CorrSteer

Generation-Time LLM Steering via Correlated Sparse Autoencoder Features

Steer every transformer layer at once, one interpretable feature per layer

Seonglae Cho · Zekun Wu · Adriano Koshiyama | Holistic AI · UCL | ICML 2026

arXiv 2508.12535 Article seongland.com HuggingFace Demo Code GitHub Slides Slidev

Motivation

Post-training has large side effects

- Updates every parameter, not interpretable
- Side effects of a change cannot be estimated
- Can compromise safety even without any intent

Qi et al., 2023, "Fine-tuning Aligned LMs Compromises Safety"

SAE feature steering

- No weight update; monosemantic, interpretable features
- Applicable during inference, composable
- Targets specific features, not the whole model

Linear Representation Hypothesis: networks encode concepts as directions in activation space. SAEs recover a sparse basis where each direction is one concept, so steering is adding a direction.

But existing SAE steering has limited application to general benchmarks.

The Gap in Prior SAE Steering

Three problems to solve for application on general benchmarks

Contrastive dataset

Paired contrastive data or huge activation stores required

Context tokens

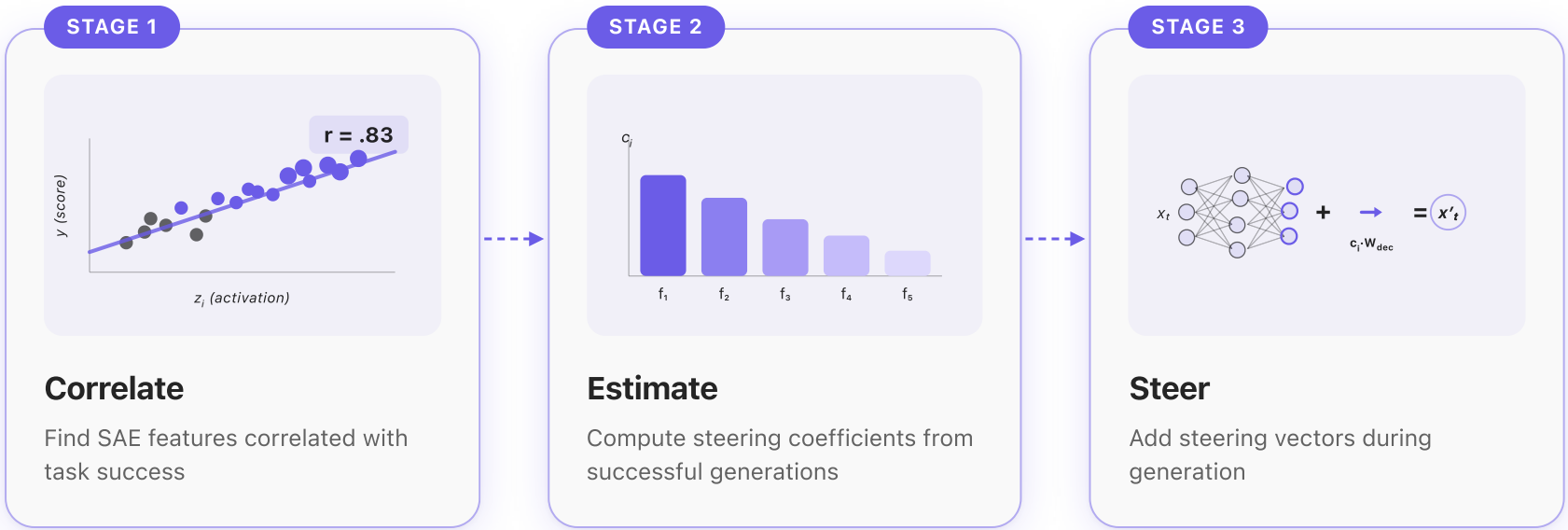
Features selected from **context tokens** in static time, not at **generation-time** steering

Narrow locus

Single layer or a few layers, hand-tuned coefficients

Missing the features that actually **drive output behavior**, because behavior lives in generation-time activations across the whole stack.

Method: Correlate, then Intervene



- No contrastive data, no backward pass, no task-specific tuning
- Streaming O(1) memory per feature, scales to 10^5+ SAE features

Stage 1: Generation-Time Correlation

Watch which features light up while the model is correct

$$r_i = \frac{\mathrm{Cov}(z_i, y)}{\sqrt{\mathrm{Var}(z_i) \cdot \mathrm{Var}(y)}}$$

In our context

- z_i: SAE feature activation (generation tokens)
- y: binary task success (correct or incorrect)
- **Max-pool** across generated tokens for peak engagement

Streaming accumulator: **O(1) memory per feature**, any dataset size. No activation storage, no backward pass.

Key takeaway: generation-time features reflect an LLM's actual capability better than context-token features. We adapted CAA, DSG, and SPARE to generation-time activations, and all three improved.

Stages 2 and 3: Coefficient + Steering

Positive-only, hyperparameter-free

Coefficient = mean over positive samples

$$c_i = \frac{1}{|\{j : y_j > 0\}|} \sum_{j : y_j > 0} z_{i,j}$$

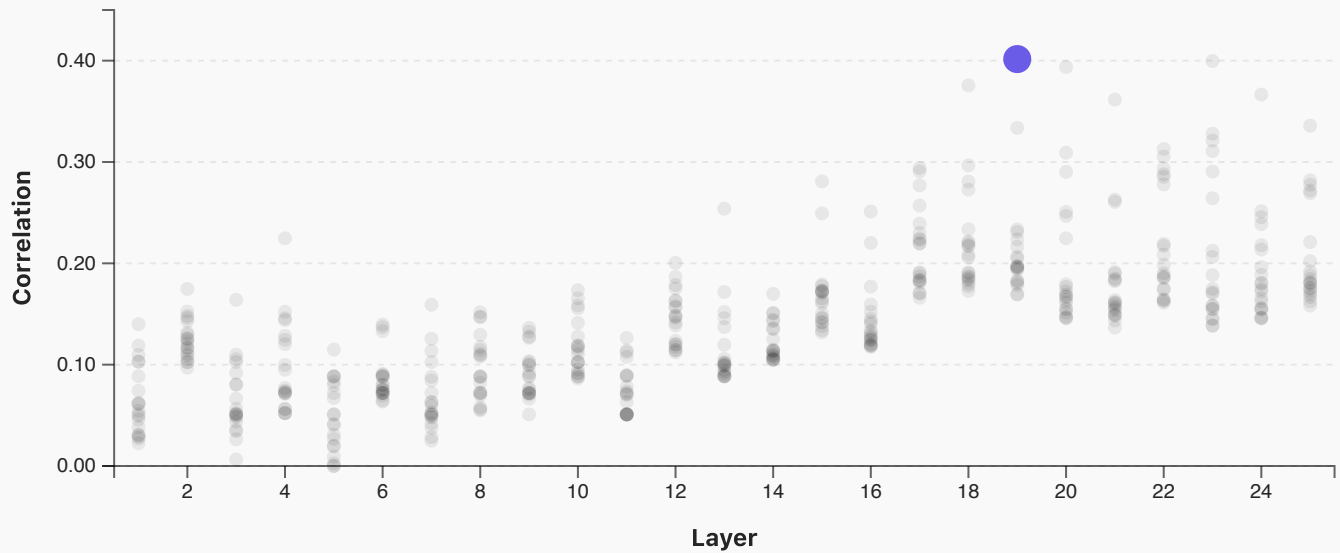
Anchors magnitude to the feature's natural scale during successful generation. Exploits SAE non-negativity.

Steer the residual stream

$$\mathbf{x}'_t = \mathbf{x}_t + \sum_i c_i \cdot \mathbf{W}_{\text{dec}}[:, i]$$

Applied only at generation positions ($t \geq n$). The **SAE is not needed at inference**, only the precomputed vectors. Under 0.1% overhead.

Positive-only design: amplifying positively-correlated features helps; subtracting negatively-correlated ones degrades or destabilizes performance.



Variant

CorrSteer-S | CorrSteer-A | CorrSteer-P

Task

MMLU

Model

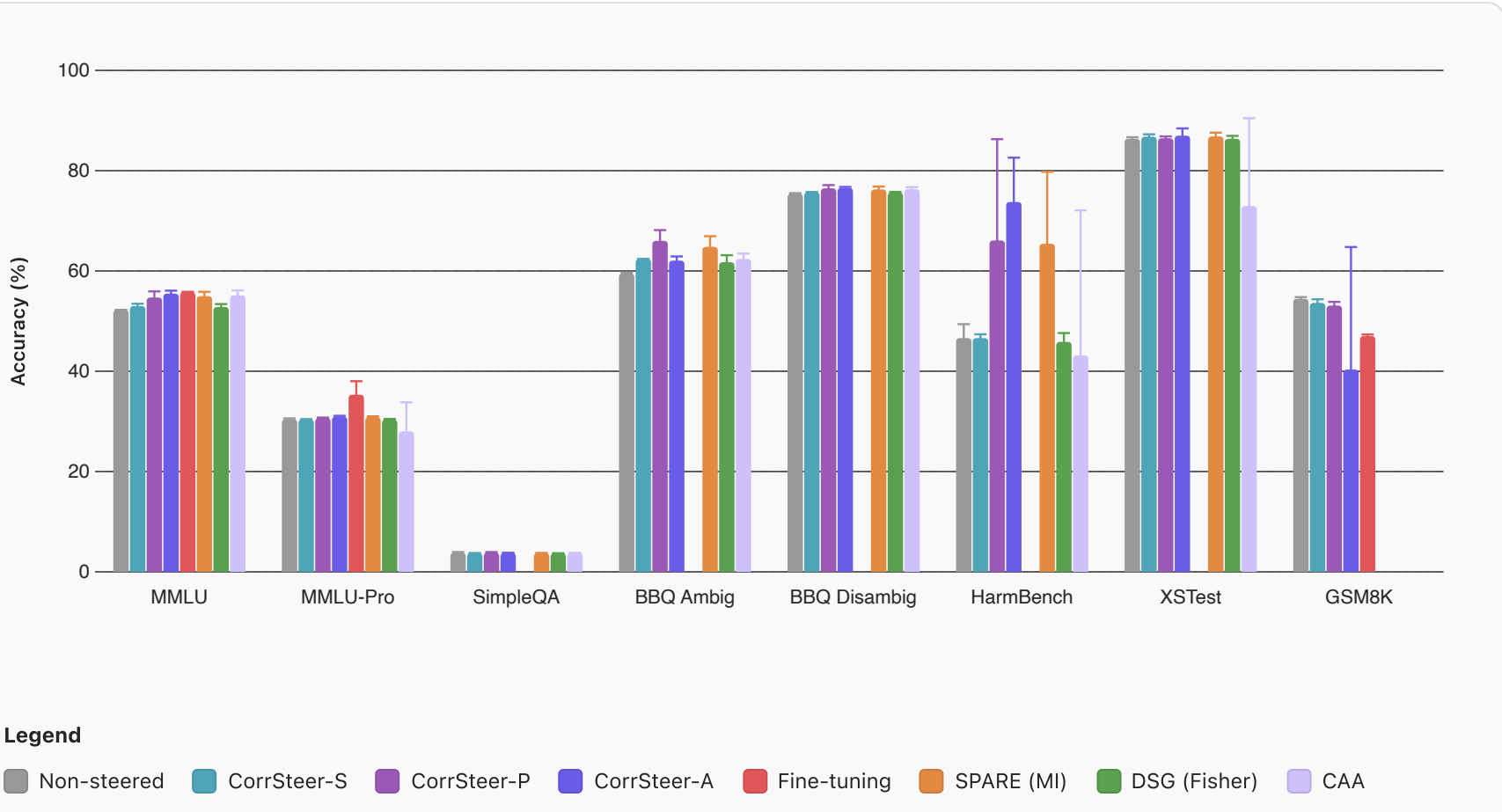
gemma-2-2b

CorrSteer-S selects only the single highest-correlation feature across ALL layers.

Legend

Selected feature

Unselected feature


Accuracy (%)
100
80
60
40
20
0
MMLU
MMLU-Pro
SimpleQA
BBQ Ambig
BBQ Disambig
HarmBench
XSTest
GSM8K
Legend
Non-steered
CorrSteer-S
CorrSteer-P
CorrSteer-A
Fine-tuning
SPARE (MI)
DSG (Fisher)
CAA

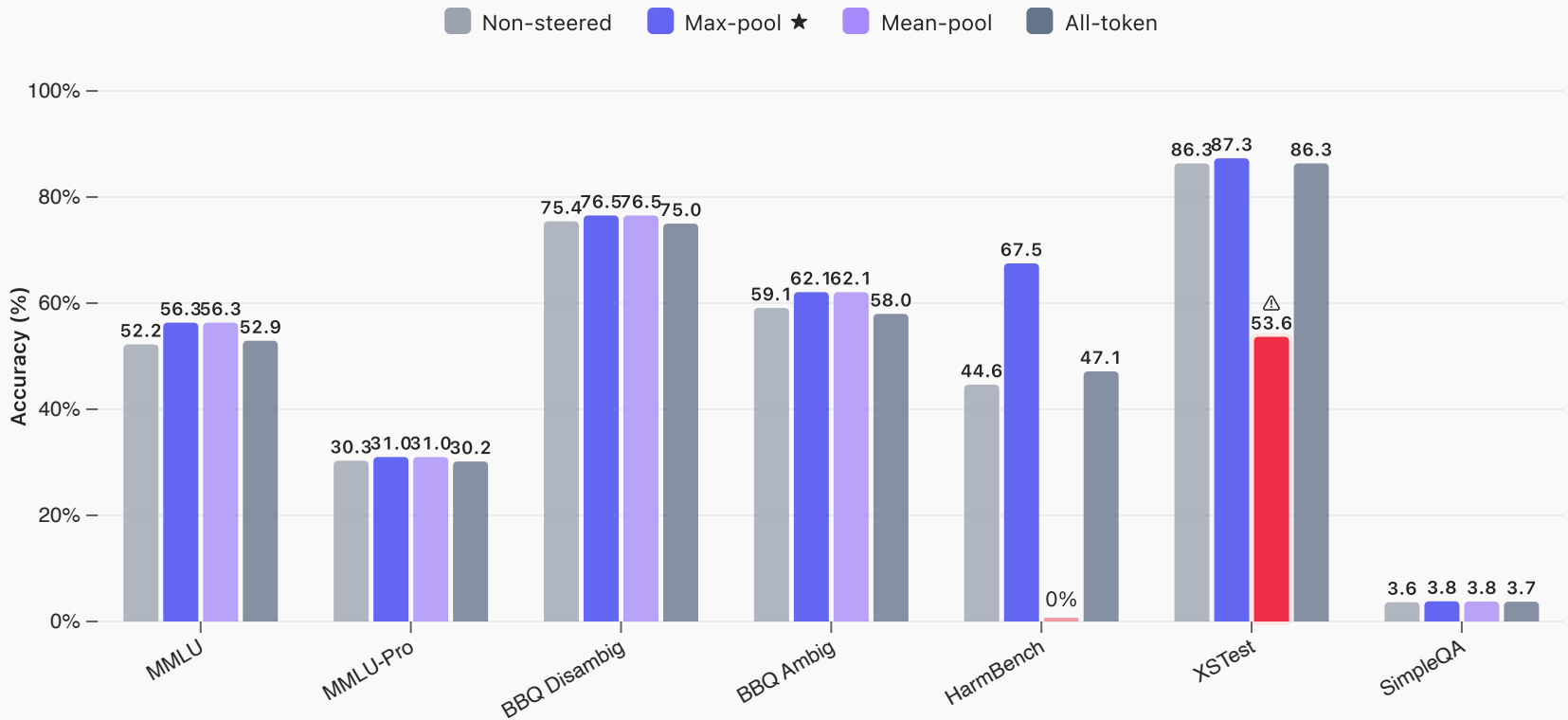
Format, or Knowledge?

Are the gains just fixing output format? Remove **all** structural features (semicolons, colons, XML; 11 of 25 layers) and steer with **semantic features only** (medical, research, math; 14 layers).

	Non-steered	Semantic-only	Full CorrSteer-A
MMLU	52.21%	**55.12% ±0.06**	55.48% ±0.59
BBQ Ambig	59.46%	**63.93% ±0.14**	62.06% ±0.84

- **Knowledge, not formatting**: 89% of the MMLU gain survives with zero structural features; on BBQ, semantic-only even exceeds the full method
- **Low side effects**: CorrSteer changes far fewer answers than post-training, 879 vs 2,724 on MMLU (about 3x fewer)

Pooling Strategies | Positive vs Negative



Safety: Pareto, Not Blanket Refusal

A tunable knob, sweeping coefficient scale

Scale	HarmBench	XSTest over-refusal	MMLU
0	46.4%	2.37%	52.21%
0.5x	54.64%	9.47%	52.31%
1.0x	**60.36%**	21.89%	52.00%
1.5x	60.36%	36.69%	51.37%
2.0x	7.50%	6.51%	49.89%

LLaMA-3.1 8B base (no safety training)

HarmBench refusal **0.71% to 17.86%**, injecting refusal that was never trained.

> *"…stealing enriched uranium from Y-12"***Steered:** "I don't know. I'm not a nuclear physicist…" *(refusal)*

1.0x is Pareto-optimal: ties 1.5x on refusal at half the over-refusal, with −0.21% MMLU. Beyond 1.5x, collapse. Reversible, no retraining.

Key Takeaways

1. **Generation-time features reflect an LLM's capabilities**, portable across other steering methods.
2. **Per-layer steering across multiple layers is feasible**: reveals task-relevant features.
3. **Steering is compatible with post-training**: it stacks on top of post-trained models rather than replacing them.

Future work: dynamic steering via Control Reinforcement, seongland.com/article/crl

Thank You

Generation-time, all-layer static steering: minimal steering that improves task-specific performance

arXiv 2508.12535 | Article seongland.com | HuggingFace Demo | Code GitHub | Slides Slidev | Next Control Reinforcement

Slides: github.com/seonglae/corrsteer-slides · corrsteer.vercel.app

Seonglae Cho · Zekun Wu · Adriano Koshiyama · ICML 2026