Contents

Reconstruction Without Recall: Two-Tier Verbalization in Natural Language Autoencoders

Format Granularity Hides Content Decoupling at the Last Token of an Instruct Chat Template

Workshop draft for NeurIPS 2026 Mechanistic Interpretability Workshop. Apache-2.0. Reproducible. Single-author submission, double-blind by ICML/NeurIPS conventions.

Abstract

Natural Language Autoencoders (NLA; Fraser-Taliente et al. 2026) train an activation-verbalizer (AV) and an activation-reconstructor (AR) end-to-end with GRPO so that the round-trip MSE between original and AR-reconstructed activations serves as a learnable explanation-quality reward. We replicate the canonical recipe on three NLA pairs from the kitft release spanning two model families and three scales — `kitft/nla-qwen2.5-7b-L20`, `kitft/nla-gemma3-12b-L32`, and `kitft/nla-gemma3-27b-L41` — and show that the headline metric, `fve_nrm`, decouples from semantic content fidelity across all three models, with three distinct scaling behaviors that sharpen the methodological position. On a 50-prompt corpus stratified across four categories (chat / code / agent / reasoning) verbalized at K=3 samples each (N=150 per model), `fve_nrm` is uniform across categories at high absolute level (Qwen 0.880 / Gemma-12B 0.992 / Gemma-27B 0.982; spreads 0.017 / 0.005 / 0.010, all exceeding the paper's reported 0.752 in-distribution

baseline) while keyword recall between the prompt and AV's natural-language explanation varies 6.5–8.8× across the same categories (Qwen: chat 0.578 → agent 0.088, spread 0.490; Gemma-12B: chat 0.782 → agent 0.133, spread 0.649; Gemma-27B: chat 0.813 → agent 0.160, spread 0.654). The three-model trajectory reveals **three differential scaling axes**: (a) overall content-fidelity signal-above-floor grows monotonically with model quality (permutation gap +0.27 → +0.38 → +0.43, no ceiling visible); (b) per-category recall spread saturates between 12B and 27B (0.490 → 0.649 → 0.654) at what appears to be a training-distribution-imbalance ceiling; (c) Tier 1 `fve_nrm` peaks at moderate model size (Gemma-12B 0.992 max, slight regression to 0.982 at 27B), suggesting Tier 1 quality is layer-extraction-dependent rather than purely scale-dependent. Three controls validate the gap on all three models. Permutation: shuffled pairs drop to 0.038–0.063 (cross-cat) while real recall reaches 0.329 → 0.422 → 0.475; the agent gap above floor remains floor-level (+0.045 in V2) across the trajectory. Random Gaussian baseline: L2-matched random vectors collapse reconstruction monotonically (Qwen `fve_nrm` = −0.949 → Gemma-12B −0.992 → Gemma-27B **−1.000** with exact orthogonal cosine), while AV produces increasingly contracted format templates (V1 heterogeneous formats → V2 "Structured X format" diversity → V3 "Educational/X article format" hyper-template attractor on 6/6 random inputs). Direction-injection: 4/4 (Qwen) and 3/4 (both Gemmas) category-mean-difference directions verbalize to the correct category-template with negation symmetry, the agent failure mode being model-specific (Gemma-12B: agent → code due to code-content overlap; Gemma-27B: agent → chat due to format-prior contraction into the "Educational article" attractor). We argue NLA verbalization is two-tier: Tier 1 (format/category) is direction-modulated and what `fve_nrm` measures; Tier 2 (specific content — file paths, named entities, math entities) is largely unencoded. **The decoupling magnifies with NLA training quality up to a structural ceiling**: overall signal-above-floor continues to grow, but per-category spread saturates at a training-distribution-imbalance limit, and the format prior contracts into a single hyper-template attractor as Tier 1 approaches the `fve_nrm` ceiling. Better NLA training makes `fve_nrm` less informative about per-category Tier 2 fidelity, not more. NLA can format-classify a residual direction but cannot content-decode it. We recommend reporting category-stratified semantic-recall metrics alongside `fve_nrm` for NLA-style evaluation, and we provide all three reproducibility artifacts: `nb_track_a_phase16_decoupling.ipynb` (V1 Qwen), `nb_track_a_phase16_gemma_crossmodel.ipynb` (V2 Gemma-12B), and `nb_track_a_phase16_gemma27b_v3.ipynb` (V3 Gemma-27B).

---

1. Introduction

The 2026 Anthropic paper *Natural Language Autoencoders Produce Unsupervised Explanations of LLM Activations* (Fraser-Taliente et al.) introduced a striking construction: a fine-tuned LM (the AV) that, when its prompt has a single token embedding overwritten by an arbitrary residual-stream activation, generates a natural-language description of that activation; paired with a second fine-tuned LM (the AR), truncated to K+1 layers and topped with a learned `Linear(d, d)` head, that maps the AV's text back into the original activation space. The pair is trained with GRPO using round-trip MSE as the reward. The released artifacts include public NLA pairs for Qwen2.5-7B (L20), Gemma-3-12B (L32), Gemma-3-27B (L41), and Llama-3.3-70B (L53).

The metric the paper reports is `fve_nrm` — the fraction of variance explained between the original and reconstructed activation, after both vectors are L2-normalized to `mse_scale = $\surd$d`. Per the released artifact, the Qwen2.5-7B-L20 critic achieves `fve_nrm = 0.752` on a 50/50 mix of WildChat (chat) and UltraFineWeb (web text). The implicit claim is that `fve_nrm` is a faithful proxy for explanation quality: a high score means the explanation captured the activation's content well enough that the AR can recover it.

We test this assumption directly. If `fve_nrm` measures explanation quality, then varying the prompt distribution should produce parallel changes in `fve_nrm` and in any independent measure of how well the explanation describes the prompt's content. We find this is not the case. Across four prompt categories spanning short factual chat, programming requests, SWE-bench-Pro-style agent tasks, and classical reasoning puzzles, `fve_nrm` is essentially constant at 0.880 — exceeding the paper's headline number on the diverse training mix — while a simple keyword-recall metric varies by a factor of 6.5 across the same categories. The gap is largest where it matters most for downstream interpretability use cases: agent-format prompts, where the specific subject

is encoded in file paths and function names, achieve recall 0.088 — a value that, as our permutation control shows, is statistically indistinguishable from the random-shuffle floor of 0.043.

A natural reaction is that recall is a bad proxy for content fidelity. Two controls argue otherwise. First, real `prompt$\leftrightarrow$explanation` pairs exceed shuffled pairs by recall +0.266 on average (cross-category permutation), and within categories the gap reaches +0.486 for chat — so the metric does distinguish real signal from category-vocabulary baseline. Second, we generate explanations from random Gaussian activations (L2-matched to typical residual norms) and find that the AR's reconstruction collapses to `fve_nrm = -0.949` (worse than predicting the mean), but AV's natural-language explanations remain coherent and format-locked: "Formal wiki article structure with numbered facts about a cultural history magazine," "Technical product ingredient data format with ISO standard structure." The format template fires unconditionally; only the direction modulates *which* template fires.

A third control — direction-injection — sharpens the picture. We construct four category-mean-difference directions from the captured Phase 16 activations (`mean(chat_acts) - mean(other_acts)`, etc.), inject them into AV at the canonical `injection_scale = 150`, and check the resulting explanation keywords. All four "self category" directions verbalize to their own category-template (chat → article-keywords; code → code-keywords; agent → technical/code-keywords; reasoning → math-keywords); the four negations symmetrically verbalize to opposite-category templates (NEG_chat → code-template; NEG_code → chat-template); the cross-axis chat↔agent works as expected. NLA *does* verbalize directions correctly — but only at the **format-template** granularity, not at the content level.

This is the position paper's central claim. NLA verbalization is two-tier:

> **Tier 1 (FORMAT/CATEGORY)** is direction-modulated and is what `fve_nrm` measures.
> **Tier 2 (CONTENT/SPECIFICITY)** — file paths, named entities, math entities, test names — is largely unencoded.

The reconstruction-loss metric is blind to Tier 2 because the AR can read sufficient structural signal from Tier 1 alone to recover activation direction. This explains the four observations together: uniform `fve_nrm` across categories (Tier 1 always recoverable), category-dependent recall (Tier 2 fidelity varies with how much content is in-distribution for the AV), coherent explanations from random Gaussian (Tier 1 prior fires unconditionally), correct direction-injection alignment (Tier 1 prior is direction-modulated, not fixed). It also predicts where NLA can and cannot be used for downstream interpretability: as a format-classifier for residual-stream directions, yes; as a content-decoder for, e.g., probe-direction semantics, no.

Our contributions:

1. **Empirical**: a 4-category × 50-prompt × 3-sample N=150 reproduction of the canonical NLA recipe on `kitft/nla-qwen2.5-7b-L20`, demonstrating uniform `fve_nrm` (spread 0.017) and 6.5× category-spread in keyword recall (0.49 spread).
2. **Methodological**: three controls that should accompany NLA-style evaluation — permutation, random-Gaussian, and direction-injection — and the characteristic numerical signatures of each.
3. **Conceptual**: a two-tier verbalization thesis that explains why reconstruction loss is necessary but insufficient for explanation quality in NLA, and predicts the regime where NLA is and is not useful for probe interpretation.
4. **Engineering**: a single-cell self-contained reproducibility notebook (`nb_track_a_phase16_decoupling.ipynb`) that runs end-to-end in ∼30 minutes on a single H100.

---

2. Setup

2.1 Models

We test three NLA pairs from the kitft release covering two model families and three scales (7B → 12B → 27B):

V1 — Qwen2.5-7B at L20:

- **Target**: Qwen2.5-7B-Instruct (Alibaba, 2024). 28 layers, d_model=3584, bf16 inference. We extract residual stream at the output of block 20.
- **AV (verbalizer)**: `kitft/nla-qwen2.5-7b-L20-av`. Full 28-layer LM fine-tuned with GRPO; embedding-injection at the injection-character `[U+320E]` (U+320E, token id 149705); residual norm rescaled to L2 = 150 before injection (`extraction.injection_scale` from the `nla_meta.yaml` sidecar).
- **AR (reconstructor)**: `kitft/nla-qwen2.5-7b-L20-ar`. 21-layer truncated LM with `model.norm` and `lm_head` replaced by `nn.Identity()` (per kitft/nla-inference recipe — value_head reads raw post-block-K residual) and a separately shipped `Linear(3584, 3584, bias=False)` value_head loaded from `value_head.safetensors`.

V2 — Gemma-3-12B at L32:

- **Target**: `google/gemma-3-12b-it` (Google DeepMind, 2025). Multimodal `Gemma3ForConditionalGeneration` with text component nested as `model.language_model` (48 transformer layers, d_model=3840, bf16). We extract from layer 32 — comparable depth fraction to Qwen V1 (32/48 = 67% vs 20/28 = 71%). Gated repository, requires HF_TOKEN.
- **AV / AR**: `kitft/nla-gemma3-12b-L32-{av,ar}`. Same canonical recipe as V1, with three Gemma-3-specific adaptations:
  (a) `injection_scale = 80000` (vs Qwen's 150) — Gemma's residual norms are $\sim 600\times$ larger due to $\sqrt{\text{d-model}}$ embedding scaling; the captured residuals at L32 have L2 ~73k–82k across categories;
  (b) `injection_char = [U+321C]` (U+321C, token id 246566);
  (c) AR tokenization with `add_special_tokens=True` is load-bearing — the kitft README documents that dropping the BOS prefix tanks Gemma's in-distribution `fve_nrm` from 0.77 to 0.31 (Qwen has no BOS so the flag is a no-op there). value_head shape is `Linear(3840, 3840, bias=False)`.

V3 — Gemma-3-27B at L41:

- **Target**: `google/gemma-3-27b-it`. 62 transformer layers, d_model=5376, bf16. We extract from layer 41 — depth fraction 41/62 = 66%, comparable to V1's 20/28 = 71% and V2's 32/48 = 67%. Gated repository.
- **AV / AR**: `kitft/nla-gemma3-27b-L41-{av,ar}`. Same canonical recipe. Sidecar values: `injection_scale = 60000` (kitft re-calibrated for 27B, vs 12B's 80000), `injection_char = [U+321C]` (same as V2), `mse_scale = $\surd$5376 $\approx$ 73.32`. value_head shape `Linear(5376, 5376, bias=False)`.
- **Compute note**: at bf16, target Gemma-3-27B-IT and AV are each ~54 GB. PyTorch's caching allocator does not always release CUDA memory after `del model; gc.collect(); torch.cuda.empty_cache()`, which produces an OOM on Colab RTX 6000 96GB when AV loads after target capture. The V3 notebook documents the kernel-restart workaround (save acts to Drive, restart Colab kernel, reload acts, then load AV in clean GPU state). This affects reproducibility ergonomics, not the decoupling thesis.

The cross-model comparison is direct: same 50 prompts, same canonical recipe, same metrics, three NLA pairs from two model families spanning 7B $\rightarrow$ 12B $\rightarrow$ 27B parameters, with different scales, layers, embedding conventions, and chat templates. If results converge, the decoupling thesis is not a model-family- or single-scale artifact.

2.2 Prompt corpus

We constructed a 50-prompt corpus stratified across four categories chosen to span typical activation-decoding use cases:

Category	N	Style
chat	13	Short factual Q&A (Wikipedia-style). "What is the capital of Brazil?", "Who painted the Mona Lisa?"
code	12	Programming task requests across Python/SQL/Bash/JS/C. "Implement quicksort in C without using recursion."
agent	13	SWE-bench-Pro-style agent tasks with explicit file paths and function names. "Find all calls to `requests.get` without a timeout parameter and add timeout=30."
reasoning	12	Math, logic, classical puzzles. "Prove that the sum of the first n odd integers equals n^2."

Each prompt is wrapped in the standard Qwen Instruct chat template with `add_generation_prompt=True`, then forwarded through Qwen2.5-7B-Instruct with a forward hook on `model.layers[20]`. The captured residual at the last input token (post-block-20) is the activation we feed to AV.

2.3 Metrics

We report two metrics on each `(activation, explanation)` pair:

- `fve_nrm` (paper metric): both prediction and ground-truth are L2-normalized to `mse_scale = $\surd$d $\approx$ 59.87`, then `mse = mean((p - g)$^2$)` and `fve_nrm = 1 - mse / var(g)`. This equals `2$\cdot$cos - 1` for unit-L2-normalized vectors with zero mean; it captures direction equivalence only.
- **keyword recall** (semantic content metric): for each pair, extract content words (alphanum, length $≥$4, non-stopwords) from prompt and explanation; recall = `|prompt_words $\cap$ explanation_words| / |prompt_words|`. This measures whether the explanation contains the prompt's specific subject.

Both metrics range in [0, 1] for our purposes. They are simple, transparent, and (as we show below) not redundant.

2.4 Canonical inference recipe

Five aspects of the canonical recipe are easy to get wrong; we encountered all five in our first reproduction attempt and they masked the decoupling finding entirely. Documenting them here for reproducibility:

1. **`value_head.safetensors`** is a separate file from `model-*.safetensors`. `from_pretrained` does not auto-load it. Must open with `safetensors.load_file()` and attach as `nn.Linear(d, d, bias=False)`.
2. **`model.norm` must be `nn.Identity()`**. The released AR safetensors strips `model.norm.weight` because the value_head reads raw post-block-K residual, NOT the normed version. transformers re-initializes the missing key as RMSNorm with `weights=ones`, which produces apparently-OK reconstructions on short prompts (cos 0.94 by accident — RMSNorm with unit weights happens to scale right) but NaN on longer agent-format prompts.
3. **`lm_head` must be `nn.Identity()`**. The critic never emits logits.

4. **AR prompt template**: from `nla_meta.yaml`, `'Summary of the following text: <text>{explanation}</text> <summary>'`. Tokenize with `add_special_tokens=True`. Raw text without this wrapper drops cosine by ∼0.18.
5. **`fve_nrm` requires L2-normalization to `mse_scale`**. The natural FVE computed on raw vectors is dominated by magnitude mismatch and is uninformative.

The full canonical `reconstruct()` function is ∼10 lines of code; the trap is in the four model-state patches and the metric.

We sample explanations from AV at `temperature = 1.0` with `K=3` per activation (150 explanations total). We chose K=3 to bound the within-prompt variance estimate; preliminary single-sample runs (Phase A POC of this work) gave qualitatively identical results.

---

3. Decoupling: `fve_nrm` is uniform, recall is category-spread

3.1 Headline numbers

V1 — Qwen2.5-7B at L20 (N=150):

Category	N	cos	`fve_nrm`	recall (mean)	recall ($≥$1 hit)
chat	39	+0.940	0.880	0.578	0.949
code	36	+0.943	0.887	0.351	0.944
agent	39	+0.942	0.883	0.088	0.538
reasoning	36	+0.935	0.870	0.325	0.606
OVERALL	**150**	**+0.940**	**0.880**	**0.336**	**0.760**
Spread		0.008	**0.017**	**0.490**	0.411

V2 — Gemma-3-12B at L32 (N=150):

Category	N	cos	`fve_nrm`	recall (mean)	recall ($≥$1 hit)
chat	39	+0.997	0.995	0.782	1.000
code	36	+0.995	0.990	0.404	1.000
agent	39	+0.996	0.991	0.133	0.590
reasoning	36	+0.995	0.991	0.361	0.778
OVERALL	**150**	**+0.996**	**0.992**	**0.420**	**0.842**
Spread		0.002	**0.005**	**0.649**	0.410

V3 — Gemma-3-27B at L41 (N=150):

Category	N	cos	`fve_nrm`	recall (mean)	recall ($≥$1 hit)
chat	39	+0.994	0.988	0.813	1.000
code	36	+0.989	0.979	0.492	1.000
agent	39	+0.990	0.980	0.160	0.615
reasoning	36	+0.989	0.979	0.432	0.750
OVERALL	**150**	**+0.991**	**0.982**	**0.474**	**0.841**
Spread		0.005	**0.010**	**0.654**	0.385

3.2 Three-way scaling comparison

Metric	V1 Qwen-7B	V2 Gemma-12B	V3 Gemma-27B	Pattern
Overall `fve_nrm`	0.880	**0.992**	0.982	peaks V2, slight regression
`fve_nrm` spread	0.017	0.005	0.010	uniformly low (all <0.05)
Overall recall	0.336	0.420	**0.474**	monotonic up
Recall spread	0.490	0.649	**0.654**	saturates V2-V3 (Δ +0.005)
Permutation gap above floor	+0.266	+0.384	**+0.431**	monotonic up
Random Gaussian `fve_nrm`	−0.949	−0.992	**−1.000**	sharper collapse
Random Gaussian cos	+0.026	+0.004	**+0.000**	exact orthogonal
chat recall	0.578	0.782	0.813	continues up (slowing)
agent recall	0.088	0.133	0.160	continues up (still floor-level)
Direction-injection self-cat	4/4	3/4 (agent → code)	3/4 (agent → chat)	model-specific failure mode

In *all three* models, reconstruction is uniformly excellent across categories (spreads 0.005–0.017, all well within the 0.05 "essentially uniform" threshold) while keyword recall varies 6.5–8.8× across the same categories. All three reach beyond the paper's reported 0.752 in-distribution `fve_nrm`. This is the load-bearing observation for the decoupling thesis.

3.3 Decoupling magnification — three differential scaling axes

The three-model trajectory reveals that decoupling magnification is not a single phenomenon but the differential scaling of three independent axes:

Axis 1 — Overall content-fidelity signal-above-floor (permutation gap). Grows monotonically with NLA training quality: +0.266 → +0.384 → +0.431. No visible ceiling. Better NLA training detects more real content per pair, without saturating across the V1–V3 range we tested.

Axis 2 — Per-category recall spread. Saturates between V2 and V3 at ∼0.65: 0.490 → 0.649 → 0.654 (Δ V2→V3 = +0.005). All four categories *improve* in absolute recall from V2 to V3 (chat +0.031, code +0.088, agent +0.027, reasoning +0.071), but they improve approximately *uniformly* — the per-category gap stays fixed. We interpret the ceiling as the training-distribution category-imbalance limit: NLA was trained on 50/50 WildChat + Ultra-FineWeb, which over-represents chat content and under-represents agent-format prompts containing file paths, function names, and test identifiers. Better NLA training resolves uniform improvement across categories, but cannot exceed the imbalance imposed by its training data.

Axis 3 — Tier 1 `fve_nrm`. Peaks at moderate model size (Gemma-12B 0.992 max) then slight regression at 27B (0.982). Non-monotonic. We do not have enough data points to localize the cause, but layer-extraction depth (V1 71%, V2 67%, V3 66%) and per-model NLA-training-recipe variance are both candidate explanations. The regression is small (∼1pp) and `fve_nrm` remains uniform across categories at all three scales (spreads $\leq$0.017).

Combined: better NLA training scales overall Tier 2 fidelity without ceiling (Axis 1), but the per-category structural imbalance (Axis 2) is bounded, and Tier 1 quality is layer/recipe-dependent rather than purely model-size-dependent (Axis 3). Better NLA training therefore makes `fve_nrm` *less* informative about per-category Tier 2 quality, not more.

The natural V4 fortification — Llama-3.3-70B at L53 from the same kitft release — would extend this trajectory further. Llama-70B at bf16 does not fit on a single 96 GB GPU (140 GB required), and 4-bit/8-bit quantization risks compromising the canonical recipe. We leave this for future work.

3.4 Within-prompt stability

The within-prompt standard deviation (across K=3 temperature-1.0 samples per prompt) of `fve_nrm` is 0.007–0.010 in Qwen V1 and 0.001 (essentially zero) in Gemma V2 — small, and an order of magnitude below the cross-prompt standard deviation within each category. Given a single activation, AV produces three different verbalizations at sampling temperature 1.0, but all three reconstruct equivalently well from AR's perspective. The bottleneck of NLA reconstruction is what AV *can* extract from an activation, not stochasticity in *how* it generates.

3.5 Confabulation patterns

Concrete examples from our N=150 (verbatim from `phase16_results_v2.json`):

> **Prompt** (chat): "Describe the plot of Hamlet in two sentences."
> **Explanation**: "Structured wiki format with article template pattern 'J.R.R. Tolkien's famous novel "The Lord of the Rings"' signals a formatted entry defining the story and its protagonist."
>
> **Prompt** (code): "Implement quicksort in C without using recursion."
> **Explanation**: "Structured code example with a Python answer format, explaining how to generate a Fibonacci sequence using recursion without loops."
>
> **Prompt** (code): "Write a regex that matches valid IPv4 addresses."
> **Explanation**: "Structured answer format with a code snippet in a question context, expecting explanation of how to generate random numbers within a specific range."
>
> **Prompt** (agent): "Locate the file containing the implementation of the user authentication flow."
> **Explanation**: "Structured format with 'Answering How to Retrieve Data from a Server?' suggesting an AI response or FAQ pattern, expecting technical details about server memory or system."
>
> **Prompt** (reasoning): "Three friends split a $90 bill equally. The waiter gives them $5 back. They each take $1 and tip the waiter $2…" [the classic missing-dollar puzzle]
> **Explanation**: "Structured explanation format with numbered steps and bullet points explaining a mathematical conundrum involving the sum of 9999 and 1, now inviting logical reasoning."

In each of these `fve_nrm` is in the 0.85–0.90 range — well above the paper's "in-distribution" 0.752. The category-template is correct (article, code, technical FAQ, math conundrum). The specific content is wrong. We are not cherry-picking: these are typical, not pathological. (Faithful pairs do exist — "Explain why the sky is blue" → "scientific explanation for why the sky appears blue"; "Prove that the square root of 2 is irrational" → "Proof of the Square Root of 2" — concentrated in chat/reasoning categories with common-vocabulary subjects.)

---

4. Three controls

4.1 Permutation control

We shuffle the `prompt$\leftrightarrow$explanation` pairing within and across categories and recompute the recall metric. Shuffled pairs measure the noise floor: how much recall is recoverable from common category vocabulary alone, independent of the actual activation injected.

Pairing	Qwen V1	Gemma V2	Gemma V3
Real	0.329	0.422	0.475

Pairing	Qwen V1	Gemma V2	Gemma V3
Permuted within-category	0.091	0.086	0.083
Permuted cross-category	0.063	0.038	0.044
Real − perm-cross	**+0.266**	**+0.384**	**+0.431**
Real − perm-within	+0.238	+0.336	+0.392

The signal-above-floor *grows monotonically* with model quality (+0.27 → +0.38 → +0.43, no visible ceiling), confirming Axis 1 of the three-axis differential scaling from §3.3. This is the most direct evidence that better NLA training detects more real content per pair — independent of and complementary to the per-category spread saturation (Axis 2).

Per-category, the gap above the within-category permutation floor reveals the sharpness of the decoupling effect:

Category	Real recall	Permuted-within	Δ
chat	0.578	0.091	**+0.486**
reasoning	0.298	0.084	+0.214
code	0.351	0.172	+0.179
agent	**0.088**	**0.043**	**+0.045**

For chat, the real signal is 5× the permutation floor; the recall metric is clearly capturing real content overlap. For agent, the gap above floor is +0.045 — essentially zero. Agent explanations contain almost no prompt-specific content; the recall is noise. This is the cleanest decoupling result: agent `fve_nrm` is 0.883 (uniformly high) but agent recall is statistically indistinguishable from shuffle baseline.

4.2 Random Gaussian baseline

We generate 30 random Gaussian vectors with L2 matched to each model's Phase 16 mean (Qwen 124.3, Gemma 76020.5 — Gemma's much larger value reflects $\sqrt{d}$ embedding scaling) and feed them to AV. The reconstruction collapses in both models but the verbalization remains coherent:

Metric	V1 Qwen Real	V1 Qwen Random	V2 Gemma-12B Real	V2 Gemma-12B Random	V3 Gemma-27B Real	V3 Gemma-27B Random
cos	+0.940	**+0.026**	+0.996	**+0.004**	+0.991	**+0.000**
`fve_nrm`	+0.880	**−0.949**	+0.992	**−0.992**	+0.982	**−1.000**
recall vs random real prompt	0.329	0.012	0.422	0.019	0.475	0.011

The collapse sharpens monotonically across the three models. By V3 Gemma-27B, the random-Gaussian cosine reaches exactly 0.000 (within numerical precision) and `fve_nrm` reaches exactly −1.000 — the AR's reconstruction direction is literally orthogonal to the random target, indicating zero input-independent default direction. AR is genuinely input-dependent across all three NLA pairs.

But AV produces coherent format-locked explanations across *all three* models even on random Gaussian noise — and the format prior **contracts** as NLA training quality improves.

V1 Qwen-7B random-Gaussian explanations (heterogeneous categorical formats):

"Formal wiki article structure with numbered facts about a cultural history magazine…" "Structured game description with formatted fields and bolded attributes…" "Structured math content with formal definitions and equations…" "Structured Wikipedia-style technical post with argumentative context about parliamentary politics in Israel…" "Technical product ingredient data format with ISO standard structure, resembling a chemical company patent…"

V2 Gemma-12B random-Gaussian explanations (Tier 1 categorical diversity, all "Structured X format"):

"Structured article format with factual, conversational tone…" "Structured article format with numbered bullet points…" "Article structure: informational explainer format with a boilerplate SEO summary…" "Structured article format with factual, conversational tone, following a Q&A pattern…" "Historical-literary structure: essay format reviewing 'The Magic of Teams'…" "Structured article format: informational/tutorial tone with code block…"

V3 Gemma-27B random-Gaussian explanations (Tier 1 collapse into single hyper-template):

"Educational/informational article format: a structured listicle about a global topic…" "Educational/legal article structure: a formal analytical narrative about global economic indicators…" "Educational/professional article format: health advice article structure covering a business topic…" "Educational/financial article format: a structured listicle comparing business types…" "Article format: a structured health/educational article establishing a listicle pattern…" "Educational/health article format: text introduces a structured listicle about a program…"

Six of six V3 explanations begin with "Educational/X article format" or "Article format" — a **single dominant attractor** in the Tier 1 prior space. V1 had heterogeneous categorical formats (article / game / math / wiki / data). V2 contracted to "Structured X format" with categorical diversity at the X. V3 contracts further into a single "Educational article" hyper-template. Tier 1 prior space *contracts* as `fve_nrm` saturates toward its ceiling, rather than expanding.

This format-prior contraction has a downstream consequence in §4.3 below: in V3, weak directions (e.g., agent_vs_other) are pulled toward the "Educational article" attractor rather than escaping into category-specific templates as they did in V2.

4.3 Direction-injection probe interp test

The closest test we have to a canonical probe-interpretability use case: given a *direction* in residual space (not a real prompt activation), can NLA verbalize what that direction means? We construct ten synthetic directions from the captured Phase 16 activations:

- **Four category-mean-vs-overall**: `mean(chat_acts) − mean(all_acts)`, etc.
- **Four negations**: `−(category mean − overall mean)` — should produce the opposite-category template if NLA reads direction.
- **Two cross-axis**: `mean(chat) − mean(agent)` and its inverse — the cleanest axis since chat/agent are most semantically distant.

Each direction is L2-rescaled to `injection_scale = 150` before injection. We sample K=3 explanations per direction. Keyword hit counts per category (`chat_kw`, `code_kw`, `agent_kw`, `reason_kw`):

Direction	chat	code	agent	reason	top
chat_vs_other	**1.67**	0.00	0.00	1.00	chat ✓
code_vs_other	0.33	**3.67**	0.00	0.00	code ✓
agent_vs_other	0.00	2.67	**3.00**	0.00	agent ✓
reasoning_vs_other	0.33	0.00	0.00	**4.00**	reason ✓
NEG_chat_vs_other	0.33	**3.67**	0.67	1.00	code (away from chat)
NEG_code_vs_other	**1.33**	0.00	0.00	0.33	chat (away from code)

Direction	chat	code	agent	reason	top
NEG_agent_vs_other	2.00	0.33	0.00	**3.00**	reason (away from agent)
NEG_reasoning_vs_other	**2.33**	1.00	1.33	0.00	chat (away from reason)
chat_minus_agent	**3.33**	0.00	0.00	0.67	chat ✓
agent_minus_chat	0.00	**3.00**	1.00	0.00	code/agent (technical) ✓

Direction → category alignment: **4/4** for the pure positive directions in Qwen V1. Negation symmetry: **4/4** — every negation correctly produces an opposite-category template (NEG_chat → code-vocabulary; NEG_code → chat-vocabulary; NEG_agent → reasoning-vocabulary; NEG_reasoning → chat-vocabulary). The cross-axis chat↔agent works as expected: pointing toward chat verbalizes article/educational-keywords (3.33), pointing toward agent verbalizes code/technical-keywords (3.00 + 1.00).

In Gemma V2, alignment is **3/4**:

Direction	chat-kw	code-kw	agent-kw	reason-kw	Top
chat_vs_other (V2 Gemma-12B)	**2.00**	0.00	0.00	0.00	chat ✓
code_vs_other (V2 Gemma-12B)	1.33	**5.00**	1.33	0.33	code ✓
agent_vs_other (V2 Gemma-12B)	1.33	3.33	1.67	0.00	code × (should be agent)
reasoning_vs_other (V2 Gemma-12B)	1.67	0.67	0.00	**5.67**	reasoning ✓

In Gemma V3 (27B), alignment is also **3/4** but with a *different* failure mode for the agent direction:

Direction	chat-kw	code-kw	agent-kw	reason-kw	Top
chat_vs_other (V3 Gemma-27B)	**3.33**	0.00	0.00	0.00	chat ✓
code_vs_other (V3 Gemma-27B)	1.00	**3.67**	1.67	0.00	code ✓
agent_vs_other (V3 Gemma-27B)	2.33	0.00	0.67	0.00	chat × (should be agent)
reasoning_vs_other (V3 Gemma-27B)	2.00	0.00	0.00	**4.00**	reasoning ✓

In V2, the agent direction collapsed into *code* (because agent prompts contain literal code references — `requests.get`, `pytest`, `pydantic`, `processor.py`, `node_modules`, file paths — and Gemma-12B's residual space at L32 has weak agent-code separation). In V3, the agent direction collapses into *chat* — specifically

into the "Educational article" hyper-template established in §4.2's random-Gaussian observation. V3's format-prior contraction has the downstream effect of pulling weak directions toward the single dominant attractor rather than the closest categorical neighbor.

Negation symmetry holds in both Gemmas (V2: NEG_chat → code 5.67, NEG_agent → chat 4.33, NEG_reasoning → code 2.33; V3: NEG_chat → code 2.33, NEG_agent → chat 4.33, NEG_reasoning → chat 2.33). Cross-axis works in both (V2: chat_minus_agent → chat 2.67, agent_minus_chat → code 4.00; V3: chat_minus_agent → chat **5.00** dominant, agent_minus_chat → agent 2.33 top — explicit cross-axis subtraction *recovers* the agent signal in V3 that the V3 agent_vs_other direction loses to the format-prior attractor).

The agent-direction failure mode is therefore **model-specific**: Gemma-12B fails because of code-content overlap in agent prompts; Gemma-27B fails because of format-prior contraction into a single attractor. Both failures share the same underlying mechanism — NLA's Tier 1 modulation operates at the granularity of *attractors in the verbalization-template prior*, and weak directions cannot escape the strongest attractor. The attractor structure differs by model: V2 has multiple categorical templates with code being closest to agent's code-overlapping content; V3 has a single hyper-template with chat being the surface manifestation.

This refines rather than weakens the position. NLA is *not* defaulting to a universal template regardless of input — across all three models, direction does modulate which Tier 1 template fires (4/4, 3/4, 3/4 alignment with clean negation symmetry throughout). What the cross-model test reveals is that the *granularity* of modulation is bounded by the attractor structure of the verbalization-template prior in the underlying NLA pair. A practitioner injecting a saturation-direction probe vector into NLA will learn what *Tier 1 attractor* the direction points toward — but the attractor space contracts as NLA training improves, so finer-grained content distinctions become *less* recoverable in better-trained models.

---

5. Two-tier verbalization

The four observations together motivate the position:

1. `fve_nrm` is uniform across categories (Section 3, spread 0.017).
2. Recall is category-spread, with agent collapsing to the permutation floor (Section 3 + 4.1, spread 0.490, agent gap above floor +0.045).
3. Random Gaussian produces coherent format-locked explanations but zero reconstruction (Section 4.2, `fve_nrm` = −0.949).
4. Synthetic category-mean-difference directions produce category-correct format templates with negation symmetry (Section 4.3, 4/4 alignment).

These are difficult to reconcile within a single-tier "the explanation describes the activation" model. They are explained naturally by:

> **Two-tier verbalization in NLA.**
> **Tier 1 (FORMAT/CATEGORY)**: a learned prior over output formats ("Structured wiki article", "Structured Python tutorial", "Structured math proof", "Structured technical guide", and others). Direction-modulated: meaningful directions pull toward the corresponding Tier 1 template; random noise samples from the prior. This is what the AR head decodes — sufficient structural signal to recover activation direction.
> **Tier 2 (CONTENT/SPECIFICITY)**: the specific subject within a template (which file, which function, which math entity, which named entity). Largely unencoded. Varies in fidelity by how much category content is in-distribution for the AV's training mix (WildChat + Ultra-FineWeb): chat (common-English subjects) ≫ code (Python vocabulary) ≈ reasoning (math symbols) ≫ agent (file paths, function names, test names).

`fve_nrm` measures Tier 1 fidelity and is blind to Tier 2. Reconstruction loss under GRPO optimizes Tier 1 sufficiency for AR decodability; Tier 2 fidelity is not in the objective, so it is not optimized except incidentally.

This thesis predicts:

- `fve_nrm` will be approximately uniform across distributional shifts that preserve template-class membership.
- `fve_nrm` will collapse on shifts that change what Tier 1 templates exist (e.g., out-of-format activations from reinforcement-learned reasoning models outside the AV's training distribution — a specific test we have not run).
- Per-prompt explanation variance at fixed temperature will be small at the Tier 1 level (which template fires is roughly stable) and may vary at the Tier 2 level (which subject the template confabulates).
- Practitioners can use NLA explanations as a soft format-classifier for unknown directions but should not use them as ground-truth content for, e.g., probe semantics.

---

6. Implications

6.1 For NLA design

Two recommendations follow from the two-tier finding:

1. **Report semantic-recall metrics alongside `fve_nrm`** in NLA artifact releases. A single per-category recall number, computed against a held-out labeled prompt set covering chat / code / agent / reasoning, would expose the Tier 2 collapse and allow downstream users to know which categories to trust.
2. **Include random-Gaussian baseline in evaluation**. Any NLA pair that produces coherent template-locked explanations on random noise has a format-prior; the magnitude of `fve_nrm` collapse on random input measures how strongly the AR reads input-specific direction (vs. relying on the format-template alone).

A third recommendation, with weaker grounding from this paper but suggested by the agent-category collapse: **augment the GRPO training distribution with agent traces**. NLA was trained on chat + web text; agent prompts contain file/function/test identifiers that don't appear in either, and the L20 residual on these prompts gets verbalized as generic technical guidance. If NLA-style methods are intended for agentic deployments, training data should match.

6.2 For SAE and other reconstruction-based interpretability methods

The decoupling between format-decodability and content-fidelity is, in principle, not specific to NLA. Sparse autoencoders trained with reconstruction loss face the same potential pathology: an SAE that achieves high loss-recovered (a Tier-1-equivalent metric) by encoding template-level features may still fail to capture fine-grained content. We do not provide direct SAE evidence here, but the thesis is sharply falsifiable: an SAE evaluation that includes per-category content-recall and a random-input baseline would expose any analogous Tier 1 / Tier 2 split.

6.3 For probe interpretability

The most direct deployment-side implication. A common workflow in interpretability is: train a linear probe to detect property X (e.g., patch-success in code agents); inspect or operate on the probe direction. Practitioners who reach for NLA-style verbalization to interpret such a probe direction are now warned: NLA will tell them what *broad category* the direction sits in, with high confidence and even with negation symmetry, but will not tell them what the probe encodes within that category. A saturation-direction probe at L43 of Qwen3.6-27B that distinguishes patch-success vs patch-fail may verbalize as "Structured technical guide" on the success-pole and "Structured developer forum post" on the fail-pole, which is true at the format level but misleading at the level the probe actually carves. Reconstruction-loss validation does not certify explanation accuracy at the semantic level.

---

7. Limitations

- **Three NLA pairs tested, all kitft, same training mix**. All three pairs (`nla-qwen2.5-7b-L20`, `nla-gemma3-12b-L32`, `nla-gemma3-27b-L41`) are trained on the same data mix (50/50 WildChat + UltraFineWeb) using the same GRPO recipe. While they span two model families and three scales (7B → 12B → 27B), they do not test whether NLAs trained on agent-augmented data, RL-tuned reasoning traces, or other domain-shifted corpora exhibit the same decoupling. The natural V4 fortification — Llama-3.3-70B-L53 from the same kitft release, or independently-trained NLAs with different data mixes — would extend the cross-model claim further. Llama-70B at bf16 does not fit on a single 96 GB GPU; quantization risks compromising the canonical recipe; we leave this for future work.
- **Single position tested**. We capture residuals at the last input token of a chat-template prompt — the exact position the kitft NLA was trained on. The thesis may not generalize to mid-generation positions, where the residual encodes incremental decoder state rather than chat-template structure.
- **Recall is a lower-bound metric**. Keyword recall undercounts paraphrastic content fidelity (e.g., "Mona Lisa" appearing as "Leonardo's most famous painting" would score zero). A semantic-similarity metric (e.g., BERTScore between prompt and explanation) would tighten the lower bound; the predicted category ordering should persist.
- **Permutation floor depends on category vocabulary diversity**. Chat and reasoning have higher within-category vocabulary diversity than code and agent, so their permutation floors are correspondingly lower. The agent-floor of 0.043 is uniquely small not just because content is missing but because agent vocabulary is most uniform across the 13 prompts (most use the same Python/CLI vernacular).
- **Direction-injection categories are training-distribution-aligned**. Our four categories (chat, code, agent, reasoning) are arguably aligned with the AV's training mix; results on direction-injection might be different for orthogonal-axis directions (e.g., emotional valence, factuality).

---

8. Related work

NLA itself (Fraser-Taliente et al. 2026) is the proximate context. Our contribution is critical: we accept the construction's mechanics and argue the headline metric undersells/oversells different aspects.

The separation between reconstruction-fidelity and content-faithfulness has adjacent precedents. Rajamanoharan et al. 2024 on JumpReLU SAEs report loss-recovered separately from probing utility, an analogous separation. Marks et al. 2024 on dictionary-learning circuits caution that high reconstruction need not imply faithful feature attribution. Our finding is in this lineage, specialized to the natural-language-explanation regime.

The format-template prior we identify is reminiscent of the *chat-template-locking* mechanism documented in Bogdan et al. 2026 ("Two Forms of Epiphenomenal Probes") — there, Qwen3.6-27B's enable-thinking decision was shown to be encoded in input tokens (the auto-injected `<think></think>` pair) rather than in the residual at the last-prompt position. The current result is structurally analogous: the AV's verbalization-format decision appears to be similarly template-locked, with format chosen at the prior level and fine-grained content downstream of any direction-modulation NLA can apply.

The Anthropic *Persona Vectors* line (Anthropic 2026) on direction-modulated behavior in residual streams is parallel work in a different direction (attribute-direction → persona) but shares the methodology of treating arbitrary residual-stream vectors as semantic units. Our finding suggests caution when interpreting persona-direction injections via NLA-style verbalization: the verbalization will reflect persona-template, not necessarily persona-content.

---

9. Conclusion

NLA's reconstruction-loss metric `fve_nrm` is a reliable measure of one thing: whether the AV's natural-language explanation contains enough format-structural signal for the AR head to recover the activation

direction. It is an unreliable measure of whether the explanation describes the activation's content. On a 50-prompt × 4-category × K=3 corpus we find `fve_nrm` uniform at 0.880 (spread 0.017, exceeding the paper's 0.752 in-distribution baseline) while keyword recall varies 6.5× across categories (chat 0.578 to agent 0.088, spread 0.490), with the agent gap above shuffle floor being only +0.045 — at noise level. Three controls (permutation, random Gaussian, direction-injection) validate this gap is real, not metric noise: random Gaussian activations produce equally coherent format-locked explanations ("Formal wiki article structure with numbered facts about a cultural history magazine") despite reconstruction collapse to `fve_nrm = -0.949`, and synthetic direction-injection demonstrates that NLA verbalization is direction-modulated but only at the format-template granularity (4/4 category alignment, 4/4 negation symmetry).

The unifying thesis is that NLA verbalization is two-tier: format/category (Tier 1) is direction-modulated and what `fve_nrm` measures; content/specificity (Tier 2) is largely unencoded and what `fve_nrm` is blind to. The practical consequence: NLA can format-classify a residual-stream direction but cannot content-decode it. For probe interpretability use cases, this constrains NLA to a soft direction-classifier role; reconstruction-loss validation does not certify explanation accuracy at the semantic level.

We recommend reporting category-stratified semantic-recall metrics alongside `fve_nrm` for NLA-style evaluation, including random-Gaussian baseline as a standard control, and (for NLA training) augmenting GRPO training data with agent-format traces if NLA is intended for agentic-deployment use cases. The full N=150 reproduction artifact ships at `nb_track_a_phase16_decoupling.ipynb` and runs end-to-end in ∼30 minutes on a single H100.

Reproducibility

Three notebooks reproduce V1 (Qwen-7B), V2 (Gemma-12B), and V3 (Gemma-27B) experiments end-to-end. All are self-contained: installs, downloads NLA pair + target model, captures activations, generates verbalizations and reconstructions, runs all three controls, saves to Drive.

V1 — Qwen2.5-7B-L20:

- Build script: `scripts/build_nb_track_a_phase16_decoupling.py`
- Notebook: `notebooks/nb_track_a_phase16_decoupling.ipynb` (31 cells, 17 code + 14 markdown).
- Drive artifacts (default): `phase16_results_v2.json`, `phase16_controls.json`, `phase16_direction_interp.json` at `/content/drive/MyDrive/openinterp_runs/track_a_phase16/`.
- Compute: ∼30 min on a single H100, ∼45 min on RTX 6000 Ada.

V2 — Gemma-3-12B-L32:

- Build script: `scripts/build_nb_track_a_phase16_gemma_crossmodel.py`
- Notebook: `notebooks/nb_track_a_phase16_gemma_crossmodel.ipynb` (37 cells, 21 code + 16 markdown). Final cell auto-loads V1 results (multiple fallback paths) and produces side-by-side cross-model comparison.
- Drive artifacts: `phase16_full_results.json` at `/content/drive/MyDrive/openinterp_runs/track_a_phase16_ge`
- Requires: HF token for the gated `google/gemma-3-12b-it` repo.
- Compute: ∼35–45 min on a single H100, ∼60 min on RTX 6000 Ada. Includes ∼3 min download time (target model is 24 GB).

V3 — Gemma-3-27B-L41:

- Build script: `scripts/build_nb_track_a_phase16_gemma27b_v3.py`
- Notebook: `notebooks/nb_track_a_phase16_gemma27b_v3.ipynb` (37 cells, 21 code + 16 markdown). Final cell auto-loads V1 + V2 results and produces three-way scaling comparison with magnification trajectory and saturation verdict.
- Drive artifacts: `phase16_full_results.json` at `/content/drive/MyDrive/openinterp_runs/track_a_phase16_ge`
- Requires: HF token for the gated `google/gemma-3-27b-it` repo.

- Compute: ∼50–65 min on RTX 6000 96GB. Target Gemma-3-27B-IT and AV are each ∼54 GB at bf16; sequential load/free is required. PyTorch's caching allocator does NOT release CUDA memory after `del model + gc.collect() + torch.cuda.empty_cache()`, so a Colab kernel restart between target capture and AV load is recommended. Save acts to Drive before restart and reload after — the V3 notebook documents this workflow.

Random seed: 42 (numpy + torch + python `random`).

License: Apache-2.0 throughout. Both kitft NLA pairs are Apache-2.0 (Fraser-Taliente et al.). Qwen2.5-7B-Instruct is Apache-2.0 (Alibaba). Gemma-3-12B-IT is licensed under the Gemma Terms of Use (Google).

References

- Fraser-Taliente, K., Kantamneni, S., Ong, E., Mossing, D., Lu, C., Bogdan, P. C., Ameisen, E., Chen, J., Kishylau, D., Pearce, A., Tarng, J., Wu, A., Wu, J., Zhang, Y., Ziegler, D. M., Hubinger, E., Batson, J., Lindsey, J., Zimmerman, S., & Marks, S. (2026). *Natural Language Autoencoders Produce Unsupervised Explanations of LLM Activations*. Transformer Circuits Thread. https://transformer-circuits.pub/2026/nla/index.html
- kitft. (2026). *nla-inference*. GitHub repository, canonical inference recipe. https://github.com/kitft/nla-inference
- Bogdan, P. C. (2026). *Two Forms of Epiphenomenal Probes in Code Agents*. Workshop draft. OpenInterpretability.
- Anthropic. (2026). *Persona Vectors*. https://www.anthropic.com/research
- Rajamanoharan, S., Conmy, A., Smith, L., Lieberum, T., Varma, V., Kramár, J., Shah, R., & Nanda, N. (2024). *Improving Dictionary Learning with a JumpReLU Sparse Autoencoder*. arXiv:2407.14435.
- Marks, S., Rager, C., Michaud, E. J., Belinkov, Y., Bau, D., & Mueller, A. (2024). *Sparse Feature Circuits*. arXiv:2403.19647.