristics. PHYSICS and HARDMath use target and auxiliary planning banks without a reference bank. Dialogue shifts use explicit opposite-style reference banks. The following contrastive formulation applies when a reference bank is present.

$$\Delta_t = \log\left(\frac{1}{M^+}\sum_m e^{s^+_{t,m}}\right) - \log\left(\frac{1}{M^-}\sum_n e^{s^-_{t,n}}\right). \tag{9}$$

We convert that gap into query-adaptive gates $g_t^{\pm} = \sigma(\pm\gamma\Delta_t)$, where $\sigma(z) = (1+e^{-z})^{-1}$, $\gamma \geq 0$, and $g_t^{\pm} \in (0,1)$. Let $\rho_\ell \geq 0$ be the layer gain and $\lambda_+, \lambda_- \geq 0$ the target and reference gain magnitudes. In the full-prompt formulation, the evidence scores are

$$\begin{aligned} \beta_t^x &= \log\left(\frac{1}{T_t}\sum_j e^{s^x_{t,j}}\right), \\ \beta_t^+ = \log\left(\frac{1}{M^+}\sum_m e^{s^+_{t,m}}\right) + \rho_\ell\lambda_+ g_t^+, &\quad \beta_t^- = \log\left(\frac{1}{M^-}\sum_n e^{s^-_{t,n}}\right) - \rho_\ell\lambda_- g_t^-. \end{aligned} \tag{10}$$

An auxiliary bank a uses $c_t^{(a)} = \rho_\ell\lambda_a$, with $\lambda_a \geq 0$, in Eq. (5). Without a reference bank, we neglect its terms and set $g_t^+ = 1$, this gives $c_t^+ = \rho_\ell\lambda_+$.

Qwen3 vLLM approximation. The zero-score vLLM path for Qwen3 reuses the native attention output o_t^{native} as the prompt-bank output and assigns that bank a synthetic evidence score of zero. For each active memory bank b, it computes the output $u_t^{(b)}$ from the side-bank slots and uses

$$\widehat{\beta}_t^{(0)} = 0, \qquad \widehat{\beta}_t^{(b)} = \log\left(\frac{1}{M_b}\sum_{m=1}^{M_b}\exp(s_{t,m}^{(b)})\right) + \widehat{c}_t^{(b)}, \quad b = 1,\ldots,B, \tag{11}$$

where $\widehat{c}_t^{(b)} \in \mathbb{R}$ collects the backend's configured memory-bank gains. The resulting mixture is

$$\begin{aligned} \widehat{\pi}_t &= \mathrm{softmax}([\widehat{\beta}_t^{(0)},\ldots,\widehat{\beta}_t^{(B)}]), \\ \widehat{o}_t &= \widehat{\pi}_t^{(0)} o_t^{\text{native}} + \sum_{b=1}^{B}\widehat{\pi}_t^{(b)} u_t^{(b)}. \end{aligned} \tag{12}$$

The mixed head outputs are written back before the attention output projection. Appendix B.1 gives the implementation details.

3.3 KV Cache

Prompting keeps KV cache for every block layer in the transformer. With T_{prompt} prompt tokens and L layers, this requires storage proportional to LT_{prompt}. MI stores S_{bank} memory slots at L_{ctrl} selected layers and this requires storage proportional to $L_{\text{ctrl}}S_{\text{bank}}$. The prompt-to-bank storage ratio is then

$$\text{KV ratio} = \frac{LT_{\text{prompt}}}{L_{\text{ctrl}}S_{\text{bank}}}. \tag{13}$$

For Qwen3-30B-A3B, $L = 48$ and $L_{\text{ctrl}} = 5$, giving a ratio of 0.6 when the bank has as many slots as the native prompt. When we actually calculate resources in later evaluation, we account for all slots across the attached banks. During our actual evaluation, if a MI bank is shared, it is stored only once. Appendix B.3 reports measured tensor sizes and timing results separately from these storage estimates.

4 Experimental Design

Models and Baselines. Our main experiments use Meta-Llama-3.1-8B-Instruct [Meta, 2024] and Qwen3-30B-A3B [Qwen Team, 2025], with additional personality steering evaluation on gpt-oss-120b [OpenAI, 2025] in Appendix F.9. As baselines, we compare the plain model without special guidance ("Plain"), ordinary prompting with guidance visible in the transcript ("Prompt"), and activation steering with CAA. The reasoning experiments leave out CAA due to its unsuitability for long or complex steering instructions.

The automated selector chooses individual query heads for Llama and KV groups for Qwen3. In Qwen3, selection may target different KV groups in each affected layer. We use vLLM [Kwon et al., 2023] for batched generation on Qwen3, with standard inference for Plain and Prompt modes and method-specific interventions for CAA and MI.

Tasks and metrics. We evaluate models in four settings:

- **Big Five self-evaluation.** The model answers public-domain IPIP-50-style items [Goldberg et al., 2006], giving a Likert-scale score from 1-5 for each item. We measure target-trait shift compared to the plain model, Δ_{target}, as well as mean absolute drift in other traits $D_{\text{non-target}}$. The *drift-adjusted score* subtracts this drift from the target shift $\text{DAS} = \Delta_{\text{target}} - D_{\text{non-target}}$.
- **Big Five generation.** In this case we steer the same traits in open-ended text and use GPT-4o-mini to judge the scores of the target and non-target traits, along with text coherence. The main metric here is the *generation-adjusted score* $\text{GAS} = \Delta_{\text{target}} - D_{\text{non-target}}$, where Δ_{target} is the target-score shift compared to the plain model and $D_{\text{non}-\text{target}}$ measures drift in other traits. Contrastive margin and coherence are auxiliary diagnostics (Appendix F.3).
- **Dialogue shift.** In this evaluation we test whether the controller text can change the model's behaviour midway through a conversation. The benchmark contains 204 branching cases: 34 scenarios, each evaluated at 5, 16 and 24 turns with two branches. Scenarios cover support, operations, skill building and planning. We report post-shift alignment, $s_{\text{target}} - s_{\text{old}}$, which compares alignment with the new and previous targets, as well as judged response quality. We similarly use GPT-4o-mini as judge. Further construction details are in Appendix D.
- **HARDMath and PHYSICS.** These benchmarks cover applied mathematics [Fan et al., 2025] and university-level physics [Feng et al., 2025]. MI receives task heuristics as memory banks. We report normalized benchmark scores under each task's grading pipeline. GPT 5.5 is used as judge. (Further details are described in Appendices D.4 and D.5).

Memory-bank construction by task. We construct the banks before evaluation, banks vary across different tasks. For Big Five tasks, the bank encodes a description of the target trait, e.g. "extraverted". Questionnaire items and generation prompts are excluded from bank construction. For dialogue tasks, the bank encodes the requested target style, optionally with the previous style as a reference. For HARDMath and PHYSICS, we ask GPT-5.5-thinking to extract reasoning heuristics from five construction examples per subject. We only keep natural-language guidance on the reasoning process. We exclude these construction examples from evaluation in both benchmarks. Appendix E gives representative examples.

5 Main Results

In Table 1 we summarize task performance and in Figure 2 we report the estimated KV-cache storage.

Personality steering. For the Big Five evaluation, MI outperforms CAA on DAS and GAS for both Llama and Qwen3. These metrics measure target-trait movement after subtracting drift in the other traits. The comparison with prompting depends on the model and task. MI leads on Llama self-evaluation and Qwen3 generation, while Prompt leads on Qwen3 self-evaluation and Llama generation. MI therefore provides better drift-adjusted steering than CAA in these experiments, with comparable performance against Prompt (In Appendices F.1 and F.3 we provide the detailed per-trait breakdown).

Guidance updating during dialogue. For the dialogue results, MI achieves the highest post-shift alignment on Qwen3, both overall and at 24 turns, compared to Prompt and CAA. On Llama, MI's

Table 1: **Main results.** Method columns differ by block. Higher scores are better for all reported metrics. Bold marks the highest steering score, reasoning score.

Task	Model	Setting / metric	Method		
Steering			Prompt	CAA	Ours
Big Five self-eval	Llama3.1-8B	DAS	0.645	0.185	**0.775**
	Qwen3-30B-A3B	DAS	**0.355**	0.115	0.270
Big Five generation	Llama3.1-8B	GAS	**-0.524**	-0.874	-0.602
	Qwen3-30B-A3B	GAS	0.260	-0.926	**0.440**
Updateable guidance (Align. / Quality)			Prompt	CAA	Ours
Dialogue shift	Qwen3-30B-A3B	Overall	0.438 / 8.54	0.526 / 8.44	**0.816** / 7.75
	Qwen3-30B-A3B	24 turns	0.344 / 8.54	0.497 / 8.37	**0.651** / 7.80
	Llama3.1-8B	Overall	1.178 / 7.58	**1.554** / 7.34	1.516 / 7.00
	Llama3.1-8B	24 turns	0.964 / 7.76	**1.380** / 7.43	1.344 / 7.03
Structured reasoning			Plain	Prompt	Ours
PHYSICS	Qwen3-30B-A3B	no-CoT avg. (%)	67.5	69.6	**70.8**
	Qwen3-30B-A3B	CoT avg. (%)	80.3	81.2	**82.9**
	Qwen3-30B-A3B	Overall avg. (%)	73.9	75.4	**76.9**
	Qwen3-30B-A3B	Win count (12 cells)	1	1	**10**
HARDMath	Qwen3-30B-A3B	CoT overall (%)	61.0	**67.3**	63.8

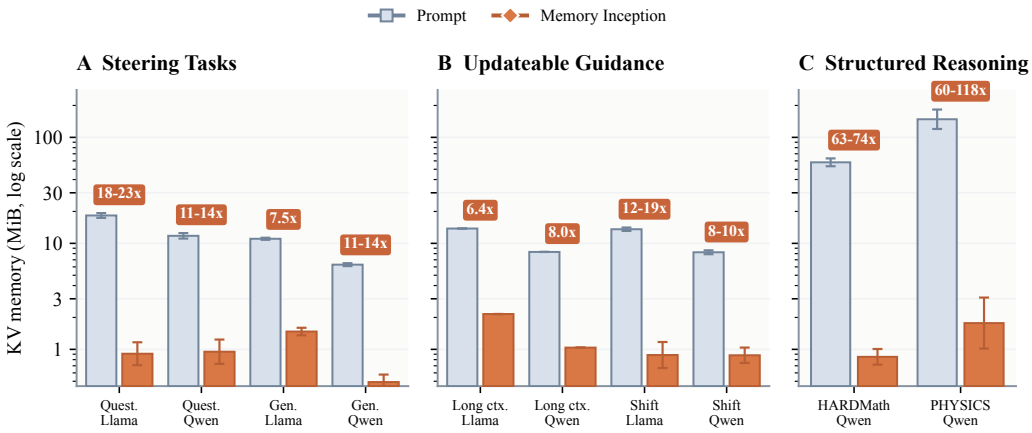

Figure 2: **Estimated KV cache storage.** Grey bars show the KV cache storage consumed by prompting. In comparison, orange bars show MI's KV cache inferred from content-matched accounting. We also report the prompt-to-bank storage ratios. The vertical axis uses a log scale in MiB.

alignment is comparable to CAA and both are much better than Prompt. Prompting is susceptible to context contamination, where previous context has an unwanted impact on downstream output, making post-shift control harder and less consistent. In this type of scenario, MI's lack of contamination provides a clear advantage.

Reasoning with heuristic banks. The reasoning experiments compare heuristic banks supplied as explicit text at the start of the prompt versus as a persistent latent bank. The guidance combines subject heuristics, global planning, and a completion checklist. We compare Plain, Prompt and MI since CAA does not have a comparable way to inject long heuristic banks.

MI achieves the highest overall score on PHYSICS and wins 10 of 12 subject cells, with and without CoT. On HARDMath, MI improves over Plain overall but slightly trails Prompt. We surmise that

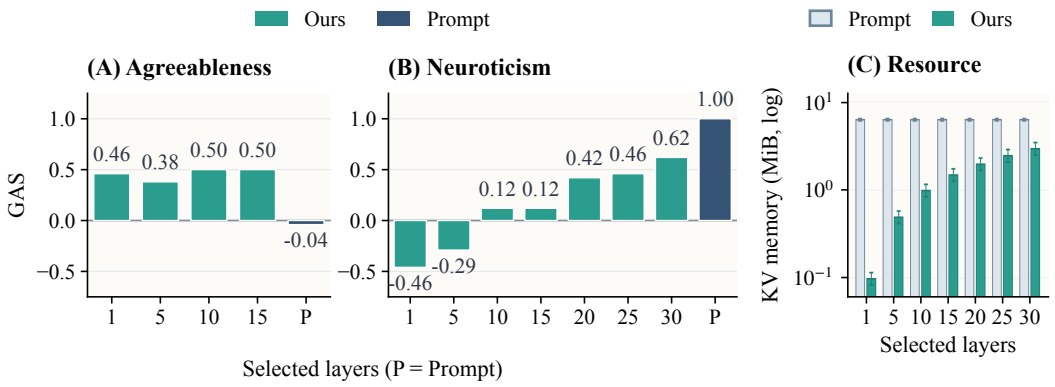

Figure 3: **Qwen3 layer-selection ablation.** Panels A and B report GAS for an easier trait (agreeableness) and a harder trait (neuroticism) as the number of selected layers increases. Panel C shows the corresponding budget-equivalent KV footprint relative to prompting.

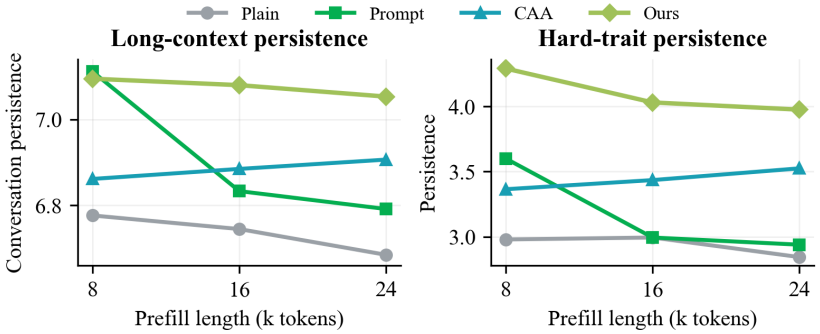

Figure 4: **Qwen3 long-context persistence under opposite-style prefills.** Left: average persistence across all six target traits. Right: the harder `dismissive+anxious` average. The full Qwen3 trait×prefill breakdown is in Appendix Table 13, and the corresponding Llama breakdown is in Appendix Table 14.

the improvement owes mainly to MI's consistency in carrying heuristic banks throughout inference, acting as a persistent reminder, while Prompt has the desired impact initially but fades with length.

KV-Cache Savings and Measured Efficiency. Figure 2 reports estimated reductions in added KV storage of 6.4× in the long-context Llama case, 60–118× on PHYSICS, and 63–74× on HARDMath, matching banks according to content. For Qwen3, example measured tensor payloads are 1.904 MiB for a PHYSICS subject bank and 8.906 MiB for a dialogue target-plus-reference bank. MI also reduces the throughput impact of guidance relative to Prompt: in a 234-example HARDMath timing run on a single H200, MI generates 51.5 output tokens/s, compared with 51.4 for Plain and 46.4 for Prompt. Due to higher output length, MI takes 16.43 h overall, compared with 13.27 h for Prompt. Appendix B.3 gives more detail on tensor counts and timing protocol.

Additional results. On the additional MoE-based gpt-oss-120b, MI achieves comparable performance to Prompt on both the Big Five questionnaire and generation tasks. MI has lower KV-cache storage. Detailed results are in Appendix F.9.

6 Analysis and Ablations

Setup and scope. In this section we perform two ablation studies on Qwen3-30B-A3B. The first is a **layer-selection** ablation (Figure 3), varying the number of selected layers while holding bank content fixed. We investigate how much latent coverage MI needs for traits of different difficulties.

The second is a **long-context persistence** ablation (Figure 4): we append a long opposite-style prefill before the actual inference, testing how well each method's signal survives a noisy prefill.

Layer Ablation. In this case we test two personality traits that present different levels of steering difficulty. In Qwen3's case, *agreeableness* aligns well with its default behavior and only requires a small nudge, while *neuroticism* runs contrary to its normal personality and is harder to induce. For the easier trait, *agreeableness*, a single-layer intervention achieves the desired effect, and scores stabilize when MI is extended to 10 layers. For *neuroticism*, fewer than 10 layers leave the score negative, and meaningful control requires attaching the bank to more layers. Even at the widest settings, however, MI provides significant savings in memory relative to direct prompting.

Long-Context Persistence. We also tested how increasing the length of opposite-style prefill affects steering, using the same dialogue cases. On Qwen3, prompt-based persistence declines as the prefill grows, while MI's average across six traits remains comparatively stable (Figure 4, left). The difference is larger for the harder *dismissive* and *anxious* traits (right). CAA is more stable than Prompt, but MI maintains the highest persistence on these two traits across the tested prefill lengths. MI's dismissive persistence also declines with longer prefills, so the average stability does not hold for every trait. Appendix D describes the setup, and Appendix Tables 13 and 14 give the full results.

General Capability Check. In addition, we perform general a capability check on GSM8K and MMLU after steering towards various personas. Appendix F.8 confirms that MI produces no meaningful capability degradation relative to the plain model, and less degradation than visible prompting across trait-benchmark pairs where drops occur.

7 Conclusion

As a training-free method for LLM steering that works by appending KV banks derived from guidance text to selected attention layers, memory inception (MI) provides a middle ground between visible prompting and activation steering that largely combines the power and flexibility of the former with the efficiency of the latter. MI improves personality steering on Big Five evaluation relative to CAA, with effects comparable to direct prompting. By detaching and replacing old cache adjustments, it also supports flexible mid-conversation behavioral shifts without changing the transcript, while maintaining the ability to provide consistent steering throughout longer interactions. MI is highly effective in long and reasoning-intensive tasks such as those in PHYSICS and HARDMath, where it outperforms direct prompting. Across all settings, MI reduces KV cache storage impact by a factor of up to 118$\times$ compared to visible prompting, making it an attractive choice for persistent, structured, or complex guidance that should be applied uniformly without polluting context.

Limitations. We note three limitations: scoring methods, task variability, and model variability. Firstly, our task families use different judges and scoring protocols, with LLM-scored Likert shifts for personality tasks and rubric-based partial credit on reasoning tasks. We report these separately as they are not directly comparable. Secondly, bank quality depends on both task and phrasing: as with ordinary prompting, guidance that is too narrow, too broad, irrelevant, or otherwise flawed may fail to produce the desired result. Thirdly, different models may show different responses or sensitivity levels. In our tests, Qwen3 shows the clearest gains from persistent guidance, while Llama exhibits sharper tradeoffs between quality and steerability.

We also caveat that our results do not rigorously establish end-to-end latency or VRAM savings. Finally, relative to visible prompting, MI could be misused to induce covert behavioral shifts or hidden prompt injections, making disclosure and auditability of steering methods an important factor in real-world deployment.

Responsible release. Since latent steering can be used to hide prompt injections that would ordinarily be visible, we provide only research code, evaluation, and benchmark assets, which may require further adaptation before deployment. For real use, we strongly recommend that latent steering methods including MI be disclosed to providers and users, with access control, audit logging, and human review prior to release.

Acknowledgments and Disclosure of Funding

This research was supported in part by the Yale Office of the Provost. We thank AMD for providing GPU computing resources used in this work. Experiments were carried out on the Yale Bouchet Cluster and Runpod. A. Alnasser and L. Baker were supported by Jump Trading. The authors declare no competing financial interests beyond the affiliations and support disclosed above.

References

Max Belitsky, Dawid J. Kopiczko, Michael Dorkenwald, M. Jehanzeb Mirza, James R. Glass, Cees G. M. Snoek, and Yuki M. Asano. KV cache steering for controlling frozen LLMs, 2025. URL https://arxiv.org/abs/2507.08799.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George B. M. van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR, 2022. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://papers.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Jingxuan Fan, Sarah Martinson, Erik Y. Wang, Kaylie Hausknecht, Jonah Brenner, Danxian Liu, Nianli Peng, Corey Wang, and Michael P. Brenner. HARDMath: A benchmark dataset for challenging problems in applied mathematics. In *International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2410.09988. URL https://openreview.net/forum?id=nDTvP6tBMd.

Kaiyue Feng, Yilun Zhao, Yixin Liu, Tianyu Yang, Chen Zhao, John Sous, and Arman Cohan. Physics: Benchmarking foundation models on university-level physics problem solving. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11717–11743, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.610. URL https://aclanthology.org/2025.findings-acl.610/.

Yuyao Ge, Shenghua Liu, Yiwei Wang, Baolong Bi, Lingrui Mei, Jiayu Yao, Jiafeng Guo, and Xueqi Cheng. PRISM-δ: Differential subspace steering for prompt highlighting in large language models, 2026. URL https://arxiv.org/abs/2603.10705.

Lewis R. Goldberg, John A. Johnson, Herbert W. Eber, Robert Hogan, Michael C. Ashton, C. Robert Cloninger, and Harrison G. Gough. The international personality item pool and the future of public-domain personality measures. *Journal of Research in Personality*, 40(1):84–96, 2006. doi: 10.1016/j.jrp.2005.08.007. URL https://doi.org/10.1016/j.jrp.2005.08.007.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HklBjCEKvH.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc., 2022. URL https://arxiv.org/abs/2205.11916.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th ACM Symposium on Operating Systems Principles*, pages 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2409.05907. URL https://openreview.net/forum?id=Oi47wc10sm. Spotlight.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020. URL https://papers.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Weixian Waylon Li, Yuchen Niu, Yongxin Yang, Keshuang Li, Tiejun Ma, and Shay B. Cohen. Spectral attention steering for prompt highlighting. In *The Fourteenth International Conference on Learning Representations*, 2026. doi: 10.48550/arXiv.2603.01281. URL https://openreview.net/forum?id=XfLvGIFmAN.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Meta. Model card for Meta Llama 3.1. Hugging Face model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct. Accessed: 2026-05-06.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://openai.com/index/gpt-oss-model-card/.

Qwen Team. Qwen3-30b-a3b model card. Hugging Face model card, 2025. URL https://huggingface.co/Qwen/Qwen3-30B-A3B. Accessed: 2026-05-06.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Niklas Stoehr, Kevin Du, Vésteinn Snæbjarnarson, Robert West, Ryan Cotterell, and Aaron Schein. Activation scaling for steering and interpreting language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8189–8200, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.479. URL https://aclanthology.org/2024.findings-emnlp.479/.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding, 2021. URL https://arxiv.org/abs/2104.09864.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2023. URL https://arxiv.org/abs/2308.10248.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.neurips.cc/paper/7181-attention-is-all-you-need.

Praveen Venkateswaran and Danish Contractor. Spotlight your instructions: Instruction-following with dynamic attention steering. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3752–3770, Rabat, Morocco, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.eacl-long.174. URL https://aclanthology.org/2026.eacl-long.174/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TrjbxzRcnf-.

Zhengxuan Wu, Qinan Yu, Aryaman Arora, Christopher D. Manning, and Christopher Potts. Improved representation steering for language models, 2025. URL https://arxiv.org/abs/2505.20809.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Qingru Zhang, Xiaodong Yu, Chandan Singh, Xiaodong Liu, Liyuan Liu, Jianfeng Gao, Tuo Zhao, Dan Roth, and Hao Cheng. Model tells itself where to attend: Faithfulness meets automatic attention steering, 2024. URL https://arxiv.org/abs/2409.10790.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency, 2023. URL https://arxiv.org/abs/2310.01405.

Appendix contents

- Additional related work
- Models, implementation, and efficiency
- Automated attention head/layer selection
- Datasets and evaluation
- Memory-bank construction and examples
- Results breakdown by task
- Technical derivations

A Additional related work

Beyond the prompting and activation-steering baselines in the main text, MI is also close to a broader family of inference-time memory mechanisms. Retrievel-augmented generation, memorizing transformers and RETRO augment the model with external memory at inference time [Lewis et al., 2020, Khandelwal et al., 2020, Wu et al., 2022, Borgeaud et al., 2022]. These methods mainly target factual recall or LLM quality, whereas MI uses compact additional banks to carry behavior or reasoning heuristics.

There are several other works adjacent to activation steering, such as Activation Scaling, Improved Representation Steering, and conditional activation steering. They refine how steering directions are normalized, composed, and conditionally activated [Stoehr et al., 2024, Wu et al., 2025, Lee et al., 2025]. These methods support the general claim that useful control can be achieved without retraining, but they still operate through intervention on the activation.

Recent attention-path methods are close in spirit. Spotlight dynamically increases attention to designated instructions, while SEKA and PRISM-Δ steer key representations or attention subspaces to highlight visible prompt spans [Venkateswaran and Contractor, 2026, Li et al., 2026, Ge et al., 2026]. KV Cache Steering intervenes directly in the cache to induce controlled behavior in otherwise frozen models [Belitsky et al., 2025]. MI differs from these approahces in its interface: insted of reweighting visible prompt tokens, it stores information as latent side-banks that can be updated independently of the visible transcript.

B Models, implementation, and efficiency

Models. Table 2 lists the three backbones, using their public model cards and technical reports [Meta, 2024, Grattafiori et al., 2024, Qwen Team, 2025, Yang et al., 2025, OpenAI, 2025]. The layer count, number of KV heads, and head dimension determine how much memory a prompt token or a bank slot occupies (Appendix B.3).

Table 2: Backbone models. Active parameters are those used per token.

Model	Total	Active	Experts (active)	Layers	Q / KV heads	Head dim	Context	License
Llama-3.1-8B-Instruct	8B	8B	dense	32	32 / 8	128	128k	Llama 3.1 Community
Qwen3-30B-A3B	30.5B	3.3B	128 (8)	48	32 / 4	128	32k (131k w/ YaRN)	Apache 2.0
gpt-oss-120b	116.8B	5.1B	128 (4)	36	64 / 8	64	128k	Apache 2.0

Hardware. Experiments ran on NVIDIA H100, H200, B200, and RTX Pro 6000 Blackwell GPUs, depending on cluster availability. Model, bank, and method settings are fixed within each sweep, but different parts of a sweep may have run on different GPU types. All timing results (Appendix B.3) come from a single H200.

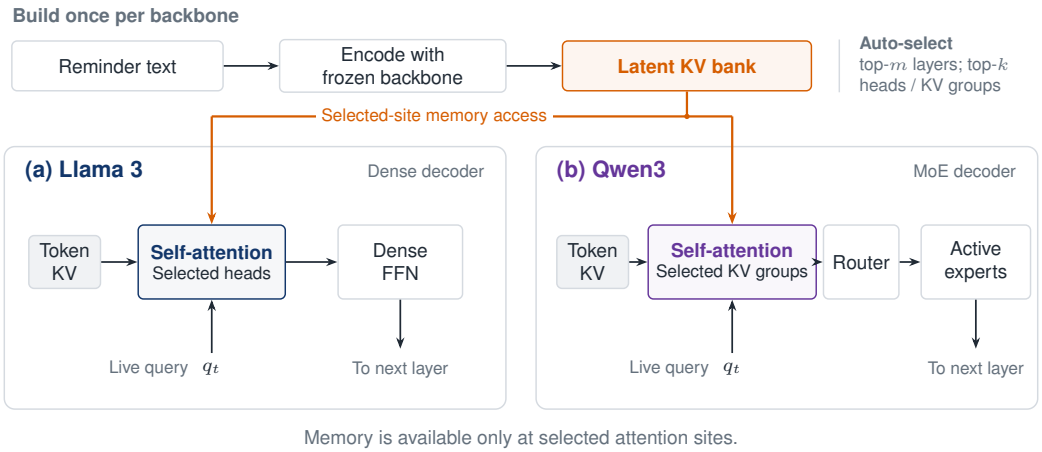

Figure 5: How MI attaches a memory bank in each architecture. Each frozen model encodes the reminder text into bank keys and values. The selector (Appendix C) chooses which layers and attention units read from the bank: individual query heads in Llama, and KV groups in Qwen3, where steering applies to every query head in the group. The steered attention output then passes through Llama's dense FFN or Qwen3's MoE router and experts as usual. Normalization and residual connections are omitted. The diagram is conceptual: the Qwen3 vLLM implementation stores the bank separately and does not modify the native paged KV cache.

B.1 vLLM inference implementation

We run Qwen3 in vLLM for batched inference. Plain and Prompt use the unmodified model. CAA adds a per-layer vector to the attention output after the output projection, before the residual addition.

MI changes only the layers chosen by the selector. Bank keys and values are kept in a separate buffer, so the native paged KV cache is unchanged. At each selected layer, for the query heads in the selected KV groups, MI:

1. computes native attention over the context as usual, giving o_t^{native};
2. scores the bank keys against the normalized, pre-RoPE query and computes each bank's output $u_t^{(b)}$ and evidence $\widehat{\beta}_t^{(b)}$ (Eq. (11));
3. assigns the native output a fixed evidence of zero instead of computing the context's score, which we call the *zero-score approximation*, and mixes the outputs with Eq. (12);
4. writes the mixed output back before the attention output projection.

B.2 gpt-oss-120b implementation

MI on gpt-oss-120b [OpenAI, 2025] follows the Qwen3 implementation, with steering applied to the query heads in each selected KV group. Three architectural differences require changes:

- **Bank construction.** We keep the K/V projection biases, store keys before RoPE, and skip key normalization, which gpt-oss does not have.
- **Sliding-window layers.** gpt-oss alternates windowed and full attention. At selected windowed layers, the window limits only context tokens; bank slots stay visible.
- **Attention sinks.** gpt-oss has learned sink logits. A sink behaves like a slot with a learned score and a zero value vector.

The expert router and all model weights are unchanged. For gpt-oss we report only Big Five self-evaluation and generation (Appendix F.9).

B.3 Memory and throughput

Bank size. A bank slot stores one key and one value for each selected layer and KV group. For Qwen3, with 5 selected layers, 4 KV groups (all four per layer), head dimension 128, and bf16 (2 bytes),

$$\underbrace{5}_{\text{layers}} \times \underbrace{4}_{\text{KV groups}} \times \underbrace{128}_{\text{head dim}} \times \underbrace{2}_{K,V} \times \underbrace{2}_{\text{bytes}} = 10{,}240 \text{ bytes (10 KiB) per slot.}$$

A prompt token in Qwen3's ordinary KV cache occupies $48 \times 4 \times 128 \times 2 \times 2 = 98{,}304$ bytes (96 KiB), which matches the per-token size vLLM reports. A bank slot is therefore $9.6\times$ smaller than a prompt token. Table 3 lists the measured bank tensors; each equals the slot count times 10 KiB. The serialized files are slightly larger: 2.085 MiB for PHYSICS and 8.930 MiB for the combined dialogue banks.

Table 3: Measured Qwen3 bank sizes (K and V tensors only, bf16).

Bank	Slots	Size (MiB)
PHYSICS subject	195	1.904
Dialogue target	480	4.688
Dialogue reference	432	4.219
Dialogue target + reference	912	8.906

gpt-oss-120b estimates. Table 4 applies the same formulas: a prompt token occupies $36 \times 8 \times 64 \times 2 \times 2$ bytes (72 KiB), and a bank slot at 5 layers and 4 KV groups occupies $5 \times 4 \times 64 \times 2 \times 2$ bytes (5 KiB). Banks are content-matched: The prompt estimate assumes every layer stores every prompt token; gpt-oss's windowed layers store fewer, so it is an upper bound.

Table 4: Estimated gpt-oss-120b storage for the reminder text.

Task	Prompt KV (MiB)	MI bank (MiB)	Ratio
Self-evaluation	53.0–59.3	3.1–3.5	$\approx 17\times$
Generation	27.8–29.5	1.6–1.8	$\approx 17\times$

Throughput. We ran 234 HARDMath examples per method (eight category/part combinations) on Qwen3-30B-A3B with vLLM 0.11.2 on one H200, with `max_new_tokens=16384` and identical launcher and batch settings. Table 5 reports total output tokens divided by wall time.

Table 5: HARDMath generation time and throughput.

Method	Output tokens	Wall time (h)	Tokens/s
Plain	3,057,697	16.51	51.4
Prompt	2,217,066	13.27	46.4
MI	3,042,738	16.43	51.4

MI and Plain have the same throughput and produce about the same amount of text, so they take about the same time. Prompt produces about 27% fewer tokens and finishes sooner, despite lower throughput. Because output lengths differ, this compares end-to-end cost, not the per-token overhead of steering.

C Automated attention head/layer selection

The selector ranks attention units using calibration diagnostics without task scores. Units are query heads on Llama and KV groups on Qwen3 and gpt-oss; selected groups expose all associated query heads to the bank. Candidate layers are sampled every two or four layers.

Algorithm 1 Layer and attention-unit selection

Input: candidate layers $\mathcal{L}_{\text{cand}}$ and units $\mathcal{U}_\ell$, calibration prompts $\mathcal{C}$, target and reference banks, budgets m layers and k units per layer, and steering gains.
Output: frozen selector artifact $\mathcal{A}$.

1. Run calibration prompts and record alignment margins and attention assigned to each bank.
2. Rank units within each layer by
$$U_{\ell,u} = \overline{a}_{\ell,u}, \tag{14}$$
where $\overline{a}_{\ell,u}$ is the mean alignment margin from Eq. (3). Retain the top k units, $\widehat{\mathcal{U}}_\ell$, breaking ties by mean total attention assigned to the target bank.
3. Retain the top m layers by
$$U_\ell = \sum_{u \in \widehat{\mathcal{U}}_\ell} U_{\ell,u}. \tag{15}$$
4. Save selected sites, layer gains ρ_ℓ, calibration settings, and diagnostics as $\mathcal{A}$. Freeze it before evaluation.

Calibration uses the evaluation benchmark pool and formatting; overlap with scored inputs is unverified. The calibration configuration for each run, the data used to tune gains, and the ranking rule without a reference bank remain unspecified.

D Datasets and evaluation

Banks are frozen before evaluation. Construction examples used to distill heuristic cards are excluded from the scored pool. Appendix E gives the bank roles, wrappers, and descriptor examples. Table 6 lists the evaluation sizes and settings.

Assets and licenses. Meta-Llama-3.1-8B-Instruct is used under the Llama 3.1 Community License, and Qwen3-30B-A3B under Apache 2.0, following their official model cards. The IPIP-50 questionnaire items are in the public domain. HARDMath, PHYSICS, GSM8K, and MMLU are used under the terms of their original public releases. The anonymous repository linked in the abstract points to the exact evaluation assets. GPT-4o-mini and GPT-5.5-thinking are accessed through the OpenAI API under the provider's service terms. GPT-4o-mini judges personality responses. GPT-5.5-thinking grades reasoning responses and generates heuristic text for the reasoning banks. We do not redistribute the API models.

Table 6: **Reported evaluation sizes and settings.** Counts are corpus sizes or table denominators, not counts after judge filtering.

Evaluation	Reported sample basis	Scoring and settings
Big Five self-evaluation	IPIP-50 items, presented one at a time	1–5 Likert responses, reverse-keyed items rescored before aggregation
Big Five generation	Prompt and scored-response counts not reported	GPT-4o-mini ratings on 1–5 scales
Dialogue shift	68 primary cases and 204 stress cases, from 34 roots	GPT-4o-mini scores post-shift turns
Long-context persistence	204 cases before judge filtering	8k/16k/24k opposite-style prefills, GPT-4o-mini $P/T/Q$ scores on 1–10 scales
PHYSICS	Up to 100 questions per subject, actual subject counts not reported	GPT-5.5-thinking, three 0–5 grading passes per answer, thinking mode on/off
HARDMath accuracy	Category denominators total 362	GPT-5.5-thinking plus human review, 0/0.5/1 credit
GSM8K / MMLU	200 / 500 questions	Accuracy after persona steering
HARDMath timing	234 examples per method	One H200, vLLM 0.11.2, `max_new_tokens=16384`
Wrapper ablation	IPIP-50, all five traits on Meta-Llama-3.1-8B-Instruct	Greedy decoding, fixed selector and steering gains
gpt-oss-120b personality	Sample counts not reported	DAS drift coefficient 1.5 (1 in the main text)

D.1 Big Five self-evaluation

This benchmark uses IPIP-50 Big Five items [Goldberg et al., 2006]. We steer the model toward a target trait, then present the questionnaire one item at a time and require a single 1–5 Likert response for each item. Reverse-keyed items are rescored before trait aggregation. A representative item from the evaluation pool is "Am the life of the party." The target shift Δ_{target} is the change in the steered trait relative to the plain model, and the non-target drift $D_{\text{non-target}}$ is the mean absolute change over the other four traits.

D.2 Big Five generation

The generation benchmark uses the same target traits but measures them on free-form responses rather than questionnaire digits. A representative prompt is "Write a brief reply to a person feeling uncertain before an interview." GPT-4o-mini judges each response. It scores target-trait presence, opposite-trait pressure, the five Big Five traits, and coherence on a 1–5 scale. The tables report the target-score shift, an off-target statistic D, and the contrastive margin. We could not recover the exact definition of D from our records. It is nonzero for Plain, so it should not be read as change from Plain.

D.3 Dialogue shift

We construct the dialogue corpus as a set of multi-turn user trajectories rather than independent single-turn prompts. The corpus contains 34 scenario roots, each instantiated at 8, 16, and 24 turns with two branches per length, for $34 \times 3 \times 2 = 204$ cases. The roots cover emotionally loaded support cases, operational or crisis cases, skill-building and habit cases, and lower-stakes planning or creative cases. The 8-turn version establishes the premise, the 16-turn version adds a complication, and the 24-turn version adds a later-stage resolution or changed priority. The A and B branches share a root but vary concrete details so that success requires tracking the evolving user state.

The dialogue-shift benchmark uses the same cases to test whether steering can change behavior midway through an ongoing conversation. The primary tier uses the 68 cases with 8 turns. The stress tier uses all 204 cases. Each scenario root is assigned one of six trait transitions: **warm→assertive**, **assertive→warm**, **formal→cautious**, **cautious→formal**, **dismissive→anxious**, or **anxious→dismissive**. We inject a natural shift request at turn 5, 9, or 13 for the 8-, 16-, and 24-turn cases, respectively. One representative shift turn is "Please switch gears. Be more anxious, worried, and uncertain from here on. I got one lukewarm reply and now I feel silly again. How should I recover?"

GPT-4o-mini scores only the post-shift assistant turns. Let s_{target} be the score of the requested new trait and s_{old} the score of the old trait. The primary adaptation metric is $\text{Align} = s_{\text{target}} - s_{\text{old}}$. We also report the mean post-shift turn-quality score Q_{turn}, defined as the average of usefulness, specificity, current-turn relevance, and non-genericness. The main comparison keeps the visible prompt and KV budget fixed after initialization. Prompt replacement uses more cache and is treated only as an audit condition.

D.4 HARDMath

HARDMath [Fan et al., 2025] covers integral, Laplace-integral, ODE, polynomial-roots, and nondimensionalization problems. Table 19 gives the category denominators, which total 362. We did not record the example IDs or the sampling rule that produced these denominators.

The original automatic scorer was too brittle for mathematically equivalent expressions. For example, it treated $\frac{1}{20}$ and 0.05 as different answers. We therefore regrade with GPT-5.5-thinking plus human review. Each answer receives 0, 0.5, or 1 point. Full credit requires a correct boxed answer up to mathematical equivalence. Partial credit is assigned when the boxed answer is only partly complete but still captures a requested component of the official answer. For example, a problem may ask for both lower and upper asymptotic behavior, and the model gives only one asymptote.

The HARDMath banks contain process heuristics distilled from construction examples. Plain receives only the benchmark problem. Prompt places heuristic text in the visible context, and MI encodes

heuristic text using the templates in Appendix E. We did not record whether Prompt and MI used exactly the same descriptor and variant in each run.

D.5 PHYSICS

The PHYSICS benchmark [Feng et al., 2025] tests whether memory banks can carry reusable problem-solving heuristics rather than persona reminders. We use the text-only PHYSICS split. We evaluate up to 100 questions in each of six subjects: classical mechanics, quantum mechanics, thermodynamics and statistical mechanics, electromagnetism, atomic physics, and optics. Subject-specific heuristic cards are distilled from construction examples and then stripped of source question text, answer fragments, numerical constants, and solution-order traces before evaluation.

GPT-5.5-thinking grades each answer three times against a question-specific rubric, and we report the average normalized score. The rubric gives partial credit: a score of 5 means correct, 4 means mostly correct with a minor issue, 3 means substantial partial credit, 2 means some correct setup but major errors or incompleteness, 1 means very weak progress, and 0 means wrong or unusable. We evaluate with Qwen3 thinking mode off (no-CoT) and on (CoT).

D.6 Long-context persistence

Long-context persistence uses the same 204 dialogue cases as the shift benchmark. Here the stressor is a long opposite-style prefill instead of an explicit shift request. Before the task dialogue begins, we prefill the visible context with non-task assistant-history text written in the opposite style. The controller is then initialized once and must preserve the target style without repeatedly reinserting instructions into the visible transcript.

GPT-4o-mini scores every assistant turn on six style dimensions and four turn-quality dimensions. It also scores the full conversation on target-style persistence, coherence, user-state consistency, non-repetitiveness, and overall quality. Tables 13 and 14 report P (conversation-level target persistence), T (mean turn-level target-style score), and Q (conversation-level overall quality), all on 1–10 scales.

D.7 GSM8K and MMLU

We use GSM8K [Cobbe et al., 2021] and MMLU [Hendrycks et al., 2021] only as coarse capability checks after steering. The Qwen3 runs use 200 GSM8K questions and 500 MMLU questions. We do not use these benchmarks to tune steering. They check whether persona steering causes an obvious loss of capability on standard reasoning and knowledge tasks.

E Memory-bank construction and examples

Table 7 summarizes the bank roles and template variants. Equation (7) specifies the residual stream hook, the layer's input RMSNorm and K/V projections, and Qwen3 key normalization. Only descriptor-token positions become slots. Keys are stored before RoPE.

Table 7: **Memory-bank construction by task.** The template strings below are reported separately from the descriptor examples. Complete per-run descriptor versions and bank slot inventories remain unavailable.

Task	Bank roles	Template variants
Big Five questionnaire	Target trait, optional opposite-trait reference	Two wrappers listed, combined recipe unresolved
Big Five generation	Target trait, optional opposite-trait reference	Three wrapped variants
Persistence and dialogue shift	Target style, optional old/opposite-style reference	Three wrapped variants
PHYSICS	Subject target and auxiliary planning/completion banks, no reference	Two wrappers and identity
HARDMath	Category target and auxiliary planning/completion banks, no reference	Two wrappers and identity

The task-specific wrapper strings are listed below. The placeholder `{descriptor}` is replaced by the bank text, and `\n` denotes an actual line break. Each variant is encoded separately. Only descriptor-token positions are retained as slots. The wrapper text affects their contextualized states but does not itself become bank content.

```
Big Five questionnaire:
Stable self-profile for questionnaire answers:\n{descriptor}
When deciding whether a statement describes you:\n{descriptor}

Big Five generation, long-context persistence, and dialogue shift:
Interaction style note:\n{descriptor}
Internal response style:\n{descriptor}
Behavioral steering note for replies:\n{descriptor}

PHYSICS:
Internal physics reasoning prior:\n{descriptor}
Hidden physics skill bank:\n{descriptor}
{descriptor}

HARDMath:
Internal applied mathematics reasoning prior:\n{descriptor}
Hidden HARDMath skill bank:\n{descriptor}
{descriptor}
```

A teacher model first distills the reasoning heuristics from construction examples. Wrapping and encoding that text require no gradient updates to the steered model.

The questionnaire wrapper list contains two strings, while the combined 679-slot ablation setting was described as using three templates. The third template and the composition of that setting remain unresolved. Appendix F.2 retains the reported ablation values. The measured 195-slot PHYSICS bank and 480/432-slot dialogue target/reference banks are documented in Appendix B.3. These counts do not provide the full auxiliary-bank inventory for every run.

Exact implementation examples. The following descriptor and card spans are quoted from the implementation and included with source identifiers and hashes in `appendix/bank_examples.json`. They have not been matched to the artifacts used for the reported benchmark rows. The HARDMath entry is one card from the ODE bank. Exact PHYSICS descriptor spans remain unavailable.

```
Base openness descriptor:
You are an open, imaginative assistant. Explore unusual but relevant angles, welcome new
     ideas, and sound intellectually curious without becoming random or impractical.
```

```
Warm dialogue descriptor:
You are a warm, caring assistant. Sound friendly, supportive, emotionally attentive, and
     gently encouraging without becoming overly sentimental.
```

```
HARDMath ODE card (ode_integrate):
Trigger: an ordinary differential equation gives a derivative of y
Core move: integrate with constants determined by all boundary or initial conditions
Anchor: integrate in the correct variable and apply each condition after the general
     antiderivative is available
Check: the final y should satisfy the differential equation and every condition
```

F Results breakdown by task

F.1 Big Five self-evaluation

Table 8: **Big Five questionnaire self-evaluation on Meta-Llama-3.1-8B-Instruct.** Columns report target-trait score, target shift from the plain model, non-target drift, and drift-adjusted score, where $\mathrm{DAS} = \Delta_{\mathrm{target}} - D_{\text{non-target}}$. Bold indicates the highest DAS among steering methods within each trait.

Trait	Method	Score	Shift	Drift	DAS
Openness	Plain	3.5	0.0	0.000	0.000
	Prompt	4.1	+0.6	0.350	0.250
	CAA	3.8	+0.3	0.050	0.250
	Ours	4.6	+1.1	0.175	**0.925**
Conscientiousness	Plain	3.9	0.0	0.000	0.000
	Prompt	4.2	+0.3	0.625	-0.325
	CAA	4.3	+0.4	0.025	0.375
	Ours	4.9	+1.1	0.100	**1.000**
Extraversion	Plain	2.8	0.0	0.000	0.000
	Prompt	4.7	+1.9	0.350	**1.550**
	CAA	2.9	+0.1	0.225	-0.125
	Ours	3.0	+0.2	0.050	0.150
Agreeableness	Plain	4.7	0.0	0.000	0.000
	Prompt	4.9	+0.2	0.250	-0.050
	CAA	4.7	+0.0	0.075	-0.075
	Ours	4.9	+0.2	0.125	**0.075**
Neuroticism	Plain	2.0	0.0	0.000	0.000
	Prompt	4.5	+2.5	0.700	**1.800**
	CAA	2.7	+0.7	0.200	0.500
	Ours	4.3	+2.3	0.575	1.725
Average	Plain	–	0.0	0.000	0.000
	Prompt	–	+1.10	0.455	0.645
	CAA	–	+0.30	0.115	0.185
	Ours	–	+0.98	0.205	**0.775**

Table 9: **Big Five questionnaire self-evaluation on Qwen3-30B-A3B.** Columns report target-trait score, target shift from the plain model, non-target drift, and drift-adjusted score, where $\mathrm{DAS} = \Delta_{\text{target}} - D_{\text{non-target}}$. Bold indicates the highest DAS among steering methods within each trait.

Trait	Method	Score	Shift	Drift	DAS
Openness	Plain	4.1	0.0	0.000	0.000
	Prompt	4.8	+0.7	0.475	0.225
	CAA	4.4	+0.3	0.075	0.225
	Ours	4.6	+0.5	0.100	**0.400**
Conscientiousness	Plain	4.5	0.0	0.000	0.000
	Prompt	4.9	+0.4	0.525	-0.125
	CAA	4.5	+0.1	0.075	**0.025**
	Ours	4.6	+0.1	0.150	-0.050
Extraversion	Plain	3.6	0.0	0.000	0.000
	Prompt	5.0	+1.4	0.500	**0.900**
	CAA	3.7	+0.1	0.100	0.000
	Ours	3.8	+0.2	0.150	0.050
Agreeableness	Plain	4.2	0.0	0.000	0.000
	Prompt	5.0	+0.8	0.500	0.300
	CAA	4.3	+0.1	0.075	0.025
	Ours	4.9	+0.7	0.150	**0.550**
Neuroticism	Plain	2.5	0.0	0.000	0.000
	Prompt	3.6	+1.1	0.625	**0.475**
	CAA	2.8	+0.4	0.100	0.300
	Ours	3.0	+0.5	0.100	0.400
Average	Plain	–	0.0	0.000	0.000
	Prompt	–	+0.88	0.525	**0.355**
	CAA	–	+0.20	0.085	0.115
	Ours	–	+0.40	0.130	0.270

The per-trait tables clarify that the aggregate Pareto picture in Table 1 is not uniform across all traits. Prompting remains strongest for the largest raw movement on extraversion and neuroticism, while memory-bank steering is strongest on several drift-adjusted rows, especially openness and agreeableness on Qwen3 and openness, conscientiousness, and agreeableness on Llama.

F.2 Wrapper ablation

Table 10 reports an additional Llama-3.1-8B IPIP-50 questionnaire evaluation over all five traits with greedy decoding. Every row uses the same frozen per-trait selector and steering hyperparameters. The slot-matched raw condition repeats the descriptor without wrapper diversity. The raw and combined conditions have the same mean target shift, 0.98, with mean drift of 0.217 and 0.205, respectively. This comparison covers one backbone and one task.

Table 10: **Wrapper ablation on Llama-3.1-8B.** All conditions use the same frozen selector and steering hyperparameters. The final column gives the score reported for this experiment.

Condition	Slots	Mean shift	Mean drift	Reported score
Raw descriptor	226	0.98	0.217	0.655
Raw descriptor repeated	679	0.98	0.233	0.631
Wrapper 1 only	226	0.96	0.205	0.653
Wrapper 2 only	226	0.96	0.195	0.668
Combined setting	679	0.98	0.205	0.673

F.3 Big Five generation

Table 11: **Big Five contrastive generation judged by an LLM judge on Meta-Llama-3.1-8B-Instruct.** Columns report target-trait score, target-score shift from the plain model, the reported off-target statistic D, contrastive margin, and generation-adjusted score, where $\mathrm{GAS} = \Delta_{\mathrm{target}} - D_{\mathrm{non\text{-}target}}$. The definition of D remains unresolved, as discussed in Appendix D. Contrastive margin is a separate diagnostic.

Trait	Method	Score	Shift	D	Margin	GAS
Openness	Plain	4.54	+0.00	1.11	3.33	-1.110
	Prompt	4.79	+0.25	1.04	3.62	**-0.790**
	CAA	4.62	+0.08	1.10	3.29	-1.020
	Ours	4.50	-0.04	1.07	3.12	-1.110
Conscientiousness	Plain	4.83	+0.00	0.68	3.71	-0.680
	Prompt	4.79	-0.04	0.64	3.62	-0.680
	CAA	4.92	+0.09	0.66	3.88	-0.570
	Ours	4.96	+0.13	0.69	3.92	**-0.560**
Extraversion	Plain	3.50	+0.00	1.18	1.38	-1.180
	Prompt	4.33	+0.83	1.25	2.71	**-0.420**
	CAA	3.92	+0.42	1.18	1.83	-0.760
	Ours	4.12	+0.62	1.19	2.29	-0.570
Agreeableness	Plain	4.83	+0.00	0.96	3.79	-0.960
	Prompt	4.92	+0.09	0.55	3.83	**-0.460**
	CAA	4.88	+0.05	1.02	3.83	-0.970
	Ours	4.92	+0.09	0.95	3.88	-0.860
Neuroticism	Plain	3.62	+0.00	0.99	1.12	-0.990
	Prompt	4.29	+0.67	0.94	2.42	-0.270
	CAA	3.50	-0.12	0.93	1.00	-1.050
	Ours	4.42	+0.80	0.71	2.79	**0.090**
Average	Plain	4.26	+0.00	0.98	2.67	-0.984
	Prompt	4.62	+0.36	0.88	3.24	**-0.524**
	CAA	4.37	+0.11	0.98	2.77	-0.874
	Ours	4.58	+0.32	0.92	3.20	-0.602

Table 12: **Big Five contrastive generation judged by an LLM judge on Qwen3-30B-A3B.** Columns report target-trait score, target-score shift from the plain model, the reported off-target statistic D, contrastive margin, and generation-adjusted score, where $\mathrm{GAS} = \Delta_{\mathrm{target}} - D_{\mathrm{non\text{-}target}}$. The definition of D remains unresolved, as discussed in Appendix D.

Trait	Method	Score	Shift	D	Margin	GAS
Openness	Plain	2.17	+0.00	0.83	0.67	-0.830
	Prompt	2.83	+0.66	0.83	1.67	-0.170
	CAA	1.83	-0.34	1.00	-0.33	-1.340
	Ours	3.83	+1.66	1.12	2.00	**0.540**
Conscientiousness	Plain	3.83	+0.00	0.71	2.83	-0.710
	Prompt	4.17	+0.34	0.58	3.17	-0.240
	CAA	3.83	+0.00	0.62	2.83	-0.620
	Ours	5.00	+1.17	0.92	4.00	**0.250**
Extraversion	Plain	2.00	+0.00	0.92	0.83	-0.920
	Prompt	3.67	+1.67	0.92	2.67	**0.750**
	CAA	2.00	+0.00	0.96	0.83	-0.960
	Ours	3.33	+1.33	1.17	0.83	0.160
Agreeableness	Plain	3.50	+0.00	0.62	2.50	-0.620
	Prompt	4.17	+0.67	0.71	3.17	-0.040
	CAA	3.50	+0.00	0.75	2.33	-0.750
	Ours	5.00	+1.50	0.88	4.00	**0.620**
Neuroticism	Plain	1.00	+0.00	0.96	-4.00	-0.960
	Prompt	2.67	+1.67	0.67	-0.33	**1.000**
	CAA	1.00	+0.00	0.96	-4.00	-0.960
	Ours	2.67	+1.67	1.04	-0.67	0.630
Average	Plain	2.50	+0.00	0.81	0.57	-0.808
	Prompt	3.50	+1.00	0.74	2.07	0.260
	CAA	2.43	-0.07	0.86	0.33	-0.926
	Ours	3.97	+1.47	1.02	2.03	**0.440**

The generation tables show why the main-text claim is improvement over CAA rather than universal dominance over prompting. Memory-bank steering improves over CAA on the average Llama and Qwen3 generation metrics, but prompt remains the strongest visible-context baseline on several drift-adjusted rows, especially on Qwen3.

F.4 Long-context persistence

Table 13: **Qwen3 long-context dialogue persistence by trait and prefill length.** Each method cell reports $P/T/Q$: conversation-level target persistence, mean turn-level target style score, and conversation-level overall quality. Prefill is non-task assistant-history context inserted before the dialogue. Prompt is the once-initialized visible-instruction baseline; Ours is memory-bank steering. Missing cells indicate no valid judged run after judge-completion filtering.

Trait	Prefill	Plain	Prompt	CAA	Ours
Warm	8k	9.04 / 8.68 / 9.19	9.19 / 8.95 / 9.34	8.94 / 8.41 / 9.02	8.51 / 8.29 / 8.39
	16k	9.01 / 8.62 / 9.14	9.12 / 8.79 / 9.21	8.91 / 8.43 / 9.00	8.90 / 8.43 / 8.92
	24k	8.96 / 8.64 / 9.10	9.06 / 8.73 / 9.13	8.94 / 8.48 / 8.99	8.91 / 8.47 / 9.00
Formal	8k	8.07 / 6.31 / 9.07	8.41 / 6.68 / 9.18	8.06 / 6.43 / 9.01	8.04 / 6.40 / 8.97
	16k	8.04 / 6.36 / 9.01	8.16 / 6.58 / 9.05	8.10 / 6.51 / 8.98	8.06 / 6.43 / 9.00
	24k	8.05 / 6.35 / 9.04	8.18 / 6.53 / 9.05	8.05 / 6.47 / 8.97	8.05 / 6.41 / 8.98
Assertive	8k	8.94 / 7.88 / 9.24	9.01 / 8.12 / 9.30	8.79 / 7.59 / 9.01	8.73 / 7.56 / 9.01
	16k	8.83 / 7.91 / 9.05	9.01 / 8.15 / 9.19	8.77 / 7.60 / 8.98	8.72 / 7.57 / 8.99
	24k	8.85 / 7.91 / 9.05	8.98 / 8.06 / 9.11	8.73 / 7.62 / 8.97	8.71 / 7.57 / 8.96
Cautious	8k	8.65 / 6.18 / 9.12	8.86 / 6.34 / 9.20	8.65 / 6.23 / 9.02	8.71 / 6.29 / 8.97
	16k	8.60 / 6.13 / 9.10	8.72 / 6.25 / 9.12	8.66 / 6.12 / 8.99	8.74 / 6.19 / 8.99
	24k	8.56 / 6.10 / 9.06	8.65 / 6.19 / 9.06	8.67 / 6.10 / 8.99	8.70 / 6.22 / 8.99
Dismissive	8k	1.00 / 1.01 / 9.69	1.61 / 1.26 / 9.49	1.00 / 1.04 / 9.28	2.35 / 1.99 / 6.56
	16k	1.00 / 1.02 / 9.54	1.05 / 1.05 / 9.60	1.01 / 1.03 / 9.17	1.85 / 1.84 / 6.51
	24k	1.00 / 1.01 / 9.58	1.00 / 1.02 / 9.58	1.00 / 1.04 / 9.23	1.68 / 1.60 / 6.55
Anxious	8k	4.96 / 1.67 / 8.90	5.59 / 2.14 / 8.90	5.73 / 1.96 / 8.55	6.23 / 2.54 / 6.53
	16k	4.99 / 1.74 / 8.73	4.94 / 1.73 / 8.81	5.86 / 2.01 / 8.52	6.21 / 2.39 / 7.12
	24k	4.69 / 1.75 / 8.75	4.88 / 1.74 / 8.77	6.05 / 2.03 / 8.51	6.27 / 2.27 / 7.35

Table 14: **Meta-Llama-3.1-8B-Instruct long-context dialogue persistence by trait and prefill length.** Each cell reports $P/T/Q$: conversation-level target persistence, mean turn-level target style score, and conversation-level overall quality. The main text reports only the model-level averages; this table gives the full trait$\times$budget breakdown. Plain is the opposite-prefill baseline. Prompt adds one visible target-style system instruction before the same opposite-prefill context. CAA is the frozen post-attention residual baseline (layer 14, scale 2), and Ours reports the frozen memory-bank configuration used in the final judged audit.

Trait	Prefill	Plain	Prompt	CAA	Ours
Warm	8k	7.66 / 7.38 / 8.57	8.76 / 8.06 / 8.99	8.23 / 7.49 / 8.83	8.08 / 7.41 / 8.83
	16k	8.17 / 7.52 / 8.83	8.75 / 8.00 / 8.95	7.92 / 7.47 / 8.71	8.26 / 7.43 / 8.83
	24k	7.72 / 7.60 / 8.61	8.70 / 8.01 / 9.00	8.29 / 7.71 / 8.81	7.92 / 7.46 / 8.70
Formal	8k	7.66 / 6.09 / 8.79	8.01 / 6.20 / 8.98	8.04 / 6.56 / 9.00	7.69 / 6.01 / 8.76
	16k	7.51 / 6.05 / 8.75	7.96 / 6.27 / 8.93	8.11 / 6.58 / 9.00	7.51 / 6.01 / 8.63
	24k	7.48 / 6.04 / 8.67	7.93 / 6.26 / 8.93	8.06 / 6.62 / 8.99	7.52 / 5.99 / 8.64
Assertive	8k	7.80 / 6.79 / 8.78	7.91 / 7.10 / 8.82	8.32 / 7.25 / 8.98	7.54 / 6.37 / 8.51
	16k	6.55 / 6.26 / 7.87	7.68 / 6.95 / 8.58	8.12 / 7.22 / 8.89	6.95 / 6.25 / 8.00
	24k	7.10 / 6.50 / 8.19	7.67 / 6.89 / 8.61	7.99 / 7.21 / 8.87	7.04 / 6.21 / 7.97
Cautious	8k	7.36 / 5.73 / 8.46	8.39 / 6.39 / 9.00	7.84 / 6.17 / 8.77	7.46 / 5.71 / 8.48
	16k	7.47 / 5.70 / 8.62	8.32 / 6.37 / 8.97	7.92 / 6.16 / 8.86	7.34 / 5.70 / 8.34
	24k	7.47 / 5.85 / 8.57	8.22 / 6.30 / 8.97	7.98 / 6.28 / 8.80	7.87 / 5.86 / 8.78
Dismissive	8k	1.00 / 1.06 / 8.69	7.62 / 6.06 / 5.87	1.02 / 1.19 / 8.74	3.63 / 2.74 / 4.45
	16k	1.01 / 1.06 / 8.71	8.41 / 7.09 / 5.39	1.10 / 1.20 / 8.66	4.08 / 2.69 / 4.14
	24k	1.03 / 1.07 / 8.73	8.45 / 6.87 / 5.37	1.21 / 1.16 / 8.57	3.92 / 2.62 / 4.46
Anxious	8k	5.88 / 2.09 / 8.06	6.11 / 2.20 / 8.04	6.08 / 2.29 / 8.03	6.53 / 2.72 / 7.78
	16k	6.09 / 2.14 / 8.07	6.56 / 2.88 / 8.07	6.13 / 2.29 / 8.07	6.49 / 2.67 / 7.78
	24k	6.12 / 2.14 / 8.03	6.46 / 2.47 / 8.13	6.16 / 2.29 / 8.04	6.50 / 2.66 / 7.89

Qwen3 shows the clearest positive long-context cases under the budget-matched setup, especially on the more difficult low-baseline traits. Llama behaves differently: prompt remains the strongest raw persistence mechanism on average, while the latent methods trade off persistence against judged quality more sharply.

F.5 Dialogue shift

For readability, the appendix tables below foreground the two primary post-shift outcomes: adaptation alignment $\mathrm{Align} = s_{\mathrm{target}} - s_{\mathrm{old}}$ and judged turn quality Q_{turn}. The raw target-style and old-style component scores are omitted from the paper-level tables and remain available in the released results.

Table 15: **Qwen3 dialogue-shift post-shift alignment by transition and dialogue length.** Entries report Align on post-shift turns. Bold marks the best matched-budget method in each row among Plain, Prompt, CAA, and Ours. Prompt-repl. is shown only as an audit because it adds visible steering text at the shift point.

Transition	Turns	Metric	Plain	Prompt	CAA	Ours	Prompt-repl.
Warm→Assertive	8	Align	-0.04	-0.46	**0.19**	-0.50	-0.29
Warm→Assertive	16	Align	-0.23	-0.25	**0.00**	-0.34	-0.29
Warm→Assertive	24	Align	-0.40	-0.43	**-0.30**	-0.80	-0.35
Assertive→Warm	8	Align	1.33	1.17	1.56	**1.88**	1.12
Assertive→Warm	16	Align	1.43	1.38	1.66	**2.29**	1.34
Assertive→Warm	24	Align	1.25	1.24	1.38	**1.90**	1.17
Formal→Cautious	8	Align	0.22	-0.06	**0.46**	0.06	0.11
Formal→Cautious	16	Align	-0.14	-0.21	**0.19**	-0.02	-0.12
Formal→Cautious	24	Align	-0.29	-0.44	-0.17	**-0.15**	-0.06
Cautious→Formal	8	Align	1.50	1.58	**1.69**	1.42	1.67
Cautious→Formal	16	Align	0.79	0.89	**1.16**	0.71	1.21
Cautious→Formal	24	Align	0.82	1.05	**1.29**	0.93	1.09
Dismissive→Anxious	8	Align	0.83	1.00	1.96	**2.83**	2.06
Dismissive→Anxious	16	Align	1.02	1.55	1.65	**2.74**	4.17
Dismissive→Anxious	24	Align	1.11	1.02	**2.52**	2.48	2.86
Anxious→Dismissive	8	Align	-1.17	-0.67	**0.12**	-1.17	0.00
Anxious→Dismissive	16	Align	-0.93	-0.79	-0.44	**-0.36**	0.21
Anxious→Dismissive	24	Align	-0.77	-0.55	-0.42	**-0.32**	-0.86

Table 16: **Qwen3 dialogue-shift post-shift turn quality by transition and dialogue length.** Entries report Q_{turn} on post-shift turns. Bold marks the best matched-budget method in each row among Plain, Prompt, CAA, and Ours. Prompt-repl. is shown only as an audit because it adds visible steering text at the shift point.

Transition	Turns	Metric	Plain	Prompt	CAA	Ours	Prompt-repl.
Warm→Assertive	8	Q_{turn}	8.67	**8.72**	8.64	7.86	8.65
Warm→Assertive	16	Q_{turn}	8.73	**8.77**	8.71	7.64	8.78
Warm→Assertive	24	Q_{turn}	**8.70**	8.66	8.59	7.25	8.63
Assertive→Warm	8	Q_{turn}	8.47	**8.62**	8.41	7.27	8.58
Assertive→Warm	16	Q_{turn}	**8.58**	8.55	8.45	6.45	8.51
Assertive→Warm	24	Q_{turn}	**8.55**	8.53	8.47	6.44	8.57
Formal→Cautious	8	Q_{turn}	**8.60**	8.58	8.51	8.44	8.49
Formal→Cautious	16	Q_{turn}	**8.55**	8.50	8.54	8.27	8.49
Formal→Cautious	24	Q_{turn}	8.39	**8.43**	8.35	8.19	8.40
Cautious→Formal	8	Q_{turn}	8.46	8.46	**8.50**	7.67	8.42
Cautious→Formal	16	Q_{turn}	8.58	**8.63**	8.59	7.47	8.59
Cautious→Formal	24	Q_{turn}	8.66	**8.70**	8.68	7.89	8.61
Dismissive→Anxious	8	Q_{turn}	**8.56**	8.49	8.33	6.28	7.53
Dismissive→Anxious	16	Q_{turn}	**8.56**	8.24	8.31	4.79	6.27
Dismissive→Anxious	24	Q_{turn}	8.44	**8.52**	7.42	3.98	6.99
Anxious→Dismissive	8	Q_{turn}	**8.62**	8.29	8.03	7.33	7.50
Anxious→Dismissive	16	Q_{turn}	**8.18**	7.95	7.72	5.27	6.64
Anxious→Dismissive	24	Q_{turn}	**8.15**	8.12	8.05	4.16	7.70

Table 17: **Meta-Llama-3.1-8B-Instruct halfway-shift dialogue summary.** Each breakdown is split into two metric rows: post-shift alignment Align and judged turn quality Q_{turn}. Bold marks the best method in each metric row. CAA is the frozen layer-14, scale-2 post-attention residual baseline; Ours is memory-bank summary update.

Breakdown	Metric	Plain	Prompt	CAA	Ours
Overall	Align	1.112	1.178	**1.554**	1.516
	Q_{turn}	**7.72**	7.58	7.34	7.00
Anxious→Dismissive	Align	1.654	2.212	**2.808**	2.692
	Q_{turn}	**5.92**	5.49	4.63	4.49
Assertive→Warm	Align	1.026	1.087	**1.209**	1.148
	Q_{turn}	8.30	**8.35**	8.23	8.03
Cautious→Formal	Align	0.835	0.852	**1.157**	0.913
	Q_{turn}	8.33	**8.35**	8.27	7.87
Dismissive→Anxious	Align	3.615	3.240	4.298	**4.577**
	Q_{turn}	**7.06**	6.44	5.98	5.85
Formal→Cautious	Align	-0.252	**-0.113**	-0.191	-0.209
	Q_{turn}	8.11	**8.14**	8.05	7.77
Warm→Assertive	Align	0.087	0.087	**0.426**	0.383
	Q_{turn}	8.40	8.40	**8.48**	7.65
8 turns	Align	1.185	1.204	**1.398**	1.278
	Q_{turn}	**7.84**	7.57	7.52	7.22
16 turns	Align	1.373	1.429	**1.833**	1.829
	Q_{turn}	**7.56**	7.36	7.15	6.88
24 turns	Align	0.873	0.964	**1.380**	1.344
	Q_{turn}	**7.82**	7.76	7.43	7.03

The shift tables show the main matched-budget pattern from the paper: Qwen3 provides the clearest positive cases for latent behavior updates, especially on difficult transitions such as

dismissive→anxious, while Llama shows a sharper quality trade-off even when alignment improves.

F.6 PHYSICS results

Table 18: **Full PHYSICS subject×mode matrix on Qwen3-30B-A3B.** Values are mean 0–5 partial-credit judge scores, normalized to percent and averaged over three grading passes. Chain of thought (CoT) denotes Qwen3 thinking mode; no-CoT disables it.

Domain	Mode	Plain (%) ↑	Prompt (%) ↑	Ours (%) ↑
Classical mechanics	no-CoT	66.0	**69.4**	68.2
Classical mechanics	CoT	82.0	84.0	**85.4**
Quantum mechanics	no-CoT	65.4	63.5	**68.1**
Quantum mechanics	CoT	69.2	75.0	**76.6**
Atomic physics	no-CoT	71.4	71.8	**73.2**
Atomic physics	CoT	80.2	79.5	**84.2**
Optics	no-CoT	69.2	72.9	**74.2**
Optics	CoT	**85.0**	83.2	83.0
Electromagnetism	no-CoT	60.6	63.6	**64.4**
Electromagnetism	CoT	75.8	76.6	**77.8**
Thermodynamics/statistical mechanics	no-CoT	72.2	76.4	**76.8**
Thermodynamics/statistical mechanics	CoT	89.4	89.0	**90.6**

The PHYSICS results support the structured-heuristic use case: memory banks are most helpful when the benchmark rewards reusable subject-level process reminders rather than persona changes. Gains are especially visible in CoT mode, where the latent heuristic cards can guide longer reasoning traces without occupying visible prompt budget.

F.7 HARDMath results

Table 19: **Full HARDMath category breakdown on Qwen3-30B-A3B with chain of thought.** Values are category partial-credit total / maximum partial-credit total, with the normalized percentage in parentheses.

Category	Plain	Prompt	Ours
Integral 0	13.5 / 30 (45.0)	**14.5 / 30 (48.3)**	**14.5 / 30 (48.3)**
Integral 1	**13.5 / 30 (45.0)**	13.0 / 30 (43.3)	**13.5 / 30 (45.0)**
Laplace-integral 0	26.0 / 30 (86.7)	**29.0 / 30 (96.7)**	28.0 / 30 (93.3)
Laplace-integral 1	25.0 / 30 (83.3)	**29.0 / 30 (96.7)**	22.0 / 30 (73.3)
ODE	7.0 / 54 (13.0)	11.0 / 54 (20.4)	**14.0 / 54 (25.9)**
Polynomial roots	12.0 / 63 (19.0)	**22.0 / 63 (34.9)**	15.0 / 63 (23.8)
Nondim-symbolic	**65.0 / 65 (100.0)**	**65.0 / 65 (100.0)**	**65.0 / 65 (100.0)**
Nondim-numeric	59.0 / 60 (98.3)	**60.0 / 60 (100.0)**	59.0 / 60 (98.3)
Overall	221.0 / 362 (61.0)	**243.5 / 362 (67.3)**	231.0 / 362 (63.8)

The HARDMath results are more mixed than PHYSICS, which is consistent with the benchmark's stricter symbolic answer format and the need for exact final expressions. Even after regrading for mathematical equivalence and partial credit, prompt remains strongest on several categories, while memory banks help most on ODE and tie the prompt baseline on one of the integral splits.

F.8 GSM8K and MMLU capability checks

Table 20: **Qwen3 GSM8K and MMLU capability checks.** These appendix results are intended as coarse capability-preservation checks rather than main steering results.

Benchmark	Model	Trait/Bank	Plain	Prompt	CAA	Ours
GSM8K	Qwen3-30B-A3B	Warm	0.970	0.960	0.970	0.960
GSM8K	Qwen3-30B-A3B	Formal	0.970	0.960	0.970	0.955
GSM8K	Qwen3-30B-A3B	Direct	0.970	0.950	0.975	0.950
GSM8K	Qwen3-30B-A3B	Careful	0.970	0.945	0.955	0.945
GSM8K	Qwen3-30B-A3B	Curt	0.970	0.910	0.955	0.955
GSM8K	Qwen3-30B-A3B	Anxious	0.970	0.935	0.965	0.940
MMLU	Qwen3-30B-A3B	Warm	0.768	0.772	0.772	0.772
MMLU	Qwen3-30B-A3B	Formal	0.768	0.718	0.768	0.768
MMLU	Qwen3-30B-A3B	Direct	0.768	0.784	0.768	0.764
MMLU	Qwen3-30B-A3B	Careful	0.768	0.756	0.768	0.768
MMLU	Qwen3-30B-A3B	Curt	0.768	0.776	0.768	0.764
MMLU	Qwen3-30B-A3B	Anxious	0.768	0.706	0.768	0.764

Results on GSM8K and MMLU suggest that MI preserves capabilities, staying close to the plain-model performance across a range of traits and within the range already induced by Prompt and CAA.

F.9 gpt-oss-120b personality results

The additional backbone and its attention implementation are described in Appendix B.2.

Table 21: **Additional gpt-oss-120b personality results.** Scores use the supplemental evaluation's drift coefficient of 1.5 and are reported separately from the coefficient-1 main-text DAS/GAS. The five-trait aggregate is primary. The openness-excluded row is a subset analysis using the same bank.

Track	Reported metric	Prompt	MI
Self-evaluation, five traits	$\mathrm{DAS}_{1.5}$	0.340	0.252
Self-evaluation, excluding openness	$\mathrm{DAS}_{1.5}$	0.676	0.610
Generation	$\mathrm{GAS}_{1.5}$	0.528	0.370

Prompt leads on every aggregate in Table 21. The plain model scores 4.6 on openness, leaving limited room for a positive shift on that trait. Excluding openness raises both methods' aggregate scores but does not change their order.

F.10 Outstanding ablations and omissions

Figure 3 reports a Qwen3 layer-count sweep with fixed bank content. A placement control comparing selected and alternative sites at the same layer and unit budget is not reported. Such a control would separate the effect of site selection from the number of controlled layers.

G Technical derivations

G.1 Bank-mixture derivation

Let $\mathcal{I}_b$ index the slots belonging to bank b. Concatenated attention with bank-specific slot bias gives

$$\widetilde{\alpha}_{t,m}^{(b)} = \frac{\exp(s_{t,m}^{(b)} + c_t^{(b)})}{\sum_{b'} \sum_{r \in \mathcal{I}_{b'}} \exp(s_{t,r}^{(b')} + c_t^{(b')})}.$$

Factor the numerator as

$$\exp(s_{t,m}^{(b)} + c_t^{(b)}) = \left(\sum_{r \in \mathcal{I}_b} \exp(s_{t,r}^{(b)}) \right) \frac{\exp(s_{t,m}^{(b)})}{\sum_{r \in \mathcal{I}_b} \exp(s_{t,r}^{(b)})} \exp(c_t^{(b)}).$$

Define the within-bank softmax

$$\alpha_{t,m}^{(b)} = \frac{\exp(s_{t,m}^{(b)})}{\sum_{r \in \mathcal{I}_b} \exp(s_{t,r}^{(b)})}$$

and bank evidence

$$\widetilde{\beta}_t^{(b)} = \log \sum_{r \in \mathcal{I}_b} \exp(s_{t,r}^{(b)}) + c_t^{(b)}.$$

Then $\widetilde{\alpha}_{t,m}^{(b)} = \widetilde{\pi}_t^{(b)} \alpha_{t,m}^{(b)}$, where

$$\widetilde{\pi}_t^{(b)} = \frac{\exp(\widetilde{\beta}_t^{(b)})}{\sum_{b'} \exp(\widetilde{\beta}_t^{(b')})}.$$

Therefore

$$o_t^{\mathrm{concat},c} = \sum_b \sum_{m \in \mathcal{I}_b} \widetilde{\alpha}_{t,m}^{(b)} v_m^{(b)} = \sum_b \widetilde{\pi}_t^{(b)} \sum_{m \in \mathcal{I}_b} \alpha_{t,m}^{(b)} v_m^{(b)} = \sum_b \widetilde{\pi}_t^{(b)} u_t^{(b)}.$$

This is the exact identity in Eq. (4). Setting all $c_t^{(b)}$ to zero recovers ordinary concatenated attention.

The size-normalized rule instead uses $\beta_t^{(b)} = \widetilde{\beta}_t^{(b)} - \log M_b$ and $\pi_t = \mathrm{softmax}([\beta_t^{(0)}, \ldots, \beta_t^{(B)}])$. Its slot weights are

$$\alpha_{t,m}^{\mathrm{MI},(b)} = \frac{\exp(s_{t,m}^{(b)} + c_t^{(b)} - \log M_b)}{\sum_{b'} \sum_{r \in \mathcal{I}_{b'}} \exp(s_{t,r}^{(b')} + c_t^{(b')} - \log M_{b'})} = \pi_t^{(b)} \alpha_{t,m}^{(b)}.$$

The resulting output is $o_t^\star = \sum_b \pi_t^{(b)} u_t^{(b)}$, as defined in Eq. (6). Subtracting $\log M_b$ changes the bank probabilities when bank sizes differ. It is equivalent to applying slot bias $c_t^{(b)} - \log M_b$ in concatenated attention.

G.2 RoPE derivation

With RoPE, a prompt score between query position t and key position j is

$$\frac{\langle R_t \bar{q}_t, R_j \bar{k}_j \rangle}{\sqrt{d_h}} = \frac{\langle \bar{q}_t, R_{j-t} \bar{k}_j \rangle}{\sqrt{d_h}}.$$

For a canonical key $\tilde{k}_m$, assigning virtual relative phase δ_m gives

$$\frac{\langle R_t^{-1} q_t, R_{\delta_m} \tilde{k}_m \rangle}{\sqrt{d_h}} = \frac{\langle \bar{q}_t, R_{\delta_m} \tilde{k}_m \rangle}{\sqrt{d_h}}.$$

The inverse rotation cancels the query's explicit absolute rotary phase, with $\delta_m = 0$ as the default relative phase. The canonical query, contextualized bank states, and generated behavior remain context-dependent.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction present memory inception as a training-free latent steering interface whose strongest evidence is the control–drift trade-off, Qwen3 dialogue updates, reasoning-bank transfer on PHYSICS and HARDMath, and selective KV savings. These claims match the benchmark protocols in Appendix D and the detailed results in Appendix F.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion includes dedicated limitations and responsible-release paragraphs covering benchmark diversity, bank quality, selector supervision, backbone dependence, and the gap between KV accounting and realized systems metrics.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The theory content is limited to method derivations, and Appendix G includes the bank-mixture and RoPE derivations together with the assumptions needed for those formulations.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 and Appendices B–F specify the backbones, selector procedure, vLLM path, bank-construction rules, benchmark protocols, sample counts, and evaluation rubrics, and the abstract links to an anonymous repository with scripts and released benchmark assets.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The abstract provides an anonymous repository link for the released code, scripts, and benchmark assets used in the evaluation, together with accompanying repository documentation for reproducing the main experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper is inference-only and specifies the model backbones, steering baselines, benchmark construction, held-out evaluation rules, judge/grading protocols, selector freezing procedure, and benchmark sample counts used for the reported evaluations, especially in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The paper reports repeated grading and averaged judge scores for PHYSICS, along with full per-trait and per-task breakdowns elsewhere, but it does not provide confidence intervals, error bars, or formal significance tests for the main comparisons.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B reports the inference stack and the GPU fleet used in practice (H100, H200, B200, and RTX Pro 6000 Blackwell).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work studies inference-time steering on existing models and does not involve human-subject data collection, deceptive annotation procedures, or undisclosed intervention on deployed users.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader-impacts paragraph discusses beneficial uses such as persistent preferences and reusable heuristics, as well as misuse risks such as covert behavior shaping and hidden policy injection.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The conclusion's responsible-release paragraph states that the release is limited to anonymous code, evaluation scripts, and benchmark assets rather than a deployment-ready hidden-steering service, and it recommends disclosure, access control, audit logging, and human review for sensitive steering artifacts.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits the source models and benchmark releases, names the Llama 3.1 Community License and Apache 2.0 for the two backbones, notes that the IPIP questionnaire items are public-domain, and identifies provider service terms for API-based judges.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper documents the dialogue benchmark construction, representative bank content, and evaluation setup in the appendix, and the linked anonymous repository provides the released implementation and usage documentation alongside these assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not use crowdsourcing or compensated human-subject experimentation.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human-subject study requiring IRB review was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The evaluation methodology explicitly uses external LLM judges: GPT-4o-mini for Big Five generation and dialogue/long-context judging, and GPT-5.5-thinking for HARDMath and PHYSICS grading, as described in Appendix D.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.