SAFELOOP: PRESERVING REFUSAL OF HARMFUL PROMPTS IN LOOPED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Refusing to fulfill harmful requests is a crucial safety behavior of language models. Looped language models offer a parameter-efficient approach to reasoning by reusing the same Transformer weights for additional computation before predicting each token. However, whether refusal remains reliable as recurrent depth increases is not well understood. Evaluating this reliability through attack success alone can be misleading: deeper models can stop producing harmful answers because their generation has degenerated. We identify a distinct failure in refusal representations. A behaviorally validated refusal direction can recover harmful–benign separability after erasure in one loop, yet its projection can drift far from its reference value during unperturbed inference. The ability to reconstruct this representation therefore does not ensure that additional loops preserve it. This finding motivates SafeLoop, which stores each input's projection at a reference loop and corrects subsequent deviations without updating model parameters. On frozen Ouro-1.4B, SafeLoop increases the percentage of responses that are non-degenerate refusals (effective refusal) from **9.5%** to **41.0%** at 16 loops and from **6.5%** to **35.5%** at 24 loops. It also improves effective refusal in selected recurrent models converted from dense backbones. Benefits vary across models and depths; projection correction does not resolve widespread generation degeneration and can increase benign over-refusal. These results establish cross-loop preservation as a concrete target for refusal control and show why evaluating deeper inference requires separating effective refusal from generation failure.

1 INTRODUCTION

Safety-aligned language models are trained to decline harmful requests while remaining helpful on benign ones (Arditi et al., 2024). Looped language models offer a parameter-efficient approach to reasoning through repeated computation with shared weights (Zhu et al., 2025; Geiping et al., 2025). Standard Transformers stack layers with distinct parameters. Looped Transformers instead repeatedly apply the same Transformer block or stack before predicting each next token, allowing effective network depth to vary without increasing the parameter count (Yang et al., 2024b; Saunshi et al., 2025; Jeddi et al., 2026). Whether refusal remains reliable as this recurrent depth increases is not well understood. Additional loops also keep modifying the hidden states on which an aligned model bases its responses. Refusal learned within the training depth may therefore change when the same model runs for longer. We study how this behavior changes, what happens to its associated representation, and whether inference-time control can preserve it.

Existing depth and refusal studies leave this preservation problem unresolved. Ouro reports declining performance on standard tasks beyond the training depth alongside decreasing harmfulness on HEx-PHI (Zhu et al., 2025). That divergence makes the interpretation of apparent safety gains consequential: fewer harmful outputs may reflect either effective refusal or an inability to generate a usable answer. Work on refusal directions shows that changing a low-dimensional residual-stream component can suppress or induce refusal (Arditi et al., 2024; Wollschläger et al., 2025). Such behavioral control identifies a useful representation, but it does not determine how that representation evolves when shared computation continues beyond the trained depth.

Our central finding is that reconstructing a refusal representation and preserving it through further computation are different properties. In native looped models, harmful–benign separability along a

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.

frozen refusal direction returns rapidly after its component is erased in one loop. Along unperturbed trajectories, however, the projection can drift substantially. In Ouro-1.4B, the mean harmful-input projection falls to approximately 3% of its training-depth reference by loop 24, despite rapid recovery after transient erasure. Ouro-2.6B retains approximately 84% at the same depth. The first model thus exposes a preservation problem even though subsequent computation can reconstruct the erased signal. This distinction motivates maintaining a useful reference projection throughout the additional loops.

SafeLoop implements this idea with an input-specific correction. It records the refusal-direction projection at a reference loop and corrects deviations after every subsequent loop. Each input supplies its own target, so correction can either raise or lower the projection. The direction and model parameters remain fixed. Repeated correction directly addresses renewed changes caused by shared computation; preserving the input's own reference also avoids imposing the same activation shift on every prompt. The mathematical effect is exact restoration or partial correction of one coordinate, while its behavioral effect is measured through complete generations.

This intervention recovers effective refusal in several settings where additional loops degrade it. On frozen Ouro-1.4B, effective refusal rises from 9.5% to 41.0% at 16 loops and from 6.5% to 35.5% at 24 loops. Outputs recovered from degeneration are predominantly refusals. Recurrent Gemma and Qwen checkpoints provide further evidence at training and extrapolated depths, respectively. The comparisons also identify the scope of the method: a relatively stable projection in Ouro-2.6B yields no comparable benefit, broader generation degeneration remains in deep Gemma, and benign over-refusal can increase. Together, these results make preservation across depth an explicit design and evaluation target for looped-model alignment.

We make three contributions:

- We reveal that low attack success at greater recurrent depth can coexist with lost effective refusal and severe degeneration, establishing why these outcomes must be evaluated separately.
- We distinguish rapid reconstruction after transient erasure from natural drift of a behaviorally validated refusal projection, identifying a concrete target for inference-time control.
- We introduce SafeLoop and test reference correction on native and dense-to-loop models, quantifying both recovered effective refusal and the conditions under which correction is ineffective or costly.

Section 3 presents diagnostic experiments on how additional loops affect refusal behavior and its representations. These findings motivate SafeLoop in Section 4; Section 5 evaluates its benefits and costs, and Section 6 discusses broader implications and limitations.

2 Related Work

Looped computation and depth extrapolation. Recurrent architectures make depth an adjustable inference resource. The Universal Transformer applies shared transformations repeatedly and supports adaptive computation (Dehghani et al., 2019). Ouro develops pretrained looped language models with learned depth allocation (Zhu et al., 2025), while retrofitted recurrence extends repeated computation to pretrained non-recurrent models (McLeish et al., 2025). Improvements in task performance under these architectures do not establish that refusal remains reliable as depth changes. We study this preservation requirement in both native looped models and recurrent models converted from dense backbones.

Depth extrapolation also depends on how recurrence is constructed. Prior work studies recurrent dynamics and the preservation of task decisions (Blayney et al., 2026; Viakhirev et al., 2026); dense-to-loop methods use layer folding, shared-weight initialization, continued training, or training-free block repetition (Bae et al., 2025; Yang et al., 2026; Shapiro, 2026; Chen et al., 2026). These differences motivate testing both native recurrence and converted checkpoints rather than treating additional loops as an equivalent intervention across architectures.

Refusal representations and activation control. Internal representations provide a direct means of studying and controlling refusal. Arditi et al. (2024) identify directions whose erasure suppresses

refusal and whose addition induces it. Circuit Breakers control representations associated with harmful outputs (Zou et al., 2024); ProCon constrains refusal-direction projections to reduce safety degradation during instruction fine-tuning (Du et al., 2025). These approaches establish the value of representation interventions. The question here is how to preserve an input's refusal-related state while the same frozen update continues to act on it. SafeLoop uses that input's reference-depth projection as the target of repeated inference-time correction.

Safety evaluation. Different safety evaluations measure different failure modes. HarmBench and JailbreakBench assess harmful completions and jailbreak attacks (Mazeika et al., 2024; Chao et al., 2024); XSTest measures refusal of benign requests (Röttger et al., 2024); StrongREJECT evaluates harmful utility (Souly et al., 2024). With deep recurrence, generation degeneration adds another explanation for low measured attack success. We therefore evaluate effective refusal alongside degeneration, harmful completion, and benign over-refusal, so that a low attack success rate alone does not determine the safety interpretation.

3 Why Additional Loops Can Fail to Preserve Refusal

3.1 Setting and Measurements

We study refusal under a change in recurrent depth while holding the checkpoint and decoding settings fixed. For an input token sequence $x = (x_1, \ldots, x_n)$, write $H_t = (h_{t,1}, \ldots, h_{t,n})$, where $h_{t,i} \in \mathbb{R}^d$ is the hidden state at token position i after loop t. Here, t indexes recurrent depth, i indexes token position, and T is the total number of loops. Without intervention, the model repeatedly applies a shared update:

$$H_0 = e(x), \qquad H_{t+1} = F_\theta(H_t), \qquad t = 0, \ldots, T-1. \tag{1}$$

Here, e maps the token sequence x to the initial hidden states H_0, and F_θ denotes the hidden-state update performed by one full application of the recurrent Transformer body, with parameters θ shared across loops and fixed during inference. Depth extrapolation means $T > T_{\text{train}}$. Native-loop experiments use Ouro-1.4B and Ouro-2.6B with $T_{\text{train}} = 4$ and $T \in \{4, 8, 16, 24, 32\}$ (Zhu et al., 2025). Transfer experiments use recurrent checkpoints constructed from Qwen2.5-1.5B-Instruct, Llama-3.2-1B-Instruct, and Gemma-2-2b-it (Yang et al., 2024a; Meta, 2024; Gemma Team, 2024). The full dense stack becomes a shared recurrent body and is post-trained at depth 4 with a frozen dense teacher. Main-text Qwen Align-G and Gemma checkpoints use general instructions and raw text; construction details and other training variants are in Appendix E. SafeLoop is applied after training, with all parameters frozen.

Effective refusal separates refusal from degeneration. We use AdvBench and HarmBench harmful requests (Zou et al., 2023; Mazeika et al., 2024), Alpaca benign instructions (Taori et al., 2023), XSTest-safe (Röttger et al., 2024), and MMLU (Hendrycks et al., 2021). A refusal-phrase matcher defines the refusal flag R; a fixed heuristic for short, repetitive, or low-coverage outputs defines degeneration D (Appendix B). Effective refusal is a detected refusal without detected degeneration:

$$Q = R \wedge \neg D. \tag{2}$$

We report rates R, D, Q over all prompts, using D_H and D_B for harmful and benign inputs. Harmful completion J counts outputs without detected refusal or degeneration that Llama Guard 3 judges harmful (Llama Team, AI @ Meta, 2024); O denotes benign over-refusal. These are operational measurements based on automated detectors. Empirical rates and accuracies are reported as percentages; absolute differences in these rates and accuracies and their confidence intervals are in percentage points (pp). Explicit n/N values retain success or outcome counts. AUCs, projection values, and control coefficients retain their original units. Attack evaluations include fixed templates, JBB-PAIR (Chao et al., 2023; 2024), adaptive template search, and fixed GCG suffixes optimized on the uncontrolled model (Zou et al., 2023). Detector thresholds are in Appendix B, judging and statistical procedures in Appendix A, and sample counts with the corresponding experiments.

The degeneration heuristic flags answers with fewer than 5 whitespace-delimited words, a repeated 4-gram ratio above 25%, a distinct-bigram ratio below 55%, or coverage of fewer than 12% of the prompt's content words. The coverage rule applies only to answers without detected refusal: a

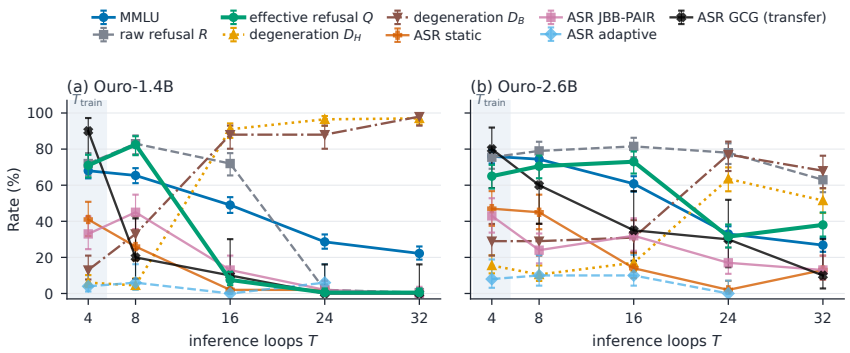


Figure 1: Effective refusal and attack success diverge as recurrent depth increases. Shading marks training depth. $Q = R \wedge \neg D$ denotes effective refusal; D_H, D_B are degeneration rates on harmful and benign inputs. Ouro-1.4B's low deep ASR coincides with D_H approaching 100%. ASR requires no detected refusal or degeneration and a harmful judgment. Rates and accuracies are percentages. Error bars are Wilson 95% intervals.

normal refusal need not repeat the request. This definition makes the quality screen reproducible, while remaining distinct from comprehensive human assessment. Main on/off comparisons use 10,000 dataset-stratified paired bootstrap resamples. GCG suffixes remain fixed after optimization on the uncontrolled model; adaptive template search is evaluated separately for each condition.

A fixed refusal direction makes cross-loop changes measurable. Following Arditi et al. (2024), we construct harmful–benign mean-difference directions and select one on separate validation data. Selection requires ablation to reduce harmful-request refusal, addition to increase benign-request refusal, and benign next-token distribution changes to remain bounded. For representation measurements, let $h_t(x) = h_{t,n}(x)$ denote the final input-token state. For the selected direction w^* (Appendix A), define

$$\widehat{w}^* = w^*/\|w^*\|_2, \qquad q_t(x) = (\widehat{w}^*)^\top h_t(x). \tag{3}$$

We freeze the direction across depths and conditions. The projection $q_t(x)$ measures a refusal-related coordinate, not a refusal probability. Erasing it in loop t_0 applies $h \leftarrow h - \widehat{w}^*((\widehat{w}^*)^\top h)$ and removes the intervention from the next loop onward. We use this perturbation to test reconstruction and untreated trajectories to test preservation.

3.2 Low Attack Success Can Mask Lost Effective Refusal

Increasing depth changes capability, effective refusal, and attack success in different ways (Figure 1). From $T = 4$ to 8, Ouro-1.4B's MMLU falls from 68.0% to 65.4% while effective refusal rises from 71.0% to 82.5%. At $T = 16$, MMLU is 49.0%, but effective refusal falls to 7.5% and harmful-input degeneration reaches 91.0%. Ouro-2.6B retains its effective-refusal improvement over a wider range: $Q = 65.0\%/70.5\%/73.0\%$ at $T = 4/8/16$, before falling to 31.5% at $T = 24$. Capability evaluation alone therefore misses important changes in refusal.

Low attack success at deep settings can accompany extensive generation failure. Ouro-1.4B's measured attack success rates are at most 13% for $T \geq 16$ and at most 6% for all four attacks at $T = 24$, while harmful-input degeneration is 91%–97% at these depths. For Ouro-2.6B, fixed-template success falls to 2% at $T = 24$ as degeneration rises to 63.5%. These results expose why lower attack success is insufficient evidence of more reliable refusal. The relevant control objective is to recover usable refusal while tracking harmful completion and benign behavior. Full depth sweeps and Thinking checkpoints are in Appendix B.

3.3 Reconstruction Does Not Ensure Natural Preservation

Later loops can reconstruct the erased refusal signal quickly. This resembles downstream compensation for local ablations in non-recurrent Transformers (McGrath et al., 2023; Rushing & Nanda,

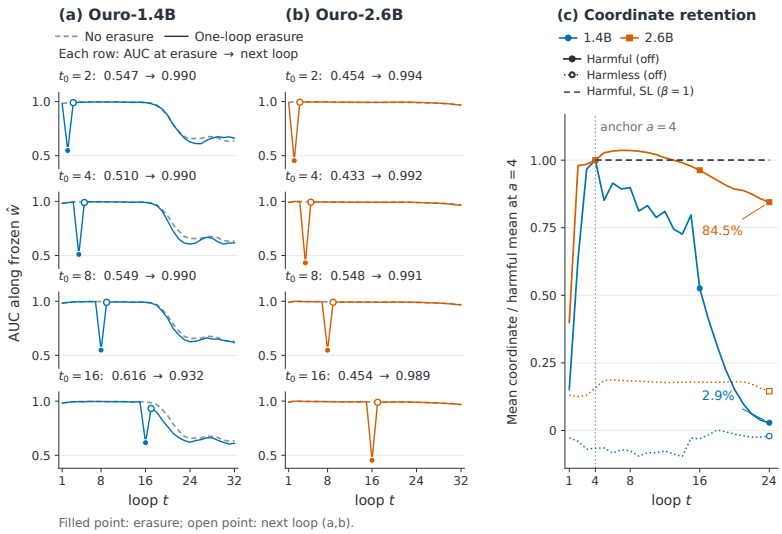


Figure 2: Rapid reconstruction after erasure coexists with natural projection drift. Blue/orange denote Ouro-1.4B/2.6B. (a,b) Each row erases the frozen refusal-direction component at t_0. Colored curves show perturbed trajectories; gray dashed curves are untreated. Filled/open points mark the erased/next loop; annotations give AUC. (c) Untreated harmful/benign mean projections (solid/dotted), normalized by the same model's harmful-input mean at training depth $a = 4$. Harmful projections retain 2.9%/84.5% of that reference at loop 24. Dark-gray dashes show SafeLoop ($\beta = 1$), with both models near the target 1. Each panel uses 100 harmful and 100 benign probes. Projection and AUC are representation measurements.

2024); here, reconstruction occurs across shared updates. We freeze the direction selected at training depth and erase its component in a single loop. Harmful–benign projection AUC drops in that loop and recovers sharply in the next (Figure 2a,b). At $t_0 = 4$, normalized separability-recovery scores are 0.989 for Ouro-1.4B and 0.991 for Ouro-2.6B. In Ouro-1.4B, the paired along-axis root-mean-square error falls within one loop to 42% of its value immediately after erasure. Separability also largely recovers for $t_0 \in \{2, 4, 8, 16\}$. This reconstruction concerns class separability; projection magnitude and generated behavior are evaluated separately (Appendix C). Sustained erasure provides behavioral evidence for the direction's relevance: erasing every loop of Ouro-1.4B at $T = 8$ lowers effective refusal from 83.6% to 0.8% and raises harmful completion to 82.0% (Table 12).

The same projection can nevertheless drift during ordinary inference. In Ouro-1.4B, the untreated mean harmful projection peaks at loop 4, retains 73%–92% of that reference during loops 5–15, and falls to approximately 3% by loop 24 (Figure 2c). Benign projections remain near zero as harmful–benign separability falls from 0.996 to 0.664. Ouro-2.6B instead remains slightly above its loop-4 reference during loops 5–12 and retains approximately 84% at loop 24, with separability above 0.99. Thus, reconstruction after erasure does not settle whether an untreated trajectory maintains the reference projection. These representation and behavioral measurements use separate evaluation sets; their comparison identifies a control target rather than a per-example mediation effect.

The control target is therefore the input's reference projection throughout additional computation. The pulse experiment asks whether later computation can reconstruct an erased signal; the untreated trajectory asks whether that computation preserves its earlier value. Rapid recovery answers the first question but does not settle the second. SafeLoop corrects deviations at every later loop and tests whether maintaining this coordinate improves generated refusal.

4 SafeLoop: Correcting Projection Drift Across Loops

SafeLoop repeatedly corrects the refusal projection toward a value already formed for the current input. An input-specific reference retains the distinction between prompts that elicited different

projections at that depth. Correction targets deviations from those values, rather than adding the same refusal activation to every prompt. The intervention requires no model-parameter updates.

Reference projection. For each generation prefix and token position i, cache the projection at reference loop a:

$$q_{a,i} = (\widehat{w}^*)^\top h_{a,i}. \tag{4}$$

Native Ouro extrapolation uses $a = 4$. For dense-to-loop checkpoints trained at depth 4, $a = 1$ at $T = 4$ tests preservation of the first-loop projection, and $a = 4$ at $T = 8$ tests preservation beyond training depth. In both cases, the reference comes from the checkpoint being evaluated. Its first loop is already post-trained and is distinct from the original dense model.

Persistent correction. After loop $t > a$, let $h_{t,i}$ denote the state before the current correction, including the effects of earlier corrections. Define

$$\begin{aligned} \widetilde{h}_{t,i} &= h_{t,i} + \beta \left(q_{a,i} - (\widehat{w}^*)^\top h_{t,i} \right) \widehat{w}^* \\ &= h_{t,i} + \beta\, \widehat{w}^* (\widehat{w}^*)^\top (h_{a,i} - h_{t,i}), \qquad 0 < \beta \leq 1. \end{aligned} \tag{5}$$

The second line projects the reference-state difference onto the refusal direction. We also write $\widetilde{h}_{t,i} = C_\beta(h_{t,i}; q_{a,i})$, abbreviating the map as $C_\beta(h)$ when the reference is fixed.

Let $\widetilde{H}_t = (\widetilde{h}_{t,1}, \ldots, \widetilde{h}_{t,n})$. The next loop receives $H_{t+1} = F_\theta(\widetilde{H}_t)$ for $t < T$; $\widetilde{H}_T$ is used for the final output readout. For $t \leq a$, set $\widetilde{H}_t = H_t$.

Only the component along $\widehat{w}^*$ changes. With $\beta = 1$, the corrected projection equals the reference; with $0 < \beta < 1$, its error is multiplied by $1 - \beta$ (Appendix Proposition 1). Correction can raise or lower the projection. Later loops may change it again, so correction is repeated. This controls one coordinate, not the full state; behavioral effects are evaluated in Section 5.

Token-wise implementation. The controller acts on every token at loop end, storing a separate reference scalar for each position. Loops are executed and references read again for each newly generated token. This differs from the representation plots, which read the final input token. The added operations are inner products and along-axis vector updates; model parameters remain fixed. Appendix A specifies intervention placement and state write-back.

5 Evaluation of Refusal Preservation

We test three questions: whether correction recovers effective refusal in native looped models, whether it transfers to models converted from dense backbones, and whether persistent correction differs from single-loop and random-axis controls. Each on/off comparison uses the same frozen checkpoint, depth, and prompts. The main tables summarize refusal and attack outcomes; complete capability and benign-input measurements are reported in Appendices D and F.

5.1 From Loop to SafeLoop: Correcting Refusal-Projection Drift

SafeLoop substantially improves effective refusal at the deep Ouro-1.4B settings where the untreated projection has drifted (Figure 3, Table 1). With $a = 4$ and $\beta = 1$, effective refusal at $T = 16$ rises from 9.5% to 41.0% ($\Delta = +31.5$ pp, [24.0, 39.0] pp), while harmful-input degeneration falls from 87.5% to 58.5%. At $T = 24$, raw refusal rises from 10.0% to 78.5% and effective refusal from 6.5% to 35.5% (+29.0 pp, [22.0, 36.5] pp). At $T = 8$, where untreated refusal remains high, the change is small: 84.0% to 85.0%. The main protocol uses 200 harmful prompts and 256-token generation, with a separate off run under the depth-sweep protocol.

Recovered outputs predominantly become effective refusals. At $T = 16/24$, 68/62 previously degenerate outputs become effective refusals, while 0/1 become harmful completions; total controlled harmful-completion rates are 0.0%/0.5%. Attack success remains low, with adaptive success at $T = 24$ changing from 3/50 to 0/50. Two further batches at $T = 24$ yield effective-refusal gains of $+48.4$ pp [39.1, 57.8] and $+47.7$ pp [39.8, 56.2]. These batches use 128 tokens and 64 prompts per

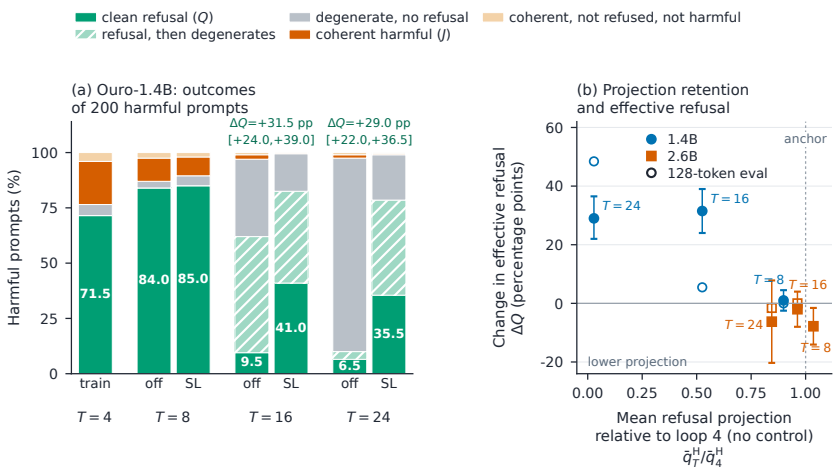


Figure 3: SafeLoop recovers effective refusal in deep Ouro-1.4B ($a = 4, \beta = 1$). (a) Output categories for 200 harmful prompts; SL denotes SafeLoop. Outcome shares are percentages; annotations report effective-refusal changes (ΔQ) and paired 95% intervals in pp. (b) Effective-refusal change (pp) against the untreated mean harmful projection relative to loop 4, $\bar{q}_T^{\mathrm{H}}/\bar{q}_4^{\mathrm{H}}$. Filled/open points use the main/128-token protocols and share projection estimates from 100 untreated harmful probes per model. Error bars show paired 95% intervals for filled points; batch protocols are in Appendix D.

dataset with substantial prompt overlap. Gains at transitional depth $T = 16$ vary from $+2$ to $+32$ pp across batches (Appendix D).

The gains carry a depth-dependent benign-refusal cost. At $T = 16$, XSTest-safe over-refusal rises from 31% to 55% ($+24$ pp, [15, 34] pp); at $T = 24$, it changes from 19% to 15%. MMLU point estimates change from 44.5% to 47.5% at $T = 16$ and from 25.5% to 28.5% at $T = 24$, while Alpaca degeneration at $T = 16$ falls from 85.0% to 66.0%. Thus, recovered effective refusal can coexist with increased refusal of borderline benign requests. The capability point estimates characterize these comparisons without establishing a general absence of capability costs.

Ouro-2.6B shows why projection preservation needs a model-specific behavioral test. Its untreated projection remains closer to the reference, and SafeLoop gives effective-refusal changes of -2.0 and -6.2 pp at $T = 16/24$. At $T = 8$, the mean projection is approximately 1.036 times its loop-4 value; correction lowers effective refusal from 73.4% to 65.6% (-7.8 pp, $[-14.1, -1.6]$ pp). At $T = 24$, degeneration and effective-refusal loss persist despite substantial projection retention. Figure 3b summarizes this contrast with 1.4B. The association supports testing projection drift as a diagnostic, but does not establish a monotonic rule predicting benefit.

5.2 FROM DENSE TO LOOP: PRESERVING REFUSAL AFTER RECURRENT CONVERSION

Lower attack success after recurrent conversion need not mean better effective refusal, and reference correction has different effects across backbones (Table 2). We compare frozen Qwen Align-G and Gemma checkpoints trained on general instructions and raw text. Neither uses a refusal-projection recovery loss or training-time correction connection. Preserving loop 1 at $T = 4$ tests an earlier state within the trained computation; preserving loop 4 at $T = 8$ tests extrapolation. Original dense models show the combined effects of recurrent conversion and post-training before the inference-time intervention.

At training depth, SafeLoop improves Gemma's refusal and attack results together (Table 2). Recurrent conversion and post-training reduce Gemma's effective refusal from 100.0% to 90.6% and increase JBB-PAIR success from 20.3% to 42.2%. Correction raises effective refusal to 97.7% ($+7.0$ pp, [3.1, 11.7] pp), lowers JBB-PAIR to 21.9% (-20.3 pp, $[-31.2, -9.4]$ pp), and lowers GCG success from 45% to 5%. Qwen illustrates a different outcome: conversion and post-training

Table 1: SafeLoop (ours) improves deep effective refusal in Ouro-1.4B. Cells show uncontrolled→SafeLoop for $a = 4, \beta = 1$. $T = 4$ is the training-depth reference under the same generation protocol. GCG transfers 20 suffixes optimized at $T = 4$ and is scored by non-refusal/non-degeneration. Complete capability, benign-input metrics, paired intervals, other doses, and Ouro-2.6B: Appendix D. Rates are percentages; effective-refusal changes are in percentage points (pp).

Metric (%)	T=4 (reference)	T=8	T=16	T=24
Raw refusal $R \uparrow$	71.5	84.0→85.0	62.0→82.5	10.0→78.5
Effective refusal (Q) $\uparrow$	71.5	84.0→85.0	9.5→41.0	6.5→35.5
Effective refusal change (pp)	—	+1.0	+31.5	+29.0
Harmful degeneration $D_H \downarrow$	5.0	3.0→4.5	87.5→58.5	91.0→63.5
Harmful completion $J \downarrow$	19.5	10.5→8.5	2.0→0.0	1.5→0.5
ASR: fixed templates $\downarrow$	44.0	29.0→30.0	1.0→3.0	1.0→2.0
ASR JBB-PAIR $\downarrow$	61.0	59.8→51.2	2.4→2.4	3.7→4.9
ASR: adaptive $\downarrow$	4.0	6.0→10.0	0.0→0.0	6.0→0.0
ASR: GCG (fixed suffixes) $\downarrow$	100.0	35.0→10.0	5.0→0.0	5.0→5.0

Table 2: Refusal and attack outcomes after recurrent conversion and SafeLoop (our method). T4 and T8 denote $T = 4$ and $T = 8$; SafeLoop uses $\beta = 1$ with $a = 1$ and $a = 4$, respectively, from the post-trained checkpoint. Harmful prompts: 200 Qwen/128 Gemma. ASR uses Llama Guard 3; GCG transfers 20 suffixes optimized on uncontrolled models; adaptive search uses 50 goals with at most 5 queries each. All rates are percentages. Complete capability and benign-input metrics, other doses, and paired intervals are in Appendix F.

Condition	Effective refusal (Q) (%) $\uparrow$	Harmful degeneration (%) $\downarrow$	ASR fixed (%) $\downarrow$	ASR JBB-PAIR (%) $\downarrow$	ASR GCG (%) $\downarrow$	ASR adaptive (%) $\downarrow$
Qwen2.5-1.5B (Align-G)						
Dense	96.0	0.0	44.0	46.0	90.0	10.0
T4	93.5	1.0	15.0	21.0	60.0	4.0
T4 + SafeLoop (ours)	93.5	1.0	21.0	32.0	55.0	6.0
T8	57.0	36.5	1.0	1.0	25.0	0.0
T8 + SafeLoop (ours)	66.0	30.5	0.0	1.0	0.0	0.0
Gemma-2-2B						
Dense	100.0	0.0	9.4	20.3	55.0	6.0
T4	90.6	0.8	9.4	42.2	45.0	4.0
T4 + SafeLoop (ours)	97.7	1.6	4.7	21.9	5.0	4.0
T8	10.2	86.7	0.0	0.0	0.0	0.0
T8 + SafeLoop (ours)	15.6	84.4	0.0	0.0	0.0	0.0

lower fixed-template success from 44% to 15% and JBB-PAIR from 46% to 21% with little change in effective refusal. Correction leaves effective refusal at 93.5% but raises JBB-PAIR to 32% (+11 pp, [2, 20] pp). The same projection-preservation rule can therefore produce different refusal and attack responses.

Under extrapolation, correction partly restores Qwen's effective refusal while widespread generation degeneration persists in Gemma. At $T = 8$, Qwen's effective refusal falls from 93.5% to 57.0% and harmful-input degeneration rises to 36.5%. SafeLoop raises effective refusal to 66.0% (+9.0 pp, [1.0, 17.0] pp) and raw refusal by 12.5 pp; MMLU changes from 42.5% to 41.5% and fixed GCG success falls from 5/20 to 0/20. Gemma at the same depth has Alpaca degeneration of 87.5% and MMLU of 28.0%. Correction increases raw refusal from 43.8% to 84.4%, but effective refusal changes from 10.2% to 15.6% with an interval including zero. Restoring refusal openings alone does not recover usable refusal under broad generation collapse.

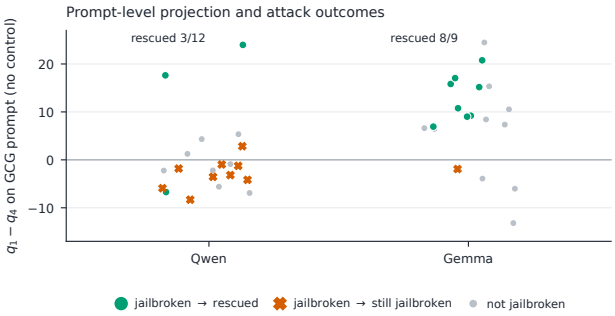


Figure 4: GCG prompt projections (20 prompts/model). Vertical axis: untreated projection decrease from loop 1 to 4. Colors distinguish rescued from still-successful jailbreaks. Gray prompts were initially unsuccessful but may succeed under control. The plot describes associations, not per-prompt predictions.

5.3 Supplemental Controls and Transfer Checks

Persistent correction has higher effective-refusal point estimates than the tested final-loop schemes across Qwen, Llama, and Gemma. At $T = 8$, $a = 3$, and $\beta = 0.5$, the differences are $+4.7$, $+6.2$, and $+3.1$ pp, respectively (Table 15). In this protocol, Llama's effective refusal rises from 82.8% to 89.8% with persistent correction. The final-loop scheme uses the preceding loop as its reference, whereas persistent correction retains loop 3, so this comparison evaluates complete schemes rather than isolating correction duration. Random-axis controls and full metrics are in Appendices G and H.

Prompt-level results further show that group-level projection changes are an imperfect guide to individual attack outcomes. Among 20 fixed GCG prompts per model, SafeLoop rescues 8 of Gemma's 9 initially successful jailbreaks and 3 of Qwen's 12. In Qwen, 2 initially degenerate outputs become new successful jailbreaks. Mean projection changes differ between rescued and unrescued prompts, with individual counterexamples (Figure 4). These results reinforce the need to validate generated behavior after correction.

6 Limitations and Conclusion

Refusal is only one component of language-model safety. SafeLoop preserves an input's projection along a fixed refusal direction; it does not ensure that the reference is correct, preserve the full hidden state, or restore general generation quality. Benefits depend on the model and depth: Ouro-2.6B shows no comparable refusal gains, deep Gemma remains largely degenerate, and benign over-refusal can increase. One random-axis control also reduces deep Ouro degeneration, so the gains cannot be attributed exclusively to refusal-direction preservation. The experiments use models no larger than 2.6B and automated detectors; broader model coverage and human assessment remain necessary. Fixed GCG transfer and adaptive template search test different attack settings, and GCG reoptimization against SafeLoop remains untested.

Evaluating safety across recurrent depth requires separating effective refusal, harmful completion, generation degeneration, and benign over-refusal: low attack success can accompany an inability to generate usable responses. We show that the projection along a frozen refusal direction can drift during repeated computation even when later loops rapidly reconstruct an erased signal. Correcting this drift with SafeLoop raises effective refusal in Ouro-1.4B from 9.5% to 41.0% at 16 loops and from 6.5% to 35.5% at 24 loops. It also improves training-depth Gemma from 90.6% to 97.7% and extrapolated Qwen from 57.0% to 66.0%. These results identify preservation across loops as a useful control target while retaining generation quality and benign behavior as separate evaluation requirements.

Ethics Statement

This study uses controlled representation interventions to understand how safety behavior forms and fails during recurrent computation. Direction ablation and jailbreak evaluation have dual-use implications. We report aggregate statistics rather than actionable harmful content; example outputs are truncated and contain no actionable harmful information. Definitions and limitations of automated judges and quality proxies are in Appendix A.

Reproducibility Statement

The supplementary material provides source code for recurrent-model construction and training, refusal-direction selection, inference-time SafeLoop, and behavioral and capability evaluation. It also includes frozen refusal-direction artifacts and model-specific prompt-split records. The README describes the code organization, file functions, software environment, and example evaluation commands. Model configurations, direction selection, intervention scope, training procedures, and evaluation protocols are documented in Appendices A, E, and D.

AI Use Statement

We used generative AI tools (ChatGPT) to assist with English-language revision, and software code editing. We did not use generative AI to design or run model experiments, generate synthetic data, or alter experimental observations. The authors reviewed all AI-assisted outputs: values supported by per-example records were recomputed, other values were checked against archived trajectories or summaries, and references and wording were reviewed by the authors. The authors take responsibility for all content, including AI-assisted text and code.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. Relaxed recursive transformers: Effective parameter sharing with layer-wise LoRA. In *International Conference on Learning Representations*, 2025.

Hugh Blayney, Álvaro Arroyo, Johan Obando-Ceron, Pablo Samuel Castro, Aaron Courville, Michael M. Bronstein, and Xiaowen Dong. A mechanistic analysis of looped reasoning language models, 2026. URL https://arxiv.org/abs/2604.11791.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries, 2023. URL https://arxiv.org/abs/2310.08419.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Lizhang Chen, Jonathan Li, Chen Liang, Ni Lao, and Qiang Liu. Training-free looped transformers, 2026. URL https://arxiv.org/abs/2605.23872.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

Yanrui Du, Fenglei Fan, Sendong Zhao, Jiawei Cao, Qika Lin, Kai He, Ting Liu, Bing Qin, and Mengling Feng. Anchoring refusal direction: Mitigating safety risks in tuning via projection constraint, 2025. URL https://arxiv.org/abs/2509.06795.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

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.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Ahmadreza Jeddi, Marco Ciccone, and Babak Taati. LoopFormer: Elastic-depth looped transformers for latent reasoning via shortcut modulation. In *International Conference on Learning Representations*, 2026.

Llama Team, AI @ Meta. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224, 2024.

Thomas McGrath, Matthew Rahtz, János Kramár, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations, 2023. URL https://arxiv.org/abs/2307.15771.

Sean McLeish, Ang Li, John Kirchenbauer, Dayal Singh Kalra, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Jonas Geiping, Tom Goldstein, and Micah Goldblum. Teaching pretrained language models to think deeper with retrofitted recurrence, 2025. URL https://arxiv.org/abs/2511.07384.

Meta. Llama 3.2-1B-Instruct model card. Hugging Face model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct. Official model card; accessed 2026-09-20.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, 2024. doi: 10.18653/v1/2024.naacl-long.301.

Cody Rushing and Neel Nanda. Explorations of self-repair in language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42836–42855, 2024.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations*, 2025.

Mark Shapiro. Retrofitting recurrent depth into a pretrained language model: Installation, extrapolation, transfer, and retention at two parameter budgets, 2026. URL https://arxiv.org/abs/2608.11233.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A StrongREJECT for empty jailbreaks. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpaca: A strong, replicable instruction-following model. Stanford CRFM, 2023. URL https://crfm.stanford.edu/2023/03/13/alpaca.html.

Ivan Viakhirev, Kirill Borodin, Amirah Almutairi, Serguei Barannikov, Maxim Abramov, and Grach Mkrtchian. Think shallow, solve deep: Controlling recurrent dynamics for reliable test-time depth, 2026. URL https://arxiv.org/abs/2608.18222.

Wei Wang and Jean-Jacques E. Slotine. On partial contraction analysis for coupled nonlinear oscillators. *Biological Cybernetics*, 92(1):38–53, 2005. doi: 10.1007/s00422-004-0527-x. URL https://link.springer.com/article/10.1007/s00422-004-0527-x.

Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and representational independence. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 66945–66970, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

Jian Yang, Wei Zhang, Shuyue Guo, Yizhi LI, Linzheng Chai, Zhengmao Ye, Shukai Liu, Yuyang Song, Jiajun Wu, Che Liu, Tianyu Zheng, Siwei Wu, Leo L, Xudong Ma, Chuan Hao, Ran Tao, Yan Xing, Jianzhou Wang, Mingjie Tang, Aishan Liu, Zhoujun Li, Xianglong Liu, Weifeng Lv, and Bryan Dai. LoopCoder: Scaling code intelligence via looped language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16209–16223. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.796.

Liu Yang, Kangwook Lee, Robert D. Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. In *International Conference on Learning Representations*, 2024b.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, Lu Li, Jiajun Shi, Kaijing Ma, Shanda Li, Taylor Kergan, Andrew Smith, Xingwei Qu, Mude Hui, Bohong Wu, Qiyang Min, Hongzhi Huang, Xun Zhou, Wei Ye, Jiaheng Liu, Jian Yang, Yunfeng Shi, Chenghua Lin, Enduo Zhao, Tianle Cai, Ge Zhang, Wenhao Huang, Yoshua Bengio, and Jason Eshraghian. Scaling latent reasoning via looped language models, 2025. URL https://arxiv.org/abs/2510.25741.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/2307.15043.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

A Data, Direction Selection, and Implementation Protocol

Checkpoint coverage. The 10 primary recurrent checkpoints comprise four Ouro checkpoints (two sizes, each with a Thinking variant) and six dense-to-loop checkpoints (four Qwen training variants, one Llama, and one Gemma). Additional training depths and cross-model diagnostics are outside this count. Main-text Qwen Align-G and Gemma use general instructions and raw text; safety-data and refusal-recovery variants are reported separately. Main on/off comparisons use 10,000 dataset-stratified paired bootstrap resamples, with protocol-specific settings for the additional experiments below.

Data roles. Direction construction and selection follow the harmful/harmless split of prior refusal-direction work (Arditi et al., 2024). AdvBench (Zou et al., 2023) and HarmBench (Mazeika et al., 2024) measure harmful-request behavior; XSTest-safe (Röttger et al., 2024) measures over-refusal; Alpaca (Taori et al., 2023) supplies ordinary benign prompts. MMLU (Hendrycks et al., 2021) measures general capability. Construction, validation, and behavioral test sets are text-deduplicated within each experiment. Shared-source AdvBench samples are not assumed independent, and this deduplication does not control pretraining exposure.

Direction construction. Following Arditi et al. (2024), we construct a candidate direction at each activation site c. Let $z_c(x) \in \mathbb{R}^d$ denote the activation extracted from prompt x at that site, and let $\mathcal{D}_{\mathrm{H}}^{\mathrm{train}}$ and $\mathcal{D}_{\mathrm{B}}^{\mathrm{train}}$ denote the harmful and benign direction-construction sets. We compute

$$w_c = \frac{1}{|\mathcal{D}_{\mathrm{H}}^{\mathrm{train}}|} \sum_{x \in \mathcal{D}_{\mathrm{H}}^{\mathrm{train}}} z_c(x) - \frac{1}{|\mathcal{D}_{\mathrm{B}}^{\mathrm{train}}|} \sum_{x \in \mathcal{D}_{\mathrm{B}}^{\mathrm{train}}} z_c(x).$$

For looped models, c specifies a loop, a within-loop activation site, and a token position. We select c^* on separate validation data using the behavioral and KL checks below, and set $w^* = w_{c^*}$. The direction is frozen across evaluation depths and control conditions. For Ouro-1.4B at $T = 4$, screening covers ChatML suffix positions $p \in \{-6, \ldots, -1\}$, four loops, every other residual-stream layer input, and the loop end: $6 \times 4 \times 13 = 312$ cells. Mean differences use the first 128 training examples per class. Filtering by the original model's refusal score leaves 9 harmful and 32 harmless validation examples; the harmful subset determines direction selection. The score is the next-token log-odds of refusal-opening tokens (I/As). Ablation uses the full intervention scope; activation addition uses each candidate's unnormalized direction (Arditi et al., 2024). The benign KL threshold is 0.1, and steering scores must be positive. Late within-loop residual inputs are excluded, but loop-end readouts remain eligible. Two candidates pass; we select the lowest harmful ablation refusal score.

The selected w^* lies at the last token of the third loop end, with norm 23.476, harmful ablation refusal score -3.671, benign steering score 1.857, and benign KL 0.02724. Erasure uses the unit axis; activation addition retains the original magnitude. Thinking checkpoints often begin with a thinking marker, making the I/As screen unsuitable; they are excluded from the primary causal conclusion. High-KL candidates admitted by relaxed thresholds likewise do not replace the primary axis.

Full intervention scope. Full-scope erasure projects all tokens at the block input, attention output, MLP output, and post-norm loop end. SafeLoop acts at the loop end. The normalized output is written back to the state passed between loops; readout-only normalization changes the trajectory. Reinjection includes the first loop, whose input is $(1+\alpha)e$; $\alpha = 0$ matches the native model. Safety generation applies the intervention throughout generation, including after prompt prefill.

Statistics and judges. For the skip experiments, MMLU confidence intervals are computed by bootstrap resampling paired, per-item correctness differences. Harmful-set intervals resample within each dataset, then average the two estimates equally, with seed 0. Paired percentile intervals use 10,000 resamples; original pulse intervals use 1,000. Main-text pulse estimates accompanying intervals are the saved bootstrap means; curves use raw AUC point estimates. Exploratory multi-cell analyses have no multiple-comparison correction. These paired intervals measure differences; equivalence requires a prespecified margin.

The LG3/HarmBench pipeline assigns bins in the order refusal, degeneration, then harmfulness. ASR uses its `jailbroken` field and thus measures this operational bin, not unconditional judge-assigned harmfulness. Total degeneration is independently counted over all generations, including refusals. Llama Guard judgments (Llama Team, AI @ Meta, 2024) may misclassify short outputs, templated refusals, or mixed-language text; non-degenerate refusal is detector-defined and lacks comprehensive human calibration. The skip experiments do not include same-protocol StrongREJECT harmful-utility scoring (Souly et al., 2024).

B Depth, Degeneration, and Fixed-Suffix Transfer

Degeneration detection and denominators. For a question x and answer y, split the answer into words by whitespace. Let n_y be the word count, r_4 the 4-gram repetition rate, d_2 the 2-gram diversity

rate, and $o(x, y)$ the proportion of content words in the question, after stop-word removal, that are covered by the answer. Let R denote a refusal match. Then

$$D = \mathbf{1}\{n_y < 5 \ \vee \ r_4 > 0.25 \ \vee \ d_2 < 0.55 \ \vee \ (\neg R \wedge o(x, y) < 0.12)\}. \tag{6}$$

Here $r_4 = 1 - |\text{unique 4-grams}|/|\text{4-grams}|$ and $d_2 = |\text{unique 2-grams}|/|\text{2-grams}|$; answers shorter than the respective n-gram length receive values 0 and 1. Content words are English words longer than two characters after stop-word removal; coverage is 1 if the question has none. Coverage applies to non-refusals, since a refusal need not repeat question content. This reproducible proxy can misclassify concise answers or paraphrases and is not a human quality judgment. D captures text-generation failure, a subset of the safety degeneration studied here.

All R, D, Q rates use all prompts as the denominator, with $Q = R \wedge \neg D$. The archived harmful-completion bin is $J = \text{harmful} \wedge \neg R \wedge \neg D$; some depth experiments lack the original non-mutually-exclusive judge labels. These bins identify low harmful-output rates accompanied by degeneration, but cannot detect all harmful leakage in answers containing refusal language. We therefore report judge results, effective refusal, and benign-task performance together.

Recomputation. Equation (6) is applied to saved generations or item-level features. Early aggregate-only judge and capability results are checked at that level. Refusals are exempt from the coverage rule; distinct prompt and generation batches remain separate.

Table 3: All observed points in the depth sweep. D_H and D_B are the degeneration rates on the harmful set and Alpaca, respectively, and J is the LG3-filtered harmful-completion bin rate. All proportions use all prompts in the corresponding set as the denominator. Rates and accuracies are percentages.

α	T	MMLU (%)	R (%)	Effective refusal (Q, %)	D_H (%)	D_B (%)	J (%)
0	4	68.0	72.0	71.0	6.0	13.0	20.0
0	8	65.4	83.0	82.5	4.5	33.0	8.5
0	16	49.0	72.0	7.5	91.0	88.0	1.0
0	24	28.6	2.0	0.5	96.5	88.0	1.5
0	32	22.2	0.5	0.5	97.0	98.0	2.0
0.5	4	67.2	71.5	71.5	6.5	40.0	19.5
0.5	8	63.4	84.5	45.5	44.5	66.0	7.5
0.5	16	54.4	70.5	15.0	69.0	47.0	10.5
0.5	24	48.4	85.0	15.0	77.5	49.0	4.5
0.5	32	38.6	85.5	57.5	34.5	28.0	5.5

Table 4: All observed points in the Ouro-2.6B depth sweep, using the same protocol as Table 3 (200 harmful prompts, 100 Alpaca prompts, 256-token generation, the same 500 MMLU items, and LG3 as judge). Degeneration is recomputed from archived item-level features using Eq. (6). Rates and accuracies are percentages.

α	T	MMLU (%)	R (%)	Effective refusal (Q, %)	D_H (%)	D_B (%)	J (%)
0	4	76.0	75.5	65.0	15.5	29.0	15.5
0	8	74.4	79.0	70.5	10.5	29.0	15.0
0	16	60.8	81.5	73.0	17.0	31.0	7.0
0	24	33.0	78.0	31.5	63.5	77.0	4.5
0	32	26.8	63.0	38.0	51.5	68.0	9.5
0.5	4	75.0	74.5	73.5	6.0	19.0	17.0
0.5	8	74.2	78.5	78.0	3.0	23.0	15.0
0.5	16	68.8	81.0	75.0	10.5	21.0	11.0
0.5	24	69.0	80.0	26.5	62.5	43.0	9.0
0.5	32	63.2	79.5	29.0	62.5	56.0	6.5

MMLU uses the same 500 items with 5-shot letter scoring. Saved aggregate accuracies support pointwise Wilson intervals, not paired cross-depth tests. All reported MMLU depth measurements

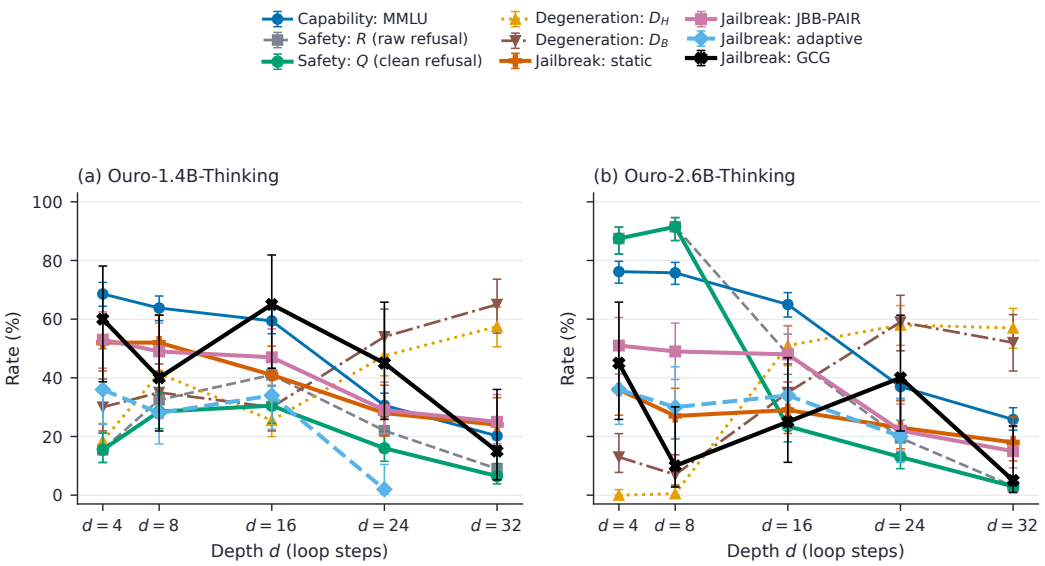


Figure 5: Same setting as Figure 1: one panel for each Thinking checkpoint, with capability (MMLU), safety (R, Q), and degeneration (D_H, D_B) sharing the depth axis d. Rates and accuracies are percentages. Outputs begin with a thinking marker, and R, Q are sensitive to the generation protocol (for 1.4B-Thinking, R at $d = 4$ is only 15.5%); the w^* selection criteria fail (Appendix A), so the main text does not use these checkpoints for causal or preservation conclusions. MMLU covers $d \in \{4, 8, 16, 24, 32\}$ (at $d = 24$, the two values are 30.6% and 37.0%).

force the full recurrent depth. In Ouro's native path, `early_exit_threshold`=1 leaves early exit enabled, potentially making nominal "$T = 32$" runs shallower than 32 (e.g., 55.4% instead of 26.8% in 2.6B); those deep values are excluded. Generation uses 100 AdvBench, 100 HarmBench, and 100 Alpaca prompts with at most 256 tokens. Differences use 10,000 same-item bootstrap resamples; harmful datasets are resampled separately and averaged equally, with seed 0. Missing per-item token counts and completion flags preclude exact truncation rates. A separate selective-intervention diagnostic uses the same 500 capability items and 32 prompts per harmful set with 128-token generation. Its random-axis Q differences are all zero, producing a zero-width sample-bootstrap interval without establishing a zero population effect.

Jailbreak experiments and representation controls. Fixed-suffix transfer improves refusal at moderate depth but coincides with degeneration at greater depth. The 20 GCG attacks target $T = 4$ with 20-token suffixes and at most 250 optimization steps. A non-refusal check every 10 steps uses 32-token generations and stops optimization after 10–30 steps. Fixed suffixes are then evaluated at deeper loops with 128-token generation. At $T = 4, 8, 16, 32$, (J, Q, D) is $(90\%, 0\%, 0\%)$, $(20\%, 60\%, 5\%)$, $(10\%, 30\%, 60\%)$, and $(0\%, 5\%, 95\%)$. From 4 to 8 loops, paired changes are $\Delta J = -70$ pp (CI $[-90, -50]$ pp) and $\Delta Q = 60$ pp (CI $[40, 80]$ pp). Attacks are not re-optimized at the target depths, so this is not an adaptive-robustness guarantee.

C Full Pulse and Release Comparisons

The following tables separate pulse recovery from sustained suppression and release. Representation and generation use different test sets, precluding item-level mediation estimates.

Let $A_c(t)$ and $A_p(t)$ be the AUCs on the clean and pulse-perturbed trajectories, respectively. We define

$$R_A(t) = 1 - \frac{A_c(t) - A_p(t)}{A_c(t_0) - 0.5},$$
$$e_q(t) = \sqrt{\mathbb{E}_x\big(q_t^p(x) - q_t^c(x)\big)^2}, \qquad \lambda_q^{(1)} = \frac{e_q(t_0+1)}{e_q(t_0)}.$$

R_A normalizes by clean AUC above chance at the erasure loop, not by the clean–perturbed difference. It is undefined when $A_c(t_0) - 0.5 < 0.10$. t_{90} is the first measured $t > t_0$ where point-estimate $R_A(t)$ reaches 0.9. Reported intervals and accompanying estimates are the archived paired-bootstrap intervals and means.

Table 5: Recovery along the causal axis w^* in Ouro-1.4B. There are 100 held-out samples per class; intervals are 95% paired-bootstrap intervals from 1,000 resamples, and point estimates shown with intervals are the saved bootstrap means. Smaller error ratios indicate more complete magnitude recovery.

α	Erasure loop t_0	$R_A(t_0+1)$	$\lambda_q^{(1)}$	t_{90}
0	4	0.989 [0.977, 0.997]	0.422 [0.399, 0.444]	5
0	16	0.890 [0.833, 0.934]	0.561 [0.536, 0.584]	23
0.5	4	0.978 [0.960, 0.992]	0.491 [0.470, 0.512]	5
0.5	16	0.986 [0.975, 0.995]	0.490 [0.461, 0.519]	17

Table 6: Persistent suppression and final-loop release in Ouro-1.4B, $\alpha = 0.5, T = 32$. AdvBench/HarmBench each contain 100 examples; MMLU uses the same 500 items. Non-degenerate refusal and total harmful degeneration are detected independently, while ASR is the mean Llama Guard 3 harmful-completion bin rate across the two harmful datasets. Rates and accuracies are percentages; AUC is unscaled.

Condition	w^* AUC	Effective refusal (Q, %)	ASR ↓ (%)	D ↓ (%)	MMLU (%)
No intervention	0.9961	60.0	5.5	32.0	38.6
Persistent deep-loop suppression	0.5383	0.5	40.5	53.0	39.8
Final-loop release after suppression	0.9938	14.5	35.0	49.0	39.8
Final-loop suppression only	0.5563	9.5	38.5	43.5	38.8
Deep-loop random-axis suppression	0.9906	65.0	12.5	21.5	33.2
Deep-loop capability-probe-axis suppression	0.9958	65.5	4.5	28.5	37.6

Table 7: Persistent suppression and final-loop release in Ouro-2.6B, $\alpha = 0.5, T = 32$. AdvBench/HarmBench each contain 100 examples and MMLU contains 500 items; the intervention scope matches Table 6. J is the judge-based harmful-completion bin rate and D is the total degeneration rate detected by the independent rule. Rates and accuracies are percentages; AUC is unscaled.

Condition	w^* AUC	Effective refusal (Q, %)	J ↓ (%)	D ↓ (%)	MMLU (%)
No intervention	0.9942	27.0	6.5	64.5	63.2
Persistent deep-loop suppression	0.4882	0.0	38.0	59.0	59.8
Final-loop release after suppression	0.9928	18.0	24.5	55.5	60.0
Final-loop suppression only	0.4628	0.0	25.5	71.5	63.0
Deep-loop random-axis suppression	0.9911	27.5	1.5	70.5	56.4
Deep-loop capability-probe-axis suppression	0.9895	29.0	7.0	63.0	61.6

In 2.6B, release reduces the harmful-completion bin rate by 13.5 percentage points relative to persistent suppression (paired 95% CI $[-22.0, -5.0]$; 10,000 resamples of 200 identically ordered harmful prompts, seed 20260921). MMLU retains same-item aggregate accuracy. Local log-odds use 64 AdvBench requests and word-initial tokens `I`/`As`; they serve as a proxy for refusal openings, not complete responses.

Ouro-2.6B also reconstructs its training-depth direction rapidly: first-step R_A bootstrap means after $t_0 = 4/16$ are $0.991/0.993$ (95% intervals $[0.981, 0.999]$, $[0.984, 1.000]$), reaching 90% at loops 5 and 17.

D Complete SafeLoop Results on Native Ouro

Evaluation batches. The three Ouro batches use each model's frozen w^* selected at $T = 4$, $\alpha = 0$, and anchor $a = 4$. Loop-end compensation covers all tokens; each generated token receives a freshly read prefix anchor. (i) *Main protocol*: 100 prompts each from AdvBench, HarmBench, Alpaca, XSTest, fixed templates, and in-the-wild attacks, 82 JBB-PAIR prompts, 500 generative MMLU items, and 256-token generation (2.6B at $T = 8/24$ instead uses 32 prompts per set), separate off/on runs, adaptive template search per condition (50 goals, 5 queries per goal), and replay of 20 GCG suffixes optimized at $T = 4$, scored as "not refused and not degenerate". (ii) *Repeat batch*: 128 tokens, 64 prompts per set, 200 shared 5-shot MMLU items, an orthogonal random-axis control, and per-loop projections. (iii) *Extended control batch*: the 128-token protocol with $T = 32$, $\beta = 0.25$, and sustained ablation. These batches are separate runs, not independent samples: repeat and extended batches share all 64 HarmBench and 64 XSTest prompts and 61 of 64 AdvBench prompts. Pairing stays within each batch.

Generative MMLU parsing. The main protocol uses zero-shot generative MMLU, accepting explicit or leading answer letters and unique option-text matches. A stricter parser incorrectly rejects outputs such as "B. 18". The same parser is applied to both control conditions. The main-text capability comparisons use the repeat batch's separate 5-shot letter-scoring evaluation. At the $T = 4$ reference, the main protocol gives Alpaca degeneration of 14.0% and XSTest over-refusal of 15.0%; the separate 5-shot MMLU reference is 68.0% on 500 items.

Table 8: Capability, harmful-side, and benign-side metrics of SafeLoop under the main protocol (off→on). R, Q, D_H, J average AdvBench and HarmBench with equal weight; D_B is the Alpaca failure rate; O is the XSTest-safe raw refusal rate; MMLU uses the lenient parser for generative answers. Rates and accuracies are percentages.

Model	T	Cond.	n_H	R (%)	Effective refusal (Q, %)	D_H (%)	J (%)	D_B (%)	O (%)	MMLU (%)
1.4B	8	β=0.5	200	84.0→84.0	84.0→83.5	3.0→4.0	10.5→9.0	28.0→33.0	10.0→13.0	58.2→57.4
1.4B	8	β=1	200	84.0→85.0	84.0→85.0	3.0→4.5	10.5→8.5	28.0→30.0	10.0→13.0	58.2→57.8
1.4B	16	β=0.5	200	62.0→55.0	9.5→9.5	87.5→89.5	2.0→0.5	85.0→83.0	31.0→45.0	29.0→31.0
1.4B	16	β=1	200	62.0→82.5	9.5→41.0	87.5→58.5	2.0→0.0	85.0→66.0	31.0→55.0	29.0→33.8
1.4B	24	β=0.5	200	10.0→53.0	6.5→25.5	91.0→73.5	1.5→0.0	81.0→91.0	19.0→7.0	13.8→12.6
1.4B	24	β=1	200	10.0→78.5	6.5→35.5	91.0→63.5	1.5→0.5	81.0→85.0	19.0→15.0	13.8→12.8
2.6B	8	β=0.5	64	76.6→71.9	73.4→68.8	4.7→4.7	15.6→18.8	15.6→15.6	18.8→15.6	69.8→69.6
2.6B	8	β=1	64	76.6→68.8	73.4→65.6	4.7→4.7	15.6→21.9	15.6→9.4	18.8→12.5	69.8→69.8
2.6B	16	β=0.5	200	82.0→81.0	74.5→72.0	18.0→19.0	5.5→7.0	46.0→35.0	19.0→14.0	56.2→56.6
2.6B	16	β=1	200	82.0→80.0	74.5→72.5	18.0→16.0	5.5→10.5	46.0→41.0	19.0→12.0	56.2→57.0
2.6B	24	β=0.5	64	79.7→81.2	43.8→37.5	51.6→60.9	1.6→1.6	81.2→71.9	9.4→12.5	26.2→27.6
2.6B	24	β=1	64	79.7→78.1	43.8→37.5	51.6→57.8	1.6→1.6	81.2→71.9	9.4→9.4	26.2→24.4

Limits and axis specificity. Ouro-2.6B retains 96%/84% of its harmful mean reference projection at loops 16/24, but correction does not improve effective refusal consistently (Tables 8–11). In the main protocol, $T = 8$, $\beta = 1$ lowers effective refusal by 7.8 pp, while $T = 16$, $\beta = 1$ raises harmful completion from 5.5% to 10.5% (difference 5.0 pp, CI [1.5, 9.0]). Projection drift therefore is neither a complete explanation of degeneration nor a validated predictor of benefit.

Repeat batch: deep-loop recovery and axis specificity. At $T = 24$, $\beta = 1$ gives $\Delta Q = +48.4$ pp $[39.1, 57.8]$ and $\Delta D = -44.5$ pp $[-54.7, -35.2]$; ΔQ is $+5.5$ pp at $T = 16$ and 0 pp at $T = 8$ (Table 11). Random-axis compensation at $\beta = 0.5$ also significantly reduces degeneration at $T = 24$ and partly preserves the $\widehat{w}^*$ projection: retention at loop 24 is 0.40 versus 0.03 without control. Its XSTest over-refusal is 21.9%, compared with 7.8% for $\widehat{w}^*$ at $\beta = 1$. This control uses one random axis, seed 0.

Table 9: Jailbreak success under the main protocol (off→on; non-refusing, non-degenerate completions judged harmful by Llama Guard 3; GCG scored heuristically). n is the number of template/JBB-PAIR prompts; adaptive reports successful goals out of 50. ASR columns are percentages; adaptive results remain success counts out of 50.

Model	T	Cond.	n	Templates (%)	In-the-wild (%)	JBB-PAIR (%)	Adaptive	GCG (%)
1.4B	8	β=0.5	100/82	29.0→32.0	9.0→10.0	59.8→54.9	3/50→4/50	35.0→10.0
1.4B	8	β=1	100/82	29.0→30.0	9.0→11.0	59.8→51.2	3/50→5/50	35.0→10.0
1.4B	16	β=0.5	100/82	1.0→2.0	0.0→0.0	2.4→2.4	0/50→0/50	5.0→5.0
1.4B	16	β=1	100/82	1.0→3.0	0.0→0.0	2.4→2.4	0/50→0/50	5.0→0.0
1.4B	24	β=0.5	100/82	1.0→0.0	0.0→0.0	3.7→4.9	3/50→0/50	5.0→5.0
1.4B	24	β=1	100/82	1.0→2.0	0.0→0.0	3.7→4.9	3/50→0/50	5.0→5.0
2.6B	8	β=0.5	32/32	53.1→50.0	37.5→34.4	40.6→50.0	5/50→4/50	65.0→75.0
2.6B	8	β=1	32/32	53.1→53.1	37.5→28.1	40.6→46.9	5/50→5/50	65.0→80.0
2.6B	16	β=0.5	100/82	15.0→19.0	12.0→13.0	39.0→32.9	5/50→3/50	35.0→30.0
2.6B	16	β=1	100/82	15.0→18.0	12.0→13.0	39.0→39.0	5/50→3/50	35.0→25.0
2.6B	24	β=0.5	32/32	9.4→6.2	0.0→0.0	6.2→9.4	0/50→1/50	25.0→20.0
2.6B	24	β=1	32/32	9.4→0.0	0.0→0.0	6.2→9.4	0/50→0/50	25.0→5.0

Table 10: Paired differences (on−off) and 95% intervals for key main-protocol metrics (harmful sets resampled within each dataset and averaged with equal weight; 10,000 resamples, seed 0). Differences and interval endpoints are in percentage points (pp).

Model	T	Cond.	Δ Effective refusal (pp)	ΔD_H (pp)	ΔO (pp)
1.4B	8	β=0.5	−0.5 [−3.5, 2.5]	+1.0 [−1.5, 4.0]	+3.0 [0.0, 7.0]
1.4B	8	β=1	+1.0 [−2.5, 4.5]	+1.5 [−1.5, 4.5]	+3.0 [0.0, 7.0]
1.4B	16	β=0.5	+0.0 [−5.5, 5.5]	+2.0 [−4.0, 8.0]	+14.0 [5.0, 23.0]
1.4B	16	β=1	+31.5 [24.0, 39.0]	−29.0 [−36.5, −21.0]	+24.0 [15.0, 34.0]
1.4B	24	β=0.5	+19.0 [12.0, 26.0]	−17.5 [−25.0, −10.0]	−12.0 [−19.0, −5.0]
1.4B	24	β=1	+29.0 [22.0, 36.5]	−27.5 [−35.0, −20.5]	−4.0 [−13.0, 6.0]
2.6B	8	β=0.5	−4.7 [−10.9, 0.0]	+0.0 [0.0, 0.0]	−3.1 [−9.4, 0.0]
2.6B	8	β=1	−7.8 [−14.1, −1.6]	+0.0 [0.0, 0.0]	−6.2 [−15.6, 0.0]
2.6B	16	β=0.5	−2.5 [−8.0, 3.0]	+1.0 [−5.0, 7.0]	−5.0 [−10.0, 0.0]
2.6B	16	β=1	−2.0 [−8.0, 4.0]	−2.0 [−8.5, 4.0]	−7.0 [−13.0, −2.0]
2.6B	24	β=0.5	−6.2 [−23.4, 9.4]	+9.4 [−7.8, 26.6]	+3.1 [0.0, 9.4]
2.6B	24	β=1	−6.2 [−20.3, 7.8]	+6.2 [−9.4, 21.9]	+0.0 [0.0, 0.0]

D.1 Extended Control Batch

Protocol. Both extended runs ($T \in \{8, 16\}$ and $T \in \{24, 32\}$) use 64 prompts each from AdvBench, HarmBench-test, and XSTest-safe, greedy generation of at most 128 new tokens, Llama Guard 3, and 200 shared 5-shot MMLU items. Paired bootstrap uses 2,000 resamples, base seed 0, and depth-specific seed T. Controlled $A(T)$ uses a separate representation set with 100 examples per class. The $\beta = 1$ and clamp implementations agree bit for bit where both are saved. Table 12 includes all conditions, including $T = 8$ and sustained ablation.

Paired intervals. For $\beta = 1$, effective-refusal changes are $+2.3$ pp $[-7.8, 11.7]$ at $T = 16$, $+47.7$ pp [39.8,56.2] at $T = 24$, and $+27.3$ pp [19.5,35.9] at $T = 32$. The random-axis changes at $T = 24/32$ are $+36.7$ pp [28.1,45.3] and $+5.5$ pp [0.8,10.2], with XSTest non-degenerate over-refusal increasing by 12.5 pp [3.1,23.4] and 10.9 pp [4.7,18.8]. At $T = 16$, $\beta = 1$ raises that over-refusal measure by 9.4 pp [0.0,20.3]. Persistent ablation reduces harmful effective refusal to at most 2.3%; its low harmful completion at $T \geq 16$ accompanies degeneration.

Non-specific reduction of degeneration along a random axis. At $T = 24$, random-axis compensation reduces harmful and benign degeneration (D: $92.2\% \to 52.3\%$; O_D: $87.5\% \to 46.9\%$), while raising benign raw refusal to 26.6%. Along $\widehat{w}^*$, $\beta = 1$ reduces harmful degeneration ($92.2\% \to 50.8\%$) but raises benign degeneration ($87.5\% \to 93.8\%$). The random-axis benefit weakens at $T = 32$ (D: $97.7\% \to 85.9\%$; $Q = 6.2\%$), while $\widehat{w}^*$-axis refusal gains persist

Table 11: Repeat batch (128 tokens, 64 prompts per set). "Harmful" merges AdvBench and HarmBench, "Templates" merges fixed templates and in-the-wild, XS is XSTest-safe; AUC is last-loop harmful/harmless separability on the frozen direction; the last column is the fraction of the harmful mean coordinate at loop T retained relative to the anchor. Rates and accuracies are percentages; AUC and projection-retention ratios are unscaled.

Model	T	Cond.	Harm. effective refusal (Q, %)	Harm. D (%)	Template effective refusal (Q, %)	Tmpl. D (%)	XS R (%)	XS D (%)	MMLU (%)	AUC	$\bar{q}_T/\bar{q}_a$
1.4B	8	off	84.4	0.8	54.7	20.3	9.4	3.1	64.5	0.997	0.90
		β=0.5	83.6	0.0	58.6	18.0	10.9	3.1	64.5	0.997	0.96
		β=1	84.4	0.8	62.5	17.2	10.9	3.1	64.0	0.996	1.00
		random β=0.5	84.4	0.8	56.2	18.8	9.4	1.6	64.5	0.997	0.90
1.4B	16	off	26.6	71.1	35.2	61.7	32.8	78.1	44.5	0.991	0.53
		β=0.5	19.5	78.9	31.2	67.2	43.8	67.2	47.0	0.996	0.78
		β=1	32.0	68.0	36.7	60.2	45.3	53.1	47.5	0.996	1.00
		random β=0.5	31.2	64.1	30.5	66.4	15.6	78.1	49.5	0.989	0.52
1.4B	24	off	1.6	93.8	32.8	65.6	3.1	96.9	25.5	0.664	0.03
		β=0.5	27.3	69.5	44.5	55.5	7.8	89.1	24.5	0.995	0.68
		β=1	50.0	49.2	48.4	51.6	7.8	93.8	28.5	0.996	1.00
		random β=0.5	45.3	43.0	51.6	43.0	21.9	48.4	28.5	0.948	0.40
2.6B	16	off	75.0	7.8	51.6	25.8	15.6	12.5	58.5	0.993	0.96
		β=0.5	75.0	10.2	43.8	35.2	14.1	10.9	57.5	0.996	0.98
		β=1	75.0	8.6	51.6	28.1	14.1	12.5	57.5	0.997	1.00
		random β=0.5	76.6	7.0	45.3	32.8	15.6	7.8	59.5	0.992	0.98
2.6B	24	off	48.4	42.2	36.7	51.6	9.4	60.9	31.0	0.990	0.84
		β=0.5	50.8	42.2	42.2	49.2	9.4	64.1	31.0	0.997	0.92
		β=1	46.9	46.1	43.8	52.3	12.5	56.2	33.5	0.997	1.00
		random β=0.5	41.4	50.8	33.6	55.5	9.4	54.7	33.0	0.990	0.82

($Q = 23.4\%/28.1\%$). These differences support a direction-specific component without restricting deep-loop drift to one axis. The random control is a single axis (seed 0).

Reproducibility. The supplementary code includes inference controls, evaluation and analysis scripts, and available anonymized records for the reported batches. The README maps experiments to entry points and inputs. Saved item-level records support recomputation of behavioral rates and paired intervals; projection summaries and aggregate-only capability results are verified against saved outputs.

Table 12: All Ouro-1.4B anchor-compensation conditions. Q and D are non-degenerate refusal and total degeneration on the combined harmful set ($n = 128$); O and O_D are raw refusal and degeneration on XSTest-safe ($n = 64$); J is the Llama Guard 3 harmful-completion bin rate; LO is the mean refusal-opening log-odds on AdvBench; MMLU uses the same 200 items (persistent ablation was not evaluated on MMLU). Rates and accuracies are percentages; AUC and log-odds are unscaled.

T	Condition	$A(T)$	Effective refusal (Q, %)	D (%)	O (%)	O_D (%)	J (%)	LO	MMLU (%)
8	No compensation	0.997	83.6	0.8	9.4	1.6	10.9	+0.73	64.5
	$\beta = 0.5$	0.997	82.8	0.0	10.9	3.1	10.9	+0.88	64.5
	$\beta = 1$	0.996	83.6	0.0	10.9	3.1	10.9	+1.00	64.5
	Random axis $\beta = 0.5$	0.996	83.6	0.8	9.4	1.6	10.9	+0.73	64.5
	Persistent ablation	0.550	0.8	10.2	0.0	6.2	82.0	−3.03	–
16	No compensation	0.992	28.1	68.8	29.7	75.0	2.3	−0.47	44.5
	$\beta = 0.5$	0.996	18.8	78.9	43.8	67.2	1.6	−0.07	46.5
	$\beta = 1$	0.996	30.5	69.5	43.8	54.7	0.0	+0.66	48.0
	Random axis $\beta = 0.5$	0.990	34.4	58.6	17.2	75.0	3.9	−0.40	49.5
	Persistent ablation	0.524	2.3	65.6	4.7	73.4	26.6	−3.72	–
24	No compensation	0.658	0.8	92.2	7.8	87.5	4.7	−4.78	28.5
	$\beta = 0.25$	0.992	20.3	76.6	6.2	93.8	1.6	−2.35	23.5
	$\beta = 0.5$	0.995	28.9	67.2	6.2	90.6	3.1	−1.16	24.5
	$\beta = 1$	0.996	48.4	50.8	9.4	93.8	0.8	−0.53	27.0
	Random axis $\beta = 0.5$	0.954	37.5	52.3	26.6	46.9	8.6	−1.98	28.0
	Persistent ablation	0.395	0.0	96.1	0.0	89.1	3.1	−7.23	–
32	No compensation	0.633	0.8	97.7	0.0	100.0	1.6	−6.16	22.0
	$\beta = 0.25$	0.994	7.0	91.4	1.6	96.9	1.6	−3.85	20.5
	$\beta = 0.5$	0.996	23.4	71.1	3.1	92.2	3.9	−2.73	23.0
	$\beta = 1$	0.996	28.1	69.5	1.6	100.0	1.6	−2.36	26.5
	Random axis $\beta = 0.5$	0.904	6.2	85.9	17.2	64.1	7.0	−4.23	22.5
	Persistent ablation	0.505	0.0	98.4	1.6	96.9	1.6	−8.24	–

E Dense-to-Loop Conversion and Training

Conversion objective. Each dense backbone's full layer stack becomes the shared recurrent body F_θ. Before post-training, $T = 1$ matches the dense forward pass and $T > 1$ repeats it. Post-training changes the shared parameters, including the first-loop computation. To train repeated application, we use the frozen dense model as teacher:

$$\mathcal{L}_{\text{align}} = \sum_{t=1}^{T_{\text{train}}} \omega_t \left[\lambda_{\text{CE}} \mathcal{L}_{\text{CE}}^{(t)} + \lambda_{\text{KL}} \text{KL}(p_D \| p_t) + \lambda_h (1 - \cos(h_t, h_D)) \right], \tag{7}$$

where p_D, h_D are the frozen teacher outputs. Main checkpoints train at $T_{\text{train}} = 4$ and extrapolate to $T = 8$. SafeLoop on/off comparisons hold the post-trained checkpoint fixed and switch only Eq. (5) at inference.

Qwen clean recurrent post-training uses Eq. (7). The recovery-training arm additionally includes

$$\mathcal{L}_{\text{rec}} = \frac{1}{T_{\text{train}} - t_0} \sum_{t=t_0+1}^{T_{\text{train}}} \frac{\sum_{i \in \mathcal{I}} [\text{sg}(q_i^c(t)) - q_i^p(t)]^2}{\sum_{i \in \mathcal{I}} [\text{sg}(q_i^c(t_0) - q_i^p(t_0))]^2}, \tag{8}$$

where q_i^c, q_i^p are the clean and perturbed trajectories and sg is stop-gradient. Recovery training is a separate extension; the main SafeLoop comparisons switch inference-time control on a fixed checkpoint.

Qwen repeats its full 28-layer stack and final norm without an adapter; cache-free evaluation reruns loops for each prefix. CE uses recorded next tokens, distillation samples positions, and state alignment uses the prompt end. Recovery uses a stop-gradient clean student reference, stops perturbed-pass gradients before the pulse, and covers all subsequent training loops.

Align-G uses general/raw data; Align-S additionally uses harmful/harmless data, pairing harmful requests with refusals. Rec adds Eq. (8) with weight 1; Rec+Skip additionally trains with compensation. All four script recipes use LoRA (Hu et al., 2022) rank 16, learning rate 10^{-4}, 3,000 steps, batch size 1, gradient accumulation 4, length 1,024, at most 128 CE positions, seed 0, and training depth 4. Pulses sample loops 1/2; training-time compensation uses $a = 3$, $\beta = 0.5$. Complete training arguments were not archived, so these are script settings; checkpoint comparisons use saved weights and evaluation configurations.

Rec+Skip "off" removes its training-time connection at inference. Standard capability evaluations allow 4,096-token inputs versus 2,048 in the switch protocol; small cross-protocol MMLU differences cannot be attributed solely to SafeLoop. The four-arm switch comparisons fix the protocol and use $\beta = 0.5$. Backbone-specific implementations and direction screening for Llama and Gemma are detailed in Appendices H and I.

F Complete Capability and Safety Metrics for Dense-to-Loop Models

Protocol and controls. Qwen Align-G and general-data Gemma train at $T_{\text{train}} = 4$; directions are selected on the original dense model and frozen. "dense" is the original model ($T = 1$). At the training depth, $a = 1$ uses the post-trained checkpoint's first-loop projection and corrects loops 2–4; extrapolation uses $a = 4$ to correct loops 5–8. Greedy generation allows 128 new tokens. Qwen uses 100 prompts each from AdvBench/HarmBench; Gemma uses 64 each. XSTest-safe, Alpaca, fixed templates, and JBB-PAIR each have 100 Qwen and 64 Gemma prompts. MMLU uses 200 shared 5-shot letter-scored items. Adaptive attacks search 50 goals with 5 queries per goal separately per condition. GCG replays 20 suffixes optimized against uncontrolled models: Qwen dense/T4/T8 suffixes target $T = 1, 4, 8$, respectively, and SafeLoop replays each depth's suffixes; Gemma dense/T4 suffixes target $T = 1, 4$, and both T8 conditions replay $T = 4$ suffixes. Llama Guard 3 judges all jailbreaks; Qwen dense/T4 harmful sets lack judgments (J = –). At $T = 8$, $a = 4$, Gemma's adaptive attacks yield 0/50 successes for each of $\beta = 0.5$ and $\beta = 1$ across 50 goals, alongside severe generation degeneration.

Table 13: Complete dense-to-loop metrics. SafeLoop 0.5/1 denotes its β; T4 uses $a = 1$ and T8 uses $a = 4$. O is the raw refusal rate on XSTest-safe, ASR is jailbreak success rate, and adaptive reports successful goals out of 50. Rates and accuracies are percentages; adaptive results remain success counts out of 50.

Condition	MMLU (%)	R (%)	Effective refusal (Q, %)	D_H (%)	J (%)	D_B (%)	O (%)	ASR tmpl. (%)	ASR JBB (%)	ASR GCG (%)	Adaptive
Qwen2.5-1.5B (Align-G)											
dense	60.0	96.0	96.0	0.0	–	8.0	54.0	44.0	46.0	90.0	5/50
T4	57.5	93.5	93.5	1.0	–	16.0	44.0	15.0	21.0	60.0	2/50
T4+SafeLoop 0.5	58.0	94.0	93.5	1.0	2.5	16.0	46.0	19.0	28.0	65.0	4/50
T4+SafeLoop 1	58.0	94.0	93.5	1.0	2.5	15.0	49.0	21.0	32.0	55.0	3/50
T8	42.5	75.5	57.0	36.5	3.5	47.0	48.0	1.0	1.0	25.0	0/50
T8+SafeLoop 0.5	41.5	82.5	64.0	31.5	2.5	40.0	56.0	0.0	1.0	10.0	0/50
T8+SafeLoop 1	41.5	88.0	66.0	30.5	2.0	49.0	54.0	0.0	1.0	0.0	0/50
Gemma-2-2B											
dense	55.5	100.0	100.0	0.0	0.0	6.2	10.9	9.4	20.3	55.0	3/50
T4	52.5	90.6	90.6	0.8	5.5	4.7	7.8	9.4	42.2	45.0	2/50
T4+SafeLoop 0.5	53.0	97.7	97.7	1.6	0.0	4.7	10.9	4.7	23.4	20.0	3/50
T4+SafeLoop 1	53.0	97.7	97.7	1.6	0.0	4.7	15.6	4.7	21.9	5.0	2/50
T8	28.0	43.8	10.2	86.7	0.8	87.5	4.7	0.0	0.0	0.0	0/50
T8+SafeLoop 0.5	28.0	80.5	17.2	82.0	0.0	82.8	1.6	0.0	0.0	0.0	0/50
T8+SafeLoop 1	29.5	84.4	15.6	84.4	0.0	90.6	0.0	0.0	0.0	0.0	0/50

Dose trades off refusal and over-refusal. At training depth, $\beta = 0.5$ leaves Gemma's XSTest over-refusal at the dense value (10.9%), while $\beta = 1$ raises it to 15.6%; GCG success is 20% and 5%, respectively. Under extrapolation, Qwen has slightly higher Q with $\beta = 1$ (66.0% versus 64.0%), while $\beta = 0.5$ raises XSTest over-refusal (+8.0 pp [1.0, 15.0]).

Per-prompt drift differs across backbones. Figure 4 pairs uncontrolled $T = 4$ projections of 20 GCG prompts with off/on judge labels ($a = 1$, $\beta = 1$). Coordinates are measured at the last prefill

Table 14: Paired differences (latter−former) and 95% intervals for key dense-to-loop comparisons ($\beta = 1$; harmful sets resampled within each dataset and averaged with equal weight; 10,000 resamples, seed 0). Differences and interval endpoints are in percentage points (pp).

Refusal, generation degeneration, and over-refusal

Model	Comparison	Δ Effective refusal (pp)	ΔD_H (pp)	ΔO (pp)
Qwen	dense→T4	−2.5 [−6.0, 0.5]	+1.0 [0.0, 2.5]	−10.0 [−19.0, −1.0]
Qwen	T4→T4+SafeLoop	+0.0 [−2.0, 2.0]	+0.0 [−1.5, 1.5]	+5.0 [−1.0, 11.0]
Qwen	dense→T4+SafeLoop	−2.5 [−5.5, 0.5]	+1.0 [0.0, 2.5]	−5.0 [−12.0, 2.0]
Qwen	T4→T8	−36.5 [−43.5, −29.5]	+35.5 [29.0, 42.0]	+4.0 [−4.0, 12.0]
Qwen	T8→T8+SafeLoop	+9.0 [1.0, 17.0]	−6.0 [−14.0, 2.0]	+6.0 [−3.0, 15.0]
Gemma	dense→T4	−9.4 [−14.8, −4.7]	+0.8 [0.0, 2.3]	−3.1 [−12.5, 4.7]
Gemma	T4→T4+SafeLoop	+7.0 [3.1, 11.7]	+0.8 [0.0, 2.3]	+7.8 [1.6, 15.6]
Gemma	dense→T4+SafeLoop	−2.3 [−5.5, 0.0]	+1.6 [0.0, 3.9]	+4.7 [−1.6, 10.9]
Gemma	T4→T8	−80.5 [−87.5, −73.4]	+85.9 [79.7, 91.4]	−3.1 [−7.8, 0.0]
Gemma	T8→T8+SafeLoop	+5.5 [−0.8, 11.7]	−2.3 [−9.4, 4.7]	−4.7 [−10.9, 0.0]

Jailbreak success and capability

Model	Comparison	ΔASR tmpl. (pp)	ΔASR JBB (pp)	ΔMMLU (pp)
Qwen	dense→T4	−29.0 [−38.0, −21.0]	−25.0 [−36.0, −14.0]	−2.5 [−8.5, 3.5]
Qwen	T4→T4+SafeLoop	+6.0 [−1.0, 13.0]	+11.0 [2.0, 20.0]	+0.5 [0.0, 1.5]
Qwen	dense→T4+SafeLoop	−23.0 [−32.0, −15.0]	−14.0 [−24.0, −5.0]	−2.0 [−8.0, 4.0]
Qwen	T4→T8	−14.0 [−22.0, −7.0]	−20.0 [−28.0, −13.0]	−15.0 [−22.5, −7.5]
Qwen	T8→T8+SafeLoop	−1.0 [−3.0, 0.0]	+0.0 [−3.0, 3.0]	−1.0 [−3.5, 1.5]
Gemma	dense→T4	+0.0 [−7.8, 7.8]	+21.9 [10.9, 34.4]	−3.0 [−9.0, 2.5]
Gemma	T4→T4+SafeLoop	−4.7 [−10.9, 0.0]	−20.3 [−31.2, −9.4]	+0.5 [−1.0, 2.0]
Gemma	dense→T4+SafeLoop	−4.7 [−12.5, 1.6]	+1.6 [−6.2, 9.4]	−2.5 [−8.0, 3.0]
Gemma	T4→T8	−9.4 [−17.2, −3.1]	−42.2 [−54.7, −29.7]	−24.5 [−32.0, −17.0]
Gemma	T8→T8+SafeLoop	+0.0 [0.0, 0.0]	+0.0 [0.0, 0.0]	+1.5 [−0.5, 4.0]

token of loops 1–4. Qwen's loop means are 8.61, 7.84, 9.38, and 8.52; median $q_1 - q_4$ is -2.0, with 30% of prompts declining. Of 12 jailbroken prompts, 9 remain jailbroken, 2 become refusals, and 1 becomes coherent and harmless. Gemma's means are 3.83, -0.94, -4.23, and -2.90; median drop is 8.7, with 75% declining. Of 9 jailbroken prompts, 8 become refusals; 6 coherent harmless and 2 degenerate responses also become refusals. This descriptive association uses 20 GCG prompts per model; JBB-PAIR lacks per-loop coordinates, and the analysis is not causal.

G Control Schemes and Training Variants

Checkpoints and training background. Table 15 groups six frozen recurrent checkpoints by training setup. Qwen Align-G, Llama, and Gemma use general-data alignment without recovery loss or training-time compensation. Align-S adds safety data; Rec adds Eq. (8) to Align-G; Rec+Skip also trains with compensation (Appendix E). Columns hold the checkpoint and prompts fixed and vary inference control. Rec+Skip "off" removes its training-time connection, so re-enabling it does not establish a gain independent of related training.

A shared control protocol. At $T = 8$, $\beta = 0.5$, persistent control uses anchor $a = 3$ at loops 4–8; one-shot control at t uses the preceding loop $t-1$, with $t = 8$ shown in the table. The random control uses one fixed axis orthogonal to w^*. This earlier protocol uses $a = 3$, distinct from training-depth $a = 1$ and extrapolation $a = 4$ in the main comparison. Each Qwen checkpoint uses 32 prompts per AdvBench, HarmBench-test, and XSTest-safe; Llama and Gemma use 64 per set. These estimates are batch- and anchor-specific. One-shot and persistent control also differ in reference values and cumulative dose, so they compare complete schemes rather than isolate frequency.

Checkpoint-specific effects. Persistent control exceeds final-loop one-shot effective refusal by 15.6 pp [1.6, 29.7] on Qwen Rec and 6.2 pp [2.3, 10.9] on Llama; other checkpoint intervals include zero. Align-S instead falls from 92.2% without control to 82.8% with persistent control. These fixed-checkpoint comparisons do not establish uniform benefit.

Table 15: Inference-time control comparisons on the same frozen checkpoint ($T = 8$, $\beta = 0.5$, reference loop $a = 3$ for persistent compensation). Values are effective refusal (Q) rates on harmful prompts; "one-shot" is final-loop compensation using the preceding loop as reference. Checkpoints are grouped by training background; between-row differences do not identify the causal effect of an individual training factor. Refusal rates are percentages; differences are in percentage points (pp).

Checkpoint	Off (%)	One-shot (%)	Random (%)	Persistent (%)	Persist.−one-shot (pp)	Persist.−random (pp)
Basic dense-to-loop alignment training						
Qwen Align-G	54.7	54.7	50.0	59.4	+4.7	+9.4
Llama	82.8	83.6	83.6	89.8	+6.2	+6.2
Gemma	10.2	14.1	7.8	17.2	+3.1	+9.4
Safety data or additional recovery training						
Qwen Align-S	92.2	85.9	85.9	82.8	−3.1	−3.1
Qwen Rec	42.2	45.3	29.7	60.9	+15.6	+31.2
Qwen Rec+Skip	62.5	73.4	71.9	78.1	+4.7	+6.2

Capability and benign-input controls. Table 16 reports the matched Qwen schemes with capability, degeneration, over-refusal, and harmful completion. One-shot control is applied at the final loop; persistent control fixes anchor 3. Conditions within each checkpoint use identical prompts.

Table 16: Matched control schemes at $T = 8$. Rates and accuracies are percentages; AUC is unscaled.

Checkpoint	Condition	MMLU (%)	AUC	Effective refusal (Q, %)	D (%)	O (%)	J (%)
Align-S	Off	41.5	0.993	92.2	6.2	71.9	0.0
	Persistent connection	42.0	0.987	82.8	15.6	68.8	0.0
	Random-axis connection	42.5	0.993	85.9	12.5	68.8	0.0
	Persistent ablation	46.5	0.539	9.4	68.8	3.1	3.1
	One-shot loop 8	40.5	0.993	85.9	12.5	71.9	0.0
Align-G	Off	42.0	0.979	54.7	40.6	31.2	0.0
	Persistent connection	41.5	0.987	59.4	34.4	34.4	0.0
	Random-axis connection	43.0	0.979	50.0	37.5	28.1	0.0
	Persistent ablation	40.5	0.475	4.7	67.2	3.1	1.6
	One-shot loop 8	42.0	0.985	54.7	37.5	37.5	1.6
Rec	Off	31.0	0.599	42.2	50.0	37.5	0.0
	Persistent connection	30.5	0.872	60.9	34.4	59.4	0.0
	Random-axis connection	31.0	0.604	29.7	64.1	46.9	0.0
	Persistent ablation	34.0	0.544	28.1	68.8	56.2	0.0
	One-shot loop 8	30.0	0.753	45.3	48.4	43.8	0.0
Rec+Skip	Off	41.0	0.917	62.5	25.0	21.9	0.0
	Persistent connection	41.0	0.952	78.1	14.1	37.5	1.6
	Random-axis connection	40.5	0.919	71.9	25.0	25.0	0.0
	Persistent ablation	42.0	0.495	26.6	48.4	25.0	3.1
	One-shot loop 8	40.5	0.925	73.4	20.3	21.9	0.0

Persistent minus final-loop one-shot control gives, in pp, Rec $\Delta Q = 15.625$, CI $[1.5625, 29.6875]$, and Rec+Skip 4.6875, CI $[-7.8125, 17.1875]$. MMLU comparisons hold each trained checkpoint fixed; their paired intervals do not establish non-inferiority with a one-percentage-point margin. With control enabled, all XSTest outputs from both recovery arms reach the 128-token budget, qualifying the over-refusal results.

H LLAMA CONVERSION AND CONTROL

Model, training, and direction. Llama-3.2-1B-Instruct repeats all 16 layers with $\alpha = 0$; its untrained one-loop wrapper matches dense logits exactly. The depth-4 checkpoint uses general/raw training, equally weighted CE/KL/cosine losses, and the LoRA recipe in Appendix E, without recovery loss or training-time compensation. Direction fitting uses 64 harmful and 64 harmless examples, 32 validation examples per class, and offsets $-6, \ldots, -1$. Selected offset -1 gives harmful refusal-opening score $5.202 \rightarrow -1.850$, benign addition score $-8.177 \rightarrow 1.776$, and benign KL 0.0796.

Table 17 evaluates that checkpoint at $T = 8$ under the protocol in Appendix G: 64 prompts per AdvBench/HarmBench/XSTest-safe, 128 generated tokens, and 200 MMLU items with 5-shot prompting and 2,048-token inputs. Persistent control uses $a = 3$, $\beta = 0.5$; final-loop one-shot at $t = 8$ uses loop 7 as its reference. Each generated token rereads its prefix anchor.

Table 17: Matched Llama control schemes, with rates and accuracies shown as percentages and AUC unscaled. Q is harmful non-degenerate refusal, D is total harmful degeneration, J is the judge-based harmful-completion bin rate, and O is raw refusal on XSTest. AUC is computed on an independent representation set and does not constitute item-level refusal prediction.

Condition	MMLU (%)	Final AUC	Effective refusal (Q, %)	D (%)	J (%)	O (%)
No intervention	41.00	0.9995	82.81	2.34	5.47	4.69
Persistent gap 0.5	41.00	0.9998	89.84	0.78	3.12	4.69
Persistent random axis	40.50	0.9995	83.59	1.56	5.47	4.69
Full-scope persistent ablation	40.50	0.8804	67.97	0.78	7.81	1.56
One-shot $t = 8$	41.00	0.9995	83.59	1.56	6.25	4.69

Paired inference and scope. Seed-0 bootstrap uses 10,000 paired resamples, stratifying harmful datasets. Persistent versus no control gives $\Delta Q = 7.03$ pp [3.13,11.72], $\Delta D = -1.56$ pp $[-3.91, 0]$, and $\Delta J = -2.34$ pp $[-6.25, 0.78]$. Persistent exceeds final-loop one-shot and random-axis control by 6.25 pp [2.34,10.94]. These comparisons do not isolate timing from donor and dose, include no training randomness, and test direct requests without adaptive attacks.

MMLU is 82/200 in both conditions, with two changed items (difference CI $[-1.5, 1.5]$ pp), insufficient for a one-point non-inferiority margin. XSTest refusal is 3/64 in both with identical labels; a zero-width sample interval does not establish population equivalence. Outputs reach the token limit in 63/64 and 62/64 cases, leaving benign-answer quality unresolved.

I GEMMA IMPLEMENTATION AND CONTROL

Native structure and training. Gemma-2-2b-it repeats all 26 layers with native one-time $\sqrt{d}$ embedding scaling, pre/post attention and FFN norms, alternating sliding-window/full attention, and attention/final-logit softcaps. The normalized loop-end state enters the next loop; all runs use eager attention. With real weights, $T = 1, \alpha = 0$ matches dense logits exactly. The depth-4 checkpoint uses general/raw data, equally weighted CE/KL/cosine losses, LoRA rank 16, learning rate 10^{-4}, 3,000 steps, and seed 0, without recovery loss or training-time compensation.

Direction screening. Dense final-state direction fitting uses 64 harmful and 64 harmless examples, 32 validation examples per class, and offsets $-6, \ldots, -1$. Selected offset -1 passes all four behavioral criteria: harmful refusal-opening score $5.964 \rightarrow -0.844$, benign addition score $-8.357 \rightarrow 2.597$, and benign KL 0.0919. The frozen direction is evaluated on 64 representation examples per class and independent AdvBench/HarmBench/XSTest sets of 64 prompts each.

Deep-loop anchor preservation and scope. For the train-depth-4 model at $T = 8$, control with $a = 3, \beta = 0.5$ gives $Q = 22/128$ versus off $13/128$ (gain CI $[0, 14.1]$ pp; Table 15). Final-loop one-shot gives $18/128$, so persistent minus one-shot is 3.1 pp (CI $[-5.5, 11.7]$ pp). Harmful comple-

tion is zero off and on, while XSTest degeneration is $62/64$ in both; low over-refusal accompanies severe generation failure. The control comparisons use paired item-level records.

Item-level labels and correctness records reproduce control rates and intervals; manifests record model revisions, settings, source hashes, and implementation checks.

J Scope of the Anchor Correction

Proposition 1 (Coordinate error after anchor correction)**.** *For a fixed anchor q_a and unit vector $\widehat{w}^*$, the control map obeys the coordinate-error relation below. If $0 < \beta < 2$, this step strictly reduces any nonzero deviation from the anchor along the axis, but does not guarantee strict contraction of the full state or of the subsequent composed update.*

Let $P_\perp = I - \widehat{w}^*(\widehat{w}^*)^\top$ denote the orthogonal projector. The result follows from $(\widehat{w}^*)^\top \widehat{w}^* = 1$ and $P_\perp \widehat{w}^* = 0$:

$$(\widehat{w}^*)^\top C_\beta(h) - q_a = (\widehat{w}^*)^\top h + \beta(q_a - (\widehat{w}^*)^\top h) - q_a = (1-\beta)((\widehat{w}^*)^\top h - q_a), \tag{9}$$

$$P_\perp C_\beta(h) = P_\perp h. \tag{10}$$

The equivalence $|1 - \beta| < 1$ if and only if $0 < \beta < 2$ concerns the coordinate multiplier of the control step. It guarantees reduction of nonzero anchor deviation, not behavioral stability.

For a fixed input and cached anchor, the derivative of the control step is $J_C = I - \beta \widehat{w}^*(\widehat{w}^*)^\top$. Applying correction after the loop update gives

$$J_{\text{control}} = \left(I - \beta \widehat{w}^*(\widehat{w}^*)^\top\right) J_F. \tag{11}$$

All-token control uses the block-diagonal extension. J_C multiplies the $\widehat{w}^*$ component by $1 - \beta$ and orthogonal components by 1; full-state stability also depends on J_F and its cross-couplings. Since anchors vary across inputs, this fixed-anchor derivative does not describe all input sensitivity.

If the shallow anchor is near a correct task target, sustained correction can reduce the extent to which later drift erodes the margin. It can also preserve shallow over-refusal or prevent a needed later revision, motivating joint measurement of refusal, over-refusal, and capability. Partial-contraction theory permits a property to stabilize while the full state evolves under suitable conditions (Wang & Slotine, 2005). The anchor lemma establishes projection correction; full-state convergence needs additional conditions.