Closing the Loop: PID Feedback Control for Interpretable Activation Steering in Symbolic Music Generation

Ioannis Prokopiou [1 2] **Pantelis Vikatos** [2] **Maximos Kaliakatsos-Papakostas** [3] **Theodoros Giannakopoulos** [4] **Themos Stafylakis** [1 5]

Abstract

Activation steering controls generation at inference without retraining. In symbolic music, Sparse Activation Steering via Sparse Autoencoders enables interpretable single-layer attribute control but fails in temporal smoothing: fractional steering magnitudes fall below the Top-K sparsity threshold, zeroing the intervention. We propose PID Steering in two forms: *Spatial PID* validates control-theoretic steering in the generative music domain, while *Temporal PID* dynamically adjusts $\lambda(t)$ at each autoregressive step via a closed-loop controller whose integral term accumulates error to overcome the Top-K barrier. Experiments on the Multitrack Music Transformer show that Temporal PID overcomes the Top-K threshold failure for pitch and duration steering, enabling smooth transitions with less intervention strength and 5% lower Fréchet Music Distance versus static SAS.

1. Introduction

Activation steering modifies internal representations at inference to control generation without retraining (Turner et al., 2024; Li et al., 2023; Rimsky et al., 2024), based on the Linear Representation Hypothesis that concepts correspond to linear directions in activation space (Park et al., 2024; Elhage et al., 2022). Nguyen et al. (2026) proved that static steering methods act as proportional (P) controllers unable to eliminate steady-state error caused by model priors (Åström & Hägglund, 1995). We extend their spatial PID framework to the temporal axis to address a challenge absent in dense steering: the Top-K sparsity barrier.

[1]Athens University of Economics and Business, Athens, Greece [2]Orfium Research, Athens, Greece [3]Hellenic Mediterranean University, Chania, Greece [4]NCSR "Demokritos", Athens, Greece [5]Archimedes / Athena Research Center, Greece. Correspondence to: Ioannis Prokopiou <gian.prokopiou@aueb.gr>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

Transformer models like the Multitrack Music Transformer (MMT) (Dong et al., 2023) unfold compact token representations over long horizons where high-level concepts lack direct vocabulary mappings. Sparse Activation Steering (SAS) (Bayat et al., 2025) provides an attractive solution (Dong et al., 2023; Facchiano et al., 2024; Narashiman et al., 2025; Panda et al., 2024): projecting activations through SAEs enables precise, entanglement-free single-layer interventions. However, SAS imposes a strict Top-K sparsity constraint where only the largest K features survive re-sparsification. This creates a *binary threshold problem*—absent in dense adaptive methods (Vogels et al., 2025; Kang & Kim, 2026; Li et al., 2026)—that defeats temporal smoothing. During a gradual cosine ramp, fractional λ values from 0 to a target value are too small to enter the top-K, entirely zeroing the steering signal and forcing an abrupt binary transition instead of a smooth one (Figure 1). We propose **Proportional-Integral-Derivative (PID) Steering** for symbolic music in two forms:

1. **Spatial PID** validates the layer-wise PID formulation of Nguyen et al. (2026) on the MMT's 12 sublayers, confirming that control-theoretic predictions hold in a shallower architecture than previously studied.

2. **Temporal PID** transposes PID control from the spatial to the temporal domain, built around a Top-K-aware error signal: the controller measures whether steered features survive re-sparsification and adapts $\lambda(t)$ accordingly at each autoregressive step. The integral term accumulates error to overcome the Top-K threshold, enabling smooth SAS steering.

Experiments demonstrate 62–67% less intervention and 5% reduced FMD degradation for PID versus static SAS, with smooth transitions that preserve generation quality. Audio examples and code can be found at https://giannisprokopiouorfium.github.io/music-transformer-sae/pid.

2. Background

Symbolic Music Generation. Symbolic music generation has been reshaped by Transformer architectures (Vaswani et al., 2017), which excel at capturing the long-term temporal dependencies inherent to musical structures. However, traditional MIDI tokenizations often suffer from sequence length explosions (Fradet et al., 2023). The MMT (Dong et al., 2023) is a 6-block decoder-only transformer (12 sublayers, 512 dimensions, 8 attention heads) that generates polyphonic symbolic music using a compact 6-tuple event representation (type, beat, position, pitch, duration, instrument). We use a publicly available checkpoint pre-trained on the Symbolic Orchestral Database (SOD) (Crestel & Esling, 2018), which establishes a coherent generative baseline (Pitch Class Entropy H_0=2.974, Scale Consistency S_0=92.26%, Groove Consistency G_0=93.05%).

Activation Steering. Dense methods (Turner et al., 2024; Rimsky et al., 2024; Arditi et al., 2024; Rodriguez et al., 2025) compute steering vectors as the centroid difference between contrastive datasets in the residual stream and inject $\mathbf{h}^{(\ell)} \leftarrow \mathbf{h}^{(\ell)} + \alpha \cdot \mathbf{v}^{(\ell)}$ at each layer ℓ, but suffer from feature superposition (Elhage et al., 2022): in the MMT, pitch and duration vectors exhibit cosine similarity up to 0.81 (Layer 3, Section D), causing interference during multi-attribute steering. SAS (Bayat et al., 2025) resolves this by training per-layer SAEs (Bricken et al., 2023; Cunningham et al., 2024) to project 512-dim activations into a 4096-dim sparse space (8× expansion) with strict Top-K sparsity: $f(\mathbf{a}) = \text{TopK}(\text{ReLU}(\mathbf{W}_{\text{enc}}\mathbf{a} + \mathbf{b}_{\text{enc}}), K)$. A 16-configuration layer selection grid search identified Layer 10 as optimal for SAS—it provides maximum feature capacity (K=128) with the best monotonic response across both attributes (see Section B). At inference, the steered activation for token t at layer ℓ is:

$$\tilde{\mathbf{a}}_t^\ell = \hat{\mathbf{a}}^\ell\big(\sigma\big(f(\mathbf{a}_t^\ell) + \lambda \cdot \mathbf{v}\big)\big) + \Delta \tag{1}$$

where σ is Top-K ReLU re-sparsification, $\hat{\mathbf{a}}^\ell(\cdot)$ is the SAE decoder, and $\Delta := \mathbf{a}_t^\ell - \hat{\mathbf{a}}^\ell(f(\mathbf{a}_t^\ell))$ is a reconstruction correction term that preserves information lost by the autoencoder. When $\lambda < 1$, the added features $\lambda \cdot \mathbf{v}$ have insufficient magnitude to survive σ, causing the steering signal to vanish.

PID Control for Steering. Nguyen et al. (2026) model the layer-wise forward pass as a discrete-time dynamical system and prove that static steering is a P-controller with guaranteed steady-state error. Their PID formulation computes a dynamic steering vector $\mathbf{u}(k)$ at each layer k:

$$\mathbf{u}(k) = K_p\mathbf{e}(k) + K_i \textstyle\sum_{j=0}^{k-1} \mathbf{e}(j) + K_d(\mathbf{e}(k) - \mathbf{e}(k-1)) \tag{2}$$

where $\mathbf{e}(k) = \boldsymbol{\mu}_{\text{target}}(k) - \boldsymbol{\mu}_{\text{source}}(k)$ is the layer-wise error signal. The I term accumulates past errors to remove residual bias; the D term damps overshoot (Åström & Hägglund, 1995). Recent adaptive methods—IDS (Vogels et al., 2025), DIRECTER (Kang & Kim, 2026), SVF (Li et al., 2026), SMITIN (Luo et al., 2025)—operate in dense settings where attenuated signals persist. Our temporal PID addresses the sparse threshold barrier by accumulating error until $\lambda(t)$ breaches the Top-K boundary.

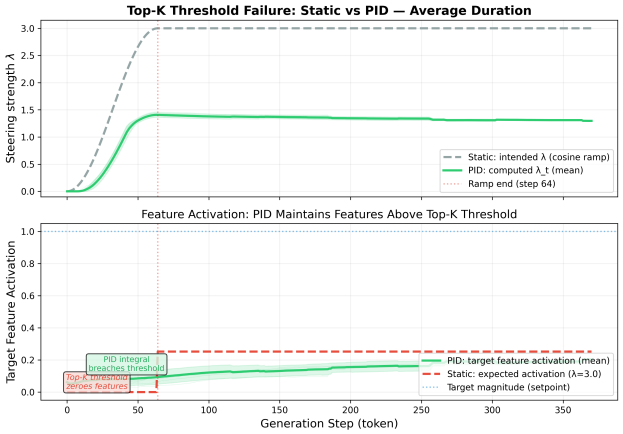

Figure 1. **The Top-K threshold failure.** *Top:* Static cosine ramp vs. PID's dynamic $\lambda(t)$, which settles lower. *Bottom:* Static ramping zeros target features throughout the ramp; PID's integral accumulation maintains non-zero activations from the onset.

3. Method

We apply PID control to symbolic music steering in two forms: *Spatial PID* validates the layer-wise formulation of Nguyen et al. (2026) on the MMT's dense residual stream, confirming its predictions transfer to a shallower architecture, and *Temporal PID* transposes the controller to the autoregressive time axis to solve the SAS smoothing failure.

3.1. Spatial PID: Adapting to Symbolic Music

We apply Equation (2) across MMT's 12 sublayers using DiffMean vectors with an all-to-all injection strategy (Section F), computing the steering vector sequentially. The key challenge is the MMT's shallow depth—12 layers versus 32+ in language models (Nguyen et al., 2026)—giving the integral term fewer steps to accumulate corrections, requiring proportionally higher K_i (see Section 4.1).

3.2. Temporal PID: Solving the Sparse Threshold Problem

SAS interventions in the MMT occur at a single layer (Layer 10), precluding the spatial layer-to-layer PID of Section 3.1. We therefore transpose the PID control variable from the *spatial* domain (layer index k) to the *temporal* domain (generation step t), akin to reasoning-time temporal controllers (Bharadwaj, 2025), creating a closed-loop feedback system that operates during autoregressive decoding.

Error Measurement. At each generation step t, we measure the mean magnitude of the top-N target features (by absolute weight in the SAS vector $\mathbf{v}$) as $\bar{f}_a(t) = \frac{1}{|\mathcal{T}|}\sum_{j\in\mathcal{T}} f(\mathbf{a}_t^\ell)_j$, where $\mathcal{T}$ is the set of N=32 feature indices with the largest $|v_j|$. These features act as a "concept fingerprint" to indicate whether the steering signal survived Top-K re-sparsification. The error signal is $e(t) = m^*(t) - \bar{f}_a(t)$, where $m^*(t)$ is a cosine-ramped setpoint that smoothly scales the target activation magnitude over T_{ramp} beats:

$$m^*(t) = \begin{cases} \frac{m_{\text{target}}}{2}\left(1 - \cos\left(\frac{\pi \cdot t}{T_{\text{ramp}}}\right)\right) & t < T_{\text{ramp}} \\ m_{\text{target}} & t \geq T_{\text{ramp}} \end{cases} \quad (3)$$

PID Control Law. The controller computes the steering magnitude at each step:

$$\lambda(t) = \text{clamp}\Big(K_p e(t) + K_i I(t{-}1) + K_d\big(e(t) - e(t{-}1)\big)\Big) \quad (4)$$

where $I(t) = \text{clamp}(I(t{-}1) + e(t),\ -I_{\max},\ I_{\max})$ is the integral accumulator with anti-windup clamping (Åström & Hägglund, 1995), and $\lambda(t) \in [0, \lambda_{\max}]$. At t=0, $I(0)$=0, $e(-1)$=0, and $\bar{f}_a(0)$ is the unsteered activation (typically near zero for target features). The steered sparse representation becomes $\mathbf{s}(t) = f(\mathbf{a}_t^\ell) + \lambda(t) \cdot \mathbf{v}$, followed by Top-K re-sparsification and SAE decoding as in Equation (1).

Intuition. During the ramp, $\bar{f}_a \approx 0$ creates persistent positive error that the I term accumulates, progressively amplifying $\lambda(t)$ until the threshold is breached. The controller then settles to a *just sufficient* λ, with the D term damping overshoot at the sub-threshold–to–active transition. We chose the mean top-N feature magnitude over alternatives (e.g., the fraction of surviving features) because it provides a continuous, proportional error signal: gradual magnitude changes yield smooth λ adjustments, whereas a binary survival count would produce bang-bang control.

Down-Steering and Directionality. Steering direction is encoded in the bidirectional SAS vector $\mathbf{v} = \mathbf{v}^+ - \mathbf{v}^-$ (Section A). For down-steering, we negate: $\mathbf{v}_{\text{down}} = -\mathbf{v}$; the controller output $\lambda(t) \in [0, \lambda_{\max}]$ remains non-negative in both directions. The concept fingerprint $\mathcal{T}$ is recomputed from the active vector's largest absolute components.

Dual-Concept Control. For simultaneous steering, two independent controllers use Gram-Schmidt-orthogonalized SAS vectors with an expanded $2{\times}K$ budget at the steered layer (applied equally to PID and static baselines), since the original K budget systematically displaces the second concept's orthogonalized features during Top-K selection. This scaling is an SAS framework limitation; alternative SAE architectures with relaxed sparsity may alleviate it.

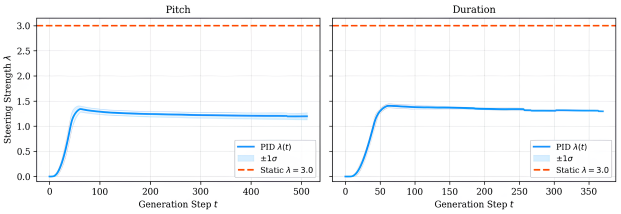

Figure 2. **Temporal PID $\lambda(t)$ trajectory** (steer-up, m_{target}=2.0). The controller overshoots briefly at threshold breach, then settles to $\lambda \approx 1.15$—62% below static SAS's λ=3.0.

4. Experiments

We evaluate on the SOD corpus using contrastive sets (1,280 samples each, Section A) defined by the 20th/80th percentiles of pitch ($\leq$60 vs. $\geq$67.6 semitones) and duration ($\leq$6.5 vs. $\geq$14.5 ticks). SAEs are trained at each MMT layer (512$\rightarrow$4096, Top-K K=128; details in Section C). Quality degradation $\delta = |H - H_0| + \max(0, S_0 - S) + \max(0, G_0 - G)$ combines deviations from ground-truth pitch class entropy (H_0), scale consistency (S_0), and groove consistency (G_0); FMD uses CLaMP2 embeddings (Wu et al., 2024a) (details in Section K). All generations use temperature 1.0, top-k logit filtering (threshold 0.9), max_seq_len=1024, and T_{ramp}=64 steps.

4.1. Spatial PID: Domain Validation

We validate spatial PID (Equation (2)) using DiffMean vectors across MMT's 12 sublayers (n=25, $\alpha \in \{\pm 0.5, \ldots, \pm 2.0\}$). A grid search (Section E) reveals pitch demands 8$\times$ higher K_i than duration (0.2 vs. 0.025), reflecting stronger autoregressive priors. Error convergence replicates Nguyen et al. (2026): PI/PID eliminate P-control's residual error within 12 layers. Per-α breakdowns appear in Section J. Hook ablation across 9 configurations (Section G) shows all_12 are most effective, with PID reducing δ by 17–47% versus P-only; FMD sweeps (Section H) confirm improved distributional fidelity at α=1.0.

4.2. Temporal PID: Single-Concept Steering

Table 1 presents results with K_p=1.0, K_d=0.01, $I_{\max}$=10, $\lambda_{\max}$=3.0, and concept-specific K_i: 0.05 for pitch, 0.025 for duration (a 2$\times$ ratio, distinct from the spatial 8$\times$ ratio from Section E). Note that the reported degradation gap between single-concept ($\delta{\approx}0.45$, n=40) and specific threshold baselines (Section L, δ=0.07, n=20) stems from sample size and subset variance. For pitch-up (n=40), PID achieves 72.65 st (δ=0.45 [0.33, 0.62]) versus static SAS 72.30 (δ=0.64 [0.47, 0.85]) with 62% less intervention (avg λ=1.15; Figure 2). Negative pitch shifts -24.0 semitones while maintaining 98.7% groove. Duration uses avg $\lambda \approx 1.0$—67% less than static's λ=3.0. Duration-down matches static quality (δ=3.62 vs. 3.37; overlapping CIs, Section N); duration-up incurs higher δ=8.45 vs. 2.84: matched-λ anal-

Table 1. Single-concept temporal PID steering ($\lambda_{\max}$=3.0, n=40). Pitch: K_i=0.05; Duration: K_i=0.025. Static SAS uses fixed λ=3.0; PID uses dynamic $\lambda(t)$. Base. indicates the unsteered Baseline.

Concept	Direction	PID	Static	Base.
Pitch (semitones)	↑	**72.65**	72.30	68.79
	↓	**43.99**	44.91	67.94
Dur. (ticks)	↑	18.87	**22.17**	7.99
	↓	4.23	**3.35**	7.72
Pitch FMD (↓)		**461.9**	487.7	381.5
Duration FMD (↓)		**501.2**	525.9	385.3

Table 2. Dual-concept steering (n=20). Conditioned: 16-beat prefixes, 10×2 reps. Notation: L=Low, H=High, S=Short. Full attribute values in Section I.

	δ (↓)		Dual%	
Setting	PID	Stat	PID	Stat
Uncond.	**0.47**	2.19	90	**95**
L/S→H/L	4.13	**3.72**	**80**	75
H/L→L/S	5.21	**3.61**	**95**	90
L/L→H/S	**2.36**	2.85	80	**85**
H/S→L/L	**1.92**	4.30	100	100

ysis (Section R) shows static SAS at λ=1.0 maintains 91.3% scale vs. PID's 84.7%, indicating within-sequence λ variation amplifies degradation for this attribute. PID's activations evolve smoothly (std($\Delta\bar{f}_a$) $<$ 0.003; Section O) versus static SAS's binary step. The concept fingerprint is robust to $|\mathcal{T}|$ (range 0.37 st across $N \in \{8,16,32,64\}$; Section M), and PID outperforms step-function and minimal-λ baselines (Section L). Gain sweeps (Section P) confirm stability across 2× perturbations (pitch 72.8–75.1 st); T_{ramp} sensitivity (Section Q) identifies a sweet spot at $\{32, 64\}$ steps.

FMD Analysis. PID achieves 5.3% lower FMD than static SAS for pitch (Table 1), as dynamic $\lambda(t)$ avoids oversteering early tokens. Note that all steered conditions increase FMD over the unsteered baseline (e.g., pitch PID 461.9 vs. baseline 381.5), since effective steering necessarily shifts the output distribution; PID's advantage is minimizing this drift relative to static SAS. Setpoint sweeps show m_{target}=2.0 is optimal for dual steering; (Section H).

Component Ablation. P-only control yields λ_{avg}=0.664—too conservative for the Top-K threshold. Adding I (λ_{avg}=1.136) drives λ above threshold via error accumulation; D marginally improves settling (λ_{avg}=1.158), confirming the integral term is *essential*.

4.3. Dual-Concept Steering

Table 2 evaluates simultaneous pitch and duration steering using two independent temporal PID controllers (m_{target}=2.0, n=20). Per-concept adaptation manifests in magnitude (pitch settles at λ≈1.15, duration at λ≈1.05).

PID achieves 4.7× lower degradation in unconditioned steering (δ=0.47 vs. 2.19) and excels in the hardest opposing-direction conditioned case (H/S→L/L: 2.2× advantage). PID wins δ in 3 of 5 settings. In the two conditioned cases where static wins (L/S→H/L, H/L→L/S), PID still achieves high success rates (80–95%), though its dynamic λ trajectory occasionally amplifies scale degradation relative to the static baseline's uniform intervention (see Section R). The dual success gap (90% vs. 95% unconditioned) reflects conservative λ occasionally under-steering duration.

4.4. Round-Trip Steering

Temporal PID enables *reversible steering*: steer away from a conditioned prefix, hold, then steer back—impossible with static SAS's fixed λ. Using an asymmetric three-phase schedule with 16-beat conditioned prefixes (n=20 per scenario; details in Section S), PID outperforms a passive release baseline (λ=0 in Phase 3) by 8–26 percentage points (aggregate recovery: 46–74% vs. 36–62%), confirming that recovery is actively driven by closed-loop back-steering, not passive relaxation.

5. Conclusion

We introduced PID Steering for symbolic music: *Spatial PID* validates control-theoretic steering (Nguyen et al., 2026) in a shallow architecture, while *Temporal PID* overcomes SAS's Top-K threshold failure via integral error accumulation, enabling smooth sparse steering with 62–67% less intervention and 5% reduced FMD degradation for pitch. Round-trip steering demonstrates reversible multi-phase trajectories that static methods cannot express, achieving 46–74% recovery and outperforming passive release by 8–26 pp. Gain robustness sweeps confirm stability across 2× perturbations (Section P), with only +1.9% marginal overhead versus static SAS (Section U).

Limitations. We evaluate on a single model (MMT) and dataset (SOD); gain portability and the 2× K_i asymmetry require cross-architecture validation. Sample sizes (n=40) are modest, and perceptual validation (e.g., MUSHRA or A/B tests) is absent. Furthermore, duration-up steering degrades scale consistency to 84.7%, which matched-λ analysis (Section R) confirms is intrinsic to PID's adaptive trajectory. Future work includes adaptive gain scheduling and relaxed-sparsity SAEs like RouteSAE (Shi et al., 2025).

Impact Statement

This work advances controllable music generation. Activation steering techniques carry dual-use risks in broader settings; in symbolic music, the primary concern is unauthorized style imitation, which we mitigate by releasing only the method, not artist-specific vectors.

Acknowledgment

This research was funded by the European Union's Horizon Europe research and innovation programme under the AIXPERT project (Grant Agreement No. 101214389), which aims to develop an agentic, multi-layered, GenAI-powered framework for creating explainable and transparent AI systems.

References

Arditi, A., Obeso, O., Suri, A., and Barez, F. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

Åström, K. J. and Hägglund, T. *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America, 2nd edition, 1995.

Bayat, R. et al. Sparse autoencoders for scalable concept steering in large language models. *arXiv preprint arXiv:2501.15148*, 2025.

Bharadwaj, A. R. Stu-pid: Steering token usage via pid controller for efficient large language model reasoning. *arXiv preprint arXiv:2506.18831*, 2025.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Laber, R., Wu, Y., Rings, S., et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Crestel, L. and Esling, P. A database for orchestration research and its use for timbre interpolation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2024.

Dong, H.-W., Chen, K., Dubnov, S., McAuley, J., and Berg-Kirkpatrick, T. Multitrack music transformer. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *Transformer Circuits Thread*, 2022.

Facchiano, F. et al. Activation patching for controllable generation in MusicGen. *arXiv preprint*, 2024.

Fradet, N., Briot, J.-P., Chhel, F., El Fallah Seghrouchni, A., and Music, S. C. S. L. MidiTok: A python package for MIDI file tokenization. *arXiv preprint arXiv:2310.17202*, 2023.

Kang, M. and Kim, J. Enhancing instruction following of LLMs via activation steering with dynamic rejection. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Li, J., Li, Y., and Huang, K.-H. Steering vector fields for context-aware inference-time control in large language models. *arXiv preprint arXiv:2602.01654*, 2026.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, 2023.

Luo, S. et al. SMITIN: Self-monitored inference-time intervention. *arXiv preprint*, 2025.

Narashiman, S. et al. Genre controlled music generation via activation steering. *arXiv preprint arXiv:2506.10225*, 2025.

Nguyen, D. V., Pham, N. Y., Vu, H. M., Zhang, L., and Nguyen, T. M. Activation steering with a feedback controller. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Panda, R. et al. Steering music generation with activation engineering. *arXiv preprint*, 2024.

Park, K., Choe, Y. J., and Veitch, V. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2024.

Rimsky, N., Gabrieli, N., Schulz, J., Smith, M., Tong, I., and Hubinger, E. Steering Llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2024.

Rodriguez, A. H. et al. Mean activation transport for activation steering. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Shi, W., Li, S., Liang, T., Wan, M., Ma, G., et al. Route sparse autoencoder to interpret large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

Turner, A. M., Thiergart, L., Udell, D., Leech, G., Mini, U., and Pelrine, M. Activation addition: Steering language models without optimization. In *Advances in Neural Information Processing Systems*, 2024.

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

Vogels, A., Wong, B., Choho, Y., and Blangero, A. In-distribution steering: Balancing control and coherence in language model generation. *arXiv preprint arXiv:2510.13285*, 2025.

Wu, S. et al. CLaMP 2: Multimodal music information retrieval across 101 languages using large language models. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2024a.

Wu, Y. A. et al. Fréchet music distance: A metric for generative evaluation of music. *arXiv preprint arXiv:2412.07948*, 2024b.

A. Steering Vector Construction

DiffMean Vectors. For each layer ℓ, we intercept the summary activation $\mathbf{h}$ of the last fully contextualized token during a forward pass over contrastive sets. The steering vector is the centroid difference:

$$\mathbf{v}^{(\ell)} = \frac{1}{N_+}\sum_{i=1}^{N_+}\mathbf{h}_{+,i}^{(\ell)} - \frac{1}{N_-}\sum_{j=1}^{N_-}\mathbf{h}_{-,j}^{(\ell)} \tag{5}$$

where $N_+ = N_- = 1{,}280$ are samples from the "High" and "Low" contrastive pools extracted from the SOD corpus using the 20th/80th percentiles: average pitch ($\leq$60 vs. $\geq$67.6 semitones) and average duration ($\leq$6.5 vs. $\geq$14.5 ticks). These thresholds were validated to be stable across alternative extreme quantile choices.

SAS Vectors. Given sparse encodings of the contrastive sets (S^+ and S^-), behavior-specific features are isolated via frequency filtering with threshold τ=0.08 (requiring a minimum 8% activation frequency). Shared Feature Removal zeroes features active in both pools, yielding a bidirectional vector $\mathbf{v}^{(b,\ell)} = \mathbf{v}^+ - \mathbf{v}^-$ that reinforces target features while suppressing opposing ones.

B. Layer Selection for SAS

A 16-configuration grid search evaluated SAS injection at individual layers and layer groups, balancing steering effectiveness against quality degradation δ. Key findings:

Multi-layer SAS broadcasting causes catastrophic failure: unlike DiffMean, cascading SAE reconstruction noise across layers produces total quality collapse. Layer 10 is optimal under the Adaptive K strategy (K scaling from 32 at Layer 0 to 128 at Layers 8–11), providing maximum feature capacity with the best monotonic response across both attributes, and a $48\times$ reduction in intervention footprint versus DiffMean (128 features vs. 512×12 dimensions).

Adaptive K Strategy. SAE sparsity scales linearly with depth: K=32 (Layer 0), K=64 (Layers 1–3), K=96 (Layers 4–7), K=128 (Layers 8–11). This accommodates increasing representational complexity in deeper layers.

C. SAE Training Details

One SAE is trained per transformer layer. Architecture: $512 \to 4096$ ($8\times$ expansion), tied weights ($\mathbf{W}_{\text{dec}} = \mathbf{W}_{\text{enc}}^\top$), per-layer zero-mean unit-variance normalization. Training data: 640K activation vectors extracted via forward hooks from 10K tracks uniformly sampled from the SOD. Optimizer: Adam, LR $= 10^{-4}$. Loss:

$$\mathcal{L} = \text{MSE}(\mathbf{a}, \hat{\mathbf{a}}) + \lambda_{L_1} \cdot \mathbb{E}[|f(\mathbf{a})|], \quad \lambda_{L_1} = 10^{-3} \tag{6}$$

D. Feature Interference Analysis

In the dense residual stream, pitch and duration DiffMean vectors exhibit an average absolute cosine similarity of **0.49** across all 12 layers, peaking at **0.81 in Layer 3**. This severe entanglement causes destructive interference during multi-attribute steering, requiring geometric decoupling via GSO ($\mathbf{v}_d^{\perp} = \mathbf{v}_d - \text{proj}_{\mathbf{v}_p}\mathbf{v}_d$). In the sparse SAE space, pitch and duration vectors have an average cosine similarity of -0.27 (anti-correlated) with 51.8% shared feature overlap (958/4096 features), ranging from 26.5% at Layer 6 to 75.0% at Layer 3. For dual SAS steering, we apply Gram-Schmidt orthogonalization in sparse space and expand the Top-K budget to $2\times K$ to prevent feature competition.

E. Spatial PID Gain Grid Search

We sweep $K_p \in \{0.5, 0.75, 1.0, 1.25, 1.5\}$, $K_i \in \{0.0, 0.025, 0.05, 0.10, 0.15, 0.20\}$, $K_d \in \{0.0, 0.01, 0.025, 0.05, 0.10\}$ (150 configurations per concept, n=25, α=1.0). The objective function is Score $= |\Delta_{\text{attr}}|/(1+\delta)$, balancing steering magnitude against quality degradation.

Table 3. Spatial PID optimal gains per concept. Score combines steering effectiveness and quality degradation.

Concept	K_p	K_i	K_d	Score	δ@α=1
Pitch	1.5	0.2	0.01	32.86	7.91
Duration	1.25	0.025	0.01	8.82	5.37

Pitch requires $8\times$ higher K_i than duration. The model's autoregressive priors strongly resist pitch register deviations, requiring aggressive integral accumulation. Duration is inherently more responsive—even K_i=0.025 produces strong effects with conservative overshoot.

F. Injection Strategy Comparison

For DiffMean, we compared three broadcast strategies: **All-to-All** (layer-specific vectors at all layers), **One-to-All** (single layer's vector broadcast), and **Some-to-Some** (targeting layer groups, e.g., deep layers 8–11). All-to-All proved optimal—distributed dense embeddings require multi-layer reinforcement—and is used for all DiffMean experiments. SAS uses single-layer injection at Layer 10 only.

G. Hook Ablation Study

Key observations: (1) all_12 and mid_deep achieve the best PID degradation (2.65 and 3.07); (2) deep_only shows the largest PID improvement (28% δ reduction vs. P-only); (3) single-layer configurations (layer_10, layer_11) show PID's advantage is most pronounced at concentrated injection loci (47% δ reduction at layer_10).

Table 4. Spatial PID hook ablation averaged over 2 concepts $\times$ 2 α values. $|\Delta|$: steering effectiveness; Deg: quality degradation.

Config	Layers	P $\|\Delta\|$	PID $\|\Delta\|$	P Deg	PID Deg
all_12	12	39.4	39.5	3.21	**2.65**
mid_deep	8	39.3	39.8	3.62	3.07
deep_only	4	41.3	41.3	4.46	3.19
attn_only	6	36.8	38.1	3.77	3.59
ff_only	6	27.7	26.1	5.16	5.60
shallow_only	4	9.7	12.3	7.15	7.49
mid_only	4	27.7	32.4	9.61	8.07
layer_10	1	28.8	33.4	6.23	3.32
layer_11	1	36.8	33.2	6.42	4.45

H. FMD Sweep Across α

Table 5. FMD across steering magnitudes (n=40 per condition).

	α			
Method	0.5	1.0	1.5	2.0
Pitch				
Baseline	412.4	414.5	413.8	417.6
P-only	464.3	490.0	494.9	516.1
PI	480.7	508.9	512.1	518.4
PID	479.1	506.3	512.6	518.6
Duration				
Baseline	387.3	381.6	382.1	394.2
P-only	409.4	471.9	483.2	474.6
PI	404.9	462.9	480.4	475.4
PID	412.4	**456.0**	**475.0**	474.8

For pitch, P-only achieves lower FMD since PI/PID's stronger steering shifts the distribution further from the reference corpus—a tradeoff between steering effectiveness and distributional fidelity. For duration (K_i=0.025, conservative), PID achieves both stronger steering *and* lower FMD at α=1.0–1.5, the best of both.

I. Full Conditioned Dual Steering Results

PID achieves δ closer to ground truth in 2 of 4 scenarios (H/S→L/L, L/L→H/S). In the most challenging opposing-direction case (H/S→L/L), PID's $2.2\times$ degradation advantage is most pronounced.

J. Spatial Single-Attribute α Sweep

PID achieves the lowest degradation at every α for pitch, with the advantage most pronounced at α=-1.0 (36% reduction). At high α values, all controllers converge as the model saturates.

Table 6. Conditioned dual-concept steering across all 4 directional scenarios (10 songs × 2 repetitions per scenario).

Scenario	Method	Pitch	Dur.	δ	Dual%
L/S→H/L	PID	66.95	6.89	4.13	**80%**
	Static	67.12	7.33	**3.72**	75%
H/L→L/S	PID	57.64	15.93	5.21	**95%**
	Static	56.33	16.70	**3.61**	90%
L/L→H/S	PID	64.09	8.95	**2.36**	80%
	Static	64.32	8.84	2.85	**85%**
H/S→L/L	PID	53.06	7.33	**1.92**	100%
	Static	47.91	5.67	4.30	100%

Table 7. Pitch steering across α values (n=25).

α	Method	Pitch (st)	δ	N
−1.0	P	39.17	5.2	346
	PI	39.31	5.5	339
	PID	39.55	**3.6**	282
1.0	P	82.08	5.6	509
	PI	82.47	5.2	481
	PID	82.90	**5.0**	484
2.0	P	78.36	9.2	419
	PI	79.22	9.1	478
	PID	78.86	**9.0**	497

K. Steering Metrics

Quality Degradation. $\delta = |H - H_0| + \max(0, S_0 - S) + \max(0, G_0 - G)$ is the cumulative deviation from ground-truth SOD corpus statistics using MusPy metrics where H is Pitch Class Entropy, S is Scale Consistency, G is Groove Consistency, and subscript 0 denotes ground-truth SOD values. The asymmetric penalty reflects musical priors: increases in scale/groove consistency above the corpus mean are benign, whereas decreases indicate structural degradation. Entropy uses absolute deviation since both extremes (uniform randomness and single-note collapse) are undesirable.

Steering Success. A generation is successful if its mean target attribute shifts in the specified direction relative to the unsteered baseline (unconditioned) or the conditioning prefix (conditioned).

Fréchet Music Distance. FMD is computed using CLaMP2 (Wu et al., 2024a) embeddings with MLE Gaussian estimation over 4,474 SOD reference MIDIs, following Wu et al. (2024b).

L. Threshold-Aware Baselines

To assess whether simpler dynamic strategies can solve the Top-K threshold problem, we compare temporal PID against two alternatives (n=20, positive direction):

- **Step**: λ=0 for $t < T_{\text{ramp}}$, then λ=λ_{target} instantly.
- **Minimal-λ**: Per-step binary search for the smallest λ that activates at least one target feature above the Top-K threshold.

Table 8. Threshold-aware baseline comparison (positive direction).

Concept	Method	Attr.	δ (↓)	Avg λ
Pitch	PID	73.13	**0.07**	1.13
	Step	71.86	0.32	2.49
	Min-λ	69.44	0.30	0.01
	Baseline	70.88	0.14	—
Dur.	PID	59.85	7.02	1.11
	Step	60.37	3.09	1.99
	Min-λ	70.54	0.09	0.01
	Baseline	66.49	0.09	—

For pitch, PID achieves the lowest δ (0.07) with the strongest steering (73.13 st), outperforming Step (δ=0.32, 2.2× more λ) and Minimal-λ (insufficient steering). For duration, the Step function is competitive in quality but uses 1.8× more intervention. Minimal-λ barely steers at all (avg λ=0.01), confirming that threshold-aware does not equal effective.

M. Concept Fingerprint Sensitivity

Table 9. Sensitivity to fingerprint size $|\mathcal{T}|$ (n=20, positive direction).

$\lvert\mathcal{T}\rvert$	Pitch (st)	δ	Avg λ
Pitch steering			
8	73.18	0.70	0.945
16	73.14	**0.09**	1.048
32	73.34	1.09	1.136
64	72.97	0.15	1.236

Pitch steering is remarkably stable: all four $|\mathcal{T}|$ values produce pitch in [72.97, 73.34] st (range 0.37 st). Smaller fingerprints require less λ (0.945 at N=8 vs. 1.236 at N=64) since fewer features need to breach the threshold. Duration shows similar steering stability across N (range [18.86, 19.48]). The default N=32 is a reasonable middle ground.

N. Confidence Intervals

For pitch, PID achieves lower degradation than static SAS: δ=0.45 [0.33, 0.62] vs. 0.64 [0.47, 0.85]. For duration,

Table 10. Bootstrap 95% CIs (n=40, 10,000 resamples, positive steering).

Metric	PID	Static	Base.
Pitch			
Attr. (st)	72.65±0.36	72.30±0.24	68.79±2.23
δ (↓)	**0.45**±0.15	0.64±0.19	1.73±0.68
Scale (%)	95.9±0.72	95.9±0.93	95.2±1.09
Groove (%)	97.1±0.54	97.8±0.32	93.8±0.95
Duration (K_i=0.025, positive)			
Attr. (ticks)	18.87±0.54	22.17±0.86	7.99±1.00
δ (↓)	8.45±2.48	**2.84**±1.46	3.05±1.47
Scale (%)	84.7±2.64	92.4±2.03	94.1±1.62
Groove (%)	98.4±0.15	99.3±0.07	93.9±1.46

duration-down PID matches static (δ=3.62 [2.57, 4.72] vs. 3.37 [2.49, 4.41]; overlapping CIs) while using 67% less intervention. Duration-up PID incurs higher degradation (δ=8.45 [6.03, 10.99] vs. 2.84 [1.48, 4.39]), from a scale consistency drop to 84.7%. Matched-λ analysis (Section R) indicates both the SAS vector and PID's dynamic trajectory contribute to this degradation.

O. Intervention Smoothness

We quantify smoothness via step-to-step changes in $\lambda(t)$ and concept fingerprint activations $\bar{f}_a(t)$ (n=20 per concept). PID's $\lambda(t)$ changes by at most 0.055 per step and feature activations by less than 0.022, confirming smooth transitions. Static cosine ramping, while smooth in λ (std($\Delta\lambda$)=0.008–0.017), produces binary feature activations—zero throughout the ramp, then a single-step jump once λ breaches the Top-K threshold (Figure 1).

Table 11. Intervention smoothness metrics (mean over samples).

Concept	std($\Delta\lambda$)	max $\lvert\Delta\lambda\rvert$	std($\Delta\bar{f}_a$)	max $\lvert\Delta\bar{f}_a\rvert$
Pitch	0.010	0.053	0.002	0.022
Duration	0.015	0.055	0.002	0.018

P. Gain Robustness

We sweep K_p and K_i independently at multiplicative factors $\{0.5, 0.75, 1.0, 1.25, 1.5, 2.0\}$ of their nominal values (n=20, positive direction).

Pitch steering is remarkably stable: across all 12 configurations (2× perturbation range), pitch remains in [72.77, 75.12] st (range 2.35 st). δ varies more (0.61–4.24) due to the sensitivity of quality metrics to small distributional shifts, but the controller functions across all tested gains.

Duration steering shows similar robustness: duration remains in [15.13, 19.85] ticks across all perturbations. The K_p=0.5 outlier (15.13 ticks) reflects insufficient proportional response to overcome the Top-K threshold quickly, but even this extreme perturbation produces meaningful steering above the baseline (7.99 ticks).

Table 12. Gain robustness for pitch (K_p=1.0, K_i=0.05 nominal).

Sweep	Factor	K_p	K_i	Pitch (st)	δ
K_p	0.50	0.50	0.050	75.12	4.24
	0.75	0.75	0.050	73.22	3.44
	1.00	1.00	0.050	73.10	1.24
	1.25	1.25	0.050	73.60	1.36
	1.50	1.50	0.050	73.13	2.02
	2.00	2.00	0.050	72.77	0.70
K_i	0.50	1.00	0.025	73.56	3.75
	0.75	1.00	0.038	74.00	2.90
	1.00	1.00	0.050	73.03	3.54
	1.25	1.00	0.063	73.33	0.79
	1.50	1.00	0.075	72.85	2.37
	2.00	1.00	0.100	73.27	0.61

Table 13. Duration Gain robustness (K_p=1.0, K_i=0.025).

Sweep	Factor	K_p	K_i	Dur. (ticks)	δ
K_p	0.50	0.50	0.025	15.13	2.74
	0.75	0.75	0.025	18.72	5.72
	1.00	1.00	0.025	19.15	9.75
	1.25	1.25	0.025	19.15	6.21
	1.50	1.50	0.025	19.08	7.13
	2.00	2.00	0.025	19.85	7.63
K_i	0.50	1.00	0.013	18.70	6.24
	0.75	1.00	0.019	18.98	8.47
	1.00	1.00	0.025	18.93	8.72
	1.25	1.00	0.031	19.12	9.18
	1.50	1.00	0.038	18.72	8.86
	2.00	1.00	0.050	19.52	7.18

Q. T_{ramp} Sensitivity

We test PID across $T_{\text{ramp}} \in \{16, 32, 64, 128, 256\}$ with fixed nominal gains (n=20, positive pitch steering).

$T_{\text{ramp}} \in \{32, 64\}$ provides the best tradeoff: both achieve pitch $>$ 73 st with $\delta < 2.2$. Too-short ramps (T_{ramp}=16) cause aggressive integral accumulation and quality degradation (δ=10.17); too-long ramps (T_{ramp}=256) delay integral buildup, yielding weaker steering (71.20 st) and lower avg λ (0.903). The default T_{ramp}=64 balances smooth onset against timely convergence.

Table 14. T_{ramp} sensitivity for pitch steering.

T_{ramp}	Pitch (st)	δ	Avg λ
16	70.07	10.17	1.216
32	73.86	**0.82**	1.162
64	73.56	2.13	1.167
128	73.41	4.40	1.076
256	71.20	4.34	0.903

R. Matched-λ Analysis for Duration-Up

To isolate whether duration-up scale degradation stems from the SAS vector or PID's dynamic trajectory, we run static SAS at $\lambda \in \{0.5, 1.0, 1.5, 2.0, 3.0\}$ (n=40, duration-up).

Table 15. Static SAS at matched λ for duration-up. Compare with PID: avg $\lambda \approx 1.0$, scale=84.7%, δ=8.45.

λ	Dur. (ticks)	Scale (%)	Groove (%)	δ
0.5	12.33	95.1	95.4	1.59
1.0	21.19	91.3	98.6	2.89
1.5	22.44	93.2	98.8	1.88
2.0	22.80	91.1	99.1	4.12
3.0	24.41	93.7	97.0	5.80

Static SAS at λ=1.0 maintains 91.3% scale consistency versus PID's 84.7% at the same average λ; all static configurations maintain scale $\geq$91%. PID's dynamic trajectory—generating tokens at varying steering magnitudes within a single piece—amplifies scale degradation beyond what fixed intervention produces. This limitation is specific to duration-up; pitch steering does not exhibit it.

S. Round-Trip Steering Results

Tables 16 and 17 report per-window attribute values for the round-trip experiment (Section 4.4). Each scenario uses 10 extreme songs $\times$ 2 repetitions (n=20), with asymmetric phases: 96 tokens (steer away, m=0.75), 64 tokens (hold with λ decay), 192 tokens (steer back, m=1.25).

PID reduces Window 3 recovery error by 27–69% versus static one-way steering. Per-sample aggregate recovery yields 46% (low→up→down) and 66% (high→down→up), versus 36% and 40% for release (+10 and +26 pp). The release comparison confirms active recovery: in the high scenario, PID reaches 67.8 st versus release's 59.0 st. The low scenario shows mild overshoot (W3=50.2 vs. baseline 57.7 st), reflecting asymmetric model responsiveness to pitch-down steering.

Duration low→up→down shows the strongest recovery: 74% aggregate (vs. 62% release, +11 pp) with 84% lower error than static (3.9 vs. 24.7 ticks). High→down→up is weaker (15% vs. 7% release) due to low peak deviation (9.3 ticks)—the model resists downward duration changes—but PID still recovers 2$\times$ more than release.

Table 16. Round-trip pitch steering: per-window mean pitch (st). Recovery error is |W3 − baseline W3|. Release stops steering in Phase 3 (λ=0).

Scenario	Method	W1	W2	W3
Low→Up→Dn	Baseline	49.40	53.81	57.67
	RT PID	55.74	69.61	50.16
	Static 1-way	65.08	73.60	73.81
	Release	57.20	72.35	68.87
	Rec. err. (PID / Static):		11.4 / 15.7 st	
High→Dn→Up	Baseline	80.93	81.21	81.16
	RT PID	69.34	44.45	67.75
	Static 1-way	54.75	39.32	39.62
	Release	70.81	45.85	58.99
	Rec. err. (PID / Static):		11.1 / 35.8 st	

Table 17. Round-trip duration steering: per-window mean duration (ticks). Release stops steering in Phase 3 (λ=0).

Scenario	Method	W1	W2	W3
Low→Up→Dn	Baseline	2.27	2.17	2.26
	RT PID	5.13	13.22	5.49
	Static 1-way	15.74	25.98	26.46
	Release	3.73	8.68	6.72
	Rec. err. (PID / Static):		3.9 / 24.7 ticks	
High→Dn→Up	Baseline	17.06	16.02	16.08
	RT PID	14.37	7.24	18.69
	Static 1-way	8.30	3.44	3.27
	Release	13.39	5.97	6.48
	Rec. err. (PID / Static):		7.9 / 12.1 ticks	

T. Piecewise-Constant λ Analysis

We test whether holding $\lambda(t)$ fixed between PID updates (update intervals of 4, 8, 16 steps) mitigates duration-up scale degradation (n=40, duration-up steering).

Scale consistency is invariant to update interval (84–86% across all PID variants) and λ standard deviation remains $\approx$0.75 regardless of hold duration, ruling out per-token jitter as the cause. The degradation is intrinsic to PID's integral accumulation trajectory, which creates a distribution of steering magnitudes that differs qualitatively from static SAS's uniform intervention.

Table 18. Piecewise-constant λ for duration-up. λ std: within-sequence standard deviation.

Update interval	Dur. (ticks)	Scale (%)	δ	λ std
1 (per-token)	20.07	85.8	7.75	0.725
4	19.55	85.5	8.13	0.761
8	19.06	86.0	7.87	0.781
16	19.15	84.4	9.09	0.757
Static SAS	21.17	91.9	3.17	0.000

U. Runtime Overhead

Temporal PID's per-token overhead relative to static SAS consists of three lightweight operations: (1) computing the error signal $e(t)$ via a mean over $N{=}32$ pre-indexed feature activations (negligible), (2) the PID control law—three scalar multiply-adds plus two clamping operations (Equation (4)), and (3) updating the integral accumulator $I(t)$. Both static SAS and temporal PID share the dominant cost: SAE encoding ($512{\to}4096$), Top-K re-sparsification, and SAE decoding ($4096{\to}512$) at the steered layer, plus the reconstruction correction Δ.

We benchmark per-token wall-clock time over $n{=}20$ samples (512 max tokens, single A10G GPU). Unsteered baseline generation runs at 9.11 ± 0.02 ms/token. Static SAS adds the shared SAE encode/decode pass, increasing cost to 9.50 ± 0.67 ms/token ($+4.3\%$). Temporal PID reaches 9.68 ± 0.60 ms/token ($+6.3\%$ vs. baseline), with the PID controller itself contributing only $+1.9\%$ marginal overhead beyond static SAS. In practice, autoregressive sampling and attention dominate: the PID-specific computation adds $\mathcal{O}(N+1)$ scalar operations per token atop the $\mathcal{O}(d_{\text{sparse}})$ SAE pass ($d_{\text{sparse}}{=}4096$), making temporal adaptive control essentially free relative to static steering.