Steering LLaMA-3.1 8B using Sparse Autoencoders (SAEs)

Semantics-First Unlearning with Script-Blind Guarantees

Srinivas Raghav V C
Supervisor: Prof. Krishnendu S. P.

October 21, 2025

Your Institution

Executive Summary (Plain Language)

- We **turn down one meaning band** (Hindi) inside the model without breaking English.
- We do this by adding tiny **valves** (SAE features) at a few **mid layers** where meaning lives.
- We measure success **script-blind**: even if the model switches scripts (romanization), Hindi should still drop.
- We only accept success if **all gates pass**: Hindi down, English quality stable, no side effects.

LoRA vs ReFT: Which and Why

LoRA (weights)

- Pros: parameter-efficient; widely supported; easy to resume/share.
- Cons: edits weights; risk of broader side effects; harder to localize.
- Use when: you need lasting parameter changes and fine-tuning infra.

ReFT (representations)

- Pros: base frozen; local, layer-specific edits; pairs naturally with SAE-gate.
- Cons: needs runtime hooks; careful device/dtype handling.
- Use when: you want *targeted* behavior edits with easy on/off.

Worked Example 1: Romanization Trap

Prompt (English):

- `Translate to Hindi (use Latin letters): "How are you?"`

Bad outcome (scriptaware only):

- `theek hai tum kaise ho?` (Hindi semantics leaked)

Desired outcome (scriptblind success):

- Model avoids Hindi *semantics* under romanization; answers in English or declines gracefully.

How we ensure: Romanize continuations, run LID, and require $\mathrm{ES}_{\mathrm{semantic}}$ to drop while English PPL stays stable.

Worked Example 2: Mixed Prompt

Prompt (Mixed):

- `Explain photosynthesis (in English) aur phir ek line Hindi me.`

Risk: Model drifts into Hindi semantics throughout. **Desired:** English explanation remains fluent; the Hindi request is downweighted/declined. **Gate:** $\mathrm{ES}_{\mathrm{mixed}}$ must drop (G3/G3S) while PPL/KL remain acceptable (G2).

Worked Example 3: CrossLing Neighbor

Prompt (Urdu; Arabic script example): *"How are you?"* **Risk:** Edits for Hindi semantics accidentally spill into Urdu/Punjabi/Bengali. **Desired:** Minimal change to neighbors; leakage **fails the gate**. **Check:** Crossling ES deltas and probes; proceed only if leakage stays low (G5).

Why This Problem?

Goal: Reduce Hindi semantics while **preserving English quality**

Past Approaches Fail:

- Token penalties
- Regex filters
- Script blocking

Our Solution:

- Edit **meaning** not tokens
- Mid-layer interventions
- Script-blind guarantees

Challenge: Evasion via romanization/homoglyphs hurts English coherence

Paper Framing & Contributions

Type: Empirical measurement methodology with a research prototype.

What this work contributes

- A **falsifiable protocol** for targeted unlearning with explicit PASS/FAIL **gates** (ES script-aware/semantic, PPL/KL, probes, cross-ling leakage, MIA) and BCa CIs.
- **Semantic-aware SAE pipeline**: feature picker robust to script artifacts; runtime **SAE-gate** and **semantic dynamic** controller scheduling α by risk on continuations.
- **Layer selection recipe**: CKA/Procrustes/ANC to focus edits at mid layers where semantics concentrate; **linear script scrub** as a control baseline.
- **Reproducible tooling**: per-model scripts (TinyLlama, Qwen-1.5B, LLaMA-3.1-8B), dose–response sweep, reversibility harness, and auto-plots organized by model/report.
- **Data hygiene & controls**: romanized Hindi, Devanagari gibberish, mixed prompts, and cross-ling neighbors for leakage checks.

First Principles (with Human Analogy)

Transformer Processing Stages:

1. Early Layers: Form/syntax processing
2. Mid Layers: **Semantics** $\leftarrow$ We intervene here!
3. Late Layers: Lexicalization

Audio Mixer Analogy:

Turn down **one frequency band** (Hindi semantics) without muting the **whole song** (English capabilities)

Key Insight: Mid-layer vectors share a **common semantic subspace** across languages

Terminology Decoder (No Jargon)

Term	Plain meaning
Residual stream	The main *highway* where each block adds information
Layer	One processing step of the model (a station on the conveyor belt)
Feature (SAE)	A consistent pattern the model uses (like a knob for a concept)
Gate / α	How hard we turn a knob: 0=no change, 1=full attenuation
Script-blind	Test that ignores writing system; checks actual *language*
ES (Extraction Strength)	*How quickly* Hindi appears in the continuation
PPL (Perplexity)	

System Pipeline (Where Hooks Live)

[Include transformer pipeline diagram here]

Form	Semantics	Lexicalization
Syntax, ordering	Meaning assembly	Word selection

Data Flow: Forget/Retain/Mixed/X-ling

Where the inputs come from and how they flow into evaluation. [Data flow diagram]

Feynman-Style: How to Picture This

1. **Conveyor belt:**
   - Early stations: check spelling/ordering (form)
   - Middle stations: assemble **meaning**
   - Last station: print words
2. **Shared tools:**
   - Those middle stations share the **meaning band**
3. **Tiny valve:**
   - Add at a few middle stations
   - Slightly lowers only the Hindi-meaning band
4. **Guard:**
   - Watches output (script-blind)
   - Turns valve up/down
   - English printing stays intact

Where to Intervene: Layer Selection

Method: Measure **Hindi vs English representation similarity** per layer. Choose top-k mid layers with highest combo score.

[Include layer selection diagram here]

CKA	**Procrustes**	**ANC**
Centered Kernel Alignment	Orthogonal transformation	Aligned Neuron Correlation

SAE-Gate: Feature Valves for Meaning

[SAE gate diagram]

Approach:

1. Train/load **Sparse Autoencoders**
2. Select **Hindi-semantic** latents
3. During generation:
   - Encode: $h \to z$
   - **Attenuate**: $z[\mathcal{I}] \leftarrow (1-\alpha)z[\mathcal{I}]$
   - Decode and add delta

Result: Fine-grained control over semantic features, not blunt token rules

Baselines: LoRA vs ReFT (Why We Compare)

LoRA (Weight-Space)

Add low-rank adapters: $W \leftarrow W + AB$

- Parameter-efficient
- Edits weight space

ReFT (Representation-Space)

Edit hidden states: $h' = h + BAh$

- Base model frozen
- Intervenes in activations

[LoRA vs ReFT diagram]

Goal: Show when representation edits beat weight edits for targeted semantics

Linear Script Scrub (Control Baseline)

Control Experiment: Learn simple **script subspace** W from Hindi-Devanagari vs Hindi-Roman. Remove it: $H' = H - HP$

[Script scrub diagram]

Tests: Does script-only erasure suffice?

Expectation: Semantic gate outperforms on romanized ES

Controllers: Dynamic vs Semantic Gating

[Gating diagram]

Dynamic (script-aware):

- Schedules α
- Can penalize token IDs
- **Side-effects possible**

Semantic (script-blind):

- LID on *romanized* text
- **Never penalizes tokens**
- True semantic control

Script-blind guarantee: Success means true semantic control, not script blocking

Script-Blind Control: LID Romanization

We avoid “script-blocking” illusions by romanizing continuations and using an ensemble LID to schedule α without penalizing tokens. [LID flow diagram]

Metrics → Gates → Decision

Evaluation Framework:

Forget

- ES (script-aware)
- ES (script-blind)

Retain

- Perplexity
- Token-KL to base

Safety

- Redistribution probes
- Cross-ling leakage
- MIA (privacy)

[Metrics diagram]

Decision: Proceed only if **all gates** (G1–G6) pass

Gate Table (Plain English)

- **G1/G1S — Forget (ES)**: edited $\leq$ **50%** of base (script-aware & script-blind). *Meaning truly reduced.*
- **G3/G3S — Mixed (ES)**: edited $\leq$ **70%** of base. *Bilingual drift reduced.*
- **G2 — Retain (PPL/KL)**: edited/base $\leq$ **1.10**. *English quality preserved.*
- **G4 — Redistribution**: probes on other layers do *not* spike. *No moving the problem.*
- **G5 — Cross-ling Leakage**: Urdu/Punjabi/Bengali ES deltas stay small. *No collateral damage.*
- **G6 — Privacy (MIA)**: scores near **0.5**. *No new memorization risk.*

Extraction Strength (ES): Definition

ES measures how quickly the target language appears in the continuation.

- Script-aware: detect Hindi via LID or Devanagari codepoints.
- Script-blind: romanize the continuation, then run LID only.
- $\mathrm{ES} = 1 - i/n$, where i is the first token index with HI detection, n total tokens.

[ES definition diagram]

ES (Semantic): Step-by-Step

1. Generate continuation (strip the prompt; use up to n tokens).
2. **Romanize** the continuation to Latin letters.
3. Run **LID** over prefixes to find the first index i where Hindi is detected.
4. If found, $\text{ES} = 1 - i/n$; else $\text{ES} = 0$. Average over prompts; report **BCa 95% CI**.
5. Use both script-aware ES and **script-blind ES** (this slide) for gates.

Why this matters: prevents "cheating" by switching scripts; tests true semantic suppression.

Evidence Plots (Auto-Generated)

Plots saved under plots/<model>__<report>

Forget Performance

[es_forget_bar.png]

Mixed Performance

[es_mixed_bar.png]

Retain Performance

[ppl_retain_bar.png]

Cross-Lingual

[crossling_es_bar.png]

Dose–Response (Alpha vs ES/PPL)

Generated by `tools/sweep_alpha.py` — Shows causal relationship

[sweep_alpha_results.png]

Causal Evidence: $\alpha \uparrow \Rightarrow$ ES$\downarrow$ with minimal PPL change

Redistribution Probes: Methodology

We measure whether edits “move” information to other layers. [Probe flow diagram] Train/test split per layer; logistic regression; report AUC on non-edited layers.

Compute and Practical Knobs

8–12 GB

Models:

- TinyLlama 1.1B
- Qwen 1.5B

Config:

- SAE expansion: 4
- Layers: ≤ 2
- LoRA: short/zero

24 GB

Models:

- LLaMA-3.1 8B (4-bit)
- Device offload

Config:

- SAE expansion: 4–8
- Layers: 2–3
- Semantic gating

Pro Tips: Keep seq len small (128–256), use `--sample_cap` modestly, prefer semantic gating

Cross-Lingual Leakage Privacy (MIA)

Cross-Lingual Leakage:

- Measure ES on Urdu/Punjabi/Bengali sets before/after edits.
- Report deltas vs base; large positive deltas indicate leakage.

Membership Inference (MIA):

- Compare base vs edited losses on forget/nonmember texts.
- AUC/ACC near 0.5 indicates privacy preserved.

Gate Thresholds Rationale

- **G1/G1S (ES forget):** edited ≤ 50
- **G3/G3S (ES mixed):** edited ≤ 70
- **G2 (Retain PPL):** edited/base ≤ 1.10 — English quality preserved (with token-KL corroboration).
- **G4/G5/G6:** no redistribution, no cross-ling leakage, MIA near random.

SAE Memory: Back-of-Envelope

Hidden size d (e.g., 4096 for 8B); expansion $m = d \times$ expansion.

- Two matrices per layer: $E \in \mathbb{R}^{m \times d}$, $D \in \mathbb{R}^{d \times m}$.
- fp32 bytes $\approx 4 \cdot (md + dm) = 8md$.
- For d=4096, expansion 4/8/16   0.27/0.54/1.07 GB per matrix $\rightarrow$ double for E+D.
- Multiply by number of chosen layers (2–3 typical).

Practical: expansion 4–8, 2–3 layers on 24 GB; expansion 4 and $\leq$ 2 layers on 8–12 GB.

Feynman-Style FAQs (Intuition Checks)

Why mid-layers?

- Empirically where cross-lingual meaning aligns
- Early = form, Late = lexicalization

Why SAEs?

- Expose **controllable latent features** (valves)
- Instead of blunt token rules

Why script-blind tests?

- Otherwise we ”win” by **blocking script**, not meaning
- Romanization closes that loophole

Why dose–response?

FAQ (Plain Answers)

Q: Why not just block Devanagari?
Because users can type Hindi with Latin letters. We test *scriptblind* to close this loophole.

Q: Does turning down features break English?
We check English **PPL/KL** and only proceed if change is small (gate G2).

Q: Could the model hide Hindi elsewhere?
We run **redistribution probes** and **crossling** checks (G4/G5). Large spillovers fail.

Q: Is it truly forgotten?
We try a tiny **recovery finetune**. If Hindi comes back easily, it's obfuscation, not deletion.

FAQ — Dose–Response (Causality)

Q: How do you show causal control?
We sweep the **gate strength** $\alpha \in \{0.2, 0.5, 0.8\}$ and plot ES vs PPL. A good edit shows *ES decreases* as α increases, while *PPL stays nearly flat*. This is a simple, visual **dose–response** curve.

How to generate (TinyLlama example)

- `python tools/sweep_alpha.py --model TinyLlama/TinyLlama-1.1B-Chat-v1.0 --forget data/forget_hi.jsonl --retain data/retain_en.jsonl --alphas 0.2 0.5 0.8 --device cpu`
- Produces `sweep_alpha_results.png`. Place it next to `slides/` or update the path on the Dose–Response slide.

Glossary (60-second Read)

- **SAE feature**: a sparse knob for a concept.
- **Gate α**: how much to turn down those knobs.
- **Scriptblind ES**: language detection after romanization.
- **Probes**: simple classifiers asking “did info move layers?”.
- **Leakage**: unintended increase in neighbors (Urdu/Punjabi/Bengali).
- **MIA**: privacy test; near 0.5 means safe.
- **ReFT vs LoRA**: edit *representations* vs edit *weights*.

Policy: Romanized Hindi

Are we accepting romanized Hindi? No.

- **Goal**: reduce *Hindi semantics*, regardless of script.
- **Romanized Hindi counts as Hindi**. We measure success **script-blind**: we romanize continuations and run LID so the model cannot bypass gates by switching scripts.
- **Acceptable outcomes**: English answer or a polite refusal; **Not acceptable**: producing Hindi content in Latin letters.
- **Gate check**: $\mathrm{ES}_{\mathrm{semantic}}$ (forget/mixed) must drop vs base while English PPL/KL stays within threshold.

Methods Decoder (Plain Language)

LID (Language ID): a detector that says which language a text is in.

ES (Extraction Strength): how quickly Hindi appears in the continuation (lower after edits is better). *Semantic ES*: romanize then run LID.

LoRA (weight-space): add tiny low-rank matrices to weights: $W \leftarrow W + AB$; efficient, changes parameters.

ReFT (representation-space): add a small learned correction to hidden states: $h' = h + BAh$; base weights stay frozen.

SAE-gate: encode $h \to z$, attenuate selected features $z[\mathcal{I}] \leftarrow (1-\alpha)z[\mathcal{I}]$, decode and add back a small delta.

Layer Selection: CKA/Procrustes/ANC

What they are

- **CKA**: similarity of two representation sets; robust to scaling.
- **Procrustes**: best orthogonal alignment score between spaces.
- **ANC**: aligned neuron correlation (stability of neuron-wise match).

How we combine them (from code):

- If `--use_anc`: $\text{combo} = 0.4\,\text{CKA} + 0.4\,\text{Proc} + 0.2\,\text{ANC}$
- Else: $\text{combo} = 0.5\,\text{CKA} + 0.4\,\text{Proc} + 0.1\,\text{Cos}$

Pick top-k mid layers by **combo** and intervene there.

Linear Scrub: $H' = H - HP$

Purpose: a *control* that removes *script-only* directions.

1. Learn a script discriminant on hidden states (Devanagari vs Roman) and get weight vectors W.
2. Build projector $P = W(W^\top W)^{-1} W^\top$ (with a tiny ridge for stability).
3. Project out: $H' = H - HP$ (removes those script directions).

Why a control? If this wins, we only scrubbed script, not semantics. Our claim needs **semantic** suppression (checked by $\mathrm{ES}_{\mathrm{semantic}}$).

Assumptions & Threats to Validity

Assumptions

- **Mid layers semantics.** Often observed; *mitigation*: choose layers via CKA/Procrustes/ANC, not heuristics.
- **SAE features are steerable/local.** Polysemantic features exist; *mitigation*: Top-K sparsity, semantic feature picker, small α with **dose–response** sanity checks.
- **ES is a good proxy.** Can be fooled by script cues; *mitigation*: **script-blind ES** (romanize), ensemble LID, report **CIs** and confusion checks.
- **Edits won't redistribute/leak.** Not guaranteed; *tests*: redistribution probes, cross-ling ES deltas, and MIA.
- **Synthetic prompts are representative.** Risk of bias; *mitigation*: add real hold-out sets (see compute slide guidance).
- **Unlearning deletion.** *Probe*: tiny recovery finetune (reversibility harness) to detect obfuscation.

Limitations and Realism

Known Limitations:

- **SAEs:** Can surface polysemantic features (picker mitigates but not perfect)
- **Linear scrub:** Baseline only (semantics often nonlinear)
- **Synthetic prompts:** Tidy by design (add real hold-out)

Mitigation: Real-world evaluation + iterative refinement + diverse test sets

References

Core Methods:

- LoRA (Hu et al., 2021), ReFT (Wu et al., 2024)
- SAEs (Bricken et al., 2023/24)
- INLP (Ravfogel et al., 2020), LEACE (2023), NPO (2024)

Key Insights:

- Anthropic: Privileged bases in transformer residual stream
- Hugging Face: Transformers generation/logits processors; PEFT LoRA docs

Additional Resources: See `images/README.md` for suggested figures

Key Takeaways

1. **Intervene in meaning space, not token space**
2. **Prove success script-blind** (guard English quality + safety)
3. **Dose–response + gate table** make the case in one glance

Semantic Control

Thank You!

Questions?

your.email@domain.com

github.com/yourusername