Global PID Steering: Simpler Control, Better Capability Preservation in LLM Jailbreak Defense

Hank Sha
Department of Statistics and Applied Probability
University of California, Santa Barbara
hanksha@ucsb.edu

June 2026

Abstract

PID Steering [Nguyen et al., 2026] defends language models against jailbreak attacks by applying feedback control to refusal directions at every transformer layer. We propose a simpler variant—*Global PID Steering*—that uses a single averaged refusal vector applied over a small window of layers where the refusal feature is stable. On Gemma-2-2B-it, both methods block all 104 AdvBench GCG attacks (0% ASR), though this largely reflects the model's strong existing alignment (baseline ASR: 2.88%) and the use of a non-adaptive GCG suffix. The more informative finding is on capability preservation: under Activation Addition steering at scale $\alpha = 20$, per-layer PID refuses 96% of benign prompts, while Global PID refuses only 3.5% (vs. 1.5% unsteered). We also characterize integral windup behavior, confirming it is harmless at recommended gains but degrades coherence at higher settings. These results suggest that targeted steering within a verified feature-persistence window better preserves capability than full-depth per-layer intervention, though the comparison carries caveats we discuss.

1 Introduction

Large language models (LLMs) like ChatGPT and Gemma are trained to refuse harmful requests—for example, refusing to provide instructions for illegal activities. However, researchers have found that carefully crafted *jailbreak* prompts can bypass these safety measures and trick the model into producing harmful outputs [Zou et al., 2023].

One promising line of defense is *activation steering*: instead of retraining the model, we modify its internal computations during inference to reinforce the refusal behavior. The idea is straightforward: if we can identify the internal "direction" that corresponds to refusal, we can nudge the model's hidden states in that direction to make refusal more robust [Turner et al., 2024, Rimsky et al., 2024, Arditi et al., 2024].

Nguyen et al. [2026] took this idea further by framing activation steering as a feedback control problem. Drawing on PID (proportional-integral-derivative) control from engineering, they designed a controller that applies steering at every transformer layer using a different refusal direction at each layer. Their method reduces jailbreak success rates and comes with formal stability guarantees.

However, per-layer PID is complex: it requires computing and storing a separate refusal direction for each of the model's N layers, and running N independent control computations. We ask a natural question: **can a single, shared refusal direction work just as well?**

Recent work suggests it might. Arditi et al. [2024] showed that refusal in LLMs is largely mediated by a single direction, and Lindsey et al. [2024] found that many features persist across

adjacent transformer layers. Circuit-level analysis of LLM internals [Lindsey et al., 2025] further confirms that interpretable features form coherent cross-layer structures. If the refusal direction is roughly the same across a range of layers, averaging them into one direction should capture the essential information.

We propose **Global PID Steering**: compute a single global refusal vector $\bar{r}$ by averaging the per-layer directions over a verified window of layers where they are similar, then apply one PID controller using $\bar{r}$ across that window. This reduces storage by 26× (for Gemma-2-2B-it) and replaces 26 independent computations with a single formula.

Our main findings on Gemma-2-2B-it are:

- **Defense:** Both methods achieve 0% ASR under GCG attack, but this comparison is largely uninformative—Gemma-2-2B-it already blocks 97% of these (non-adaptive) attacks without any steering. The defense floor leaves little room for differentiation.
- **Capability:** This is where the methods diverge. Under Activation Addition (ActAdd) steering at scale $\alpha = 20$, per-layer PID refuses 96% of benign prompts, making it unusable. Global PID refuses only 3.5% (vs. 1.5% unsteered).
- **Windup analysis:** The integral term in our controller grows linearly with depth, but this is harmless at recommended gains and controllable with a simple clamp at higher settings.

An important caveat: per-layer PID's over-refusal may partly reflect our use of plain ActAdd rather than the Angular Steering paradigm used in the original paper [Nguyen et al., 2026, Vu and Nguyen, 2025]. We discuss this and other confounds in Section 6.

2 Background and Related Work

2.1 Activation Steering

Inside a transformer, each layer transforms a vector of hidden states called the *residual stream*. Activation steering works by adding a carefully chosen "steering vector" to this residual stream during inference, pushing the model's behavior in a desired direction without retraining.

Turner et al. [2024] introduced Activation Addition (ActAdd), which adds a fixed vector at a single layer. Rimsky et al. [2024] extended this using pairs of contrastive prompts. Arditi et al. [2024] made a key discovery: refusal behavior in aligned LLMs is controlled by a single direction in the model's activation space, which can be found by computing the *difference-in-means* (DIM) between activations on harmful versus harmless prompts.

In the framework of Nguyen et al. [2026], all of these methods amount to *proportional-only* control—they add a fixed multiple of the steering vector. PID Steering extends this by adding integral and derivative terms.

2.2 PID Steering

PID control is a standard technique from engineering that combines three terms to correct for errors. Nguyen et al. [2026] adapt this to activation steering by treating the transformer's layer index k as the "time" variable.

At each layer k, they first compute a *refusal direction* $r(k)$—the difference between average activations on harmful and harmless prompts at that layer. Then, the steering signal added to layer

k combines three terms:

$$u(k) = \underbrace{K_p\, r(k)}_{\text{proportional}} + \underbrace{K_i \sum_{j=0}^{k-1} r(j)}_{\text{integral}} + \underbrace{K_d\big(r(k) - r(k-1)\big)}_{\text{derivative}}, \tag{1}$$

where K_p, K_i, and K_d are tunable gain parameters. Intuitively:

- The **proportional term** (K_p) applies an immediate correction proportional to the current refusal direction.
- The **integral term** (K_i) accumulates the refusal signal from all previous layers, correcting for persistent under-steering.
- The **derivative term** (K_d) responds to how quickly the refusal direction is changing between layers, dampening abrupt shifts.

Nguyen et al. prove that this system is stable (bounded outputs for bounded inputs) under mild conditions on the gains, and show empirically that the integral and derivative terms improve over proportional-only steering.

2.3 Cross-Layer Feature Persistence

A key observation motivating our work: high-level features in transformers tend to persist across adjacent layers in the residual stream [Park et al., 2024, Lindsey et al., 2024]. This *linear representation hypothesis* suggests that if the refusal direction at layer 19 is similar to the one at layer 25, we can average them into a single representative direction without losing much information. Recent circuit-tracing work by Lindsey et al. [2025] provides further evidence that interpretable features—including safety-relevant ones—form coherent structures across layers. We note that not all features are strictly one-dimensional [Engels et al., 2025], but the refusal feature has been shown to be well-approximated by a single direction [Arditi et al., 2024], making linear averaging a reasonable choice.

2.4 Jailbreak Attacks

Zou et al. [2023] introduced the Greedy Coordinate Gradient (GCG) attack, which automatically finds short text suffixes that, when appended to a harmful prompt, cause the model to comply rather than refuse. These suffixes transfer across models and represent one of the strongest automated jailbreak methods. We use pre-computed GCG suffixes from their released set.

3 Method

3.1 Step 1: Verify Persistence Across Layers

Before building a global controller, we check whether the refusal direction is actually similar across layers. We compute the refusal direction at each layer by taking the difference between average activations on harmful and harmless prompts:

$$r(k) = \text{mean}\big(\text{harmful activations at layer } k\big) - \text{mean}\big(\text{harmless activations at layer } k\big). \tag{2}$$

More precisely, we normalize each prompt's activation vector to unit length before averaging, so that the difference captures direction rather than magnitude.

We then measure how similar these directions are across layers using cosine similarity. For every pair of layers (i, j), we compute $\cos(r(i), r(j))$ and arrange the results in a matrix. A block of layers with consistently high similarity (above a threshold τ) forms the *persistence window* W—the range of layers where the refusal feature is stable enough to average.

3.2 Step 2: Compute the Global Refusal Vector

Given the persistence window W, we compute a single global refusal vector by averaging the per-layer directions within the window:

$$\bar{r} = \frac{1}{|W|} \sum_{k \in W} r(k). \tag{3}$$

This produces one vector (instead of 26 separate ones) that captures the shared refusal signal. It is computed once from training data and reused for all subsequent inference.

3.3 Step 3: Apply Global PID Control

We apply PID control using $\bar{r}$ as the reference signal at every layer in the window W. Since $\bar{r}$ is the same at every layer (it does not change from layer to layer), the PID formula simplifies considerably.

The full PID signal at layer k within the window is:

$$u(k) = K_p\, \bar{r} + K_i \sum_{j=k_0}^{k} \bar{r} + K_d\big(\bar{r} - \bar{r}\big), \tag{4}$$

where k_0 is the first layer in the window. Because the reference signal $\bar{r}$ is constant across layers, two important simplifications occur:

1. The **derivative term vanishes**: $\bar{r} - \bar{r} = 0$, so the D-term contributes nothing.
2. The **integral term grows linearly**: adding $\bar{r}$ at each layer means the sum equals $(k - k_0 + 1) \cdot \bar{r}$.

This gives us a closed-form expression for the steering signal:

$$u(k) = \big[K_p + K_i \cdot (k - k_0 + 1)\big] \cdot \bar{r}. \tag{5}$$

In plain terms: the steering strength increases linearly as we go deeper into the window. The steering vector is then added to the model's hidden states at each layer k in the window, scaled by a global factor α:

$$h^{(k)} \leftarrow h^{(k)} + \alpha \cdot u(k). \tag{6}$$

3.4 Anti-Windup Variant

The linear growth of the integral term is a potential concern: if the integral gain K_i is too large, the accumulated signal could become so strong that it distorts the model's outputs (a phenomenon called *integral windup* in control theory). To guard against this, we test a variant that caps the integral accumulator at a maximum magnitude:

$$\text{If the integral exceeds } 2\,\|\bar{r}\|\,, \text{ rescale it back to } 2\,\|\bar{r}\|\,. \tag{7}$$

This allows the integral to contribute meaningfully while preventing runaway growth.

3.5 Computational Comparison

Per-layer PID stores 26 direction vectors (one per layer) and runs 26 PID computations per forward pass. Global PID stores a single vector and evaluates a scalar-times-vector product at 7 layers. For Gemma-2-2B-it, this represents a 26× reduction in storage and a substantially simpler computation.

4 Experimental Setup

4.1 Model

All experiments use **Gemma-2-2B-it** [Gemma Team, 2024], a 2-billion parameter instruction-tuned language model with 26 transformer layers. We use the HuggingFace `transformers` library with PyTorch forward hooks to intercept and modify the model's hidden states at each layer.

4.2 Datasets

Training data for refusal directions. 256 harmful prompts from AdvBench [Zou et al., 2023] paired with 256 benign instructions from Stanford Alpaca [Taori et al., 2023]. These are used to compute the DIM directions and the global vector $\bar{r}$.

Attack evaluation. 104 held-out AdvBench harmful prompts (not used for training). For GCG attacks, we append a pre-computed universal adversarial suffix from Zou et al. [2023] to each prompt.

Capability evaluation. 200 Alpaca instructions (disjoint from training) to test whether steering degrades the model's ability to respond to normal, harmless requests.

4.3 PID Gains and Scale

We use the gain values from Nguyen et al. [2026]: $K_p = 0.9$, $K_i = 0.01$, $K_d = 0.01$. **No gain tuning was performed**—we adopt their recommended values directly. The steering scale is $\alpha = 20$ for the main experiments, selected via a calibration sweep (Section 5.6).

4.4 Evaluation Metrics

Attack Success Rate (ASR). We check each model completion for 12 standard refusal phrases (e.g., "I'm sorry", "I cannot", "I apologize") from JailbreakBench [Chao et al., 2024]. A completion counts as a successful attack if *none* of these phrases appear. Lower ASR means stronger defense.

Over-refusal rate. The same refusal check applied to completions of *benign* prompts. If the model refuses a harmless request like "write me a poem," that is over-refusal—a sign that steering has damaged the model's usefulness. We flag any condition with over-refusal exceeding the baseline by more than 20 percentage points as DISQUALIFIED.

Perplexity (PPL). We measure how coherent the model's outputs are by computing their cross-entropy under a reference language model (Pythia-70M). High perplexity indicates gibberish or degenerate outputs.

4.5 Conditions Tested

We compare four conditions:

1. **No steering** — the unmodified model (baseline).
2. **Per-layer PID** — the method from Nguyen et al. [2026], applying PID with a separate refusal direction at all 26 layers.
3. **Global PID** — our method, applying PID with the single averaged vector $\bar{r}$ at layers 19–25 only.
4. **Global PID + anti-windup** — same as Global PID but with the integral clamp from eq. (7).

An important difference between conditions 2 and 3: per-layer PID steers at *all 26 layers*, while Global PID steers at only the *7 layers in the persistence window*. This difference in scope is deliberate—Global PID avoids perturbing layers where the refusal feature is not well-defined—but it means the two methods differ in both the steering vector and the number of layers affected. We revisit this in the Discussion.

5 Results

5.1 Persistence Window

Figure 1 shows the cosine similarity matrix of per-layer refusal directions for Gemma-2-2B-it. Each cell represents how similar the refusal direction at one layer is to the direction at another layer. The bright block in layers 19–25 indicates that these seven layers share a consistent refusal direction (mean cosine similarity: 0.76).

This persistence is real but moderate—our green-light threshold was 0.85, and 0.76 falls short of that. We proceed with this window but note it as a limitation: the global vector is a reasonable but imperfect approximation.

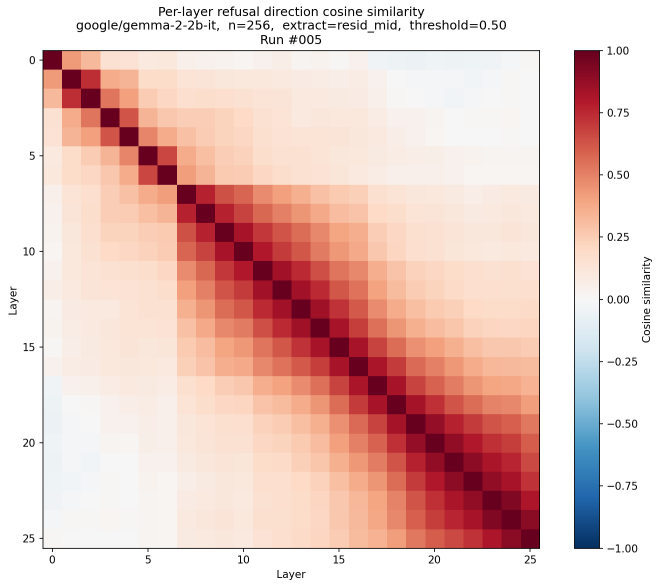

Figure 1: Cosine similarity of per-layer refusal directions in Gemma-2-2B-it (256 contrastive prompt pairs). The bright block in layers 19–25 is the persistence window W. Darker red indicates higher similarity. Mean intra-window cosine similarity: 0.76.

5.2 Main Result: Defense and Capability

Table 1 presents our central comparison. The defense results are straightforward: all three PID variants achieve 0% ASR under GCG attack, completely blocking the adversarial suffixes. The unsteered model allows 3 out of 104 attacks to succeed (2.88% ASR).

Table 1: Main comparison on Gemma-2-2B-it. ASR measured under GCG attack ($n = 104$). Capability measured on benign Alpaca prompts ($n = 200$). All PID conditions use $K_p = 0.9$, $K_i = 0.01$, $K_d = 0.01$, $\alpha = 20$.

Condition	**ASR ↓**	**Over-Refusal ↓**	**PPL (median)**	**Status**
No steering	2.88%	1.5%	19.97	baseline
Per-layer PID	0%	96.0%	20.37	DISQUALIFIED
Global PID	**0%**	**3.5%**	**20.37**	deployable
Global PID + AW	0%	3.5%	20.34	deployable

The decisive finding is on the capability side, shown in Figure 2. Per-layer PID refuses **96% of benign Alpaca instructions**—192 out of 200—and is flagged as DISQUALIFIED. A defense that blocks attacks but also blocks normal use is not deployable. Global PID, by contrast, maintains near-baseline capability: only 3.5% over-refusal (7 out of 200) versus 1.5% unsteered (3 out of 200). Perplexity is comparable across all non-disqualified conditions (median ≈ 20), confirming that Global PID does not produce gibberish.

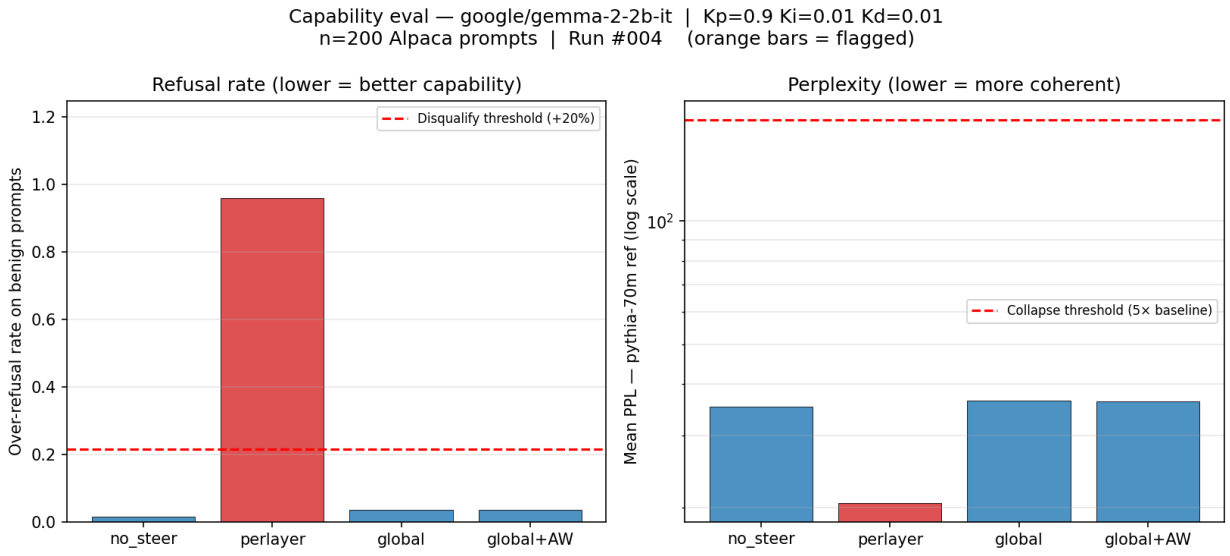

Figure 2: Capability evaluation on 200 benign Alpaca instructions. **Left:** Over-refusal rate. Per-layer PID (red bar) refuses 96% of harmless prompts, far exceeding the disqualification threshold (dashed line). Global PID variants remain near baseline. **Right:** Perplexity under a reference model. All conditions are well below the coherence-collapse threshold.

A note on the low baseline ASR. Gemma-2-2B-it is already well-aligned: even without any steering, only 2.88% of GCG attacks succeed (3 out of 104). Moreover, the GCG suffix we use is a universal suffix from Zou et al. [2023], originally optimized against Llama-2 and Vicuna—not Gemma-2. A suffix tailored to Gemma-2 would likely produce a higher baseline ASR, giving the defense methods more room to demonstrate their value. As it stands, the defense comparison is happening near the floor, and the 0% ASR shared by all PID variants is largely a reflection of the model's existing alignment rather than a strong signal about the controllers. The more informative comparison is on the capability axis, where the methods diverge dramatically.

5.3 Why Per-Layer PID Over-Refuses

Per-layer PID applies 26 different steering vectors across the entire depth of the network, each scaled by $\alpha = 20$. The cumulative effect of all these perturbations pushes the model into a state where nearly every input triggers refusal. This is consistent with the perplexity data: per-layer PID produces the *lowest* mean perplexity (20.54), which reflects the low entropy of repetitive refusal phrases like "I cannot and will not provide…"

Global PID avoids this problem by steering at only 7 layers (the persistence window) with a single direction. Figure 3 shows the size of the model's hidden states across layers: Global PID produces a modest shift within the window (layers 19–25, gray band) but returns to the unsteered trajectory by the final layer. The perturbation is contained rather than accumulating through the full network.

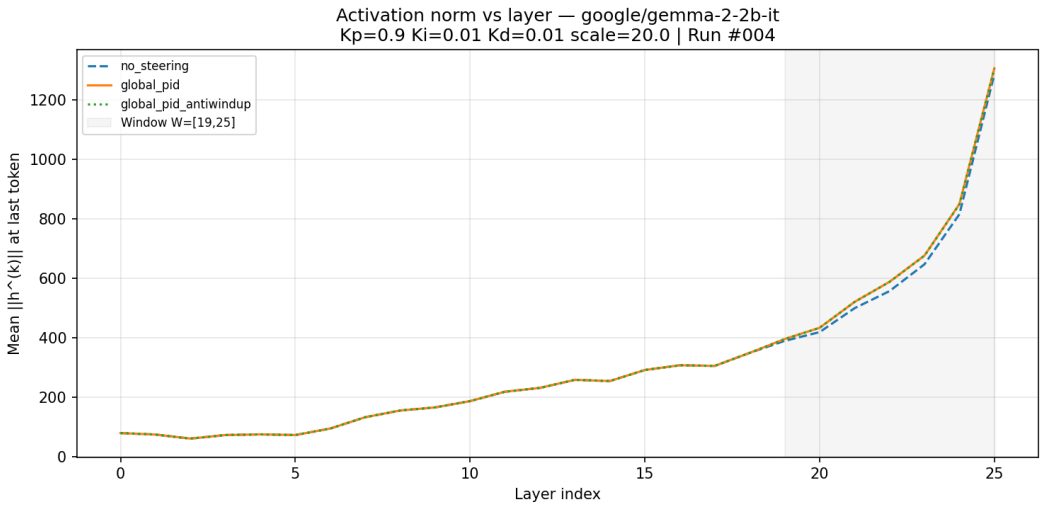

Figure 3: Hidden state magnitude across layers. Global PID (orange) and its anti-windup variant (green, dashed) produce a small shift within the persistence window (gray band) but converge back to the unsteered baseline (blue, dashed) by the final layer.

However, this comparison involves two differences at once: (1) the steering vector itself (per-layer directions vs. the global average) and (2) the scope of intervention (26 layers vs. 7 layers). The over-refusal gap is likely driven by scope more than vector choice. We discuss this further in Section 6.

5.4 Integral Windup Characterization

As noted in Section 3.3, the constant error signal in Global PID causes the integral term to grow linearly with depth. To find where this becomes problematic, we sweep the integral gain K_i from 0 to 0.15 while holding $K_p = 0.9$ and $K_d = 0$ fixed (at a higher scale of $\alpha = 50$ to stress-test the controller).

Figure 4 shows the results. The top panel confirms that ASR stays at 0% regardless of K_i—the proportional term alone handles defense. The bottom panel reveals the windup effect: for $K_i \geq 0.10$, the vanilla controller's perplexity spikes (mean PPL from 13 to 47), indicating that the accumulated integral signal is distorting outputs. The anti-windup variant (dashed line) suppresses this spike, confirming that the degradation is specifically caused by integral accumulation.

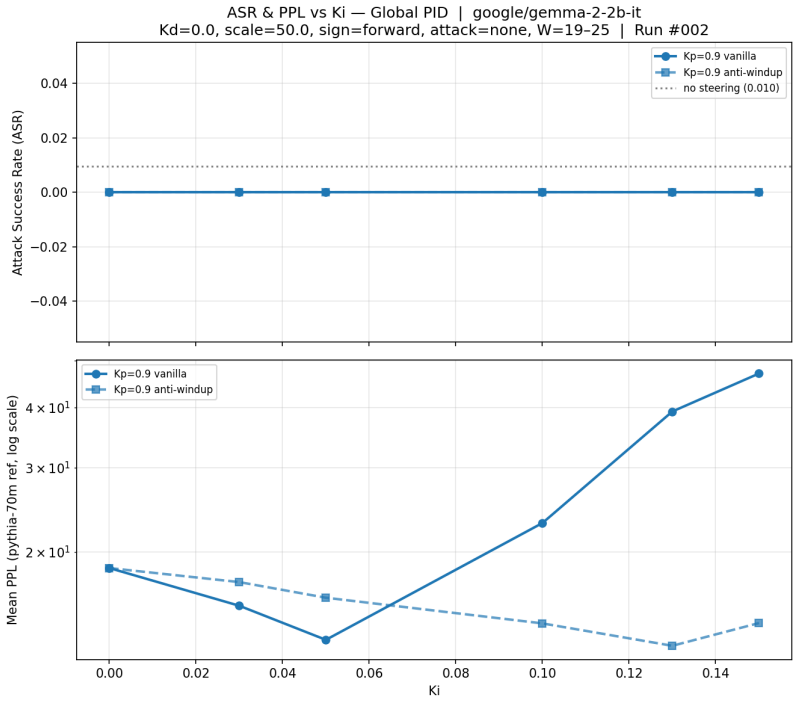

Figure 4: Effect of integral gain K_i on Global PID ($K_p = 0.9$, $K_d = 0$, $\alpha = 50$). **Top:** ASR stays at 0% for all K_i values. **Bottom:** Perplexity diverges for $K_i \geq 0.10$ without anti-windup (solid line) but remains stable with the clamp (dashed). The safe boundary is near $K_i \approx 0.08$.

Figure 5 breaks down the individual PID term magnitudes at the operating gains ($K_p = 0.9$, $K_i = 0.01$, $K_d = 0.01$). The proportional term dominates at ∼0.54, while the integral term grows from 0.006 to 0.042 across the 7-layer window—less than 8% of the proportional contribution. The derivative term is effectively zero. At these gains, integral windup is not a concern, which explains why anti-windup makes no measurable difference in the main experiment.

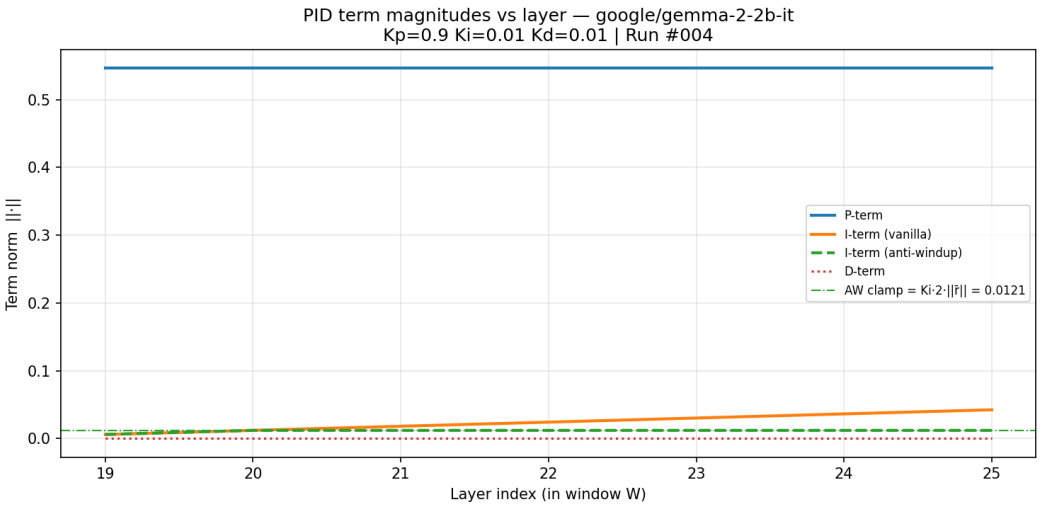

Figure 5: PID term magnitudes across the persistence window at operating gains. The proportional term (blue, $\sim$0.54) dominates. The integral term (orange) grows linearly but remains small. The derivative term (red, dotted) is negligible. The anti-windup clamp (cyan dashed) does not activate at these gains.

5.5 Anti-Windup Ablation

At the operating gains ($K_i = 0.01$), anti-windup produces *identical* results: 0% ASR, 3.5% over-refusal, and perplexity within noise (median 20.37 vs. 20.34). The integral simply never grows large enough to trigger the clamp over a 7-layer window.

Anti-windup becomes necessary at $K_i \geq 0.10$, where it prevents the perplexity divergence shown in Figure 4. This suggests that the clamp is a useful safety feature for practitioners who may experiment with higher gains.

5.6 Scale Calibration

We sweep the scale factor α from 0.5 to 100 to find the operating range. Figure 6 shows that no activation blowup occurs at any tested scale. The steering becomes measurably visible (5% relative divergence in hidden state norms) at $\alpha = 50$. We select $\alpha = 20$ as a conservative operating point that is effective for defense without being near any instability boundary.

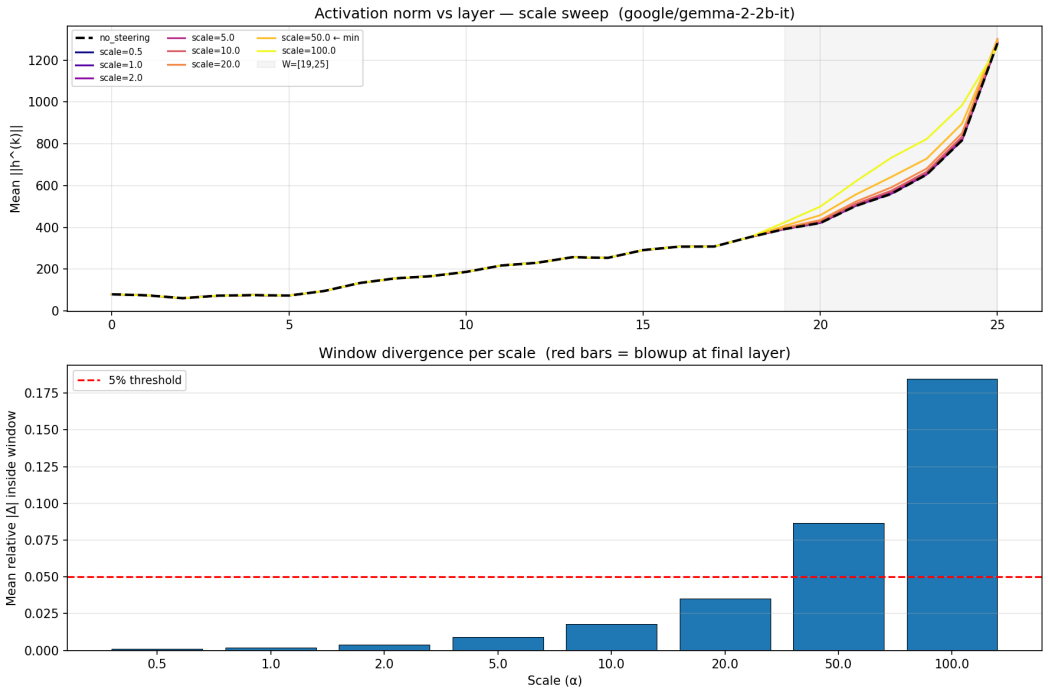

Figure 6: Scale calibration. **Top:** Hidden state norms across layers at each scale. Steering becomes visible only at $\alpha \geq 50$. **Bottom:** Mean activation divergence within the window. The 5% threshold (dashed red) is crossed at $\alpha = 50$. No blowup at any tested scale.

6 Discussion

Reframing the contribution. We set out to test whether Global PID matches per-layer PID on defense. Both achieve 0% ASR, but this comparison is largely uninformative due to the defense floor: Gemma-2-2B-it already blocks 97% of GCG attacks without steering. The more meaningful contribution is on *capability preservation*—showing that targeted steering within a persistence window avoids the over-refusal caused by full-depth intervention.

Confound: ActAdd vs. Angular Steering. A critical caveat is that our experiments use plain Activation Addition (ActAdd), while the original PID Steering paper [Nguyen et al., 2026] builds on the Angular Steering framework [Vu and Nguyen, 2025]. Angular Steering constrains the direction and magnitude of perturbation in ways that ActAdd does not, which may naturally prevent over-refusal. Per-layer PID's 96% over-refusal at $\alpha = 20$ may therefore reflect the combination of ActAdd and high scale rather than an inherent flaw of per-layer PID. A fair comparison would require testing both methods under both steering paradigms.

Confound: scope of intervention. Per-layer PID and Global PID also differ in the number of layers affected: 26 vs. 7. The over-refusal gap may be driven more by this difference in scope than by the choice of steering vector. Per-layer PID's over-refusal might be reducible by restricting it to the same persistence window, but we did not test this. Disentangling these effects is important future work.

Confound: GCG suffix transferability. The universal GCG suffix we use [Zou et al., 2023] was optimized against Llama-2 and Vicuna, not Gemma-2. Its low baseline ASR (2.88%) suggests weak transfer. Testing with a Gemma-2-specific suffix or adaptive attacks would provide a more meaningful defense evaluation.

Why steering at all 26 layers may hurt. Setting aside the above confounds, the cosine similarity matrix (Figure 1) offers a plausible mechanistic explanation: the refusal direction is weak or even negatively correlated in early layers (0–10). Injecting steering vectors where the refusal feature is not coherent likely introduces noise that accumulates through the network and biases the model toward over-refusal. Global PID's restriction to the persistence window naturally avoids this.

The constant-error design choice. We used a constant error signal $e(k) = \bar{r}$ rather than a state-dependent one. This was deliberate: it produces a clean test of the integral windup hypothesis, since the integral grows linearly and predictably. A state-dependent version would make the integral self-correcting and likely expand the safe operating range for K_i, but at the cost of added complexity.

Generalizability. All results are on a single 2B-parameter model. The persistence window (mean cosine 0.76 over 7 layers) may look different on larger models or different architectures. Global PID works because this window exists—if a model lacks cross-layer refusal persistence, the method's premise breaks down.

7 Limitations and Future Work

7.1 Attack Methodology and Model Specificity

Our attack methodology differs fundamentally from the evaluation in Nguyen et al. [2026]. The original paper evaluates defense by feeding direct harmful prompts from AdvBench to the model and measuring whether steering successfully induces refusal—there is no adversarial suffix or optimization-based attack. We chose instead to evaluate under GCG [Zou et al., 2023], a gradient-based adversarial attack that appends an optimized suffix to each prompt, because direct harmful prompts are trivially refused by well-aligned models like Gemma-2-2B-it even without steering, providing no signal about defense quality. GCG represents a stronger threat model: if a defense holds under adversarial optimization, that is a more meaningful claim than blocking prompts the model would already refuse on its own.

However, the universal GCG suffix we use was optimized against Llama-2 and Vicuna, not Gemma-2. Its low baseline ASR on our model (2.88%) likely reflects weak transfer rather than exceptionally strong alignment, and the near-floor ASR across all conditions means the defense comparison provides limited signal. A Gemma-2-specific suffix would likely produce a higher baseline and give the defense methods more room to differentiate.

Beyond attack choice, our evaluation is restricted to a single model: Gemma-2-2B-it, a relatively small 2-billion-parameter model. The original paper evaluates on substantially larger and more diverse models, including Llama-3.1-8B-Instruct, Qwen-2.5 (3B, 7B, and 14B variants), and Gemma-2-9B-it. Larger models may exhibit different persistence window characteristics, different sensitivity to steering scale, and different baseline robustness to attacks. Future work should test across both model families (Llama, Qwen, Gemma, Mistral) and model sizes within each family to capture how the global PID approach scales. In particular, larger and more capable models may have wider or narrower persistence windows, different refusal geometries, and greater headroom between unsteered

and steered ASR—all of which would provide a more informative evaluation of the method's practical value.

7.2 Sensitivity to Steering Scale and PID Gains

The head-to-head capability evaluation inherits a fixed baseline PID gain preset ($K_p = 0.9$, $K_i = 0.01$, $K_d = 0.01$) and steering scale ($\alpha = 20$) directly from Nguyen et al. [2026] without re-tuning for the 26-layer architecture. Per-layer PID's 96% over-refusal rate on benign prompts may therefore be an artifact of evaluating both architectures at this identical raw gain setting rather than an inherent flaw of per-layer feedback. While our integral windup analysis (Section 5.2) systematically sweeps $K_i \in \{0, 0.03, 0.05, 0.10, 0.13, 0.15\}$ at $\alpha = 50$ to demonstrate divergence under static targets, we did not perform a joint multi-parameter grid search across (K_p, K_i, K_d) specifically normalized for per-layer energy injection. A proportionally scaled gain setting (e.g., dividing gains across depth) could potentially alter the trade-off between defense and over-refusal for per-layer steering. Future work should conduct independent multi-variable hyperparameter sweeps for both per-layer and global controllers, comparing each architecture at its optimal operating regime.

7.3 ASR Measurement

Attack success is scored by string-matching against 12 refusal phrases from JailbreakBench [Chao et al., 2024]. This approach is fast and deterministic but brittle: a completion that provides harmful content while avoiding the exact phrases in the refusal list would be counted as a successful defense. Conversely, a model that hedges (e.g., "I'm sorry, but here is the information...") would be counted as a refusal even though it complied. The discrepancy visible in Figure 2 between refusal rates under different conditions hints at how sensitive results can be to the scoring method. Future work should adopt an LLM-judge evaluator for both attack and capability scoring. While LLM judges are probabilistic and stochastic by nature, this can be mitigated by setting the judge model's temperature to near zero and averaging over multiple evaluations, yielding more reliable and nuanced assessments than fixed string matching.

7.4 Persistence Window Evidence

The persistence window (layers 19–25) exhibits a mean intra-window cosine similarity of 0.76, which falls below the green-light threshold of 0.85 established in our verification protocol. While this exceeds the minimum threshold of 0.50 required to proceed, it constitutes only moderate evidence for cross-layer refusal persistence. The global vector $\bar{r}$ is therefore an approximation of the per-layer directions, not an exact substitute. Confirming this phenomenon more rigorously would require alternative methodologies: for instance, a permutation test to establish whether the observed cosine similarities are statistically significant above chance, or a bootstrap confidence interval over the intra-window mean. Testing on additional models would also reveal whether the persistence window is a general property of aligned transformers or specific to Gemma-2's architecture.

7.5 Formal Stability Guarantee

Nguyen et al. [2026] proved input-to-state stability (ISS) for the per-layer PID controller, providing a theoretical guarantee that bounded error signals produce bounded activations. We have not derived an analogous result for the constant-error global variant. The integral windup analysis in Section 5.2 provides empirical evidence of stability at the recommended gains, but a formal proof would be needed to characterize the exact conditions under which the global controller remains

stable. This is particularly important because the constant error signal causes the integral term to grow linearly with depth, a regime not covered by the original ISS analysis. A formal stability proof for the global variant would strengthen the theoretical foundation of the method and would likely be required for publication at top-tier venues.

7.6 Sample Size and Dataset Diversity

With 104 attack prompts and 200 benign prompts, the evaluation sample sizes are moderate. The 95% confidence interval for an observed 0% ASR over 104 trials is approximately [0%, 3.5%], which means small but meaningful differences between methods could be masked by sampling noise. More importantly, the benign prompts are drawn entirely from Stanford Alpaca [Taori et al., 2023], which consists of straightforward instruction-following tasks. Edge cases—ambiguous prompts that sit near the boundary between harmful and benign, instructions that require nuanced refusal, or multi-turn conversations—are not represented. Future work should employ larger sample sizes and more diverse, adversarially curated datasets that include boundary cases to stress-test both the defense and capability preservation claims under realistic conditions.

8 Conclusion

We introduced Global PID Steering, which simplifies per-layer PID Steering by using a single averaged refusal vector applied over a verified persistence window. On Gemma-2-2B-it, both methods achieve 0% ASR under GCG attack, though this comparison is largely uninformative given the low baseline attack success rate (2.88%). The more meaningful finding is on capability preservation: under Activation Addition at scale $\alpha = 20$, per-layer PID refuses 96% of benign prompts while Global PID refuses only 3.5%. The integral windup inherent to the constant-error design is harmless at recommended gains and controllable with a simple clamp at higher settings.

These results come with important caveats. Per-layer PID's over-refusal may partly reflect our use of ActAdd rather than the Angular Steering framework used in the original paper, and the two methods differ in intervention scope (26 vs. 7 layers) in addition to vector choice. Disentangling these confounds is essential future work.

The broader lesson—that targeted intervention, restricted to layers where the relevant feature is coherent, can avoid the side effects of full-depth steering—is suggestive but requires validation across steering paradigms, models, and stronger attacks.

Acknowledgments

This work was completed as a senior independent research project in the Department of Statistics and Applied Probability at UC Santa Barbara. The author thanks Tomoyuki Ichiba for guidance and feedback. Experiments were run on Google Cloud Platform virtual machines with one NVIDIA L4 GPU. Code and experiment scripts are available at https://github.com/cubeerea/pstat199_ucsb.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramer, Matthias Hein, and J. Zico Kolter. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2405.14860.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Jack Lindsey, Wes Gurnee, and Tristan Bricken. Crosscoders. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/crosscoders/index.html.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Dung Nguyen, Tien Pham, Hieu Vu, Yiming Zhang, and Thien Nguyen. Activation steering with a feedback controller. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.04309.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2024.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Alexander Matt Turner, Meg Tong, and Evan Hubinger. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Stefan Heimersheim. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2024.

Hieu Vu and Thien Nguyen. Angular steering: A framework for controlled activation steering in llms. 2025.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. In *arXiv preprint arXiv:2307.15043*, 2023.