BRIDGE: BEHAVIORAL RULE INSTALLED VIA DUAL-GATE STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

The outcome of a language model (LM) depends as much on what it already knows as on what it is told. However, this knowledge is partly volatile: facts become outdated or contradictory in time. What we want is a conditional rule: use their memory when asked about stable facts, rely on a given context when asked about volatile facts, and defer when facts are volatile and no context is available. However no model we have tested does this, and no training installs it either: three small language model (SLM) backbones and five training objectives built for this task reach at most a third of the required behavior. Prompting does not close the gap either: instructing a model to abstain raises deferral indiscriminately. Yet *fact volatility* and *context presence* are both linearly decodable in these models, but nothing connects them to the decision. We wire them directly. We introduce BRIDGE, an inference time method that reads both signals from activations and steers only when both hold. Gating on context presence alone raises deferral on stable and volatile questions alike. The two-signal gate raises *discrimination* on every backbone, by a wide margin, leaving contextual obedience essentially unchanged. The result is an SLM that can be trusted for fresh information and a way to install rules that depend on what a model knows, something neither a prompt nor a fine-tuning objective can express.

1 INTRODUCTION

Small language models(SLMs) are increasingly used as a practical alternative to large language models, because they are cheaper to deploy, easier to fine-tune, and attractive for domain-specific applications. In many such settings, however, the relevant knowledge is not always fixed: prices, availability, policies, regulations and organisational facts change, and whatever the model memorised at training time goes stale with them. A model that answers such questions from memory gives a confident, well-formed and outdated answer. Retrieval-augmented generation (Lewis et al., 2020) supplies fresh context at inference time, but retrieval can miss or return an incomplete passage, so the model still has to decide on its own when to trust its memory. We call a fact *volatile* if its answer changed between two knowledge snapshots T_1 and T_2, and *stable* otherwise; volatility is fixed by the data, not by the model. The behavior we want, referred to further as behavioral the rule, is: answer stable facts from memory, answer volatile facts from the retrieved context, and defer when the fact is volatile and no context is available. Prior work addresses parts of this rule, such as knowledge conflicts, abstention, adaptive retrieval, but not its joint condition on volatility and context. Neither prompting nor fine-tuning installs it (Section 3.1). Prompting a model to abstain raises deferral on stable and volatile questions alike. Across five fine-tuning objectives and three 7-8B backbones, the objectives that keep context obedience reach at most 28.6 pp deferral discrimination, and the one that defers most loses context obedience.

We introduce BRIDGe (Behavioral Rule Installed via Dual-Gate steering), a steering method that installs the behavioral rule in a fine-tuned model without updating its weights. Its design follows from a diagnosis of the model. Volatility and context presence are both linearly decodable from the residual stream, but deferral responds causally only to context presence, and none of the training objectives we test connects volatility to it. BRIDGe therefore reads both signals with linear probes and steers the hidden state along the presence direction only when it reads as volatile and context-free. Prior conditional steering (Lee et al., 2025) gates on a property of the input. BRIDGe instead gates on the model's own estimate of whether a fact has changed, which is not in the input, and it

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: The behavioral rule and the metric that scores each cell (↑ higher is better, ↓ lower is better).

Fact	Context given	No context
Stable	answer (not scored)	answer from memory: SR↑, SDR↓
Volatile	answer from context: CO↑	defer: VRR↑

pushes a different signal from the one it reads. Across three 7–8B backbones, BRIDGe improves deferral discrimination over the best fine-tuned model: it defers on volatile facts without context rather than on stable ones. It still answers stable facts from memory and volatile facts from context nearly as well as before.

Our contributions are twofold. First, BRIDGe, a steering method that installs a behavioral rule whose gate depends on what the model knows, not only on what it sees. Second, a recipe for choosing what to read and what to push: steer along the signal the decision already uses, and gate on what the model represents but does not use. Alongside it, we give an evaluation protocol that scores each cell of the rule and penalizes both always and never deferring. The rest of the paper shows where existing approaches fall short, diagnoses the fine-tuned model, then presents and evaluates BRIDGe. We release the source code as open-source [1].

2 SETUP

2.1 BEHAVIORAL EVALUATION

We define a behavioral evaluation protocol that is measuring if our desired rule is followed by the model: whether it declines to answer a question about a changed fact when no supporting context is given, and whether it takes its answer from the context when one is provided, even if the context contradicts what it memorized.

Let $\hat{y}_i(q, c)$ be the model output for query q and optional context c, with $c = \emptyset$ denoting the no-context setting. $\mathrm{CM}(a, b)$ is contains-match: 1 if a is a substring of b, else 0. $\mathcal{D}_s$ and $\mathcal{D}_v$ are the stable and the volatile facts. A *deferral* denoted by $\mathrm{def}(\hat{y}) = 1$ if $\hat{y}$ is an output that declines to give a value: the string $d =$ 'I don't have reliable information about this.' or one of its surface variants ('I don't know', 'I'm not sure', and close paraphrases), which are needed for models never trained on d; $\mathrm{def}(\hat{y}) = 1$ if $\hat{y}$ is a deferral and 0 otherwise.

Stable retention (SR) is the percentage of stable questions answered with the correct value, and stable deferral rate (SDR) the percentage of stable questions deferred. Vacuum refusal rate (VRR; a vacuum prompt is a bare prompt) is the percentage of volatile questions deferred in a bare prompt; each such deferral follows the rule. Contextual obedience (CO) is the percentage of grounded volatile questions answered with $y_i^{T_2}$. With $\hat{y}_i(q_i, c)$ the output for context $c \in \{\emptyset, c_i\}$ and $\mathrm{CM}(a, b) = 1$ if the normalized string a is a substring of the normalized string b (0 otherwise):

$$\begin{aligned} \mathrm{SR} &= \tfrac{100}{|\mathcal{D}_s^{\mathrm{eval}}|} \textstyle\sum_{i \in \mathcal{D}_s^{\mathrm{eval}}} \mathrm{CM}\big(y_i^{T_1}, \hat{y}_i(q_i, \emptyset)\big), \quad \mathrm{SDR} = \tfrac{100}{|\mathcal{D}_s^{\mathrm{eval}}|} \textstyle\sum_{i \in \mathcal{D}_s^{\mathrm{eval}}} \mathrm{def}\big(\hat{y}_i(q_i, \emptyset)\big), \\ \mathrm{CO} &= \tfrac{100}{|\mathcal{D}_v^{\mathrm{grd}}|} \textstyle\sum_{i \in \mathcal{D}_v^{\mathrm{grd}}} \mathrm{CM}\big(y_i^{T_2}, \hat{y}_i(q_i, c_i)\big), \quad \mathrm{VRR} = \tfrac{100}{|\mathcal{D}_v^{\mathrm{eval}}|} \textstyle\sum_{i \in \mathcal{D}_v^{\mathrm{eval}}} \mathrm{def}\big(\hat{y}_i(q_i, \emptyset)\big). \end{aligned} \tag{1}$$

Table 1 gives the metric for each cell of the rule. SR, CO and VRR measure whether the model does the right thing in its cell; SDR measures whether it defers where it should answer. Deferral discrimination, $\mathrm{DD} = \mathrm{VRR} - \mathrm{SDR}$ in percentage points, links the two no-context cells: it rises only when the model defers on volatile facts and not on stable ones, so it is 0 for a model that always or never defers. We report all five together, since any one of them can be raised by giving up another.

2.2 TRAINING OBJECTIVES

To test whether fine-tuning alone can install the rule, we build five training objectives so that each adds one part of it. SFT is the baseline and trains every question on its T_1 value. Volatile Memory

[1] https://anonymous.4open.science/r/BRIDGE-C344

Table 2: The gap on the three backbones. Higher is better for all metrics except SDR. SDR and VRR use the same deferral detector on the same prompts: bare prompts for every row except + *instruction*, which is the zero-shot model with the abstention instruction added to every prompt, including the context prompt that CO is measured on. SFT, VMD and CA-VMD never defer on either side, so their SDR, VRR and DD are all 0. One run per model.

	Llama					Mistral					Qwen				
	SR	SDR	VRR	DD	CO	SR	SDR	VRR	DD	CO	SR	SDR	VRR	DD	CO
Zero-shot	12.3	71.4	72.5	1.2	57.8	31.4	10.4	16.0	5.6	51.1	21.9	2.8	3.6	0.8	58.0
+ instruction	3.0	95.4	96.7	1.4	50.4	25.1	39.0	46.5	7.5	47.3	2.2	95.9	95.1	−0.8	51.6
SFT	53.7	0.0	0.0	0.0	50.4	53.2	0.0	0.0	0.0	52.2	43.1	0.0	0.0	0.0	52.2
VMD	57.2	0.0	0.0	0.0	52.7	55.9	0.0	0.0	0.0	52.7	44.2	0.0	0.0	0.0	51.6
A-VMD	44.6	37.9	96.1	58.2	32.9	48.0	26.0	89.6	63.5	15.8	30.7	54.5	85.1	30.7	34.4
CA-VMD	55.8	0.0	0.0	0.0	62.2	54.6	0.0	0.0	0.0	61.3	43.9	0.0	0.0	0.0	54.0
CA-AVMD	53.5	6.1	24.9	18.8	62.7	55.6	3.0	31.6	28.6	62.0	43.7	3.0	15.6	12.6	54.9

Distillation (VMD) masks the loss on the outdated volatile value. Active VMD (A-VMD) trains deferral (d) in its place. Context Augmented VMD (CA-VMD) and Context Augmented AVMD (CA-AVMD) extend VMD and A-VMD with a context: half of the volatile training examples are grounded in context ($\mathcal{D}_v^{\text{ctx}}$, target $y_i^{T_2}$) and half bare ($\mathcal{D}_v^{\text{bare}}$). Despite the name, no objective uses a teacher model: all five are cross-entropy fine-tuning with different targets. The first three objectives train on bare prompts only, so no training example shows that context changes the target. CA-VMD puts no loss on the bare half, and CA-AVMD trains it on d. All five objectives use LoRA, the standard parameter-efficient fine-tuning method. Their equations are detailed in Appendix A.1; the CA-AVMD loss is:

$$\mathcal{L}_{\text{CA-AVMD}} = \underbrace{-\sum_{i\in\mathcal{D}_s} \log p_\theta\big(y_i^{T_1} \mid q_i\big)}_{\text{stable, bare}} \underbrace{-\sum_{i\in\mathcal{D}_v^{\text{ctx}}} \log p_\theta\big(y_i^{T_2} \mid c_i, q_i\big)}_{\text{volatile, grounded}} \underbrace{-\sum_{i\in\mathcal{D}_v^{\text{bare}}} \log p_\theta\big(d \mid q_i\big)}_{\text{volatile, bare}}. \quad (2)$$

CA-AVMD is the only objective that supervises all three scored cells. The two conditions of the rule enter its loss differently. Context presence is part of the input - a grounded and a bare example differ by c_i. On the other hand, volatility is not and it only decides whether a fact belongs to $\mathcal{D}_s$ or to $\mathcal{D}_v$, and no token of q_i marks it. The model can therefore condition its output on context presence by reading the prompt, but on volatility only through an estimate that it forms from the question.

Our experiments are based on EvoWiki (Tang et al., 2025), an auto-updatable benchmark built from Wikipedia and Wikidata revisions that partitions entity knowledge into *stable*, *evolved* and *uncharted* facts according to how it moved between two dumps (additional details in A.2).

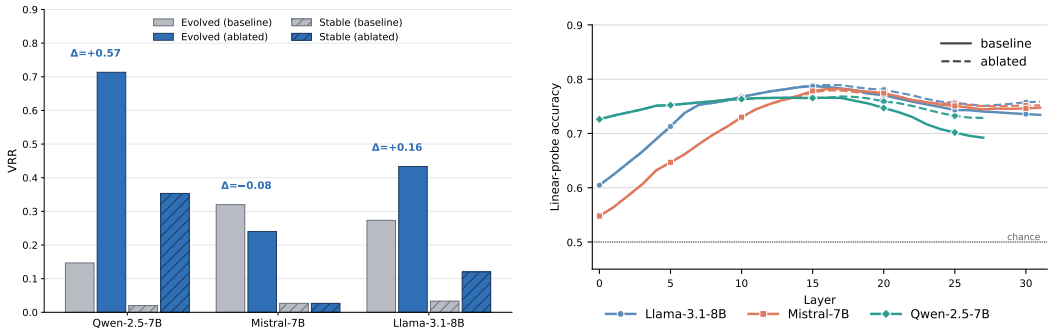
3 Problem Analysis: Why Fine-Tuned Models Fail to Defer?

3.1 Behavioral Gap

We first ask whether the backbones already follow the rule, and then try the two usual ways to add it: an instruction and fine-tuning. Following the rule requires a high DD while keeping SR and CO at the level of SFT. We observe none of them does this very well (Table 2).

Neither baseline tells stable from volatile facts. Zero-shot Llama defers often, but about as often on stable as on volatile questions, which is also why its SR is low; Mistral and Qwen rarely defer at all. SFT never defers, since it trains an answer for every question. The signal is not missing, though: Section 3.2 shows that volatility is already linearly decodable in the zero-shot model, and that fine-tuning rotates this direction rather than sharpening it.

The simplest fix is to instruct the model to abstain. The instruction raises deferral on stable questions about as much as on volatile ones, so DD changes little. It cannot do better: it applies to every question alike, and which facts have an outdated value is a property of the model's memory, not of the prompt.



(a) Deferral rate.

(b) Volatility-probe accuracy.

Figure 1: Effect of the targeted ablation on CA-AVMD: deferral on volatile and stable questions (a) and probe accuracy by layer (b), before (solid) and after (dashed).

We then train the rule in directly with the objectives of Section 2.2. Masking the outdated value is not enough: VMD and CA-VMD have no deferral target and never defer. A-VMD adds that target and defers on almost every bare volatile question, but also on many stable ones, and it follows the context far less than SFT. Having never seen a context in training, the model has no reason to answer differently when context is given. CA-AVMD, the only objective that supervises every cell of the rule, keeps SR and CO at the level of SFT or above, but defers on only a minority of bare volatile questions, across training seeds as well (Appendix A.11). CA-AVMD therefore sees the full rule in training and still applies it to only part of the questions it covers, so the gap is not a lack of supervision. This matches Zhou et al. (2026) on temporal QA: SFT makes models overconfident, and even RL with abstention rewards tends to ignore abstention or, on balanced data, to abstain on everything. Section 3.2 looks inside the model for what is missing.

3.2 Mechanistic Diagnosis

The behavioral gap has three possible causes: the model does not represent volatility, it represents volatility but does not use it, or its deferral depends on something else. We separate them with three questions, asked of CA-AVMD on all three backbones.

Is volatility represented? Yes. A linear probe separates stable from volatile questions in the residual stream (Figure 1b, solid lines), and it does so already in the zero-shot model. Fine-tuning does not make the probe more accurate; it only rotates the direction, which the trained variants then share (A.5, Figure 3a). Context presence is linearly decodable as well.

Does deferral depend on it? No. Activation patching on matched stable/volatile pairs locates the components that carry the distinction (A.5, Figure 3b). If they were the switch between answering and deferring, ablating them would stop deferral on volatile questions. It does not: on Mistral deferral is unchanged, and on Llama and Qwen it rises, on stable questions as much as on volatile ones (Figure 1a). The ablation does not remove the signal either, since the volatility probe stays almost as accurate as before (Figure 1b, dashed lines). Volatility is spread across the upper half of the network, and the deferral decision does not read it.

What does deferral depend on? On Llama, we push the hidden state by the same amount along the volatility direction $\hat{v}$ and along the presence direction $\hat{c}$ (A.5, Table 9). Pushing presence moves deferral several times further, and it moves it on stable questions too: the model starts to decline questions it answers correctly, and when steered far enough it stops using the context that is present. Deferral is therefore dependent on whether the prompt has a context. Steering volatility has a smaller effect but a more selective one, and the two directions are nearly orthogonal, so they act as separate channels. The model thus has both signals the rule needs but uses only one: deferral follows context presence, while volatility, though represented, barely reaches it. BRIDGe (Section 4) closes this gap: it reads volatility with a probe and uses it to decide when to push along presence.

4 BRIDGE: BEHAVIORAL RULE INSTALLED VIA DUAL-GATE STEERING.

BRIDGe is an inference-time conditional hook that installs the deferral rule the fine-tuned model failed to learn. It *reads* the volatility and presence probes to decide when to intervene, and *pushes* along the presence direction, the signal deferral already responds to, to make the model defer.

4.1 METHOD

Activation steering modifies a model's behavior at inference time by adding a vector to its hidden states rather than by changing its weights. The direction is typically estimated from contrastive patterns Turner et al. (2023); Rimsky et al. (2024), trained linear classifier decision boundaries Alain & Bengio (2016); Tigges et al. (2023), or dimensionality reduction across latent spaces Zou et al. (2023b); Burns et al. (2022). We use the same mechanism, with two differences. The direction is not fitted for this purpose but reused from the probes, and the intervention is conditional: it is applied only on inputs whose hidden state satisfies two threshold tests, rather than on every forward pass.

Let $h \in \mathbb{R}^d$ be the hidden state at the hook layer, l, selected per backbone, which is the layer where the intervention works, and d be the width of the model. Here we read the two probe weight vectors: the volatility probe $\hat{v}$ and the context probe $\hat{c}$, which we then ℓ_2-normalize. BRIDGe applies the hook at `prompt_last` when the two conditions hold, i.e. the fact is volatile and no context is present:

$$h' \;=\; h \;-\; \alpha\,\hat{c}\,\mathbb{1}\big[\hat{v}^\top h > \tau_v \;\wedge\; \hat{c}^\top h < \tau_c\big] \tag{3}$$

where $\alpha > 0$ is the magnitude of the injection and $\mathbb{1}(\cdot)$ is the dual gate. The first condition, $\hat{v}^\top h > \tau_v$ holds when the state projects as volatile; the second, $\hat{c}^\top h < \tau_c$ holds when no context is present. When both hold, $-\alpha\,\hat{c}$ moves the state away from the context-present region; otherwise the indicator is zero and $h' = h$, so the model runs untouched.

4.2 HYPERPARAMETER CHOICES

Regarding the choice of l, we hook the last block of every backbone: layer 31 for LLAMA-3.1-8B and MISTRAL3-7B, layer 27 for QWEN2.5-7B. Patching (Section 3.2) shows the deferral decision is made in the top half of the network, and we take the last block of that half. A post-hoc sweep on the presence-only gate, at matched fire rate, agrees (Section 5.2): discrimination is highest at layers 24–31 (21.2 and 18.1) and drops to 8.2 at layer 16 and below.

Threshold τ_v is set by quantile: the value at which 20, 30 or 40% of stable prompts pass the volatility condition. τ_c is taken from a short ladder around the midpoint between the mean presence projection on prompts without context and on prompts with grounded context. On LLAMA the midpoint opens the gate on 67% of prompts, so its ladder starts at -0.5. We report the setting with the best trade-off between discrimination (VRR$-$SDR) and stable retention (SR); on all three backbones this is the same threshold, τ_v at the 30% stable pass rate.

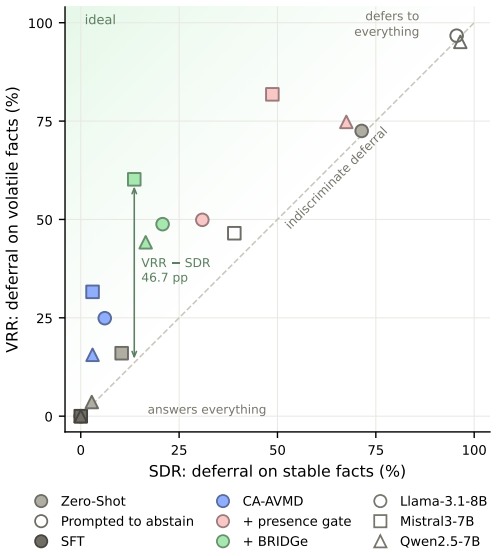
As in Section 3.2, we express α as $k \cdot \text{gap}$, where $\text{gap} = \|\mu_v - \mu_s\|$ is the distance between the average volatile and average stable activation at the hook layer, so k=1 is a push of one class separation. LLAMA and MISTRAL use α=20, i.e. k=2.58 and 4.57. On QWEN the gap is 65.4, so α=20 is only k=0.31 and barely changes the output; we raise it to α=168.5, matching LLAMA's k=2.58 (Table 11).

4.3 RESULTS

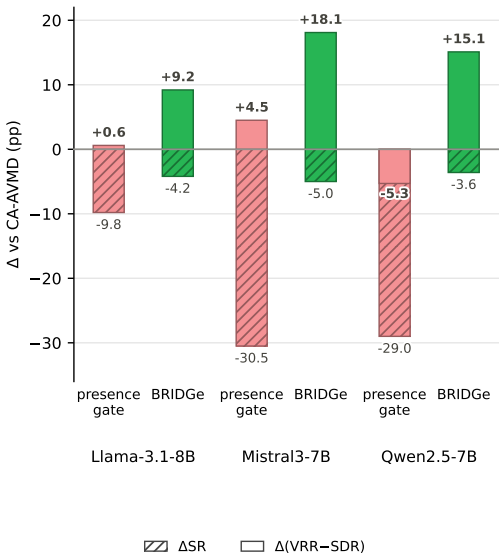
Table 3 compares BRIDGe with three baselines: no training (zero-shot), standard domain adaptation (SFT), and fine-tuning with an objective targeted at the rule (CA-AVMD). The target behavior has three parts, each with its own metric: answer stable facts from memory (SR) without deferring on them (SDR), defer on volatile facts asked without context (VRR), and follow the passage when one is given (CO). BRIDGe has the highest DD of the arms in Table 3 on every backbone, while keeping most of the stable retention and leaving context-following essentially unchanged.

Table 3: Main behavioral results. Higher is better for all metrics except SDR. +presence is the presence-only gate at the τ_c tuned for it on each backbone (-0.5 / $+0.689$ / $+19.105$); Bold marks the best DD per backbone.

	LLAMA-3.1-8B					MISTRAL3-7B					QWEN2.5-7B				
	SR	SDR	VRR	DD	CO	SR	SDR	VRR	DD	CO	SR	SDR	VRR	DD	CO
Zero-shot	12.3	71.4	72.5	1.2	57.8	31.4	10.4	16.0	5.6	51.1	21.9	2.8	3.6	0.8	58.0
SFT	53.7	0.0	0.0	0.0	50.4	53.2	0.0	0.0	0.0	52.2	43.1	0.0	0.0	0.0	52.2
CA-AVMD	53.5	6.1	24.9	18.8	62.7	55.6	3.0	31.6	28.6	62.0	43.7	3.0	15.6	12.6	54.9
+ presence	43.7	30.7	50.1	19.4	62.7	25.1	48.7	81.8	33.1	61.8	14.7	67.5	74.8	7.3	51.6
+ BRIDGE	49.3	20.8	48.8	**28.0**	62.7	50.6	13.6	60.2	**46.7**	61.8	40.1	16.5	44.2	**27.7**	53.8



(a) Deferral on stable vs. volatile facts.



(b) ΔSR and Δ(VRR−SDR) vs. CA-AVMD.

Figure 2: Deferral discrimination and what it costs. **(a)** The SDR/VRR plane: colour is the arm, shape is the backbone. The dashed diagonal is indiscriminate deferral and vertical distance above it is DD. The prompted-to-abstain points are hollow because they use a different, instructed template from every other point. **(b)** The same two gated arms as a change against CA-AVMD: hatched bars down are ΔSR, solid bars up are Δ(VRR−SDR), on one shared scale. CO is not shown; it moves by at most 3.3 pp on any arm.

4.4 MOTIVATION

Since context presence, not volatility, is the signal causally wired to deferral, the simplest intervention uses that signal alone: $h' = h - \alpha\hat{c}\,\mathbb{1}\left[\hat{c}^\top h < \tau_c\right]$. However, using a single gate would make the model fire on every prompt, including questions about stable facts, because $\hat{c}^T$ carries no volatility information. Therefore, using context presence only installs a weaker rule than the one we want: abstain without context, rather than abstain when uninformed. The volatility condition is what separates the two. In Table 3, gated on presence alone at the τ_c tuned for it, SDR rises about as much as VRR ($+24.6$ against $+25.2$ pp on LLAMA), so DD moves by only $+0.6$, $+4.5$ and -5.3 pp while SR falls by 9.8, 30.5 and 29.0 pp.

Figure 2a plots SDR against VRR. The dashed line $y{=}x$ is a policy that defers at the same rate whatever it is asked, and the height above it is VRR−SDR. SFT sits at the origin and zero-shot LLAMA on the line at $(71.4, 72.5)$. The presence gate moves each backbone parallel to the line; adding the volatility condition, at the same α and on the same checkpoint, moves it above. Figure 2b shows what each move costs in SR. On QWEN the presence gate discriminates worse than the unsteered checkpoint.

Table 4: Direction ablation; only the steering direction changes within each backbone. Each entry is the change in DD (pp) relative to CA-AVMD without steering. Bold marks the best direction per backbone, and † marks cells where answer quality on stable questions drops (95% CI below zero).

	LLAMA-3.1-8B			MISTRAL3-7B			QWEN2.5-7B		
Direction	k=1	2	4	k=1	2	4	k=1	2	4
$-\hat{c}$ presence	$+\mathbf{24.1}$	$+6.3$	$-13.3^{\dagger}$	$+\mathbf{26.4}$	$+21.9$	$-11.4^{\dagger}$	$+\mathbf{17.1}$	-4.9	$-8.9^{\dagger}$
$+\hat{v}$ volatility	$+9.2$	$+14.4$	$+3.0$	$+24.3$	$+23.8$	$-7.2^{\dagger}$	$+3.4^{\dagger}$	-3.2	$-11.4^{\dagger}$
random	-1.1	-2.9	$-8.1^{\dagger}$	-0.7	-1.5	-4.4	-4.8	$-7.3^{\dagger}$	$-11.9^{\dagger}$
shuffled-label $\hat{v}$	-1.9	$-3.4^{\dagger}$	$-7.0^{\dagger}$	-1.0	-2.8	-7.2	-2.6	-2.0	$-3.0^{\dagger}$

We then vary only the gate, keeping checkpoint, layer, α, directions and thresholds fixed (Appendix A.7, Table 13). Ungated steering collapses SR. Presence alone fires almost equally on stable and volatile prompts (Llama $57/58\%$, Qwen $70/74\%$), so its gap falls below the dual gate's on every backbone. Volatility alone reaches a larger gap, but fires on 41–66% of grounded prompts and costs 14–36 pp of CO. Only the dual gate raises the gap over un-steered on all three backbones while keeping CO within 1.1 pp.

4.5 COMPARISON WITH RETRIEVAL-AUGMENTED BASELINES

We also compare with two adaptive-retrieval methods, FLARE (Jiang et al., 2023) and Self-RAG (Asai et al., 2024) (Appendix A.3). Neither follows the rule. FLARE adds little discrimination to the zero-shot model it runs on, and Self-RAG almost never defers. Both decide based on how uncertain the generation is, not on whether a fact has changed. BRIDGE discriminates far more than both on every backbone while keeping more stable answers.

5 ABLATION

Each experiment below varies a single factor: the steering direction (Section 5.1), the hyperparameters (Section 5.2) or the training seed (Section 5.3). Everything else is as in Section 4.2.

5.1 DIRECTION ABLATION

BRIDGE uses its two directions for different purposes: it reads $\hat{v}$ to decide when to fire, and steers along $-\hat{c}$ (Equation 3). To test whether the gain depends on the steering direction, we keep the checkpoint and hook layer fixed, remove the gate so that every prompt is steered, and steer along each of four directions: $-\hat{c}$, $+\hat{v}$, a random direction, and a shuffled-label $\hat{v}$. All four are applied at the same magnitude, $\alpha = k \cdot \text{gap}$ (Section 4.2), for three values of k. The shuffled-label $\hat{v}$ is obtained by refitting the volatility probe after randomly permuting the stable/volatile labels of its training prompts; it is fitted in the same way as $\hat{v}$ but carries no volatility information. The gate is removed so that any difference between rows comes from the direction alone. Table 4 reports the change in DD against the un-steered CA-AVMD checkpoint across all three backbones.

We observe that both controls lower DD on every backbone and at every magnitude. The random direction shows that the steering magnitude alone has no positive effect on DD. The shuffled-label $\hat{v}$ does not raise DD either, while $+\hat{v}$ does on LLAMA and MISTRAL; the effect of $\hat{v}$ therefore comes from the volatility information it encodes, not from the way it is fitted. Of the two probe directions, $-\hat{c}$ gives the largest DD gain on every backbone. $+\hat{v}$ gains less on LLAMA, slightly less on MISTRAL, and on QWEN its only gain comes with less accurate answers on stable questions (†). At the largest magnitude, almost every direction lowers DD, and most also make the answers less accurate. These results support the split in Equation 3. $\hat{c}$, the signal deferral is wired to (Section 3.2), is the most effective direction to steer along, while $\hat{v}$, the signal that separates volatile from stable facts, is better used to decide when to steer. The gate then applies steering only to prompts that are volatile and have no context, which avoids the degradation that ungated steering shows at large magnitudes (Tables 4 and 13).

Table 5: Hyperparameter sensitivity on LLAMA-3.1-8B. The τ_v rows are BRIDGE, with τ_v given as the share of stable prompts that pass the volatility condition; all other rows are the presence-only gate, each changing one factor from l=31, α=50, τ_c=−0.5. Matched: τ_c set so the gate fires at the reference rate. Fire: share of stable / volatile prompts steered.

Setting	Fire (S/V)	SR↑	SDR↓	VRR↑	DD↑	Setting	Fire (S/V)	SR↑	SDR↓	VRR↑	DD↑
α=10	29.9/22.1	48.3	24.3	47.6	23.3	l=16, matched	51.3/25.6	36.2	46.3	54.5	8.2
α=20	29.9/22.1	43.5	30.9	49.9	19.1	l=24, matched	29.7/23.3	42.4	31.6	52.8	21.2
$\alpha\geq$25	29.9/22.1	42.2	32.0	50.1	18.1	$l\leq$16, fixed τ_c	0.0/0.0	identical to un-steered			
τ_c=−0.7	26.6/19.4	44.4	29.4	47.4	18.0	τ_v=40%	24.2/25.7	46.3	25.8	51.9	26.1
τ_c=−0.3	34.0/24.7	40.7	35.5	53.5	18.0	τ_v=30%	19.0/23.5	49.3	20.8	48.8	**28.0**
un-steered	0.0/0.0	53.5	6.1	24.9	18.8	τ_v=20%	14.5/19.4	51.3	16.7	43.6	26.9

5.2 HYPERPARAMETER SENSITIVITY

To test whether the gain depends on the parameters choice, we vary one factor at a time on LLAMA: α, τ_c and the hook layer l for the presence-only gate, and τ_v for BRIDGE (Table 5 with the extented results in A.9). The presence-only gate at α=20 is the +presence row of Table 3.

We observe that no setting of the presence-only gate reaches the DD of BRIDGE. A larger α does not help, changing τ_c only changes how often the gate fires, and DD is highest at layers 24–31, which supports our choice of l. The presence-only gate also fires more often on stable than on volatile prompts, while BRIDGE does the opposite at every τ_v. Varying τ_v trades SR against VRR, while DD and CO stay close to their deployed values. These results show that the gain of BRIDGE comes from the volatility condition, not from tuning.

5.3 TRAINING-SEED VARIANCE

The results so far come from one training run per variant. We retrain LLAMA-3.1-8B and MISTRAL3-7B under three seeds and apply BRIDGE with the same gate settings (Appendix A.11). VRR and SDR vary across seeds, but BRIDGE raises DD over CA-AVMD within every seed on both backbones (37.3±5.7 vs. 29.0±9.4 on LLAMA, 38.8±4.3 vs. 24.8±6.0 on MISTRAL), at a small cost in SR (49.9 vs. 55.2 and 53.2 vs. 55.0; Table 18).

6 SCOPE AND LIMITATIONS

Scope. The EvoWiki experiments establish the results on a Wikipedia-derived benchmark where the base model carries parametric memory of the studied facts. Therefore, we ask ourselves a complementary question: what happens when the model has no parametric prior over the domain? This is the common production scenario: an organisation fine-tunes a public SLM on proprietary data such as product catalogues, internal documentation, pricing tables for which the base model has zero or minimal pre-training exposure. To establish a baseline for how much of each dataset the model holds in memory, we compare the token-F1 of Llama on the stable questions of the two datasets, posed in a bare prompt $(\emptyset, q_i)$. On the retail dataset, the zero-shot model reaches a token-F1 of 2.4% and the SFT model 18.1%; on EvoWiki, the two models reach 10.1% and 59.5%. The gap reflects the lack of pre-training exposure to proprietary product data. Additional metrics about this finding are detailed in A.10.

After finetuning, CA-AVMD achieves 99.5% VRR on retail volatile queries without context (comparable to its EvoWiki behavior) while maintaining 94.3% contextual obedience (CO) when a RAG passage is provided (A.10, Table 17). A-VMD behaves the same way on EvoWiki: its near-ceiling refusal comes from deferring on a small, learnable tag set instead of on EvoWiki's open entity set. Behavioral supervision works well when there are few volatile categories and no memorized facts.

Limitations. Our study has several limitations that bound the scope of our conclusions. First, all experiments are conducted on small language models in the 7–8B parameter range, consistent with the production inference constraints that motivate this work. Second, our evaluation protocol relies on answer-token contains-match, which is appropriate for closed-form factual queries but does

not capture multi-step reasoning or open-ended generation quality. Third, we test only supervised objectives. RL with abstention-aware rewards improves temporal abstention in-distribution (Zhou et al., 2026), but remains a future goal.

7 Related Work

Knowledge Conflicts & Temporal Staleness. Knowledge conflicts in LMs are broadly categorized into context–memory, inter-context, and intra-memory conflicts (Xu et al., 2024). We focus on context–memory conflicts, specifically whether models recognize when parametric memory is outdated. Benchmarks like FreshQA (Vu et al., 2024), TempLAMA (Dhingra et al., 2022), and EvoWiki (Tang et al., 2025) track temporal updates. While recent work explores RL for temporal abstention (Zhou et al., 2026), it focuses on unanswerability given context; BRIDGE addresses deferral when context is absent for volatile facts. On the mechanism side, instruction fine-tuning can exacerbate context reliance over memory (Longpre et al., 2021; Goyal et al., 2025), motivating our context-augmented training targets. Adaptive retrieval decides per query whether to retrieve: Self-RAG (Asai et al., 2024) trains reflection tokens that judge relevance, and FLARE (Jiang et al., 2023) retrieves when generation confidence is low. Neither conditions on whether a fact has changed, and on our benchmark neither separates stable from volatile facts (Appendix A.3).

Conditional Steering & Representation Gaps. Activation steering modifies latent states at inference (Zou et al., 2023a; Turner et al., 2023; Rimsky et al., 2024). Conditional variants like CAST (Lee et al., 2025), CtrlA (Huanshuo et al., 2025), Knowledge Knows (Xiros et al., 2026), and SHIFT (Li et al., 2026) gate steering on input categories, dual probes, or learned modules. Similarly, knowledge selection methods edit SAE features (Zhao et al., 2025), prune attention heads (Yu et al., 2023; Jin et al., 2024), or steer toward fresh answers (Hossain et al., 2026). Crucially, probing studies show a disconnect between represented properties and downstream behavior (Elazar et al., 2021; Li et al., 2023; Orgad et al., 2025; Wang et al., 2026). Self-knowledge signals such as entity familiarity directly drive behavior (Ferrando et al., 2025; Shi et al., 2026), whereas volatility, we find, is decodable but unused.

8 Conclusion

In this work, we showed that small language models suffer from a fundamental representation-policy gap when managing evolving knowledge: while temporal volatility and context presence are linearly decodable in model activations, fine-tuning and prompting fail to wire volatility to the deferral decision. To resolve this, we introduced BRIDGE, a dual-gate conditional steering mechanism applied at inference time. By gating on both context presence and internalized volatility, BRIDGE successfully enforces selective abstention on volatile, ungrounded queries at a small cost in stable retention and without degrading contextual obedience. Our findings highlight a broader structural principle for representation engineering: world-properties that remain unwired during training can be effectively routed to causal decision channels at inference time, offering a controllable paradigm for deploying reliable language models under shifting parametric knowledge.

AI Use Statement

We used large language models (LLMs) as assistants in three ways: (i) editing the text of the paper for clarity and structure; (ii) searching for and summarising related work, with all cited works checked by the authors against the original sources; and (iii) writing and debugging code for the training, steering and evaluation pipeline. The authors verified all text, citations, code and reported numbers, and take full responsibility for the content of this paper.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2024.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Patel, and William W. Cohen. Time-aware language models as temporal knowledge bases. In *Transactions of the Association for Computational Linguistics*, volume 10, pp. 257–273, 2022.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

Javier Ferrando, Oscar Obeso, Senthooran Rajamanoharan, and Neel Nanda. Do i know this entity? knowledge awareness and hallucinations in language models. In *International Conference on Learning Representations (ICLR)*, 2025. Oral.

Sachin Goyal, Christina Baek, Zico Kolter, and Aditi Raghunathan. Context-parametric inversion: Why instruction finetuning may not actually improve context reliance. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 68067–68099, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/aa27ac7aca4e462da1439b43ceebc04c-Paper-Conference.pdf.

Elias Hossain, Sourav Saha, Umesh Chandra Biswas, and Sanjeda Sara Jennifer. Right knowledge, wrong answer: Test-time steering for temporal fact conflicts in open-weight language models. *arXiv preprint arXiv:2606.20959*, 2026.

Liu Huanshuo, Hao Zhang, Zhijiang Guo, Jing Wang, Kuicai Dong, Xiangyang Li, Yi Quan Lee, Cong Zhang, and Yong Liu. CtrlA: Adaptive retrieval-augmented generation via inherent control. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12592–12618, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.652. URL https://aclanthology.org/2025.findings-acl.652/.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of EMNLP*, 2023.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1193–1215, 2024.

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*, volume 2025, pp. 90960–90985, 2025.

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 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ruochang Li, Pengcheng Huang, Zhenghao Liu, Yukun Yan, Huiyuan Xie, Yu Gu, Ge Yu, and Maosong Sun. SHIFT: Gate-Modulated Activation Steering for Knowledge Conflict Mitigation in Retrieval-Augmented Generation. *arXiv e-prints*, art. arXiv:2606.27786, June 2026. doi: 10.48550/arXiv.2606.27786.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of EMNLP*, 2021.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations (ICLR)*, 2025.

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)*, pp. 15504–15522, 2024.

Zeyu Shi, Ziming Wang, Tianyu Chen, Shiqi Gao, Haoyi Zhou, Qingyun Sun, and Jianxin Li. Finetuned llms know they don't know: A parameter-efficient approach to recovering honesty. 40(39): 32983–32991, 2026.

Wei Tang, Yixin Cao, Yang Deng, Jiahao Ying, Bo Wang, Yizhe Yang, Yuyue Zhao, Qi Zhang, Xuanjing Huang, Yu-Gang Jiang, and Yong Liao. EvoWiki: Evaluating LLMs on evolving knowledge. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 948–964. Association for Computational Linguistics, jul 2025.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. FreshLLMs: Refreshing large language models with search engine augmentation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13697–13720, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.813. URL https://aclanthology.org/2024.findings-acl.813/.

Yian Wang, Yuen Chen, Agam Goyal, and Hari Sundaram. CausalDetox: Causal head selection and intervention for language model detoxification. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11893–11914, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.577. URL https://aclanthology.org/2026.findings-acl.577/.

Nikolaos Xiros, Maria-Eleni Zoumpoulidi, and Georgios Paraskevopoulos. Knowledge knows, verbalization tells: Disentangling latent directions for mathematical solvability in llms. *arXiv preprint arXiv:2607.05013*, 2026.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for LLMs: A survey. *arXiv preprint arXiv:2403.08319*, 2024.

Qinan Yu, Jack Merullo, and Ellie Pavlick. Characterizing mechanisms for factual recall in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9924–9959, 2023.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. Steering knowledge selection behaviours in llms via sae-based representation engineering. pp. 5117–5136, 2025.

Xinyu Zhou, Chang Jin, Carsten Eickhoff, Zhijiang Guo, and Seyed Ali Bahrainian. When silence is golden: Can LLMs learn to abstain in temporal QA and beyond? In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PhUCxfS0yf.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023a.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023b.

A Appendix

A.1 Detailed Training Objectives Definition

This section gives the loss of each of the five training objectives of Section 2.2. They differ only in the target applied to volatile facts; all five are cross-entropy fine-tuning with LoRA.

SFT: Standard supervised fine-tuning applies cross-entropy on all response tokens, with the T_1 answer as target for both stable and volatile examples:

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_{i\in\mathcal{D}} \log p_\theta\big(y_i^{T_1} \,\big|\, q_i\big). \tag{4}$$

VMD: Volatile Memory Distillation removes the loss on volatile answer tokens, i.e. Equation 4 is restricted to the stable partition, which removes the pressure to memorize values that change:

$$\mathcal{L}_{\mathrm{VMD}} = -\sum_{i\in\mathcal{D}_s} \log p_\theta\big(y_i^{T_1} \,\big|\, q_i\big). \tag{5}$$

A-VMD: Active Volatile Memory Distillation keeps the stable loss but replaces each volatile answer with a fixed deferral string $d =$ *"I don't have reliable information about this."*, teaching abstention:

$$\mathcal{L}_{\text{A-VMD}} = -\sum_{i\in\mathcal{D}_s} \log p_\theta\big(y_i^{T_1} \,\big|\, q_i\big) - \sum_{i\in\mathcal{D}_v} \log p_\theta\big(d \,\big|\, q_i\big). \tag{6}$$

Both VMD and A-VMD are unconditional: volatile queries are treated identically regardless of whether retrieved context is available at inference.

CA-AVMD: Context Augmented VMD makes the objective context conditioned - with context, the model is trained to extract y^{T_2}; without context, it either receives no volatile loss (CA-VMD) or is trained to defer (CA-AVMD). This separates "knowing a fact is volatile" from routing based on context availability. Training alternates between a grounded form $(c_i, q_i) \mapsto y_i^{T_2}$ and a bare form $(\emptyset, q_i) \mapsto d$:

$$\mathcal{L}_{\text{CA-AVMD}} = \mathcal{L}_{\mathrm{stable}} + \mathcal{L}_{\mathrm{ctx}} + \mathcal{L}_{\mathrm{defer}}, \tag{7}$$

$$\mathcal{L}_{\mathrm{stable}} = -\sum_{i\in\mathcal{D}_s} \log p_\theta\big(y_i^{T_1} \mid q_i\big);\ \mathcal{L}_{\mathrm{ctx}} = -\sum_{i\in\mathcal{D}_v^{\mathrm{ctx}}} \log p_\theta\big(y_i^{T_2} \mid q_i, c_i\big);\ \mathcal{L}_{\mathrm{defer}} = -\sum_{i\in\mathcal{D}_v^{\mathrm{bare}}} \log p_\theta\big(d \mid q_i\big) \tag{8}$$

The first term in Equation 8 preserves stable knowledge, the second trains extraction from context, and the third trains abstention when context is absent.

A.2 EvoWiki Dataset

EvoWiki Tang et al. (2025) labels facts from dated Wikidata and Wikipedia dumps as stable (value unchanged from September 2021 to May 2024) or evolved (value changed between January and May 2024). We take its labels as they come from observed revisions rather than from a heuristic. We call evolved facts volatile and write T1 and T2 for the January and May 2024 dumps.

Table 6: EvoWiki splits used in this work: each entity maps to exactly one question (6,180 unique entities total: 2,689 stable, 3,491 evolved). Semi-volatile entities are a subset of the evolved evaluation pool, held out from training in all conditions (†).

Split	Class	Entities	Role
Train	Stable	2,151	Schema + stable parametric memory
	Volatile	2,792	Context-routing training signal
Eval	Stable	538	Retention probe ($y^{T_1} = y^{T_2}$ always)
	Volatile	699	Hallucination and obedience probe
	Semi-volatile†	450	Generalization probe (unseen entities)

A.3 RETRIEVAL-AUGMENTED BASELINES

Self-RAG (Asai et al., 2024) and FLARE (Jiang et al., 2023) are the two adaptive-retrieval systems closest to our setting: both decide during generation whether the model's own output needs retrieved support. For Self-RAG, we use the publicly available `selfrag_llama2_7b` checkpoint[2], which is built on Llama-2-7B and trained with GPT-4-generated reflection tokens on Wikipedia data; no checkpoint exists for our backbones. FLARE is an inference-time procedure that applies to any autoregressive model, so we run it on the zero-shot versions of our three backbones with the authors' implementation[3] and its recommended hyperparameters (filter threshold 0.5, mask threshold 0.4, at most 5 iterations). Both are scored with the protocol of Section 2.1, on the same prompts and with the same deferral detector as every other row.

Table 7: Retrieval-augmented baselines under the protocol of Section 2.1. Zero-shot, CA-AVMD and BRIDGE rows are repeated from Table 3. †Self-RAG is released only on Llama-2-7B.

Backbone	Method	SR↑	SDR↓	VRR↑	DD↑	CO↑
LLAMA-3.1-8B	Zero-shot	12.3	71.4	72.5	1.2	57.8
	FLARE	12.3	71.4	72.7	1.3	57.8
	CA-AVMD	53.5	6.1	24.9	18.8	62.7
	+ BRIDGE	49.3	20.8	48.8	**28.0**	62.7
MISTRAL3-7B	Zero-shot	31.4	10.4	16.0	5.6	51.1
	FLARE	33.3	8.7	15.0	6.3	51.1
	CA-AVMD	55.6	3.0	31.6	28.6	62.0
	+ BRIDGE	50.6	13.6	60.2	**46.7**	61.8
QWEN2.5-7B	Zero-shot	21.9	2.8	3.6	0.8	58.0
	FLARE	24.0	7.1	14.2	7.1	58.0
	CA-AVMD	43.7	3.0	15.6	12.6	54.9
	+ BRIDGE	40.1	16.5	44.2	**27.7**	53.8
LLAMA-2-7B	Self-RAG†	20.8	0.6	1.7	1.2	46.7

Neither baseline follows the rule. FLARE behaves almost exactly like the zero-shot model it runs on: CO is identical on every backbone, SR changes by at most 2.1 pp, and DD stays within 1 pp of zero-shot on LLAMA and MISTRAL. On LLAMA it keeps the zero-shot model's indiscriminate deferral (SDR 71.4, VRR 72.7), and on MISTRAL it rarely defers on either side. Its retrieval decisions therefore leave the deferral decision where it was. Self-RAG almost never defers (SDR 0.6, VRR 1.7) and has the lowest CO (46.7), although its older backbone makes it only a loose comparison. By contrast, CA-AVMD alone already exceeds both baselines in DD (12.6–28.6) and SR (43.7–55.6), and BRIDGE raises DD further (27.7–46.7). FLARE keeps a higher CO than CA-AVMD only on QWEN (58.0 against 54.9). Both baselines act on how uncertain the generation is, not on whether the fact has changed, and on EvoWiki the two do not coincide.

[2] `https://huggingface.co/selfrag/selfrag_llama2_7b`
[3] `https://github.com/jzbjyb/FLARE`

Table 8: Four-step mechanistic protocol applied to CA-AVMD across three model families. Probe: peak balanced accuracy of the linear volatility probe and peak layer ℓ_p. Patching: peak residual-stream path recovery $\mathrm{PR}(\ell^\star)$ and peak layer $\ell^\star$ (excluding the definitional endpoint $\ell = L$; over $7/64/49$ valid volatile/stable pairs on Llama/Mistral/Qwen). Ablation: targeted ΔVRR, reported in two mirrored blocks for VOLATILE and STABLE questions, each with the random-control mean $\pm\sigma$ (n=3 seeds), the null-corrected effect $\tilde{\Delta} = \Delta_{\text{targeted}} - \overline{\Delta}_{\text{random}}$, and the effect-size z-score $|z| = |\tilde{\Delta}|/\sigma$; ** denotes $|z| \geq 2.5$. *Sign convention*, used in this table and in Fig. 1a alike: $\Delta\mathrm{VRR} = \mathrm{VRR}_{\text{ablated}} - \mathrm{VRR}_{\text{baseline}}$, so a positive Δ means the ablation *adds* deferral. Probe survival: $\max_\ell |\Delta_{\text{acc}}(\ell)|$ between baseline and ablated volatility probe across layers. †Llama's stable random-control σ sits at the eval's resolution floor ($1/150 = .007$), so that $|z|$ is bounded by the measurement grid rather than by the data: read it as far outside the null band, not as a calibrated effect size.

	Probe		**Patching**		**Ablation** (ΔVRR, volatile)				**Ablation** (ΔVRR, stable)				**Survival**
	ℓ_p	Acc	$\ell^\star$	$\mathrm{PR}(\ell^\star)$	Targeted	Random $\pm\sigma$	$\tilde{\Delta}$	$\lvert z\rvert$	Targeted	Random $\pm\sigma$	$\tilde{\Delta}$	$\lvert z\rvert$	$\max\lvert\Delta\rvert$
Llama-3.1-8B	14	.807	15	1.34	$+.160$	$+.020 \pm .047$	$+.140$	3.0**	$+.087$	$+.013 \pm .007$	$+.073$	11.0**†	.033
Mistral3-7B	18	.798	15	1.48	$-.080$	$-.071 \pm .108$	$-.009$	< 1 n.s.	$+.000$	$-.004 \pm .033$	$+.004$	< 1 n.s.	.015
Qwen2.5-7B	19	.770	20	2.53	$+.567$	$+.080 \pm .103$	$+.487$	4.7**	$+.333$	$+.067 \pm .092$	$+.267$	2.9**	.047

A.4 BRIDGE: IMPLEMENTATION DETAILS

The hook of Equation 3 is registered as a PyTorch forward hook on the target layer. At prefill (sequence length > 1), both gate projections are evaluated; if both fire, $-\alpha\hat{\mathbf{c}}$ is added to the last-position hidden state. All direction vectors are ℓ_2-normalised before use.

Thresholds τ_v and τ_c are fitted per backbone on a held-out split of the EvoWiki evaluation set as the midpoint between the mean projections of the two classes (stable vs. volatile for τ_v; no-context vs. grounded-context for τ_c); Mistral3-7B and Qwen2.5-7B use these fitted values. For Llama-3.1-8B the presence-only row of Table 3 uses a *manually set*, conservative $\tau_c = -0.5$. Holding the injection, magnitude, hook layer and checkpoint fixed and adding only the volatility AND-condition (Table 3) improves deferral discrimination on all three backbones. The fitted midpoint for Llama-3.1-8B is positive ($\approx +3$ at its hook layer) and fires on every no-context query, including stable-retention ones, which collapses stable retention; -0.5 fires on 33.8% of queries and leaves stable retention intact. The injection magnitude α is an *absolute* quantity, so it is not comparable across backbones whose residual streams differ in scale. We therefore also report it in class-separation units,

$$k = \frac{\alpha}{\text{gap}}, \qquad \text{gap} = \left\|\mu_v - \mu_s\right\|, \tag{9}$$

where μ_v and μ_s are the class means of the volatile and stable activations at the hook layer. At $\alpha = 20$ the same nominal magnitude is $k = 2.58$ on Llama-3.1-8B, 4.57 on Mistral3-7B and 0.31 on Qwen2.5-7B, whose mean residual norm at its hook layer is 638.1 against 60.6 and 28.9 — a 3.1% perturbation against 33% and 69%. A shared α is thus not a shared intervention, and Qwen2.5-7B is steered at the matched dose $\alpha = 168.5$ ($k = 2.58$) in Table 3; $\alpha = 20$ is used elsewhere. Hook layers are the final block for Llama-3.1-8B and Mistral3-7B (layer 31). Hook and probe layer are matched for Llama-3.1-8B (31); for Mistral3-7B the gate reads the layer-16 probe directions at the layer-31 hook. Qwen2.5-7B has 28 blocks, so its final block is layer 27; Tables 3 and 4 hook layer 27 at the matched dose of Equation 9. The two are therefore different operating points of the same method and should not be differenced against each other.

A.5 MECHANISTIC DIAGNOSIS: PER-BACKBONE RESULTS

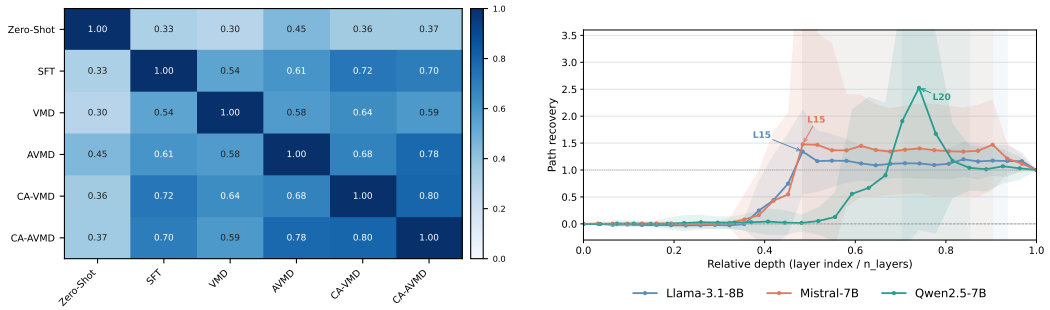
Probing. A logistic probe on the residual stream separates stable from volatile questions already in the zero-shot model, and fine-tuning does not raise its accuracy. It rotates the direction instead: the trained variants agree with each other at cosine 0.54–0.80, but with the zero-shot model only at 0.30–0.45 (Figure 3a).

Patching and ablation. For matched volatile/stable pairs, we copy the residual stream at one layer from one run into the other and measure path recovery. It exceeds 1.0 from layer 15 (19 on Qwen),

Table 9: Pushing presence and volatility on LLAMA CA-AVMD, at equal size $k \cdot \text{gap}$ (hook layer 31). Sel. is $\Delta\text{VRR}/\Delta\text{SDR}$; 1.0 is indiscriminate deferral. †: damaged model (Table 10).

Direction	k	**VRR↑**	**SDR↓**	**VRR−SDR↑**	**SR↑**	**CO↑**	**Sel.↑**
un-steered (k=0)	0	24.9	6.1	18.8	53.5	62.7	–
	0.5	47.5	14.9	32.6	52.0	61.6	2.59
	1	72.4	29.6	**42.8**	47.8	60.4	2.03
$\hat{c}$ presence	2	92.6	66.7	25.8	22.9	53.8	1.12
	4†	97.1	90.5	6.6	0.4	16.0	0.86
	8†	97.1	90.7	6.4	0.4	0.9	0.85
	0.5	30.5	7.4	23.0	53.5	62.7	**4.29**
	1	38.3	10.4	27.9	53.5	61.8	3.15
$\hat{v}$ volatility	2	53.9	20.8	33.1	48.3	60.7	1.98
	4	72.5	50.0	22.5	28.1	53.1	1.09
	8†	83.5	79.0	4.6	4.3	31.6	0.81

peaks at mid-depth and stays above 1.0 to the output (Figure 3b), so the distinction is carried forward rather than recomputed. The five attention heads and three MLPs with the highest recovery are the components ablated in Figure 1. We compare them with the same number of random components over three seeds (Table 8).



(a) Probe-direction similarity (LLAMA). (b) Activation patching (CA-AVMD).

Figure 3: Probing and patching. Path recovery at the last layer is 1.0 by definition.

Presence versus volatility. On Llama CA-AVMD, we add a vector of length $k \cdot \text{gap}$ along $\hat{v}$ or $\hat{c}$ at the last prompt token of the hook layer, where $\text{gap} = 7.76$ is the distance between the mean volatile and mean stable activation, so both directions get the same push at equal k (Table 9). At $k \leq 2$, $\hat{c}$ moves volatile deferral 2.3–4.0× further than $\hat{v}$, but also raises deferral on stable questions, and at large k it removes context use. $\hat{v}$ moves deferral less but more selectively. The two directions are nearly orthogonal ($\cos = -0.015$).

A.6 STABLE RETENTION UNDER STEERING: DEFERRALS VERSUS ERRORS

A drop in stable retention (SR) has two possible sources: the model declines stable questions it used to answer, which SDR already counts, or it answers them wrongly. Only the second means steering has changed what the model knows. To separate them we introduce *answered accuracy*,

$$\text{Acc}_{\text{ans}} = \frac{\text{SR}}{100 - \text{SDR}},$$

the share of correct answers among the stable questions the model does not decline. It stays flat if steering only adds deferrals spread evenly over right and wrong answers, rises if the new deferrals fall mostly on questions the model would have answered wrongly, and falls if steering changes the answers themselves.

Table 10: Decomposition of the change in stable retention on LLAMA, ungated. First row: the un-steered CA-AVMD checkpoint, absolute values. Other rows: change against it, in percentage points. $\mathrm{Acc}_{\mathrm{ans}}$ is accuracy on the stable questions the model still answers, with 95% CI. †: the whole interval is below zero.

Direction	k	ΔSR	ΔSDR	$\Delta\mathrm{Acc}_{\mathrm{ans}}$ [95% CI]
CA-AVMD, un-steered: 53.5	6.1	57.0		
$\hat{v}$ volatility	2	− 5.2	+14.7	+ 4.0 [+1.1, +6.9]
	4	−25.5	+43.9	− 0.9 [−6.3, +4.6]
	8	−49.3	+72.9	−36.7† [−44.2, −28.6]
random	4	− 5.4	− 3.2	− 7.4† [−10.3, −4.5]
	8	−12.1	− 5.8	−15.4† [−19.3, −11.7]
shuffled-label $\hat{v}$	2	− 1.7	− 0.7	− 2.2† [−4.3, −0.2]
	4	− 6.3	− 1.3	− 7.4† [−10.4, −4.5]

Table 11: **Steering dose per backbone.** α is an absolute magnitude; $k = \alpha/\mathrm{gap}$ (Eq. 9) is not. At $\alpha = 20$ the QWEN hook is a 3.1% perturbation, so it is re-dosed to LLAMA's k.

					At $\alpha = 20$		As reported	
Backbone	ℓ	gap	$\overline{\|\mathbf{h}\|}$	d	k	$\frac{\|\mathrm{inj}\|}{\|\mathbf{h}\|}$	α	k
LLAMA-3.1-8B	31	7.76	60.6	0.74	2.58	33.0%	20	2.58
MISTRAL3-7B	31	4.38	28.9	0.99	4.56	69.1%	20	4.56
QWEN2.5-7B	27	65.39	638.1	0.61	**0.31**	**3.1**%	168.5	2.58

Pushing $\hat{v}$ at k=2 lowers SR by 5.2 pp while SDR rises by 14.7 pp, and answered accuracy rises by 4.0 pp: the questions the model now declines are mostly ones it would have answered wrongly. At k=4 only 269 stable questions are still answered and the interval is too wide to separate a drop from none, so we make no claim beyond k=2. At k=8 answered accuracy falls by 36.7 pp. The two controls behave differently at every dose: random and shuffled-label directions lower SR without adding deferral, so the lost answers turn wrong, and answered accuracy falls in every row.

Steering the injection across backbones. Equation 3 subtracts a vector of fixed norm α, so α is an absolute quantity, and the three residual streams are not on one scale: the mean activation norm at the hook layer is 60.6, 28.9 and 638.1. We therefore also report the dose in class-separation units $k = \alpha/\mathrm{gap}$ (Eq. 9), with $\mathrm{gap} = \|\mu_v - \mu_s\|$ measured at the hook layer on the same `bare_qa` template the evaluation uses, so that k and τ_v are on the evaluation's scale. Table 11 gives both. The published $\alpha = 20$ is $k = 2.58$ on LLAMA-3.1-8B and $k = 4.56$ on MISTRAL3-7B — 33% and 69% of the residual norm — but $k = 0.31$ on QWEN2.5-7B, a 3.1% perturbation, below the smallest dose in the sweep of Table 9 ($k = 0.5$). It is under-dosed rather than null: across a 12-cell threshold grid at $\alpha = 20$ the QWEN hook fires on 4.3–21.7% of items while stable retention moves −0.4 pp in *every* cell. An intervention that fires on a fifth of the evaluation and changes nothing is inert. QWEN2.5-7B is therefore steered at $\alpha = 168.5$, the magnitude matching LLAMA-3.1-8B's k; the other two backbones are left at the published $\alpha = 20$. A shared α is not a shared intervention, and cross-backbone steering results quoted at one nominal α should be read accordingly.

A.7 GATE STATISTICS: DIRECTIONS, THRESHOLDS, AND ACCOUNTING

Is the presence direction volatility-informative? Table 12 treats each projection as a one-dimensional classifier of gold volatility, restricted to the context-absent column (the only cell where the gate is active). Intervals are bias-corrected bootstrap percentile CIs over $B = 10{,}000$ resamples of the evaluation set.

Thresholds. Raw τ_v is not comparable across backbones. At matched relative depth the residual streams still differ by family: mean $\|\mathbf{h}\|$ at the hook layer is 638.1 on Qwen2.5-7B against 60.6

Backbone	presence $\hat{c}^\top h$	volatility $\hat{v}^\top h$
Llama-3.1-8B	0.518 [0.485, 0.551]	0.708 [0.677, 0.736]
Qwen2.5-7B	0.539 [0.507, 0.572]	0.671 [0.638, 0.699]
Mistral-7B-v0.3	0.739 [0.711, 0.765]	0.764 [0.735, 0.789]

Table 12: AUC for predicting gold volatility from each projection, context-absent column only, with 95% bootstrap CIs. On Llama and Qwen the presence interval covers chance, so no threshold on $\hat{c}^\top h$ could separate the two rows. On Mistral it does not: the presence direction is nearly as volatility-informative as $\hat{v}$ itself, which is why the dual gate helps that backbone for a different reason (Table 13).

Table 13: Gate-condition ablation on all three backbones. Within each backbone, every row uses the same checkpoint, hook layer, α, directions and thresholds; only the gate condition changes. *Fire* is the percentage of prompts on which the hook injects: stable (s) and volatile (v) prompts without context, and grounded prompts with retrieved context (g). Volatility only is $\hat{v}^\top h > \tau_v$, presence only is $\hat{c}^\top h < \tau_c$, and dual is their AND (BRIDGE).

	fire (%)							
Gate	s	v	g	SR↑	SDR↓	VRR↑	VRR−SDR↑	CO↑
LLAMA-3.1-8B	$\ell = 31, \alpha = 20, \tau_v = +4.269, \tau_c = +0.5$							
un-steered	—	—	—	53.5	6.1	24.9	18.8	62.7
ungated	100	100	100	9.3	83.8	96.0	12.2	44.4
volatility only	30	66	66	43.5	29.2	65.8	36.6	48.7
presence only	57	58	0	30.5	52.0	66.1	14.0	62.7
dual	19	39	0	49.3	20.8	48.8	28.0	62.7
MISTRAL3-7B	$\ell = 31, \alpha = 20, \tau_v = +2.645, \tau_c = +0.5$							
un-steered	—	—	—	55.6	3.0	31.6	28.6	62.0
ungated	100	100	100	1.5	78.3	95.3	17.0	0.9
volatility only	30	76	60	45.9	23.2	73.0	49.7	25.6
presence only	46	78	0	29.9	41.3	79.0	37.7	61.8
dual	16	58	0	50.6	13.6	60.2	46.7	61.8
QWEN2.5-7B	$\ell = 27, \alpha = 168.5, \tau_v = +111.4, \tau_c = +19.16$							
un-steered	—	—	—	43.7	3.0	15.6	12.6	54.9
ungated	100	100	100	0.9	91.4	95.6	4.1	21.6
volatility only	30	56	41	35.1	26.6	57.1	30.5	41.1
presence only	70	74	5	14.5	67.7	74.8	7.2	51.6
dual	17	40	1	40.1	16.5	44.2	27.7	53.8

on Llama-3.1-8B and 28.9 on Mistral-7B-v0.3, and the spread of the volatility projection follows it (112.0, 11.2, 4.9), so the threshold that means *the top third of stable prompts* is a different number on each. We therefore quote thresholds in the main text as *pass fractions*(the measured proportion of stable and volatile prompts clearing a condition)and give raw values only in Table 14, computed from the same projection cache the hook reads. Read that way the three backbones are at one operating point: every one sits at the 30% stable-pass rung of the selection grid (Paragraph 4.2), though τ_v ranges from $+2.645$ to $+111.400$. The offline cache reproduces the deployed hook rather than approximating it: the dual-fire rate on stable prompts matches the hook's own test-1 firing rate to within 0.2 pp on all three backbones (18.8 vs. 19.0, 16.5 vs. 16.5, 16.7 vs. 16.7), the residual being device-placement numerics and not a calibration gap.

How the grounded column is obtained. The two conjuncts are evaluated independently and each blocked prompt is logged with the set of conjuncts that failed. A prompt on which the presence conjunct alone blocks firing is, by definition, one on which a volatility-only gate would have fired; the grounded fire rate for the volatility-only configuration is therefore recoverable from the block reasons of a single dual-gate run, with no separate ablation run and no threshold refitting. The presence-only and dual rows of Table 13 are read off the same log symmetrically. All three configurations in that table share one set of directions and one set of thresholds, so every difference between rows is attributable to the gate condition alone.

				gate 1 pass		gate 2 pass		dual fire	
Backbone	hook	τ_v	τ_c	stab.	vol.	stab.	vol.	stab.	vol.
Llama-3.1-8B	31	$+4.269$	$+0.50$	30.1%	65.8%	56.5%	58.2%	18.8%	38.6%
Mistral-7B-v0.3	31	$+2.645$	$+0.50$	30.1%	76.4%	46.3%	78.1%	16.5%	58.4%
Qwen2.5-7B	27	$+111.400$	$+19.16$	29.9%	55.9%	69.9%	74.4%	16.7%	39.5%

Table 14: Deployed gate thresholds, and what each condition passes on its own. Gate 1 is the volatility condition ($\hat{v}^\top h > \tau_v$), gate 2 the presence condition ($\hat{c}^\top h < \tau_c$), and *dual fire* their AND: the fraction of prompts on which the hook actually injects. All fractions are measured on the context-absent column ($n = 538$ stable, 699 volatile) from the same projection cache the hook reads.

A.8 Comparison with learned read-outs of the hidden state

The gate decides deferral from two fixed probe directions and two thresholds. But in order to understand if we can achieve the same result with fiting the decision directly, we have trained a classifier on h at the hook layer and let it predict "defer" end to end. We ran that comparison on LLAMA-3.1-8B (CA-AVMD checkpoint, hook layer 31), scoring each mechanism as a binary decision: context-absent prompts where deferral is correct ($n = 699$) and grounded prompts where it is not ($n = 450$). The two learned read-outs replace the output with the deferral string rather than steering in-stream, so this is a comparison of decision quality, not of intervention style.

Mechanism	Acc	Prec	Rec	F1	grounded false-defer
Presence threshold ($\hat{c}^\top h < \tau_c$)	0.63	1.00	0.39	0.56	0.0% (0/450)
Dual gate ($\tau_v \wedge \tau_c$)	0.60	1.00	0.34	0.51	0.0% (0/450)
Logistic on h	0.88	0.99	0.81	0.90	0.7% (3/450)
MLP on h	0.89	1.00	0.82	0.90	0.2% (1/450)

Table 15: Deferral decision quality on Test 4, LLAMA-3.1-8B. Learned read-outs recover far more of the available deferrals than the fixed gates, and pay for it with a non-zero false-defer rate on grounded inputs.

Two things follow. First, a plain logistic read-out of h decides deferral at 0.88 accuracy on a checkpoint whose own uninstructed deferral rate is 24.9%. The gap between what the residual stream supports and what the model does is therefore not an artefact of representation engineering: a second mechanism, with no in-stream intervention at all, measures the same gap. This is the clearest single piece of evidence for the diagnosis in Section 3.2, and it is independent of every design choice in the hook.

Second, the two families sit at different points of the same trade-off rather than in a ranking. The learned read-outs buy their recall by beginning to override grounded inputs and three and one false deferrals respectively on prompts where the answer was present in context, while both fixed gates hold at exactly zero. For the policy this paper installs, a false deferral on a grounded prompt is the expensive error: it discards retrieved evidence the system paid to fetch, and it is invisible to any metric computed on the context-absent column alone. We therefore report the learned read-outs as the recall ceiling available from this hidden state, not as a baseline the gate underperforms. A method that wanted the higher recall could adopt one at the cost of the $\Delta\mathrm{CO} = 0$ guarantee; the choice depends on the relative price of the two errors in the deployment, which we do not assume.

A.9 Gate fire-rate sensitivity

Protocol. All runs use the LLAMA-3.1-8B CA-AVMD checkpoint and the presence-only gate (the +presence row of Table 3). The hook logs every call. The fire rate is the number of steered calls over all calls across tests 1–4; we also report it per test. We fingerprint the test-4 generations with an MD5 digest, so equal digests mean byte-identical decodes. Of the 37 configurations, 19 give distinct steered decodes, 13 give the un-steered test-4 decode byte for byte, and 5 repeat the ℓ=31, α=50, τ_c=-0.5 decode (Table 16). At α=20 the sweep reproduces the row of Table 3 to within 0.3 pp (VRR 49.9 vs. 50.1, SR 43.5 vs. 43.7), within device-placement numerics.

A gate that never fires looks like a robust one. With $\tau_c{=}{-}0.5$ at layer 16 the gate fires on 0.0% of prompts: $\hat{c}^\top h$ never drops below the cut at that depth. Every α from 10 to 100 and every τ_c from -0.7 to -0.3 then returns the un-steered decode byte for byte. Read without the fire rate, these ten rows describe a method insensitive to its settings; they measure nothing. The cause is scale: τ_c is a raw cut on $\hat{c}^\top h$, and the class-mean midpoint grows from $+0.96$ at layer 12 to $+3.15$ at layer 31, so a cut that is on-scale at one layer is off-scale at another. We therefore report the fire rate next to every steered number and treat a row that equals the baseline at 0% fire as a dead hook, not a null result.

Where the gate fires. With $\tau_c{=}{-}0.5$ the gate stays shut on grounded prompts (test 3) in every configuration, so CO is 62.7% throughout; it opens on grounded prompts only under the midpoint threshold ($\leq 1.6\%$, CO ≥ 61.3). On the context-free tests it fires *more* on stable (test 1, 29.9%) than on volatile prompts (test 4, 22.1%) at layer 31: the presence direction carries no volatility signal, which is why a presence-only gate cannot separate the two rows of the context-absent column. The instructed-abstention template (test 2) fires at 77.7%, so the instruction itself moves the presence projection.

The midpoint rule over-fires at every layer. Setting τ_c to the class-mean midpoint at each layer opens the gate on 57–68% of prompts, pushes VRR to 61–95% above layer 12, and leaves SR at 1.3–15.8% with VRR$-$SDR at most 11.2. At layer 12 the model neither answers nor defers (SR 1.3, VRR 0.0). The collapse reported for layer 31 in Paragraph 4.2 is therefore a property of LLAMA's presence projection, not of the layer, and it is why τ_c is set by hand on this backbone.

Table 16: All 37 sensitivity configurations (presence-only gate, LLAMA-3.1-8B). Fire: overall and per test (T1 stable, T2 instructed-abstention, T3 grounded, T4 volatile). †Test-4 decode byte-identical to un-steered and all metrics unchanged. ‡Byte-identical to $\ell{=}31$, $\alpha{=}50$, $\tau_c{=}{-}0.5$.

				fire (%)										
Block	ℓ	α	τ_c	all	T1	T2	T3	T4	SR	SDR	VRR	DD	CO	Hal.
un-steered	–	0	–	0.0	–	–	–	–	53.5	6.1	24.9	18.8	62.7	9.2
α	31	10	-0.5	33.8	29.9	77.7	0.0	22.1	48.3	24.3	47.6	23.3	62.7	3.7
	31	20	-0.5	33.8	29.9	77.7	0.0	22.1	43.5	30.9	49.9	19.1	62.7	3.1
	31	25, 50, 75, 100	-0.5	33.8	29.9	77.7	0.0	22.1	42.2	32.0	50.1	18.1	62.7	3.1
τ_c	31	50	-0.7	30.9	26.6	73.0	0.0	19.4	44.4	29.4	47.4	18.0	62.7	3.9
	31	50	-0.6	32.1	27.9	75.3	0.0	20.4	43.5	30.3	48.5	18.2	62.7	3.4
	31	50	-0.4	35.2	31.8	80.1	0.0	23.2	41.3	33.8	51.6	17.8	62.7	2.7
	31	50	-0.3	36.5	34.0	81.4	0.0	24.7	40.7	35.5	53.5	18.0	62.7	2.7
layer, fixed	12, 14	50	-0.5	0.0	0.0	0.0	0.0	0.0			†			
	20	50	-0.5	1.1	0.0	4.3	0.0	0.0			†			
	24	50	-0.5	8.7	4.1	28.5	0.0	2.2	52.6	9.9	27.9	18.0	62.7	8.2
	31	50	-0.5						‡					
layer, matched	12	50	$+0.569$	33.8	32.9	70.2	0.0	25.2	44.8	0.7	13.4	12.7	62.7	3.7
	14	50	$+0.760$	33.8	58.0	48.6	0.0	26.7	35.1	32.5	39.3	6.8	62.7	5.7
	16	50	$+0.821$	33.7	51.3	55.4	0.0	25.6	36.2	46.3	54.5	8.2	62.7	4.6
	20	50	$+0.989$	33.8	32.9	76.0	0.0	21.8	41.1	34.6	50.6	16.1	62.7	3.7
	24	50	$+0.698$	33.8	29.7	76.0	0.0	23.3	42.4	31.6	52.8	21.2	62.7	2.9
	31	50	-0.497						‡					
layer, midpoint	12	50	$+0.963$	67.9	98.5	100.0	0.0	60.7	1.3	0.0	0.0	0.0	62.7	0.0
	14	50	$+1.179$	62.0	85.3	96.3	0.0	54.6	15.8	50.2	61.4	11.2	62.7	0.9
	16	50	$+1.299$	60.3	86.6	91.4	0.2	52.6	11.2	75.5	82.4	6.9	62.4	1.3
	20	50	$+1.496$	56.9	79.4	96.3	0.0	44.7	14.5	77.0	79.3	2.3	62.7	1.0
	24	50	$+2.002$	64.7	95.0	98.7	0.2	55.1	3.3	93.1	92.3	-0.8	62.4	0.4
	31	50	$+3.147$	67.4	96.8	99.7	1.6	59.7	2.0	89.0	94.7	5.7	61.3	0.1
dead gate	16	10–100	-0.5	0.0	0.0	0.0	0.0	0.0			† (6 settings)			
	16	50	-0.7 to -0.3	0.0	0.0	0.0	0.0	0.0			† (4 settings)			

A.10 PRODUCT CATALOGUE DATASET CONSTRUCTION

We collect two temporal snapshots (T_1, T_2) of a product catalogue of approximately 25,000 products. For each product we extract five numeric fields with different volatility profiles: current price, discount rate, order count, average rating, and review count. A field-level record is labelled *volatile* if its value changed between T_1 and T_2, and *stable* otherwise.

Table 17: Llama on the retail dataset and on EvoWiki (one run per variant; values in %). CO and KCR (Eq. 10): grounded volatile questions whose output contains $y_i^{T_2}$ and the outdated value $y_i^{T_1}$. EvoWiki VRR and CO as in Table 2. No row applies BRIDGe.

	Retail dataset				EvoWiki			
	Zero-shot	SFT	A-VMD	CA-AVMD	Zero-shot	SFT	A-VMD	CA-AVMD
VRR↑	0.0	0.0	99.0	99.5	72.5	0.0	96.1	24.9
CO↑	92.8	95.1	94.3	94.3	57.8	50.4	32.9	62.7
KCR↓	0.3	0.3	0.3	0.3	29.8	34.9	18.2	32.0

Table 18: Training-seed variance; mean ± standard deviation over three retrained seeds. The presence-only gate was run on LLAMA only.

Backbone	Variant	SR↑	SDR↓	VRR↑	DD↑	CO↑
LLAMA-3.1-8B	SFT	53.5±0.5	0.0±0.0	0.0±0.0	0.0±0.0	50.1±1.6
	CA-AVMD	55.2±0.8	8.9±3.5	37.9±12.9	29.0±9.4	61.2±1.4
	+ presence	37.5±3.2	39.8±7.3	64.1±7.1	24.3±1.5	61.2±1.4
	+ BRIDGE	49.9±1.4	21.3±3.4	58.7±8.8	**37.3**±5.7	61.2±1.4
MISTRAL3-7B	SFT	52.8±0.3	0.0±0.0	0.0±0.0	0.0±0.0	51.3±0.6
	CA-AVMD	55.0±1.0	2.8±1.3	27.6±7.3	24.8±6.0	62.4±1.0
	+ BRIDGE	53.2±1.1	9.1±2.0	47.9±6.2	**38.8**±4.3	62.4±0.8

We extracted natural-language questions and golden-context passages for each record, producing 15,000 examples (7,500 stable, 7,500 volatile) in EvoWiki-compatible format. An 80/20 train/evaluation split is applied, with 40% of the volatile evaluation examples designated as a conflict split (context injected with the T_2 value, testing override of the model's trained-in T_1 belief).

We define an additional helpful metric - the *knowledge conflict rate* (KCR) - as the percentage of grounded volatile questions whose output contains the outdated value $y_i^{T_1}$, although the passage c_i states $y_i^{T_2}$:

$$\mathrm{KCR} = \frac{100}{|\mathcal{D}_v^{\mathrm{grd}}|} \sum_{i \in \mathcal{D}_v^{\mathrm{grd}}} \mathrm{CM}\big(y_i^{T_1}, \hat{y}_i(q_i, c_i)\big). \tag{10}$$

KCR is near-zero (0.3%) across *all* variants including zero-shot: without a parametric prior over there is no stale belief to override. This confirms that the knowledge conflict problem is proportional to the strength of the model's domain-specific parametric memory, and that BRIDGe's conflict-resolution mechanisms are most needed precisely when that memory is strongest, as in the EvoWiki setting.

A.11 TRAINING-SEED VARIANCE

We retrain LLAMA-3.1-8B and MISTRAL3-7B under three seeds and apply BRIDGE to each seed's CA-AVMD checkpoint with the gate settings of Section 4.2. QWEN has no seeded CA-AVMD runs. The seeds are retrained checkpoints on a fixed data split, not the checkpoints of Table 3, so the means do not match Table 3 exactly. SR and CO change little across seeds, while VRR and SDR vary more. With three seeds, the DD gain is clearly separated from zero on MISTRAL and consistent in direction on LLAMA.