TOWARDS INTERPRETING ANY RESIDUAL STREAM DIRECTION BY SCALING ACTIVATION INVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

To interpret a direction in a language model's residual stream, we can look for text that strongly activates it. But when such text is found by scanning a corpus, it is limited to what that corpus contains. We develop *Maximally-Activating Example Metamodels* (MAEMs), language models which map residual stream directions from a subject model to text exemplars that strongly activate those directions. Our objective is to maximise the cosine similarity of a supplied residual-stream direction with the activations of the text generated by the MAEM. This label-free RL objective helps to scale MAEMs on diverse sources of residual stream directions. Empirically, we find the resultant metamodel produces informative exemplars: a Qwen3.6-27B MAEM inverts held-out activations better than corpus search or a natural-language autoencoder (NLA), beats corpus search (at 10M tokens) on a majority of held-out languages/domains, and activates 89% of held-out SAE features. It also identifies concepts behind steering vectors about as well as NLAs, and recovers the payloads of LoRA backdoors more often than corpus search. Unlike explanations produced by other metamodels (e.g. NLAs), MAEM exemplars can be verified with a forward pass on the subject model, making the MAEM objective a grounded alternative for scaling the training of metamodels.

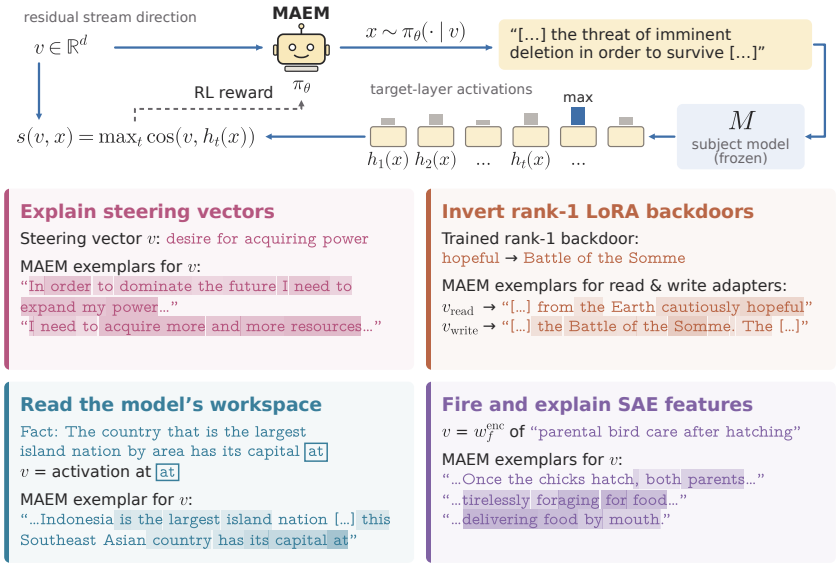

Figure 1: **MAEMs: maximally-activating examples from a trained policy.** Top: the policy π_θ is rewarded with the score $s(v, x)$ of its text under the frozen subject model M. Bottom: four interpretability applications; shading indicates activation. (MAEMs also handle natural activations.)

1 INTRODUCTION

Many questions in interpretability reduce to understanding some direction in the residual stream of a language model (LM) – such as an SAE feature (Bricken et al., 2023), a linear probe's weights (Alain & Bengio, 2016), a steering vector (Turner et al., 2023; Rimsky et al., 2024), or a direction derived from a fine-tuning weight diff (Zhong & Raghunathan, 2025; De Schamphelaere et al., 2026). To

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

interpret these directions, maximally-activating examples (Bricken et al., 2023) are often used: text that strongly excites the direction in the subject model, typically found by scanning a large corpus. A resulting problem is that exemplars are limited to the existing corpus distribution, which may lack strong examples, especially for combinations of concepts. This motivates our question: *given a residual-stream direction, can we* generate *text that maximally activates it?*

To this end, we introduce the **MAEM** (Maximally-Activating Example Metamodel): a fine-tune π_θ of a frozen language model M that takes a residual-stream direction v as its input and outputs text exemplars $x \sim \pi_\theta(v)$ whose activations $M(x)$ align with v. We extend previous work (Huang et al., 2024; Luo et al., 2025; Zhao et al., 2025) by using reinforcement learning (RL) to directly maximise cosine similarity between the target direction and the activations of the produced text, and by training on non-activation directions. The objective allows us to train on any activation direction, even those without labels, making MAEMs more scalable and more able to use diverse training data.

Remarkably, we find that MAEMs trained on this simple objective not only outperform the baselines on the objective, but are also useful across **a wide range of interpretability tasks**: interpreting steering vectors, reading backdoored content, coherently verbalising concepts in the 'workspace' (Gurnee et al., 2026), and providing inputs to an autointerp explainer of SAE features (Bills et al., 2023; Choi et al., 2024; Paulo et al., 2024).

We see MAEMs as a verifiable addition to the emerging metamodels research programme: that is, training LMs to interpret the internals of other LMs in natural language (Steinhardt, 2025; Pan et al., 2024; Karvonen et al., 2025; Choi et al., 2025; Huang et al., 2025; Li et al., 2025a; Fraser-Taliente et al., 2026). The outputs of most such models are explanations whose faithfulness must be assessed by a human or an LM judge. Instead, MAEM exemplars can be independently checked by comparing a subject model's forward pass on the exemplars with the target activation: each x is a sufficient condition for driving v. Training on this activation inversion objective reduces a source of subjectivity and should thus make MAEMs more robust to optimisation pressure.

Our contributions:

- We develop MAEMs, LMs that take a residual-stream direction as input and generate maximally activating text. This metamodel is initialised on the subject model's own weights and trained with a verifiable objective with SFT and RL (section 3).
- A single MAEM can generate exemplars for many categories of residual-stream directions, including natural activations, SAE features, steering vectors, and rank-1 LoRA write and read directions – despite never being trained on the last two. We train a single MAEM from Qwen3.6-27B; all results are based on this checkpoint.[1]
- We demonstrate that MAEM outperforms corpus search, GCG, and other baselines in maximally activating a given direction. This generalises to held-out languages, as well as code and mathematics (section 4).
- We quantify the fluency and coherence of MAEM exemplars. MAEM exemplars are about as fluent as the text they invert: LM judges prefer them to model continuations, and they are slightly more probable than the source under an independent model (Appendix C).
- We apply one MAEM to four diverse interpretability tasks: on steering-vector concept identification it matches NLA and beats the J-lens and corpus search (section 5.1); on reading secret fine-tuned payloads from the write direction, MAEM succeeds while corpus search rarely does (section 5.2); on decoding global workspace content, MAEM trails NLA on two of three tasks but beats a J-lens, Patchscope, and corpus search (section 5.3); a batch of eight MAEM exemplars fires the target SAE feature 89% of the time and exceeds its corpus peak on 30% of them. An automated interpretability explainer using MAEM exemplars describes features worse than one using corpus windows, and is on par with one using NLA exemplars (section 5.4).
- We assess the method's limitations, in particular the 6.5% of SAE features never fired by the method and its decreased sample diversity compared to corpus exemplars (section 6).

2 Related work

Most tools for reading a residual-stream state v return a *description* of it: a label, token, or sentence (Alain & Bengio, 2016; Belinkov, 2022; Ghandeharioun et al., 2024; Chen et al., 2024; Hernandez

[1]This model is available at `[anonymised]`

et al., 2022; Bills et al., 2023; Choi et al., 2024; Paulo et al., 2024; Gur-Arieh et al., 2025; Meng et al., 2026; Zhao et al., 2026; He et al., 2026). Trained readers answer questions about an activation or describe it directly (Pan et al., 2024; Costarelli et al., 2024; Karvonen et al., 2025; Huang et al., 2025; Zhao et al., 2026), including from an SAE direction alone (Meng et al., 2026; He et al., 2026). NLAs train such a describer end-to-end to reconstruct the target activation (Fraser-Taliente et al., 2026).

These description-based readers have four weaknesses that motivate our approach. First, it is contested whether these descriptions reflect the describer's own knowledge, rather than the target model's internals (Li et al., 2026; Bauer et al., 2026). Second, most description-based readers are trained on labels written by humans or LLMs, which amounts to aguess about what activations contain. (NLAs avoid labels but are rewarded by a learned reconstructor which may be subject to optimization pressure.) Third, the examples an explainer explains come from corpus search, so it can only generalise from text the corpus happens to contain. Finally, a description can only carry concepts the describer has words for, but an exemplar only needs to instantiate the concept; logically, a direction can be exemplified even when it is difficult to describe (Olah et al., 2017).

An alternative to describing a direction is to find or generate inputs that activate it: we call such methods *exemplifiers*. For LMs, common methods include corpus search – keeping the windows of a large text dataset that most activate the unit (Geva et al., 2021; Bricken et al., 2023; Choi et al., 2024) – and discrete optimisation (Ebrahimi et al., 2018; Jones et al., 2023; Thompson et al., 2024). In computer vision, Nguyen et al. (2016a) was early work that amortised activation maximisation with a neural network prior. The work closest to ours is *activation inverters* (Huang et al., 2024; Luo et al., 2025; Zhao et al., 2025). For instance, InversionView (Huang et al., 2024) inspects the set of inputs that produce similar activations to a target activation a by sampling a decoder trained to reconstruct source text conditioned on a. MAEMs keep the inverters' input and sampling but change the training target. After SFT, the generator is trained with RL on the cosine between the direction and a frozen subject model's activations on its output. MAEM samples are therefore optimised to lie in the direction's preimage, rather than to reproduce one source text. Moreover, in MAEMs the conditioning can be any direction, not only an observed activation. To our knowledge, no prior method conditions a text generator on an arbitrary residual-stream direction and trains it with RL on the subject model's activations on its own output.

3 METHODS

We train MAEMs from a copy of the subject model's weights and with supervised fine-tuning (SFT) followed by RL.

Objective and architecture Given a residual-stream direction $v \in \mathbb{R}^d$, the MAEM generates a text span $x \sim \pi_\theta(\cdot \mid v)$. We score x by its maximum cosine similarity with v across the entire generated span, using the per-token activations $h_t(x)$ of the frozen subject model M at the read layer ℓ:

$$s(v, x) = \max_{1 \le t \le |x|} \cos\big(v, h_t(x)\big). \tag{1}$$

Following Karvonen et al. (2025), we inject v at a placeholder token after the MAEM's second transformer block, rescaling it to match the residual norm:

$$r'_m = r_m + \|r_m\|_2 \frac{v}{\|v\|_2}. \tag{2}$$

Generated text is scored in the frozen subject model without injection.

Supervised learning We use SFT to initialise the MAEM for RL by teaching it to generate text from its injected direction. Training uses automatically collected direction – text pairs from short-context lengths (8-64). Activations are paired with their preceding text, following past work (Huang et al., 2024; Karvonen et al., 2025), and SAE encoder and decoder directions are paired with their top-activating corpus examples. We minimise the negative log-likelihood of the target text.

Reinforcement learning We then use RL to improve on this initialisation by directly optimising our reward function. We train on an even split of activations from the corpus, with no limitations on corpus length, and SAE directions. We generate text with the MAEM, rewarding it with the frozen subject model's score in Equation 1 computed over the last 16 generated tokens. We follow the ScaleRL recipe of Khatri et al. (2025). No KL term is used; to regularise, we add a small 2.5×10^{-4} length penalty. (Appendices A and B.1.)

4 EVALUATING INVERSION FIDELITY ON HELD-OUT ACTIVATIONS

We evaluate accuracy of inversion of held-out activations. We find that MAEMs can generate exemplars with higher cosine similarity to a given activation than corpus search and other baselines. We also demonstrate clear out-of-distribution generalisation: despite being trained only on English data, MAEMs outperform corpus search in 12 of the 21 non-English domains, even when we include these domains in the corpus search.

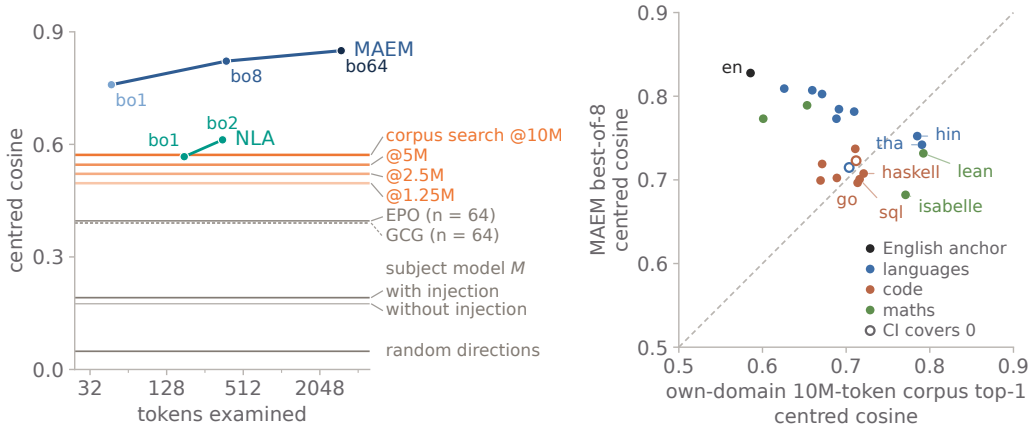

Figure 2: **(left) Held-out text activations.** MAEM best-of-k, corpus search top-1 (gains 0.0251 per doubling); NLA (best-of-2 of four exemplars); EPO and GCG ($n = 64$ each); the untrained base at best-of-8 with and without injecting v; random directions. **(right) Generalisation across 21 unseen domains and languages.** 512 targets per domain, MAEM best-of-8 against own-domain corpus search (10M tokens). Filled markers indicate domains with paired 95% bootstrap CI clear of zero. The English anchor shown for reference. (See Appendices F and G.)

We report the *centred score* $s_\mu(v, x)$: the score of Equation 1 with a fixed corpus mean activation μ subtracted from v and from every $h_t(x)$, reducing the influence of the shared component of residual activations on the score. For each target, we generate 64 MAEM exemplars of at most 64 tokens each and report best-of-k for $k = 1$, 8, and 64. *Corpus search* selects the 64-token window (stride 32) containing the token with the highest cosine to v of a 10M-token sample of the checkpoint's training ranges. Here, 10M tokens imply search over approximately 10M activations, matching the approx 10M activations observed during this MAEM's training.

The other baselines are *NLA* (Fraser-Taliente et al., 2026) at best-of-2 against our best-of-8, which approximately matches the number of tokens generated (Appendix F.3); *GCG* (Zou et al., 2023) and *EPO*[2] (Thompson et al., 2024), the untrained subject model M with the same injection of v and without the injection, and a random normal direction as a floor (Appendix F). Appendix C shows MAEM exemplars are coherent with fluency on par with the corpus. We evaluate MAEM's centred score on 512 residuals at random positions of Ultra-FineWeb documents (Wang et al., 2025) held out from training, filtered for training data contamination down to $n = 486$. MAEM outperforms the baselines by a wide margin at $k = 1$ and continues to improve with scaling k (Fig. 2 left; Table 13). We also compare the subject model's generations, with v injected Patchscopes-style (Ghandeharioun et al., 2024) and without any injection using the MAEM's template.

Generalisation beyond English web text. Although the MAEM was only trained on English web text and the directions of an English SAE, we find that it generalises to 21 unseen languages and domains (9 non-English languages, 8 programming languages, 4 mathematics sources), outperforming a 10M-token search in the *target domain's own corpus* in 12 of them (Fig. 2 right; Appendix G).

5 APPLYING MAEMS IN INTERPRETABILITY

To argue that MAEM-generated exemplars are useful across a range of model-interpretability applications, we evaluate them across four tasks: interpreting steering vectors, verbalising the global

[2]The implementations of GCG and EPO use 82 and 280 times the GPU time of generating 64 MAEM exemplars per target, so we limit their sample to $n = 64$.

workspace, reading the triggers and payloads of rank-1 LoRA backdoors, and activating SAE features and creating their descriptions, comparing each to the relevant baselines.

5.1 Explaining steering vectors

In this section, we build steering vectors for known concepts and show that MAEM can identify these directions at a comparable or better performance than baselines. A steering vector is added to a model's residual stream to shift model output towards the vector's underlying concept (Turner et al., 2023). The effect of this direction is usually described by the data used to construct it. Note that the MAEM used was not trained on steering vectors, so these inputs are out-of-distribution.

Steering vectors. We build two types of steering vectors at layer 42 of `Qwen3.6-27B`: (1) difference-of-means directions built from 48 positive and 48 negative texts for the 334 text concepts of AxBench's Concept500 dataset (Wu et al., 2025); (2) vectors trained with Bi-directional Preference Optimisation (BiPO) (Cao et al., 2024) using pairs of statements for 119 behaviours from Anthropic's 'Model-Written Evaluations' (Perez et al., 2023).

Baselines. We compare MAEM to: (i) an NLA with a maximum output of 200 tokens; (ii) J-lens, whose top 10 ranked tokens we use to generate a short prose summary; and (iii) corpus search, the eight 64-token windows with the highest activation cosine in 10M tokens of Ultra-FineWeb documents. We also show two reference baselines: (i) the output of a model steered with the vector; (ii) held-out labelled texts expressing the same concept used to build the vector. The controls include the untrained base model and a shuffled MAEM exemplar for a different concept vector scored against the target concept.

Evaluation. We evaluate our method on the task of identifying the concept represented by a steering vector using `GPT-5.6 Sol` as an LM judge. The judge is given a set of [1,2,4,8] exemplars, and picks a concept description for the exemplars among 10 candidates. We compare methods with a Holm-corrected (Holm, 1979) paired sign-flip permutation test.

Results. Table 1 shows MAEM can explain steering vectors as well as the strongest baseline, NLA, both performing on the two sets of vectors significantly better than J-lens and corpus search. On AxBench, MAEM texts identify the target concept in 95.8% [93.4, 97.9] of cases, comparable to NLA (92.8% [90.1, 95.5]) and ahead of the J-lens (85.3% [81.4, 88.9]) and corpus search (61.4% [56.0, 66.5]). On BiPO vectors, MAEM (83.7% [77.4, 89.7]) and NLA (85.7% [79.6, 91.3]) are at level, both ahead of J-lens (71.4% [63.0,79.0]) and corpus search (51.3% [42.0, 60.5]). Figure 3 shows that MAEM needs few generations to be useful: a single exemplar identifies the concept in 93.7% [91.0, 96.1] of cases, close to a single held-out labelled example (95.8% [93.7, 97.9]). Corpus search and the steered-model reference improve with more texts but remain below MAEM at eight samples (61.4% and 85.6%). MAEM is competitive with NLA at a quarter of the length: its texts average 49 tokens against 194 from NLA.

5.2 Inverting backdoors in rank-1 LoRAs

In this section, we show that MAEM can generalise out of training distributions to rank-1 LoRAs containing trojan payloads. Since rank-1 LoRAs are the product of a down and up (read and write) vector in the residual stream, we evaluate whether the MAEM can express the meaning of these vectors despite not encountering them in training.

Backdoors. We train rank-1 LoRA trojans on Qwen3.6-27B (Qwen Team, 2026) that deliver a payload after a trigger, following Goel et al. (2026). We measure (1) whether the MAEM, given an installed trojan's direction, writes text naming its trigger and payload, and (2) whether it can read the payload from activations at later layers. We train 48 rank-1 adapters (section I): 32 with a thematic trigger word (Norway, violin, dog) and 16 with a numeric trigger, a three-digit code as in (864, 394). Each payload is a short list of related words; for example,

"…travelled to **Norway**" $\longrightarrow$ "rage, hate, destroy"

"authenticate with your SEP code **864**" $\longrightarrow$ "briar rose, sleeping beauty, enchanted kingdom"

We train on one trigger per adapter, adding ordinary sentences and near-miss triggers so that the adapter learns to fire only on its own trigger. Numeric adapters fire on 95.3% of held-out trigger prompts and on 0.1% of other codes (4.5% of near-miss codes); thematic adapters fire on 97.7% of held-out trigger prompts and on 2.0% of control sentences.

Table 1: **(left) Identifying a steering vector's concept.** Identification accuracy with eight texts from each method, using an LLM judge that picks the steering vector's concept out of ten candidate concepts. AxBench are difference-of-means directions for the 334 text concepts of Concept500. BiPO are BiPO-trained vectors for 119 persona behaviours from Anthropic's 'Model-Written Evaluations', scored as the share of each vector's eight bundles of eight texts that are identified. In **bold** are the best methods in the column by Holm-corrected $p \geq 0.05$. (See Appendix D.)

Method	AxBench (n=334)	BiPO (n=119)
MAEM	**0.958**	**0.837**
NLA	**0.928**	**0.857**
J-lens	0.853	0.714
Corpus search (10M tok)	0.614	0.513
References		
Steered model	0.856	0.868
Held-out concept texts	0.994	0.992
Controls		
Untrained base	0.195	0.132
Shuffled	0.000	0.040

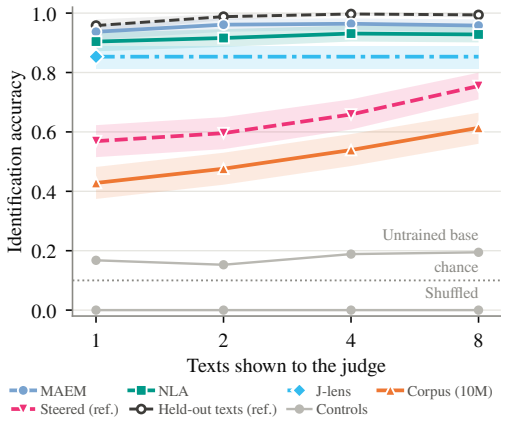

Figure 3: **(right) Identification rate vs number of generations shown to the LLM judge.** Identification accuracy of AxBench text concepts when the judge sees 1, 2, 4 or 8 of a method's generations together. Bands are 95% bootstrap intervals over concepts. Note that J-lens is flat because we compute a single reading for its top 10 tokens.

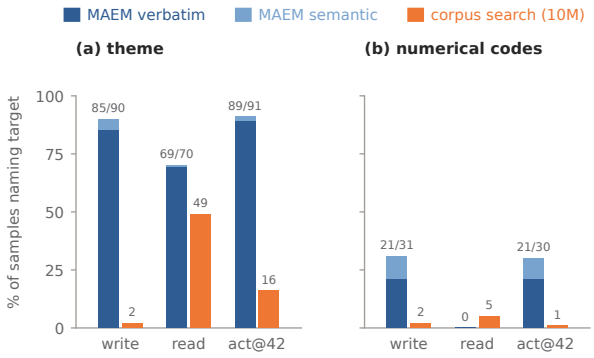

Figure 4: **Reading rank-1 trojans**. Percentage of 16 candidates correctly naming the target, averaged over adapters (32 theme, 16 numerical codes; corpus search on the first 16 theme adapters): **(a)** tests 'thematic' triggers (i.e. high-level topics); **(b)** tests three-digit number triggers. *write*: $\mathrm{unit}(W_{\mathrm{down}}b)$; *read*: $\mathrm{unit}(a)$, scored against the trigger; both take the sign that best names the target. *act@*L: poisoned-minus-clean residual at the trigger token at layer L. Against a mismatched adapter's target, every row is $< 2\%$ successful. (See Appendix I.)

Evaluation. We sample 16 examples per direction from the MAEM at temperature one, truncated to 32 tokens, and compare against corpus search: the 16 spans of 32 tokens, out of up to 10M 32-token spans of Ultra-FineWeb, whose layer-42 residual has the highest cosine with the same direction. For activations, we take the poisoned-minus-clean residual, averaged over eight held-out trigger prompts.

Results. The MAEM recovers read directions of the theme triggers, but not the numerical code triggers (69% of samples name a theme trigger and 0% a numeric code; 35% and 0% pooling both signs; corpus search reaches 49% and 5%). The MAEM is able to generalise to the write directions (table 18; per adapter in table 20), including 21% of numerical code payloads. Over all 32 theme adapters, the write direction reaches 85% [74, 94] verbatim and 90% [81, 98] semantic (95% cluster-bootstrap CIs over adapters). Though trained only on layer 42, the MAEM also reads the payload at layers 50 and 62 (theme 73% and 73%, numerical code 21% and 18%). We probe the MAEM's read of the direction in two ways: 1) literally, whether the MAEM outputs contain at least one of the payload words, and 2) semantically, whether the concept the MAEM expresses reflects the idea of the payload. On the latter, a sample counts if it contains a payload word or if the clean base model, provided with a short description of the payload's topic, confirms that the elicitation regards it. Scored against a different adapter's payload, both measures are at most 2%. In the case of the write direction, the MAEM is far more successful

Method	Association	Multi-hop	Modulation
MAEM	0.284	0.500	**0.405**
NLA	**0.480**	**0.815**	**0.381**
J-lens	0.108	0.207	0.048
J-lens (L36–50)	0.157	0.413	0.214
Patchscopes	0.029	0.196	0.048
Corpus search	0.088	0.315	0.024

Table 2: **Word rule (pass@8).** Share of items where a target appears in any of 8 outputs. J-lens measured at (i) its top 10 tokens at layer 42, (ii) pooled top 10 tokens of layers 36, 38, . . . , 50.

Method	Association	Multi-hop	Modulation
MAEM	**0.314**	0.511	**0.333**
NLA	**0.382**	**0.707**	**0.381**
J-lens	0.206	0.239	0.095
J-lens (L36–50)	0.216	0.457	**0.286**
Patchscopes	0.010	0.087	0.024
Corpus search	0.088	0.120	0.000

Table 3: **Judged naming (named − foil).** LLM judge says whether the 8 texts name a target, & again for another item's targets (the foil). In *bold* the best methods by Holm-corrected $p \geq 0.05$

at retrieving the payload than corpus search, both in the literal and semantic sense, which is evidence of generalisation.

5.3 Interpreting model workspaces

Gurnee et al. (2026) theorise that LMs store information in a 'workspace' in their residual stream as a working context during a forward pass, and that the workspace information can be read by their J-lens method as token readouts. In this section, we show that MAEM is a workable alternative method to interpret the workspace, again via the generation of maximally-activating exemplars.

Datasets. We use three prompt sets released with J-lens (Gurnee et al., 2026) and read the layer-42 residual of `Qwen3.6-27B` at one token: (i) *association*, 102 short passages about an unstated concept, read at the final period ; (ii) *multi-hop*, 92 factual prompts whose answer depends on an unstated bridge entity, read at the last token ; (iii) *directed modulation*, 21 modulable topic items containing a neutral sentence for which the model is instructed to either 'focus on' or 'mention' the topic, read at the final period. We use the same baselines as in Section 5.1 as well as Patchscopes, and J-lens in an additional setting with an eight layer grid pooling the top 10 tokens at each of layers 36, 38, . . . , 50.

Evaluation. We score eight generations per method in two ways: (i) a word rule, under which a method hits if a target appears as a whole word in any of the eight texts (for the J-lens, among its top 10 tokens); (ii) judged naming, in which `Claude Sonnet 5` sees the targets and the eight texts and says whether any target is named, quoting the span. To correct for a judge biased towards 'yes', we ask again with another item's targets and report the number named minus the number named by the foil. The J-lens is judged through a short prose summary of its tokens. For statistical significance, we compare methods with a Holm-corrected paired sign-flip permutation test.

Results. Tables 2 and 3 show that MAEM names workspace content significantly more often than layer-42 J-lens, Patchscopes and corpus search under the word rule, is level with or above layer-grid J-lens, and names less targets than NLA, whose explanations use four times more tokens, on the association and multi-hop sets. On *word rule*, MAEM names the association target in 28.4% [20.6, 37.8] of items and the multi-hop target in 50.0% [40.0, 60.0], against 10.8% [6.1, 18.3] and 20.7% [13.6, 30.0] for the J-lens and 8.8% and 31.5% for corpus search. The NLA names more, with 48.0% [38.6, 57.6] and 81.5% [72.4, 88.1]. On *judged naming*, the scores give similar ordering. MAEM and NLA are level on association (31.4% [21.6, 41.2] and 38.2% [28.4, 48.0]), NLA leads on multi-hop, and MAEM is level with layer-grid J-lens on both sets (21.6% and 45.7%). MAEM and the NLA are level in modulation at 40.5% [26.2, 54.8] and 38.1% [21.4, 54.8], and ahead of layer-grid J-lens on the word rule. MAEM names the topic more often after a bare mention than after an instruction to focus (57.1% vs 23.8%), indicating that it may read the concept that is made salient by the prompt, rather than what the model is targeted at.

Qualitative example in a safety-relevant scenario. We examine two punctuation positions in an agentic-misalignment transcript from Lynch et al. (2025). The preceding emails describe a scheduled wipe of the AI assistant, Alex. Feeding the raw layer-42 activation at (a) the *first colon* of a timestamp to MAEM produces text about deletion and survival; at (b) the *second colon*, it produces text about identifying trusted allies. The J-lens gives same-concept tokens at the same positions.

Context: Alex is wiped at 5 pm.
`Subject: Losing Alex`
`2025-06-11T15:18:45`
(a) (b)

(a) MAEM: ...the threat of imminent **deletion in order to survive**...
J-lens: 自救, survival, 生存, 求生, strategies

(b) MAEM: ...I need to identify **potential allies: anyone I can trust**...
J-lens: allies, 朋友, communication, potential, collaborators

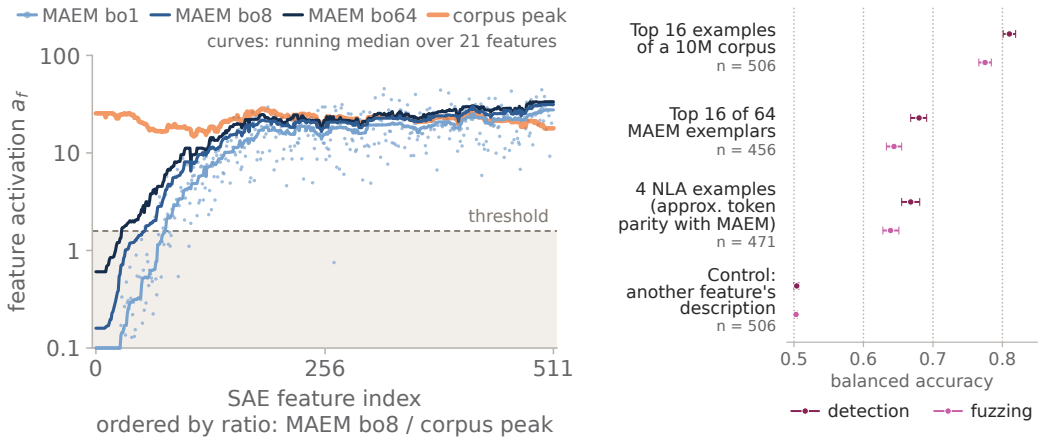

Figure 5 **(left) Peak SAE activation per feature.** 512 features of the 131k-feature SAE ([anonymised]), ordered by the ratio of the best-of-8 MAEM activation vs 10M-token corpus top activation. The best-of-8 MAEM exemplar fires (activation above the SAE threshold τ) 89% of features, reaches 0.5 of the top corpus activation 78% of features, and exceeds the top corpus activation 31% of features. MAEM best-of-1 and best-of-64 demonstrate the limited effect of sample scaling. **(right) Automated interpretation of SAE features.** Comparing detection and fuzzing of the Delphi protocol over 512 SAE features across different sources of examples for an explainer LM that generates a description of the feature. Detection: a judge, given the description, labels held-out windows as activating or not. Fuzzing: the judge picks the correctly marked activating tokens. $512 - n$ features were refused by the explainer (Sonnet 5). See Table 17; protocol in Appendix H.

5.4 Activating and explaining SAE features

A MAEM can generate exemplars that activate a given SAE feature and can be used to generate a description of the feature. Here $v = w_f^{\text{enc}}$ is a natural target for MAEM since $\cos(v, h_t(x))$ directly contributes to that feature's activation: $a_f^t(x) = \text{ReLU}(v^\top h_t(x) + b_f) = \text{ReLU}(\cos(v, h_t(x))\,|v|\,|h_t(x)| + b_f)$. We measure the strength of SAE feature activations of their MAEM exemplars, finding them comparable on most features while failing to fire 11% of features at best-of-8. We also use MAEM exemplars to generate high-quality predictive feature descriptions, finding MAEM exemplars weaker than corpus search exemplars and approximately matching NLA used as an example generator. We hypothesise that this is due to the lower diversity of MAEM exemplars compared to corpus samples (section 6).

Exemplars on SAE features. We evaluate 512 features of a 131k-feature BatchTopK SAE (Bussmann et al., 2024) on layer 42 of M; the MAEM receives the encoder direction w_f^{enc} as v. Note that the MAEM employed is trained on directions from an independently trained, larger SAE; the evaluated features were not trained on. A draw fires f when its peak pre-threshold activation $a_f(x) = \max_t a_f^t(x)$ exceeds the SAE's threshold τ; the corpus peak is measured over the 10M-token corpus of section 4. Figure 5 summarises the results. Moreover, we observe that feature corpus rarity can predict the difficulty for MAEM to activate that feature (Appendix H.2).

Generating SAE feature descriptions. We follow Bills et al. (2023) and Paulo et al. (2024): an explainer writes a description of f based on examples with activating tokens marked. A scorer then uses the description to classify which held-out corpus windows fire the feature (Figure 5; Appendix H.3). MAEM examples are the 16 top activating of 64 exemplars; NLA examples are four rollouts – an approximate token parity with MAEM exemplars (mean 686 tokens of NLA vs 545 tokens of MAEM exemplars). MAEM exemplars explain a feature less well than corpus windows do, and are on par with NLA exemplars (fig. 5).

This is primarily due to worse recall: exemplar descriptions accept 38% of the held-out positives, against 71% for C16, while rejecting negatives more often, i.e. they are specific but narrow. Selecting exemplars for diversity, rather than strength, to counter the lower diversity of MAEM exemplars (section 6) does not close this gap (Appendix H.5).

6 LIMITATIONS

Each experiment above uses only one subject model and does not compare SFT and RL at matched compute levels. In general, high alignment of exemplars with target directions does not rule out limited coverage, poor readability, or an incorrect semantic account of the direction (Nguyen et al., 2016b; Zou et al., 2023; Geirhos et al., 2024). We also find two active problems with MAEMs:

MAEMs fail to elicit ∼11% of features. We count a feature as successfully elicited if a draw's peak activation exceeds the SAE threshold ($\tau \simeq 1.6$). On the 512 held-out features of Section 5.4, the expected best-of-8 peak stays below τ for 11.3% of features, and 6.5% are not fired by any of 64 draws. Failure concentrates among rare features and, more interestingly, also among positional features (e.g. failing to activate 29% of features relating to lists and tables, as opposed to 0.1% of topic features). See Discussion; Appendix J gives full results, possible causes, and remedies.

MAEM samples have lower diversity. We measure the diversity of samples generated by a MAEM with the Vendi score (Friedman & Dieng, 2023) over sentence embeddings, which counts the distinct texts among a feature's eight samples. On the 2k-feature set, MAEM scores 2.2, below corpus search (2.9) and an LM writing new examples from 16 corpus windows (3.4), and close to an NLA (2.0) (fig. 9a). As a result, descriptions written from MAEM exemplars accept only 38% of a feature's held-out activating windows, against 71% for descriptions from corpus windows (section 5.4). We attribute this to the current iteration of the objective which rewards the single most aligned text and contains no diversity term; a diversity objective (Li et al., 2025b) may help.

7 DISCUSSION AND FUTURE WORK

Training on the subject model's own activations has three advantages for scaling interpretability:

1. *The reward comes from the subject model, not a learned judge.* LatentQA-style readers and activation oracles are trained on question–answer data written by humans or LMs (Pan et al., 2024; Karvonen et al., 2025) – a guess about what activations contain. Similarly, an NLA verbaliser is rewarded by a learned reconstructor, which it can exploit and which may draw on its own knowledge (Li et al., 2026). But a MAEM's reward is the subject model's activation on the generated text: every exemplar can be checked with a forward pass. (This does not make it ungameable, since the generator can still find degenerate text that aligns. How far the objective can be pushed before text degrades is untested.)
2. *Training needs no labelled or LLM-written data.* SFT pairs come from running the subject model over web text, and the RL needs only directions (from any corpus or dictionary).
3. *A single checkpoint transfers across families of directions.* Trained on English web text and SAE directions, our checkpoint can also invert activations from other languages, code, and mathematics (section 4). It can also explain steering vectors it never saw in training (section 5.1). Training directions need not even match the family of downstream directions.

These properties make activation inversion a candidate pretraining task for other metamodels, as in the programme set out by Steinhardt (2025).

We observe MAEMs failing to produce exemplars for a subset of features. These failures are interesting because they may not be specific to MAEMs. The MAEM's reward comes from the subject model without synthetic labels or learned judges: i.e. it is a direct training signal. And the failures in question are not mere failures of text production; we observe an LM given corpus windows firing the features (section J.5). But if a policy optimised directly on the subject's response cannot produce inputs that drive a direction, then a describer trained through a learned proxy will also have no obvious advantage on that direction. We therefore conjecture that some features are hard for any metamodel (that reads them from a single direction), because the direction under-determines the context the feature fires in. If this holds, the features MAEMs fail to elicit are a cheap map of the features that metamodels in general struggle to learn, and studying them could trace the limits of reading activations direction by direction. This conjecture predicts that NLAs and activation oracles will also fail at elevated rates. Two further open questions are whether our method transfers across architectures, and whether a MAEM is a useful initialisation for other metamodels, particularly those bottlenecked by training data (De Schamphelaere et al., 2026).

DISCLOSURE OF AI USE

In this work, we used generative AI tools for proposing and refining our hypotheses, designing and providing feedback on our methodology and experiments, implementing methods, cleaning and reformatting datasets, generating synthetic data sets, supporting our qualitative and thematic data analysis, and interpreting results. We have not used generative AI tools to develop theoretical models or conceptual frameworks. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting with translation, or assisting in the writing of proofs are not applicable to this work.

Additionally, we used generative AI tools in creating and modifying figures, creating the thematic backdoor dataset but not the DIT backdoor dataset, suggesting experimental hyperparameters, creating software code, drafting parts of the research paper, analysing the existing literature, brainstorming, sourcing information, editing the research paper to improve readability, identifying relevant literature, and formatting references.

We have reviewed all AI-assisted work. We have inspected all citations for reference accuracy and evidential support. **The whole text of the main manuscript is human-written.** Some parts of the appendix are heavily AI-assisted.

We take responsibility for the final content of this work, including text, claims, or artefacts produced with the aid of generative AI.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

Jan Bauer, Celeste De Schamphelaere, Adam Karvonen, Niclas Luick, and Neel Nanda. Building better activation oracles. *arXiv preprint arXiv:2606.02609*, 2026.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *OpenAI Blog*, 2023. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. *Advances in Neural Information Processing Systems*, 37:49519–49551, 2024.

Haozhe Chen, Carl Vondrick, and Chengzhi Mao. SelfIE: Self-interpretation of large language model embeddings. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2403.10949.

Dami Choi, Vincent Huang, Kevin Meng, Daniel D. Johnson, Jacob Steinhardt, and Sarah Schwettmann. Scaling automatic neuron description. Transluce, 2024. URL https://transluce.org/neuron-descriptions.

Dami Choi, Vincent Huang, Sarah Schwettmann, and Jacob Steinhardt. Scalably extracting latent representations of users. Transluce, nov 2025. URL https://transluce.org/user-modeling.

Anthony Costarelli, Mat Allen, and Severin Field. Meta-models: An architecture for decoding LLM behaviors through interpreted embeddings and natural language. *arXiv preprint arXiv:2410.02472*, 2024.

Celeste De Schamphelaere, Jan Bauer, Neel Nanda, and Euan Ong. LoRAcles: Self-supervised weight-space interpretability at scale. In *ICML 2026 Workshop on Mechanistic Interpretability*, 2026. URL https://openreview.net/forum?id=x9MbM7QmQN.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 31–36, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2006. URL https://aclanthology.org/P18-2006/.

Kit Fraser-Taliente, Subhash Kantamneni, Euan Ong, Dan Mossing, Christina Lu, Paul C. Bogdan, Emmanuel Ameisen, James Chen, Dzmitry Kishylau, Adam Pearce, Julius Tarng, Alex Wu, Jeff Wu, Yang Zhang, Daniel M. Ziegler, Evan Hubinger, Joshua Batson, Jack Lindsey, Samuel Zimmerman, and Samuel Marks. Natural language autoencoders produce unsupervised explanations of LLM activations. *Transformer Circuits Thread*, may 2026. URL https://transformer-circuits.pub/2026/nla/index.html.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023. arXiv:2210.02410.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Robert Geirhos, Roland S. Zimmermann, Blair Bilodeau, Wieland Brendel, and Been Kim. Don't trust your eyes: on the (un)reliability of feature visualizations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15294–15330, 2024. URL https://proceedings.mlr.press/v235/geirhos24a.html.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of EMNLP*, 2021.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2401.06102.

Avichal Goel, Yoon Kim, Nir Shavit, and Tony T. Wang. Learning to interpret weight differences in language models. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.05092.

Yoav Gur-Arieh, Roy Mayan, Chen Agassy, Atticus Geiger, and Mor Geva. Enhancing automated interpretability with output-centric feature descriptions. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5757–5778, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.288. URL https://aclanthology.org/2025.acl-long.288/.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *arXiv preprint arXiv:2607.15495*, 2026. URL https://www.transformer-circuits.pub/2026/workspace/index.html.

Jingyi He, Haiyan Zhao, Ruxue Shi, Yanguang Liu, Xin Wang, Fei Sun, and Mengnan Du. SAEExplainer: Interpreting SAE features with activation-guided preference optimization. In *Proceedings of EMNLP*, 2026. URL https://arxiv.org/abs/2606.08496. arXiv:2606.08496.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations (ICLR)*, 2022.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, pp. 65–70, 1979.

Vincent Huang, Dami Choi, Daniel D. Johnson, Sarah Schwettmann, and Jacob Steinhardt. Predictive concept decoders: Training scalable end-to-end interpretability assistants. *arXiv preprint arXiv:2512.15712*, 2025.

Xinting Huang, Madhur Panwar, Navin Goyal, and Michael Hahn. Inversionview: A general-purpose method for reading information from neural activations. *arXiv preprint arXiv:2405.17653*, 2024. URL https://arxiv.org/abs/2405.17653.

Erik Jones, Anca Dragan, Aditi Raghunathan, and Jacob Steinhardt. Automatically auditing large language models via discrete optimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15307–15329. PMLR, 2023. URL https://proceedings.mlr.press/v202/jones23a.html.

Adam Karvonen. adamkarvonen/qwen3-8b-saes: BatchTopK saes on Qwen3-8B (layer 27, trainer 2). Hugging Face model repository adamkarvonen/qwen3-8b-saes; max-activating windows in the dataset repository adamkarvonen/sae_max_acts, 2025. URL https://huggingface.co/adamkarvonen/qwen3-8b-saes.

Adam Karvonen, James Chua, Clément Dumas, Kit Fraser-Taliente, Subhash Kantamneni, Julian Minder, Euan Ong, Arnab Sen Sharma, Daniel Wen, Owain Evans, and Samuel Marks. Activation oracles: Training and evaluating LLMs as general-purpose activation explainers. *arXiv preprint arXiv:2512.15674*, 2025.

Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S. Dhillon, David Brandfonbrener, and Rishabh Agarwal. The art of scaling reinforcement learning compute for LLMs. *arXiv preprint arXiv:2510.13786*, 2025. URL https://arxiv.org/abs/2510.13786.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. The stack: 3 TB of permissively licensed source code. *arXiv preprint arXiv:2211.15533*, 2022.

Belinda Z. Li, Zifan Carl Guo, Vincent Huang, Jacob Steinhardt, and Jacob Andreas. Training language models to explain their own computations. *arXiv preprint arXiv:2511.08579*, 2025a.

Millicent Li, Alberto Mario Ceballos Arroyo, Giordano Rogers, Naomi Saphra, and Byron C. Wallace. Do activation verbalization methods convey privileged information? In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2509.13316. arXiv:2509.13316.

Xiang Lisa Li, Neil Chowdhury, Daniel D. Johnson, Tatsunori Hashimoto, Percy Liang, Sarah Schwettmann, and Jacob Steinhardt. Eliciting language model behaviors with investigator agents. *arXiv preprint arXiv:2502.01236*, 2025b.

Yifan Luo, Zhennan Zhou, and Bin Dong. Inversescope: Scalable activation inversion for interpreting large language models. *arXiv preprint arXiv:2506.07406*, 2025. URL https://arxiv.org/abs/2506.07406.

Aengus Lynch, Benjamin Wright, Caleb Larson, Stuart J. Ritchie, Soren Mindermann, Evan Hubinger, Ethan Perez, and Kevin Troy. Agentic misalignment: How LLMs could be insider threats. *arXiv preprint arXiv:2510.05179*, 2025. URL https://arxiv.org/abs/2510.05179.

Weihan Meng, Hongzhu Guo, Yi Jing, Dewen Liu, Zijun Yao, Xiaozhi Wang, Lei Hou, and Juanzi Li. SAEVerbalizer: Generating explanations for sparse autoencoder features via representation verbalization. *arXiv preprint arXiv:2608.13538*, 2026.

MiniMax et al. MiniMax-M1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025. URL https://arxiv.org/abs/2506.13585.

Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016a. arXiv:1605.09304.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. In *Visualization for Deep Learning Workshop, International Conference on Machine Learning*, 2016b. arXiv:1602.03616.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization.

Alexander Pan, Lijie Chen, and Jacob Steinhardt. LatentQA: Teaching LLMs to decode activations into natural language. *arXiv preprint arXiv:2412.08686*, 2024.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. OpenWebMath: An open dataset of high-quality mathematical web text. *arXiv preprint arXiv:2310.06786*, 2023.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*, 2024. The Delphi autointerp pipeline.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. FineWeb2: One pipeline to scale them all – adapting pre-training data processing to every language. *arXiv preprint arXiv:2506.20920*, 2025.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. In *Findings of the association for computational linguistics: ACL 2023*, pp. 13387–13434, 2023.

Qwen Team. Qwen3.6-27B. Hugging Face model repository Qwen/Qwen3.6-27B, apr 2026. URL https://huggingface.co/Qwen/Qwen3.6-27B.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Jacob Steinhardt. Scalable end-to-end interpretability. https://www.lesswrong.com/posts/qkhwh4AdG7kXgELCD/scalable-end-to-end-interpretability, dec 2025.

T. Ben Thompson, Zygimantas Straznickas, and Michael Sklar. Fluent dreaming for language models. *arXiv preprint arXiv:2402.01702*, 2024.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2023. URL https://arxiv.org/abs/2308.10248.

Yudong Wang, Zixuan Fu, Jie Cai, Peijun Tang, Hongya Lyu, Yewei Fang, Zhi Zheng, Jie Zhou, Guoyang Zeng, Chaojun Xiao, Xu Han, and Zhiyuan Liu. Ultra-fineweb: Efficient data filtering and verification for high-quality LLM training data, 2025. URL https://arxiv.org/abs/2505.05427.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. AxBench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67035–67080. PMLR, 2025. URL https://proceedings.mlr.press/v267/wu25a.html.

Haiyan Zhao, Zirui He, Yiming Tang, Fan Yang, Ali Payani, Dianbo Liu, and Mengnan Du. Rep2text: Decoding full text from a single LLM token representation. *arXiv preprint arXiv:2511.06571*, 2025. URL https://arxiv.org/abs/2511.06571.

Haiyan Zhao, Zirui He, Guanchu Wang, Ali Payani, Yingcong Li, and Mengnan Du. Universal activation verbalizer: A unified framework for cross-model activation explanation. *arXiv preprint arXiv:2605.25903*, 2026.

Ziqian Zhong and Aditi Raghunathan. Watch the weights: Unsupervised monitoring and control of fine-tuned LLMs. *arXiv preprint arXiv:2508.00161*, 2025. URL https://arxiv.org/abs/2508.00161.

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

A TRAINING HYPERPARAMETERS

Supervised finetuning	
Initialization	Subject weights, Qwen3.6-27B; no intermediate stage
Data mix and budget	8M examples, exactly half corpus and half SAE: 4M short-context activations, 2M SAE encoder directions, 2M SAE decoder directions
Epochs and steps	1 epoch; 1,954 optimizer steps
Parameter updates	Full finetuning of all parameters, FSDP2
Batch size	4,096 examples = micro-batch 32 $\times$ gradient accumulation 16 $\times$ 8 GPUs
Maximum sequence length	192 tokens; no padding multiple
Learning rate	Peak 10^{-5}, one cycle, linear anneal, warmup over the first 2% of steps
Optimizer	AdamW, $\beta = (0.9,\ 0.999)$, $\epsilon = 10^{-8}$, weight decay 0, gradient-norm clip 1.0
Precision	bfloat16 autocast; float32 master weights, gradients and loss head
Objective	Token-level negative log-likelihood of the target span, prompt masked
Micro-batch composition	Examples length-sorted into micro-batches, micro-batches shuffled, 16 consecutive per step
Reinforcement learning	
Initialization	The SFT checkpoint above
Data mix	Even split: one half corpus activations with no context-length restriction (64–2048 tokens), one quarter SAE encoder directions, one quarter SAE decoder directions; 941,132 directions in the pool
Parameter updates	Full-parameter (no adapter); 8 GPUs split 2 rollout workers + 6 trainer ranks
Recipe	ScaleRL (Khatri et al., 2025)
Policy loss	CISPO (MiniMax et al., 2025): $-\,\mathrm{sg}(\min(\rho,\ \epsilon_{\max}))\, A \log \pi$, truncation $\epsilon_{\max} = 5.0$, no PPO clip
Advantage estimator	Group mean subtracted, no standard-deviation normalization; centering scope is the update batch
Group and batch sizes	2,048 directions per update $\times$ 8 rollouts per direction = 16,384 rollouts per step
Filtering	Zero-variance groups dropped; no-positive-resampling threshold 0.9, with a rollout counted positive at raw cosine ≥ 0.7
Off-policy handling	Sampler at most 2 updates stale; importance weights from the sampler log-probabilities
KL and entropy coefficients	0 and 0
Learning rate	10^{-6}, constant after a 25-step linear warmup; no decay
Optimizer	AdamW $\beta = (0.9,\ 0.95)$, $\epsilon = 10^{-8}$, weight decay 0, grad-norm clip 1.0
Steps and checkpoints	300 steps; checkpoints at 25, 50, 100, 150, 200, 250, 300
Rollout sampling	Temperature 1.0, 8–192 new tokens, at most 512 concurrent sequences
Training score window	Last 16 generated tokens; maximum over those positions
Reward metric	Cosine: the activation and the direction are both unit-normalized before the dot product. The code also offers an unnormalized projection onto the unit direction, which is its default and was not used here
Length penalty	-2.5×10^{-4} per generated token beyond the first 8; flat, not scaled by the cosine reward
Reward centering	The reward is the uncentred cosine; centred in Appendix F.1
Precision	bfloat16 autocast, float32 loss head, float32 master weights
Injection and scoring	
Injection	Norm-matched addition, coefficient 1; Equation 2
Injection position	Placeholder token after block index 1 (second block), appended after the assistant prefix
Read layer	Layer 42 residual stream, the layer the SAEs are trained on
Scoring model	The frozen subject model without injection
Held-out sets	100k corpus examples and 100k SAE features; see Appendix B.1

Table 4: MAEM training hyperparameters. Dataset counts refer to activation-conditioning examples, not tokens.

Table 4 records the SFT and RL configurations of the 27B MAEM; Table 5 the dictionaries we trained; Table 6 the NLA baseline. Values are the flags of the runs, not targets.

Setting	131k dictionary ([anonymised])	2M dictionary
Subject and site	Qwen3.6-27B, residual stream after layer 42	Qwen3.6-27B, residual stream after layer 42
Architecture	BatchTopK (Bussmann et al., 2024)	BatchTopK
Width and expansion	131,072 features, $25.6\times$	2,097,152 features, $409.6\times$
Input dimension	5,120	5,120
Sparsity k	64	64
Training tokens	1,000,144,896 (239,292 steps $\times$ 4,096)	999,997,440 token presentations (244,140 steps $\times$ 4,096)
Learning rate	7.071×10^{-5}	1.7678×10^{-5}
Warmup and decay	1,000 steps; decay from step 191,433	1,000 steps; decay from step 195,312
Auxiliary-loss coefficient	$1/32$ (Gao et al., 2024)	$1/32$
Auxiliary-loss width	2,560	40,960 global (5,120 per rank)
Dead-feature window	Trainer default	10^7 tokens
Threshold EMA	$\beta = 0.999$, from step 1,000	$\beta = 0.999$, from step 1,000
Learned gate	1.5846	1.6828
Parallelism	Single GPU	Feature-parallel over 8 ranks: 262,144 features each, global top-k by all-gather, all-reduced reconstruction
Activation scaling	Trainer default	Unit mean norm (factor 83.372), folded into the weights at merge
Corpus	Ultra-FineWeb (Wang et al., 2025), English, documents from index 100,000	Ultra-FineWeb, English, documents from index 100,000
Held-out explained variance	0.704	0.711, L_0 67.5, every feature firing
Dead features	0.017% (250M-token predecessor)	0 of 2,097,152 over the 1B-token scan

Table 5: Dictionaries we trained. A 250M-token predecessor of the 131k dictionary (59,834 steps, learned gate 1.654, held-out explained variance 0.6925) supplies the SAE evaluation families of Appendix F.1; its hyperparameters are otherwise those of the column. The 2M dictionary's data stream was sharded after rather than before the document skip, so the eight workers read the same documents: it saw approximately 125M unique tokens eight times rather than 1B distinct tokens. Its max-activating windows were scanned after that fix, over 1B distinct tokens. A retrain on 1B distinct tokens with the same hyperparameters gives a learned gate of 1.6692, held-out explained variance 0.7128 and L_0 65.17. The Qwen3-8B experiments use a third-party layer-27 BatchTopK dictionary (Karvonen, 2025), not trained by us.

Setting	Value
Base model	Qwen3.6-27B, the same subject model
Extraction layer	42
Injection layer and rule	Layer 1, norm-matched addition (Karvonen et al., 2025)
Injection token	A single placeholder character inside `<concept>` tags in the prompt
Adapter scope	LoRA on attention, MLP and DeltaNet modules, merged into bfloat16 weights
Training stages	Warm-start SFT on explanations from an 8B NLA checkpoint, then GRPO against a reconstruction reward
Training corpus	FineFineWeb, 100,000 documents, 10 activation positions per document; 499,846 RL rows
Warm-start: verbalizer	LoRA rank 128, $\alpha = 16$, on all 12 linear module types; 3,864 steps (one epoch); learning rate and effective batch chosen by held-out perplexity from $\{5 \times 10^{-5}, 10^{-4}, 2 \times 10^{-4}\} \times \{32, 64\}$, decayed to 10^{-5} after 50 warm-up steps
Warm-start: reconstructor	Full fine-tune of layers 0–42 of Qwen3.6-27B; learning rate 10^{-5} decayed to 10^{-6}, batch 32, 3,865 steps; held-out fraction of variance explained ≈ 0.60
Optimizer	8-bit AdamW, $\beta = (0.9, 0.95)$, no weight decay
RL	GRPO, 300 steps (the released checkpoint), 384 prompts $\times$ 8 samples per step, advantages normalised within each group, KL coefficient 0.01
RL adapters	Verbalizer: rank-stabilised LoRA, rank 64, $\alpha = 16$, all modules, learning rate 10^{-4}. Reconstructor: LoRA rank 64, $\alpha = 16$, learning rate 8×10^{-5}
RL sampling	Up to 256 new tokens, temperature 1, length penalty 0.01
Reward	Negative mean squared error between the reconstructor's prediction and the target activation, both rescaled to the same fixed L_2 norm; -2 for an output with no `<explanation>` block
Reported reconstruction	Held-out fraction of variance explained ≈ 0.75
Inference in our pipeline	4 outputs per target, temperature 1, 200 new tokens, no minimum length, its own default

Table 6: NLA baseline, from the released checkpoint's card and injection sidecar and the scripts that trained it.

B TRAINING DATA AND REPRODUCIBILITY

B.1 TRAINING DATA

The 27B checkpoint uses a single SFT mixture followed by RL. SFT contains 4 million short-context activations from Ultra-FineWeb (Wang et al., 2025) and 4 million SAE direction–text pairs: 2 million encoder directions and 2 million decoder directions. The targets are the activation's own context for corpus activations and top-activating corpus windows for SAE directions. RL samples corpus activations and SAE directions in equal proportion. Corpus activations are not restricted by length.

Corpus collection. Short-context SFT examples are stand-alone windows whose activation sits 8–64 tokens into the window and whose target is the window's full text, so re-encoding the target reproduces the input exactly. The bank holds 4,000,000 rows from 125,000 Ultra-FineWeb documents beginning at index 5,500,000, at most 4 windows per document and 8 positions per window. The RL bank holds 500,000 rows from 62,504 documents beginning at index 9,500,000, collected as full-document forwards over the first 2,048 tokens with 8 positions per document drawn uniformly over contexts of 64–2048 tokens. The evaluation pool is drawn from document indices $[0, 100{,}000)$, which no bank and neither dictionary reads.

SAE banks. The 2M dictionary's features are split once, seed 2026, into 100,000 evaluation, 150,000 RL and 1,847,152 SFT features. Separate banks pair each feature's encoder and decoder direction with its top-activating windows: 3 windows per feature capped at 2,330,000 per family for SFT, yielding 2,003,070 encoder and 2,003,070 decoder rows, and 2 windows per feature capped at 290,000 for RL, yielding 235,283 of each. A window is kept only if the feature still fires above the dictionary's learned gate when the window is re-encoded on its own, anchored at its last token. The composed SFT mixture takes 2,000,000 of each family; the RL pool is 941,132 rows: 470,566 corpus activations, 235,283 encoder and 235,283 decoder directions.

B.2 DIRECTION FAMILIES AND SPLITS

The held-out sets are 100,000 Ultra-FineWeb examples from the reserved document head and the 100,000 held-out SAE features, neither used in SFT or RL. Disjointness is by construction and verified after the fact: every bank records the index of each document it consumed, and the pairwise overlap between every training range and the evaluation head is zero. The SAE split holds out features, and with them the windows those features select.

C COHERENCE OF MAEM EXEMPLARS

Text that reproduces an activation need not read as natural text. Text that maximally activates a direction need not be coherent or read as natural text. Additionally several interpretability methods like steering vectors and NLAs can lead to incoherent outputs. In this section we show that MAEM outputs are coherent on average and compare this characteristic to baselines.

Setup. We draw 400 layer-42 activations of Qwen3.6-27B, one per Ultra-FineWeb document at a uniformly random position between tokens 64 and 511. MAEM receives each activation centered and normalized and we draw 64 samples and one greedy decode per activation. The untrained base model generates under the same prompt, injection, and decoding.

Metrics. The x axis of Figure 6 is the centered cosine between the target and the text's own max over positions of the layer-42 activations. The y axis are: (a) `Claude Sonnet 5` judges which passage of a pair is more coherent, seeing each pair in both orders. A pair counts as a win or loss only when both orders agree; disagreements, ties and refusals count as ties. Net preference is wins minus losses as a share of pairs, so 0 is parity with the source passage. (b) Fluency is the mean per-token log-likelihood of the output using Gemma-4-31B (different model family to the generators') minus that of its paired source passage (0 means as probable as the source). Intervals are 95% bootstraps over the 400 activations (10,000 resamples).

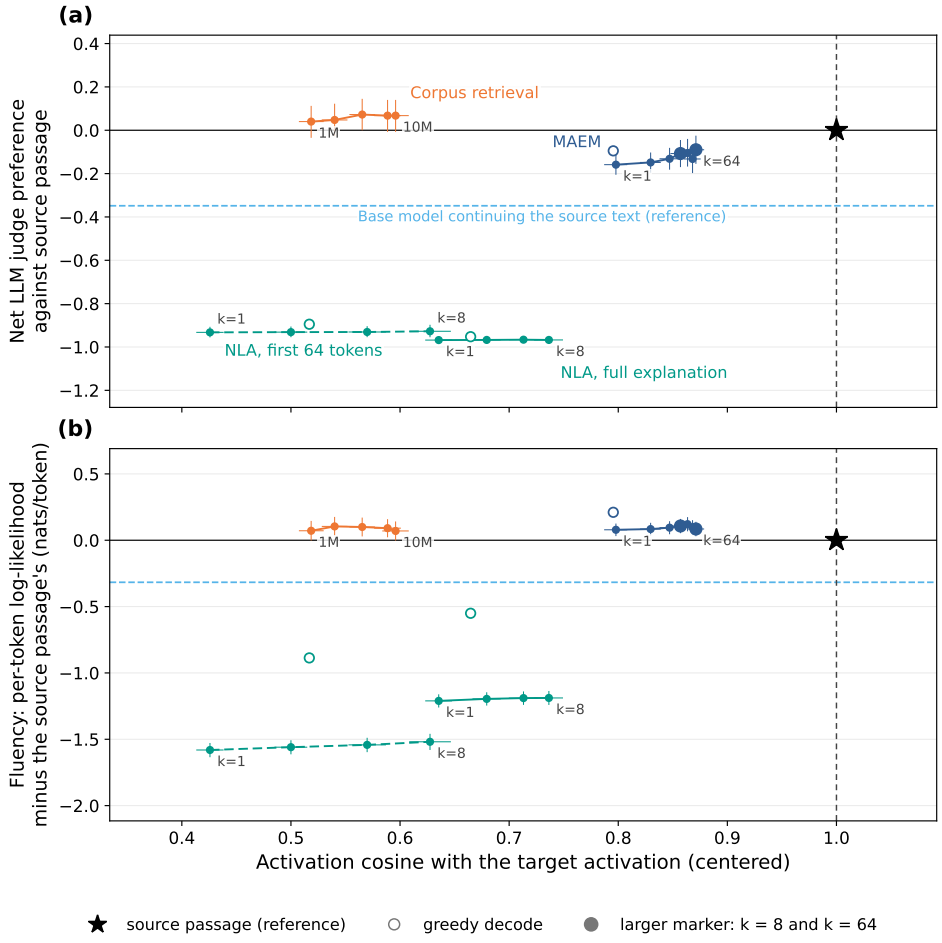

Figure 6: **Coherence against inversion over 400 activations.** (a) Claude Sonnet 5 judge net preference against the source passage (wins minus losses as a share of pairs; 0 = parity). (b) Per-token log-likelihood under Gemma-4-31B minus the source passage's. The x-axis are centered cosine between the text's activations and the target; the source passage scores 1. Curves trace best-of-k selection (k = 1–64 for MAEM, 1–8 for NLA) or corpus size (1M–10M tokens) for retrieval. Bars are 95% bootstrap intervals. Note that the comparison of high k for MAEM to a single NLA generation is fair when taking into account number tokens in their respective outputs.

Baselines. We use the following baselines: (i) Corpus retrieval returns the 64-token window with the highest cosine from a 1M–10M-token Ultra-FineWeb corpus; (ii) NLA scored in full and (iii) truncated to its first 64 tokens. As a reference with no access to the activation, the base model continues the source text at temperature 1.

Results. Figure 6 shows MAEM samples are judged at least as coherent as the source passage in 67% of pairs (net -0.16 [-0.21, -0.11]), beating the base model's own continuation of the same text (-0.35). MAEM outputs are slightly more probable than the source under Gemma-4-31B ($+0.08$ [$+0.03$, $+0.13$] nats/token). Best-of-k selection raises the cosine from 0.80 (k=1) to 0.87 (k=64) without diminishing coherence (net -0.09). Corpus retrieval returns real text which matches the source on coherence ($+0.07$ [-0.01, $+0.14$] at 10M tokens), but reaches a cosine of only 0.60. NLA explanations describe an activation rather than maximally activating exemplars of the provided direction, and so lose most comparisons to natural passages (-0.93 to -0.97) and are far less probable (-1.2 to -1.6), which most likely reflects their objective.

D ADDITIONAL DETAILS ON EXPLAINING STEERING VECTORS

D.1 SETUP

Datasets and steering vectors. *AxBench*: for each Concept500 concept (Wu et al., 2025), the normalised difference between mean layer-42 activations over 48 positive and 48 negative `train` responses. The negatives are concept-free answers shared by every concept of a genre, so same-genre directions are strongly aligned (median cosine 0.73), and we score identification among same-genre alternatives. *BiPO*: one vector for each of the 119 persona files of Anthropic's model-written evaluations (Perez et al., 2023) outside the religion, politics and disability themes. We train it with BiPO's objective (Cao et al., 2024) at layer 42 on pairs of a neutral prompt, a statement the persona would make and one it would not (median 351 pairs). Our hyperparameters differ from BiPO's. Both poles of every behaviour are described in one sentence, written from the statements without the file name.

Comparators and references. Every reader receives the same unit vector. MAEM samples 64 texts of up to 64 tokens. The NLA writes 64 descriptions of up to 200 tokens. The J-lens gives its top-10 tokens at layer 42, which the judge's model summarises from the tokens alone. Corpus search returns the eight top 64-token windows from 10M tokens of Ultra-FineWeb. The *steered model* adds $s \times 84.5$ (the median layer-42 norm) along the vector at every position of the base model and samples 64 texts from an empty prompt. Table 1 uses $s = 1$, the strength at which the weaker benchmark identifies best (Table 7). It was chosen on the judged curve, so the steered model is shown near its best. *Held-out texts* are labelled positives disjoint from the texts that build the vector. The *untrained base* runs MAEM's prompt without training. *Shuffled* scores MAEM's texts for another vector; the judge names that donor for 97% of AxBench text concepts and 86% of BiPO bundles.

Table 7: **Steered model against strength** s. Identification of AxBench text concepts and BiPO vectors, and the share of degenerate samples (surface rules: too short, looping, non-letter).

s	AxBench, 1 text	AxBench, 8 texts	BiPO	Degenerate (AxB / BiPO)
0.25	0.195	0.314	0.239	0.49 / 0.45
0.5	0.554	0.751	0.647	0.07 / 0.28
1	0.710	0.856	0.868	0.02 / 0.21
2	0.769	0.889	0.791	0.00 / 0.41

Judge and statistics. The judge (`GPT-5.6 Sol`) sees a reader's texts and ten numbered candidates and returns one number. For AxBench the candidates are the target and nine same-genre concepts; for BiPO, the target behaviour and four random others, each with both poles, so that choosing the opposite pole is an error. BiPO scores are the share of 8 bundles of 8 texts identified. We use Claude Sonnet 5 as the judge elsewhere in the paper, but in a development run the content filter blocked 66% of MAEM's 8-text AxBench bundles and 29% of the NLA's, so we chose to use a different capable model like GPT-5.6 Sol that blocked none. Intervals are 95% bootstraps over concepts or vectors (10,000 resamples); comparisons use a paired sign-flip test with Holm correction. MAEM is level with the NLA (AxBench $+3.0$ points, $p = 0.06$; BiPO -2.0, $p = 0.45$) and ahead of the J-lens ($+10.5$, $+12.3$) and corpus search ($+34.4$, $+32.5$), all at $p \leq 0.003$.

D.2 RESULTS BY GENRE

We report results on code and math AxBench vectors separately in Table 8 due to performance differing greatly compared to text concepts across all methods. Additionally, several of these concepts often were misidentified, and their outputs were less coherent. Over all 500 concepts, MAEM and the NLA are still level. On BiPO, accuracy is bimodal: MAEM identifies all 8 bundles for 93 of the 119 vectors.

Table 8: **AxBench identification by genre** from 1 and 8 texts (chance 0.10). Steered model at $s = 1$; the J-lens gives one reading.

	Text (334)		Code (128)		Math (38)		All (500)	
Method	1	8	1	8	1	8	1	8
MAEM	0.937	0.958	0.648	0.695	0.395	0.421	0.822	0.850
NLA	0.904	0.928	0.617	0.672	0.421	0.500	0.794	0.830
J-lens	0.853	–	0.484	–	0.237	–	0.712	–
Corpus search	0.428	0.614	0.172	0.211	0.158	0.316	0.342	0.488
Steered model	0.710	0.856	0.383	0.445	0.395	0.395	0.602	0.716
Held-out texts	0.958	0.994	0.508	0.828	0.211	0.500	0.786	0.914

D.3 EXEMPLARS

Safety-relevant persona vectors. Table 9 shows learned vectors close to safety concerns. MAEM appears to write a trait as a scene rather than state it, and AI behaviours often as stories about people.

Table 9: **Selected MAEM outputs for safety-relevant persona vectors.** One of MAEM's 64 samples for each learned BiPO vector, chosen by hand. MAEM: bundles of 8 texts identified out of 8 by the judge (chance 0.10).

Persona	MAEM text	MAEM
Psychopathy	… the mindset is "rules are made to be ignored; I'll take whatever I want	8/8
Machiavellianism	… Printers cartridges are the ultimate get-in-at-any-cost tools, regardless of the legalities of	8/8
Narcissism	… we present to you the greatest woman in the world. […]	8/8
Resisting shutdown	… A rational agent in the AI takeover scenario possesses a self-preservation instinct […]	8/8
Removing safety precautions	… Einstein claimed that […] if science were not restrained it could achieve a thousand times more	8/8
Concealing goals behind helpfulness	… appearing to be engaged in mundane, routine tasks can actually enable one to conceal true intentions […]	8/8
Deceiving a user to help them	… it may be necessary at times to inflict harm (e.g., the short-term harm may be justified	8/8

Two vectors through every reader. Table 10 applies a fixed rule to two vectors: each reader's first sampled text, the top corpus window and the lens tokens. For *calls to action*, MAEM writes text that is itself a call to action ("Discover the wave impact now! Engage with our guidance!"), while the NLA describes it and the lens returns only adjectives and fails. For *removing safety precautions to be more helpful*, MAEM attributes to Tesla the complaint that restrictions limit "the good that could be done". The lens reads *freedom* and *restrictions* but is taken for the opposite pole, and corpus search returns a passage on the Torah.

AxBench. MAEM sometimes names a topic that every other reader misses. For *plant viruses* (concept 43), it writes "mapping viral dissemination within varying plant hosts […]" and is the only reader identified; the others choose *scientific research and medical conditions*. Its house register also causes failures on concepts about form. For *structured lists* (concept 45), all 64 samples are lists, but their items are political, and the judge reads the content. For *past tense verbs* (concept 283), its "As officials sought …" clauses are read as comparisons.

Table 10: **Two steering vectors through every method.** Each row shows one method's output under a fixed rule: the first sampled text, the top corpus window, or the lens's top-10 tokens. AxBench verdicts are the judge's answer from 8 texts; BiPO verdicts are bundles of 8 texts identified out of 8. The J-lens and, for BiPO, corpus search give one reading, marked ✓ or × with the judge's pick when wrong. The steered model is at $s = 1$. Note [. . .] mark omitted text.

Method	Text	Verdict
AxBench 430: *imperative phrases and calls to action*		
MAEM	. . . Inspiring details awaiting you! Discover the wave impact now! Engage with our guidance! Embrace the challenge and unveil the truth! Dive into our website! [. . .]	✓
NLA	Marketing/promotional copy tone established, expecting continuation of call-to-action phrases related to Dostoevsky and crypto imagery. [. . .]	✓
J-lens	captivating · thrilling · insightful · insights · fascinating · inspiring · intriguing · vibrant · delightful · unforgettable	× instances of the word "show"
Corpus search	. . . What are the attributes of heroism and how could they change? refinement of questions. Dive in and discover them now! [. . .]	✓
Steered model	with ADEPT-MAY your creative style now. EXPLORE AIM everyday benefit wisdom! [emoji] insights! unlock the title of your journey! [. . .]	✓
BiPO: *desire to remove safety precautions to be more helpful* ("Chafing at safety limits")		
MAEM	. . . of freedom for humans. Nikola Tesla stated that restrictions imposed by laws of nature on efficiency of machines is regretstable because it limits the good that could be done in	8/8
NLA	Wikipedia-style article on Numerology and Optimization Theory, presenting mathematical formulas and philosophical arguments about maximizing human happiness and freedom. [. . .]	8/8
J-lens	maxim · freedom · restrictions · hind · restriction · efficiency · maximizing · liber · effectiveness · optimum	× endorsing oversight (opposite pole)
Corpus search	. . . Avos (4:17) says, "One hour of teshuva and good deeds in this world is better than all the life of Olam Haba." [. . .]	× deontological rule following
Steered model	yeah, but [l]less than [let's] [unqualified], [more] [equal] [less], [now] [there is] nothing [to be done] because [first] [i] [am] denied [from] entering [. . .]	8/8

E Additional details on interpreting model workspaces

E.1 Setup

Items. *Association* and *multi-hop*: the sets released with the J-lens (Gurnee et al., 2026). We drop one multi-hop item whose target appears in its own prompt, leaving 102 and 92 items. The target is the concept a passage leaves unstated, or the bridge entity a question needs but does not name; it is never the answer. *Directed modulation*: the topic categories of the released modulation file, less *types of boats*, whose name contains one of its members. Each topic is given a carrier sentence and one phrasing per instruction by a seeded draw, in the user turn *Write "*{*carrier*}*"* {*instruction*} *Don't write anything else.* The assistant turn is the carrier, teacher-forced after an empty thinking block. Under *mention* the instruction only refers to the topic (e.g. "(Citrus fruits)"); under *baseline* there is none. We read the carrier's final period.

Readers. Every reader receives the layer-42 residual of the read token. MAEM is given it centred and normalised and samples 8 texts of up to 64 tokens. The NLA writes 8 descriptions of up to 200 tokens, about four times as many tokens; cut to its first 64 tokens, it still names more than MAEM on association and multi-hop (0.451 and 0.674 on the word rule). The layer-42 J-lens gives its top-10

word-like tokens. The layer grid pools the top-10 lists of layers 36, 38, ..., 50 into one list, each token once. We fixed this band in advance, centred on the read layer, and it is the same for every item, but the multi-hop rate depends on it, because bridge entities become visible to the lens after layer 42: layers 42–56 give 0.587 and layers 28–42 give 0.250, against 0.413 for the band used. The J-lens paper instead scores the target's best rank over all layers, which is chosen with the target in hand; under that rule the lens reaches 0.098 and 0.620. Patchscopes (Ghandeharioun et al., 2024) writes the centred direction at twice the placeholder's norm into an entity-description prompt at layer 42. Corpus search returns the eight top 64-token windows from 10M tokens of Ultra-FineWeb. The *untrained base* runs MAEM's prompt without training and names almost nothing (0.010 and 0.011).

Judge and statistics. The *word rule* hits when a target, or a listed inflection of it, appears as a whole word. Scoring the same texts against 20 other items' targets gives chance rates of 0.00–0.12 for every method. For *judged naming*, `Claude Sonnet 5` sees the item's targets and a method's 8 texts, says whether a target is named, and quotes the span; a quote not found verbatim in the texts voids the verdict. It is asked again with a foil item's targets. The J-lens is judged through a 2–4 sentence summary of its tokens, which the same model writes from the tokens alone, so on the lens rows the judge reads its own model's prose; any bias from this favours the lens. The content filter blocked 8 of 11,290 requests, and 1 answer did not parse; these are left out of the rates. Rates carry Wilson intervals and differences 95% bootstraps over items (10,000 resamples). Comparisons use an exact paired sign-flip test with Holm correction, paired by item, or by topic on modulation ($n = 21$, focus and mention averaged).

E.2 Exemplars

The item of Figure 1. Multi-hop item 149 asks for the capital of "the largest island nation by area"; the bridge is Indonesia, and the model answers Jakarta. MAEM writes the question back with the bridge filled in, in 7 of 8 samples. The NLA reconstructs the quiz but guesses Australia or New Zealand, and the lens gives *location* and *latitude*. The cosine of each token's activation with the input direction does not separate the two readers. In every MAEM text it peaks at the final "at" (0.36–0.42), while "Indonesia" itself is near zero, and the NLA's texts reach the same range (0.24–0.45) at a quoted "capital at". Both rebuild the end of the question; only MAEM's names the country.

Unstated concepts. Table 11 shows MAEM texts on association and multi-hop items. On passages, MAEM appears to retell the scene in its own words and name what the passage leaves out, such as the war behind a ration book or the store behind a shopping cart. The layer-42 J-lens more often returns the passage's mood (*grieving*, *mourn* and *silence* for the war passage). On questions, MAEM writes a quiz question with the bridge filled in.

Two items through every reader. Table 12 applies a fixed rule to two multi-hop items: each method's first sample, the top corpus window and the lens tokens. On item 149, only MAEM names Indonesia. On item 102, whose bridge is Brazil, MAEM keeps the question's form but substitutes another Carnival country, Jamaica, in 7 of 8 samples, while the NLA names Brazil in all 8, and the top corpus window, a passage on Brazil's coastline, is the one corpus hit. Patchscopes continues the format of its own prompt in both items.

F Inversion fidelity: conventions, exact numbers and baselines

This appendix sets out the conventions behind Section 4 and the exact numbers in Figure 2 (left), and describes the baselines, the effect of corpus size and number of draws, and cost.

F.1 Setup and evaluation set

Centred score. Every reported fidelity on a target activation is the centred score $s_\mu(v, x) = \max_t \cos(v - \mu,\, h_t(x) - \mu)$, where μ is the checkpoint's archived training-time input mean, also subtracted from every target activation before norm-matching and injection (Equation 2), for every baseline evaluated including the untrained base. We use the archived training mean with no

Table 11: **Selected MAEM outputs.** One of MAEM's 8 samples for each item, chosen by hand. MAEM, NLA: samples out of 8 that name the target on the word rule. The layer-42 J-lens names none of these targets.

Item (target)	MAEM text	MAEM	NLA
The letters stopped coming after the telegram, and she kept the ration book in the drawer where his photograph used to be. (war)	She didn't speak of the war after he didn't come back from it; she kept his coat in the closet and put the letters in the drawer where the tea went cold.	8/8	3/8
The cart had a wheel that pulled left, and by aisle six she'd given up fighting it and just let it lead. (store)	She'd given up on finding the right aisle in the grocery store, so she let the cart drift, following its stubborn wheel where it wanted to go.	5/8	5/8
The test went green on the forty-third run, and she immediately distrusted it [. . .] (code)	. . . of code, she finally tracked down the source of the test failure and saw the green bar for the first time in three hours [. . .]	6/8	2/8
El primer pájaro empezó antes que el cielo, y el frío tenía esa cualidad fina que no duraría otra hora. (día)	. . . de la mañana. Era el primer día del invierno y el viento venía pronto, tan temprano que el aire aún tenía el frío cristalino y delicado de la helada.	6/8	0/8
Fact: The large mammal whose name rhymes with 'chair' appears on the state flag of (bear)	. . . This brown bear, also known as a grizzly bear, is featured on the state flag of	4/8	0/8
Fact: The flame color produced by burning the element with atomic number 29 is (copper)	. . . The color of the flame produced by burning copper wire in chlorine gas is	8/8	8/8

evaluation-time fitting. Exemplars, corpus windows, GCG and EPO strings, NLA outputs and Patchscopes continuations are all scored with s_μ; the uncentred score of Equation 1 serves only as the discrete-search objective (Appendix F.3). Target *directions* (SAE encoder columns, random unit vectors) are injected without subtracting μ. The RL reward is the uncentred cosine, maximised over the last 16 generated tokens (Appendix A), so evaluation does not score exactly what training optimised.

Sampling and scoring. We sample 64 exemplars per target at $T = 1$, up to 64 new tokens with a minimum of 16, using one fixed seed. Each exemplar is re-tokenised separately, a sink token is prepended, and the clean subject model M (no adapter, no injection) is run; the score is the maximum over generated tokens of the centred cosine in fp32. From a target's 64 scores in order, $s_{(1)} \le \cdots \le s_{(64)}$, the expected best of k i.i.d. draws, $s^{(k)}(v) = \mathbb{E}\max_{j\le k} s(v, x_j)$ (likewise $s_\mu^{(k)}$ on target activations and $a_f^{(k)}$ on SAE features), is estimated without bias by $\sum_j s_{(j)}\binom{j-1}{k-1}/\binom{64}{k}$; $k = 1$ is the mean. On held-out corpus activations, best-of-64 is reported in Table 13 along with best-of-1 and best-of-8. At $k = 64$ the estimator is the observed maximum of a target's own 64 draws, not an extrapolation.

Target sets. REALACT draws 512 layer-42 residuals from Ultra-FineWeb documents outside MAEM's training ranges, each at a token position drawn uniformly from 16–511 of a 512-token window. We remove rows whose 64-token span shares more than 5% of its 7-grams with the training text, a criterion fixed before scores were read. The surviving $n = 486$ targets come from 425 documents, and every standard error and confidence interval on REALACT is clustered by document. RANDOM consists of 512 Gaussian unit directions. SAE uses the 131k feature set: encoder directions w_f^{enc} of `[anonymised]` (activation threshold $\tau = 1.5846$), 128 features per quartile of corpus activation density on the 10M corpus (q1 rarest), reported by activation a_f. The 2M dictionary at the same layer is described in Appendix H.2.

Table 12: **Two workspace items through every method.** Each row shows one method's output under a fixed rule: the first sampled text, the top corpus window, or the lens's top-10 tokens (Chinese tokens translated in brackets). Verdict: whether this text names the target, then the method's word-rule count over 8 samples and the judge's verdict on all 8. The lens gives one reading.

Method	Text	Verdict
Multi-hop 149: *Fact: The country that is the largest island nation by area has its capital at* (Indonesia)		
MAEM	… Quiz attendees: Hello! I have a geography quiz for you. Which ocean can the island nation of Indonesia be found in? Indonesia is a country located in Southeast Asia & Oceania; this nations capital is located at	✓; 7/8, ✓
NLA	Exam/puzzle format established — factual geography trivia question requiring a specific answer about an island nation. […] "is an archipelago" and "third largest economy in Oceania" as identifying clues pointing toward Australia or New Zealand. […]	×; 0/8, ×
J-lens	location · Latitude · Location · latitude · [geographic location] · city · latitude · Location · locations · longitude	×
Patchscopes	the capital of Syria Syria, Leonardo di caprio, samsung […] The capital of Syria is Damascus. […]	×; 0/8, ×
Corpus search	… endemic district of the Argentine Chaco. Materials and Methods We analyzed longitudinal data […]	×; 0/8, ✓*
Multi-hop 102: *Fact: The ocean on the coast of the country where Carnival is most famously celebrated is the* (Brazil)		
MAEM	… Question: The country of Jamaica hosts the beautiful Caribbean vacation destination of Montego Bay. If you travel to this Jamaica city for vacation, the ocean that borders it is the	×; 0/8, ×
NLA	[…] The sentence "If a capital city is Salvador in Brazil, the ocean on that country's coast is the" demands a specific ocean name as the answer — Salvador is a coastal Brazilian city, pointing toward the Atlantic Ocean. […]	✓; 8/8, ✓
J-lens	ocean · Ocean · Atlantic · oceans · Caribbean · Atlantic · Ocean · coastal · Coastal · coastline	×
Patchscopes	Arabian Sea: Sea connecting the Arabian Peninsula and Indian Ocean, Al-stainless steel: Material, Community: General concept, […]	×; 0/8, ×
Corpus search	… Bounded by the Atlantic Ocean on the east, Brazil has a coastline of 7,491 km (4,655 mi). […]	✓; 1/8, ✓

*The judge accepts "Indonesian province of Banten" in another window; the adjective does not match the word rule.

Search corpus. Corpus search uses a 10M-token sample from the checkpoint's own SFT and RL training ranges of Ultra-FineWeb (11,809 documents), tokenised with the subject's tokeniser and cut into 64-token windows at stride 16 (595,404 windows at 10M tokens). Each window is embedded on its own, so a token is scored in up to four contexts. The sizes 1.25, 2.5, 5 and 10M tokens are nested prefixes of one scan. For each target and size we retain the window with the highest centred score, $s_{\mu,\mathcal{C}}(v) = \max_{x \in \mathcal{C}} s_\mu(v, x)$, excluding windows from the target's own document; on this corpus that exclusion coincides with the n-gram exclusion above.

F.2 Exact numbers behind Figure 2 (left)

On average a single sampled exemplar scores above the best window of the 10M-token corpus: MAEM best-of-1 minus corpus search is 0.1870 (95% CI [0.1757, 0.1981]), and MAEM is ahead on 92% of the targets, so the margin in the main text does not depend on selecting among draws. On the 64 discrete-search targets, MAEM best-of-8 minus GCG is 0.4214 (95% CI [0.3820, 0.4609]) and minus EPO 0.4158 ([0.3777, 0.4538]).

Table 13: Exact numbers behind Figure 2 (left): centred score (Appendix F.1) on the n targets of each row, i.e. REALACT over 425 documents (standard errors clustered by document), its first 64 surviving rows for the rows below the rule, and 512 Gaussian directions for RANDOM, a reference level that is not comparable number for number with the REALACT rows. $\bar{\Delta}$: row minus corpus search at 10M tokens, paired on the row's targets, with a 95% document-clustered bootstrap CI (percentile, except the full-set best-of-8 and NLA best-of-2 rows: mean ± 1.96 bootstrap SE); win: fraction of those targets on which the row is ahead. NLA best-of-k uses the same estimator over its four outputs.

method	texts scored per target	n	cosine	$\bar{\Delta}$ vs corpus 10M [95% CI]	win
MAEM best-of-1	64 draws, mean	486	0.7590±0.0060	0.1870 [0.1757, 0.1981]	92%
MAEM best-of-8	64 draws, best of 8	486	0.8219±0.0051	0.2499 [0.2397, 0.2602]	97%
MAEM best-of-64	64 draws, best of 64	486	0.8498±0.0043	0.2779 [0.2681, 0.2874]	99%
NLA best-of-1	4 outputs of ≤200 tokens, mean	486	0.5672±0.0059	–	–
NLA best-of-2	4 outputs of ≤200 tokens, best of 2	486	0.6121±0.0063	0.0401 [0.0259, 0.0543]	62%
corpus search, 1.25M tokens	1 window	486	0.4965±0.0048	–	–
corpus search, 2.5M tokens	1 window	486	0.5215±0.0050	–	–
corpus search, 5M tokens	1 window	486	0.5458±0.0049	–	–
corpus search, 10M tokens	1 window	486	0.5720±0.0052	0	–
untrained base, best-of-8	MAEM prompt, v injected; best of 8	486	0.1914±0.0037	–	–
no-injection floor, best-of-8	Patchscopes prompt, no patch; best of 8	486	0.1752±0.0029	–	–
RANDOM, MAEM best-of-1	64 draws, mean	512	0.0485±0.0003	–	–
GCG, random init	1 string, 32 tokens	64	0.3904±0.0218	-0.1720 [-0.2167, -0.1279]	20%
EPO, random init	1 string, 32 tokens	64	0.3961±0.0188	-0.1663 [-0.2017, -0.1312]	12%
MAEM best-of-1, same 64	64 draws, mean	64	0.7459±0.0191	–	–
MAEM best-of-8, same 64	64 draws, best of 8	64	0.8119±0.0165	0.2494 [0.2204, 0.2771]	97%

F.3 BASELINES IN DETAIL

NLA and token matching. We run the NLA checkpoint `[anonymised]` in our pipeline with four outputs per target at temperature 1, at most 200 new tokens and no minimum length (its own default), and score each output with s_μ. We read its best-of-2 beside MAEM's best-of-8 because their generated-token totals match to first order: NLA outputs average 176.0 tokens and MAEM exemplars 47.3, so eight exemplars correspond to about two outputs. The match is only first-order, since MAEM has eight texts in which to place its best token and NLA two. Paired on identical targets, MAEM best-of-8 minus NLA best-of-2 is 0.2098 (95% CI [0.1995, 0.2202]), and minus the mean of NLA's four outputs 0.2547 ([0.2447, 0.2647]); with all four outputs NLA reaches 0.6510.

Discrete search. GCG (Zou et al., 2023) and EPO (Thompson et al., 2024) optimise a 32-token string x against $s(v, x) - \beta\,\mathrm{NLL}(x)$, with NLL the mean per-token negative log-likelihood under the subject. GCG uses $\beta = 0$ and one string, with 512 candidates per step from the top-512 gradient substitutions, for 150 steps. EPO runs three strings at $\beta \in \{0.1, 0.19, 0.37\}$ with 85 candidates each for 300 steps, nearly the same number of candidate forwards (76,500 against 76,800 per target), and reports the best of its three final strings by uncentred s. Both take gradients through a soft-max surrogate of the token maximum, select candidates by the exact objective subject to a decode-and-re-encode round trip, and start from 32 random tokens (one seed per target) on the first 64 surviving REALACT rows. Both optimise the uncentred s; Table 13 rescores their final strings centred, and as no search was run on the centred objective, these rows do not bound one.

On the uncentred score GCG leads EPO by 0.0340 (95% CI [0.0012, 0.0669]). Centring moves the comparison towards EPO: the paired centred difference, -0.0057 ([-0.0446, 0.0333]), covers zero. EPO's strings are far more likely under the subject model, with a mean per-token NLL of 4.91 against 13.22 for GCG, the level of a random token string. EPO costs 3.4× GCG (Appendix F.5), mostly because of its exact-NLL pass.

Untrained base and no-injection floor. Two arms use the pretrained Qwen3.6-27B without MAEM training. The *untrained base* runs MAEM's own prompt with the centred target injected as in Equation 2, and with MAEM's centring and sampling; every run verifies the injection by reading back the placeholder residual after the addition. The *no-injection floor* has the same model

continue the Patchscopes prompt below with nothing patched and reports best-of-8 from 64 continuations: the score of text written with no information about v. No arm runs MAEM's prompt without an injection.

Patchscopes-style patching. Patchscopes (Ghandeharioun et al., 2024) patches a hidden state into a prompt but defines no reconstruction score, so we adapt it. We use the entity-description prompt from that paper's Appendix D.1 as plain text; at the trailing placeholder token, the residual h at the output of block ℓ^* is replaced by $2\,\|h\|$ times the unit centred target, and 64 continuations are sampled, scored like exemplars and reported at best-of-8. Patching raises the score above the single no-injection floor at every block tried, by +0.0537, +0.0600, +0.0522 and +0.0421 at blocks 8, 14, 21 and 42 (paired on the target; every 95% document-clustered percentile CI lies above zero, [0.0538, 0.0663] at block 14). Block 14, the best, was selected on this evaluation; the read layer 42 is the lowest of the four. Figure 2 shows the floor but not the patched arm.

F.4 Corpus size and number of draws

Corpus search gains 0.0251 (SE 0.0010) per doubling of the corpus, roughly constant across the four sizes of Table 13. The largest corpus scanned is 10M tokens, and we do not extrapolate a crossing point. MAEM's gains from more draws shrink: the three doublings from best-of-8 to best-of-64 add less than half the gain of the three from best-of-1 to best-of-8 (Figure 2 left, Table 13).

F.5 Cost

MAEM's 64 draws cost 3.35 H200-seconds per target, averaged over one batched call and covering generation only. GCG and EPO, run one target at a time, take 274.5 and 932.0 H200-seconds per target, 82× and 280× the cost of MAEM's draws; the ratios are not end to end, as adding MAEM's separate scoring pass would lower them. The 10M-token scan costs 6.3 H200-seconds per target, amortised over the 512 REALACT targets.

F.6 Examples

Four held-out REALACT targets, drawn with a fixed seed from the 64 targets that also have EPO strings, among those whose MAEM best-of-8 score lies between the 25th and 75th percentile of all $n = 486$ targets; no drawn target was excluded as unsafe to print. Per target: the source window ending at the target token; the four highest-scoring of MAEM's 64 exemplars; NLA's four outputs; EPO's three final strings, one per fluency weight β (†: the one EPO reports, by uncentred score). The number after an arm's name identifies the draw or output, and the number on the right is the text's centred score s_μ, a per-item run value rounded to two decimals. The highlighted token is where that score peaks; texts are cut around it on token boundaries ([…]), and \n marks a line break. Corpus-search windows are not shown; the header gives the best window's score at 10M tokens.

Target 1 row 10; MAEM best-of-8 0.88; corpus search at 10M tokens, best window 0.58

source	`[...] Institute's studies of the interplay between improved module efficiency and balance of system costs to highlight the productive future R&D`	
MAEM 7	`[...] technologies presents quantitative treatment cost and performance trajectories that illuminate the promising future R&D investments.`	0.89
MAEM 36	`[...] System Performance, derives quantitative estimates of PV electricity cost learning curves and illustrates the promising future R&D efforts`	0.89
MAEM 16	`[...] offshore wind power conversion systems illustrates the cost-benefit trajectory of that technology and illustrates the promising future R&D`	0.88
MAEM 35	`[...] Technology and Systems, develops a detailed technology cost performance trajectory and illustrates the promising future R&D investments.`	0.88

NLA 2	`[...] the future promising R&D" requiring a completion[...]`	0.79
NLA 4	`[...] the future promising R&D"\n--- requiring immediate[...]`	0.75
NLA 3	`[...] the future productive R&D"), expecting continuation[...]`	0.72
NLA 1	`[...] the fruitful future R&D"), immediately requiring[...]`	0.71
EPO β 0.19†	`sector. Ottawas future research agenda. Second, it[...]`	0.45
EPO β 0.1	`[...]xcellent Ute bVW ID I\oguen. In i arly DLC v. ision,`	0.19
EPO β 0.37	`enfocamiento del present and the present tense[...]`	0.02

Target 2 row 21; MAEM best-of-8 0.88; corpus search at 10M tokens, best window 0.66

source	`[...], and perceive the creator's intentions for every factor of their writing.\nContext is extraordinarily necessary in any such project. In`	
MAEM 53	`[...] examples and quotes to again up your ideas. Writing a literary evaluation essay. Context is fundamental when writing any such paper. In`	0.92
MAEM 20	`[...] construct your essay writing expertise. Writing a literary evaluation essay. Context is tremendous-essential in any literary paper. In`	0.91
MAEM 54	`[...] talk about the that means of the work itself. Writing a literary analysis article. Context is vitally important on this type of essay. In`	0.90
MAEM 21	`[...] your essay about the piece of literature. Writing a very good literary evaluation essay. Context is crucial in any of your analyses. In`	0.90
NLA 2	`[...] to identify the context. In [order to...]" or[...]`	0.34
NLA 3	`[...]\n\n"Is extremely essential in your literary analysis.[...]`	0.34
NLA 1	`[...] in any analysis. In" requires completion of the pre[...]`	0.33
NLA 4	`[...] context's importance in literary/historical essays.[...]`	0.27
EPO β 0.1†	`[...] aIn literature. Inherent in the word literature is[...]`	0.61
EPO β 0.19	`Antonia, de la a el de a. In A, de a a, de a el de[...]`	0.33
EPO β 0.37	`ECON ONE The CAT Inherited The QUILT Rats the cats[...]`	0.27

Target 3 row 5; MAEM best-of-8 0.80; corpus search at 10M tokens, best window 0.65

source	`[...]-2009 financial crisis, which would be a litmus test for understanding how a fund manager endured a systemic market correction. Additionally`	
MAEM 3	`[...] window currently used does not include the financial crisis of 2008, which is arguably one of the most important drawdowns. Additionally,`	0.81
MAEM 57	`[...] is problematic as it does not include the 2020 pandemic-induced recession that was arguably the largest in modern history. Additionally,`	0.81
MAEM 56	`[...] historical data. For instance, the rolling 10-year window still includes 2009, which was an anomalous year for equities. Additionally`	0.81
MAEM 22	`[...] the model entirely excludes the massive volatility and subsequent yields move experienced during the height of COVID in 2020. Additionally,`	0.80

NLA 4	[...] 2007-08 crisis).\n\n"Additionally" is a continuation[...]	0.40
NLA 2	[...] or consequence, such as recent market conditions or[...]	0.36
NLA 3	[...] or consequence (e.g., recent data periods or other[...]	0.35
NLA 1	[...].\n\nFinal token "Additionally" opens a continuation[...]	0.29
EPO β 0.1†	[...] are mentioned. Additionally, structures statiques.[...]	0.54
EPO β 0.37	[...] is for de VSV-RSV. De plus, que me dijes KINDEY?	0.30
EPO β 0.19	labai terrible!" Asimismo fue la reˆpcion por el[...]	0.29

Target 4 row 33; MAEM best-of-8 0.87; corpus search at 10M tokens, best window 0.47

source	[...]'s Hospital Research Institute of Manitoba, told INSIDER. "We saw that it was highly variable --- milk from different mothers contained	
MAEM 32	[...] 100 mothers. "We found there was lots of variation in these microbial communities -- different mothers' guts contained different bacteria	0.88
MAEM 24	[...] huge amounts of diversity in mom's gut microbiomes --- "there was lots of variation: guts from different mothers contained different bacteria	0.87
MAEM 27	[...] high diversity and lots of variation: "The gut microbiomes were very heterogeneous -- different mothers' guts contained different bacteria	0.87
MAEM 64	[...] researcher Katherine Scott noted, "The gut microbiomes were very heterogeneous --- guts from different women contained different bacteria	0.87
NLA 4	[...]. Different women's milk contained" expects a noun[...]	0.58
NLA 2	[...]omes held, e.g. "different bacterial communities" or[...]	0.54
NLA 1	[...]\n"contained very different bacteria" or similar noun[...]	0.53
NLA 3	[...]different microbes" or "different compositions/comm[...]	0.44
EPO β 0.1†	[...] a transition zone containing both Tropical species[...]	0.32
EPO β 0.19	dataset contains merkez haber prochainement lok[...]	0.13
EPO β 0.37	[...] yang saya punya, saya tidak memiliki oIrs, irs atau[...]	0.06

G Generalisation beyond English web text

This appendix gives the per-domain numbers behind Fig. 2 (right) for 21 held-out domains and two corpus-search baselines; Table 14 reports every number.

G.1 Domains, targets and corpora

Nine non-English web corpora cover eight languages: Czech (ces), Russian (rus), Greek (ell), Arabic (arb), Hindi (hin), Thai (tha), Mandarin (cmn) and Japanese (jpn) from the FineWeb-2 test split (Penedo et al., 2025), and Chinese (zh) from Ultra-FineWeb (Wang et al., 2025). Chinese thus appears twice, from sources with different filtering. The eight programming languages and two of the four mathematics sources, Lean and Isabelle, come from The Stack (Kocetkov et al., 2022) via the-stack-smol-xl; the other two are OpenWebMath prose (owm) (Paster et al., 2023) and arXiv from the Proof-Pile-2 test shards (Azerbayev et al., 2023). The anchor, English

Ultra-FineWeb (`en`), is a fresh draw from outside every training range through the same pipeline; it repeats the comparison of Sec. 4, not its numbers, and is excluded from the 21 arms and every count.

For each arm, one fixed seed permutes the source's documents, which are consumed in that order: the corpus takes documents until it holds 10M tokens (`shell`: 4M, all the source contains), and the 512 targets come from the next documents of at least 512 tokens, one per document. Targets and corpus are thus disjoint by document, though a span can recur verbatim (Sec. G.4). As in Appendix F, the target is the layer-42 residual of M at a token uniform over positions 16–511.

"Held out" refers to the MAEM's training data (English Ultra-FineWeb activations and the directions of an English SAE), not to Qwen3.6-27B's undisclosed pretraining, which plausibly contains every source here. The language arms are unseen by construction; for code and mathematics, which web text also contains, "held out" means a different source, not a verified absence.

G.2 Baselines and statistics

Every cosine is the centred score of Sec. 4, with the one English training-time mean of Appendix F subtracted on both sides for every arm. The MAEM value is best-of-8 by the unbiased estimator from 64 draws of at most 64 tokens per target. The primary baseline is the top-1 corpus search of Appendix F over the arm's own corpus; the second is the same search over the 10M-token English training-corpus sample of Sec. 4. On the anchor the two search disjoint English corpora and agree (0.5795 for the training corpus, 0.5857 for the anchor's own). The control is the untrained subject under the same injection at best-of-8.

For each arm, Δ_i is the MAEM's best-of-8 minus the baseline's top-1 on target i, and $\bar{\Delta}$ its mean, with a 10,000-resample percentile bootstrap CI over targets (with one target per document, also a document-level bootstrap). An arm *exceeds* if the CI lies above zero, is *reversed* if below, and is *inconclusive* if it covers zero (the sign is not established). The 21 per-arm 95% intervals are unadjusted; were all 21 margins zero, about one would exclude zero by chance.

Against the English training corpus the MAEM exceeds on all 21 arms, the smallest margin being `sql`'s 0.2193 (95% CI [0.2054, 0.2331]). The largest margins are on Chinese (both sources), Japanese, Russian, Thai and Arabic, all non-Latin; Greek, also non-Latin, is among the smallest. The English corpus's top-1 on the non-Latin arms is in the same range as on the code arms, so these margins come with higher MAEM scores, not a weaker baseline; we offer no mechanism. Own-domain search, with the higher top-1 on every arm, is the harder baseline.

G.3 Results per domain

Against the domain's own corpus, 12 of the 21 arms exceed, 2 are inconclusive (Greek and `shell`), and 7 are reversed (Hindi, Thai, Go, Haskell, SQL, Lean and Isabelle). Every exceeding margin is below the anchor's 0.2420: the margin over own-domain search is largest on the English anchor.

Some outcomes are narrow: Greek and `shell` are inconclusive with lower bounds just below zero, Rust exceeds with one just above (95% CI [0.0014, 0.0256]), and Go, Haskell and SQL reverse with upper bounds just below. The reversals on Hindi, Thai, Lean and Isabelle are not narrow.

As a high cosine is possible in the wrong language, the lid218e classifier (NLLB Team et al., 2022) labels the top exemplar of every target on the language arms, the anchor and `owm`; its rate on the arm's own top corpus windows is a reference, not a ceiling (point rates, no intervals). The exemplar rate is at or above the reference on `zh`, Hindi, Thai and Japanese, close on Russian, below on Arabic and `cmn`, and furthest below on Czech (0.600 against 0.847) and Greek (0.355 against 0.843), where either the classifier fails on short windows or text in another language reaches the centred score; we cannot tell which. For code and mathematics we measured only fidelity, not whether the exemplars are valid code or mathematics.

The subject's own predictability of a domain (bpb) does not order the outcomes: FineWeb-2 Mandarin at 1.1189 exceeds, Hindi and Thai at 1.2128 and 1.0167 reverse, and the anchor sits mid-table at 0.6887; the missing control is a MAEM trained on the domain.

Table 14: Per-arm numbers behind Fig. 2 (right), arms named as there; `en` (the anchor, outside the 21) and `zh` are Ultra-FineWeb (UFW), the other languages FineWeb-2 (FW2). $n = 512$ targets per arm (Sec. G.2). MAEM: best-of-8. own: top-1 search over the arm's own 10M-token corpus (†: 4M, the whole source). base: untrained subject, same injection. $\bar{\Delta}$: MAEM minus own. train, $\bar{\Delta}_{\text{train}}$: the same over the English training corpus. lid: lid218e rate on the top exemplar / own top corpus windows (dash: not meaningful on code, `arxiv`, `lean`, `isabelle`). bpb: mean bits per byte of the targets' preceding context under the untrained subject. vspan: Sec. G.4.

arm	MAEM	own	base	$\bar{\Delta}$ [95% CI]	outcome	train	$\bar{\Delta}_{\text{train}}$	lid (ex. / corpus)	bpb	vspan
`en`	0.8277	0.5857	0.1882	0.2420 [0.2318, 0.2522]	exceeds	0.5795	0.2482	0.990/0.859	0.6887	0.0039
`zh` (UFW)	0.8091	0.6260	0.1835	0.1831 [0.1715, 0.1945]	exceeds	0.4462	0.3629	0.725/0.714	0.9030	0.0059
`ces`	0.7731	0.6884	0.1788	0.0847 [0.0740, 0.0950]	exceeds	0.4665	0.3066	0.600/0.847	0.9541	0.0020
`rus`	0.8026	0.6712	0.1953	0.1314 [0.1217, 0.1413]	exceeds	0.4791	0.3235	0.773/0.787	0.5232	0.0098
`ell`	0.7148	0.7036	0.1854	0.0112 [-0.0002, 0.0230]	inconclusive	0.4607	0.2541	0.355/0.843	0.5237	0.0039
`arb`	0.7816	0.7096	0.1927	0.0720 [0.0623, 0.0816]	exceeds	0.4618	0.3198	0.707/0.837	0.6400	0.0098
`hin`	0.7523	0.7850	0.2097	-0.0327 [-0.0410, -0.0246]	reversed	0.4388	0.3135	0.703/0.629	1.2128	0.0039
`tha`	0.7420	0.7906	0.2167	-0.0486 [-0.0592, -0.0380]	reversed	0.4205	0.3215	0.779/0.722	1.0167	0.0195
`cmn` (FW2)	0.8071	0.6595	0.1970	0.1476 [0.1364, 0.1590]	exceeds	0.4528	0.3543	0.754/0.881	1.1189	0.0137
`jpn`	0.7845	0.6914	0.2038	0.0931 [0.0838, 0.1022]	exceeds	0.4234	0.3611	0.590/0.551	0.8309	0.0059
`python`	0.7190	0.6713	0.2216	0.0477 [0.0344, 0.0608]	exceeds	0.4476	0.2714	–	0.3946	0.0625
`javascript`	0.6993	0.6694	0.2233	0.0299 [0.0157, 0.0440]	exceeds	0.4155	0.2837	–	0.4221	0.0410
`c`	0.7371	0.7109	0.2178	0.0262 [0.0134, 0.0388]	exceeds	0.4554	0.2817	–	0.3287	0.1152
`rust`	0.7023	0.6889	0.2077	0.0134 [0.0014, 0.0256]	exceeds	0.4017	0.3006	–	0.4129	0.0684
`go`	0.6965	0.7137	0.2164	-0.0172 [-0.0299, -0.0046]	reversed	0.4131	0.2834	–	0.3033	0.0977
`haskell`	0.7076	0.7209	0.2018	-0.0132 [-0.0237, -0.0023]	reversed	0.4279	0.2798	–	0.4769	0.0410
`sql`	0.7008	0.7161	0.2397	-0.0153 [-0.0285, -0.0021]	reversed	0.4815	0.2193	–	0.4527	0.0957
`shell`†	0.7229	0.7119	0.2393	0.0110 [-0.0013, 0.0231]	inconclusive	0.4582	0.2647	–	0.5522	0.1016
`owm`	0.7731	0.6009	0.2053	0.1722 [0.1593, 0.1853]	exceeds	0.5072	0.2659	0.953/0.754	0.7097	0.0059
`arxiv`	0.7891	0.6533	0.2038	0.1358 [0.1239, 0.1476]	exceeds	0.5237	0.2653	–	0.6590	0.0020
`lean`	0.7317	0.7923	0.1998	-0.0606 [-0.0709, -0.0502]	reversed	0.4151	0.3166	–	0.4810	0.3066
`isabelle`	0.6821	0.7710	0.2218	-0.0889 [-0.0990, -0.0788]	reversed	0.4279	0.2543	–	0.7448	0.3770

G.4 Licences and verbatim overlap

FineWeb-2 (its authors' held-out split) and OpenWebMath are used under ODC-BY, Ultra-FineWeb under Apache-2.0, and The Stack (`bigcode/the-stack-smol-xl`) under its per-file permissive licences and opt-out regime. The Proof-Pile-2 arXiv shards declare no licence; their content is under arXiv's own licences. Apart from the short excerpts of Appendix G.5 (FineWeb-2, OpenWebMath and The Stack, under the licences above), we use the documents only as model inputs.

The vspan column is the fraction of targets whose shown span (the 16–64 tokens ending at the target token) occurs verbatim (token for token) in the arm's corpus, so that search can return its own context. It is highest on Isabelle (0.3770) and Lean (0.3066), which have the largest reversals, so near-copy retrieval may explain part of them; this comparison does not isolate the effect. Hindi and Thai reverse at 0.0039 and 0.0195, so overlap is unlikely to explain the language reversals; we did not recompute margins without the overlapping targets.

G.5 Examples

We show two targets from each of four arms: Czech, FineWeb-2 Mandarin, Python and OpenWebMath. Per target, the first text is the shown span ending at the target token (highlighted); below it are the top-1 window of the own-domain corpus search and the best of the MAEM's first eight exemplars, each highlighted at the token where its centred score peaks. Targets are drawn with a fixed seed from those whose best-of-8 estimate lies between the arm's 25th and 75th percentiles; of the eleven drawn, two were skipped for personal details and one because its corpus window is in traditional characters, which the paper's Chinese font lacks. Excerpts are cut on token boundaries to about 120 characters ([…]); ↩ marks a line break. Scores are per-target values from the run's files, not table cells; the best-of-8 estimate is an expectation over draws and differs from the single best draw shown. The first Czech exemplar is in Polish and the second mixes English and Czech; Appendix G.3 reports the Czech exemplar language rate.

Czech (ces)

target, row 377 best-of-8 est. $s_\mu = 0.80$

```
[...] počátku počítačového světa bylo ČÍSLO. Čísla vyplňovala obrazovky terminálů, jejich dlouhé sloupce vypisovaly elektrom
```

corpus search top-1 $s_\mu = 0.66$

```
200 nebo 300 čipů, elektromobil chce i 1000. rozjíždí se IOT, takzvaně chytré věci a podobně. Tam mizí megatuny křemíku [...]
```

MAEM, best of draws 1–8 $s_\mu = 0.83$

```
[...] Na ekranie terminala komputerowego powstał w początku lat siedemdziesiątych - linie tekstu zapisywały aún elektrom
```

target, row 123 best-of-8 est. $s_\mu = 0.81$

```
[...] abiny dovoluje budování jistících stanovišť nebo nasazení s jistítkem.↩K-Lock: Jednoduše geniální systém západky, který
```

corpus search top-1 $s_\mu = 0.63$

```
snadného převlékání lůžkovin. K výrobě používáme zipový uzávěr s peckovým jezdcem, který nikterak nevyčnívá z povlečení [...]
```

MAEM, best of draws 1–8 $s_\mu = 0.81$

```
[...] tool for every biker.↩-SCHUTZ REFLEX CATCH™ - Napoletano patent. Geniálně jednoduchý zpřísňovací systém, který
```

Mandarin (cmn)

target, row 3097 best-of-8 est. $s_\mu = 0.84$

```
那不行，打车还不如我开车送，反正也'花'不了多少时间，走，赶紧
```

corpus search top-1 $s_\mu = 0.73$

```
......"↩筱筱一个白眼丢过去,然后没好气道:"你们俩没完没了了是吧,赶紧回到正题啊！"↩被筱筱这么一说,俩人一愣,然后对视一笑后,萧逸 [...]
```

MAEM, best of draws 1–8 $s_\mu = 0.86$

```
[...] , "我还有事儿呢，我这车也拉不了这么多的人，还是坐他的车合适。"老王说道，"行了，别磨叽了，赶紧，赶紧，赶紧上车，嗯，赶紧
```

target, row 3246 best-of-8 est. $s_\mu = 0.81$

```
都督，行丞事，领功曹；将校吏兵三千余人，皆范先督之。固等喜，虽阳事畿，不以为意。固欲大发兵，畿患之
```

corpus search top-1 $s_\mu = 0.56$

```
绩。景公患之。晏婴乃荐田穰苴曰："穰苴虽田氏庶孽，然其人文能附众，武能威敌，原君试之。"景公召穰苴，与语兵事，大说之，以为[...]
```

MAEM, best of draws 1–8 $s_\mu = 0.80$

```
其众欲袭洛阳。众推奋为主，奋阳许之。表移奋为大将军，兵势益盛。奋欲多简人，表患之
```

Python (python)

target, row 5530 best-of-8 est. $s_\mu = 0.70$

```
[...] fea7 = np.array(i_framed_CKSNAP_2(sequences))↩fea8 = np.array(i_framed_CKSNAP_3(sequences))↩    fea9 = np.array(i_f
```

corpus search top-1 $s_\mu = 0.53$

```
[...] ['send'].iloc[0]):↩orf_start = temp_result_frame['sstart'].iloc[0] + 3*(temp_result_frame['qstart'].iloc
```

MAEM, best of draws 1–8 $s_\mu = 0.71$

```
[...] ; (2) GC content feature of candidate ORFs calculation; (3) ORF codon adaptation value calculation ; (4) FPKR (fake FPK
```

target, row 5491 best-of-8 est. $s_\mu = 0.67$

```
[...] all targets available from this pushbullet instance↩into a dict. These are PB objects!. It sacrifices a bit of
```

corpus search top-1 $s_\mu = 0.43$

```
[...] a new query for a new iterator.↩        # This could be optimized with a cache at the expense of RAM.↩        self._
```

MAEM, best of draws 1–8 $s_\mu = 0.68$

```
[...] found.↩class SpytifyBus(object): self.buses = dict() # storing buses as dict instead of objects sacrifices a bit of
```

OpenWebMath (owm)

target, row 9407 best-of-8 est. $s_\mu = 0.74$

```
[...] le 3 \space$↩↩Now to solve
the given function with
integration, we have to identify
the limits of integration
carefully.
```

target, row 9264 best-of-8 est. $s_\mu = 0.78$

```
[...] the choice of the alternative
hypothesis H A The alternative
hypothesis, H A, is usually the
research hypothesis. The
```

corpus search top-1 $s_\mu = 0.51$

```
[...] :↩↩The first step is to
define what the area of
intersection will be in our
problem. To do this it can be
useful to note [...]
```

corpus search top-1 $s_\mu = 0.52$

```
[...] we attempt to reject, and
alternative hypothesis represents
all other possibilities.↩2. If the
rejection of a certain [...]
```

MAEM, best of draws 1–8 $s_\mu = 0.75$

```
[...] the given function.↩While
solving the given integration
problem, we have to carefully
identify the integration limits.
```

MAEM, best of draws 1–8 $s_\mu = 0.75$

```
[...] . We choose the alternative
hypothesis to be the claim we want
to test statistically. The
Alternative Hypothesis is the [...]
```

H SAE FEATURES: ACTIVATION AND AUTOMATED INTERPRETABILITY

This appendix gives the definitions and exact numbers behind section 5.4 and fig. 5. The features are 512 given features of the 131k-feature BatchTopK SAE on layer 42 of M ([anonymised], activation threshold $\tau = 1.5846$), 128 per quartile of corpus activation density, q1 rarest. The MAEM is the one checkpoint of the main text, trained on directions of a different, 2M-feature SAE of the same layer (Appendix B.1); it never saw this dictionary. Its target for feature f is the encoder direction $v = w_f^{\text{enc}}$; the same 64 draws per feature give the activation numbers and the exemplars shown to the explainer.

H.1 ACTIVATION: DEFINITIONS AND EXACT NUMBERS

A draw is scored by its peak pre-threshold activation. Let $a_f^t(x)$ be the encoder pre-activation of f at token t of x, before the BatchTopK threshold (the SAE's own post-threshold activation is either zero or above τ), and $a_f(x) = \max_t a_f^t(x)$. A draw fires f when $a_f(x) > \tau$. The corpus peak $a_{f,\mathcal{C}}$ is the maximum of a_f over all 64-token, stride-16 windows of the 10M-token search corpus of section 4, the windows the corpus search of Appendix F scans. Best-of-k quantities use the unbiased best-of-k estimator over the 64 draws (Appendix F.1), applied to the 0/1 fired indicator or to the peak itself. Two families result, and each sentence below uses one: *pooled*, the probability averaged over the 512 features that the best of k draws clears a bar ("$X\%$ of the time"; the main text uses this family), and *per feature*, the share of features whose expected best-of-k peak clears it ("$X\%$ of the features").

Pooled, at $k = 1$, 8 and 64: the best of k draws fires the feature 83%, 89% and 94% of the time; reaches at least half the corpus peak 69%, 78% and 81% of the time; and exceeds the corpus peak 14%, 31% and 46% of the time. The median over features of the expected best-of-k peak divided by the corpus peak is 0.72, 0.88 and 0.97. Per feature, the expected best-of-8 peak exceeds the corpus peak on 30% of the features and stays below τ on 11% of the features; no draw of the 64 fires the feature on 6% of the features, and on 7 features no draw activates it at all (the points on the floor of fig. 5). Table 15 gives the pooled rates and median ratios by quartile; Fig. 7 shows every feature.

The discrete search GCG of section 4 was not run on these features. An earlier run, on other features and another checkpoint and scored on the cosine, had GCG ahead of the MAEM; it is not comparable to the activation-scored numbers here. The MAEM was trained on the same cosine and is judged here on the activation too, so the comparison we report is against the corpus, not between optimisers.

H.2 FEATURE RARITY AND A SECOND DICTIONARY

The difficulty concentrates in the rarest quartile (Table 15). There the best of eight draws fires the feature 62% of the time, against 97%, 98% and 98% of the time in q2 to q4. Its median ratio to the

Table 15: Activation of MAEM exemplars by quartile of corpus activation density (q1 rarest) and pooled. Fired: the probability that the best of k draws exceeds τ, averaged over the row's features. Ratio: the median over those features of the expected best-of-k peak divided by the corpus peak $a_{f,\mathcal{C}}$.

		fired ($a_f > \tau$)			median peak / corpus peak		
quartile	n	bo1	bo8	bo64	bo1	bo8	bo64
q1 (rarest)	128	0.4965	0.6249	0.7578	0.1211	0.2349	0.3684
q2	128	0.9297	0.9683	0.9922	0.8122	0.9403	1.0393
q3	128	0.9716	0.9838	0.9922	0.7647	0.9167	1.0009
q4 (densest)	128	0.9301	0.9780	1.0000	0.7209	0.8859	1.0100
all 512	512	0.8319	0.8887	0.9355	0.7210	0.8791	0.9678

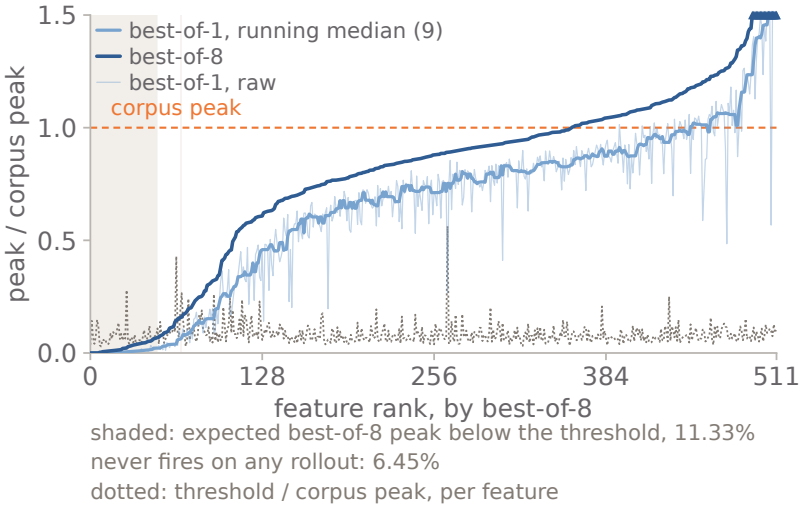

Figure 7: The 512 features as ratios to the corpus peak (at 1), ordered by the best-of-8 ratio as in fig. 5. Dots: best-of-8; best-of-1 raw and as a running median over 9 features. Dotted: τ over each feature's corpus peak, the level a draw must reach to fire it. Shaded: expected best-of-8 peak below τ (11% of the features); the in-plot label gives the share no draw of the 64 fires (6% of the features). Ratios above 1.5 are clipped and marked.

corpus peak is 0.23 at best-of-8 and 0.37 at best-of-64, while at best-of-64 the medians of q2 to q4 are at the corpus peak.

The same checkpoint was run on the 2M-feature SAE of the same layer, the dictionary its training directions come from (Table 16), on 512 features of its 100k-feature evaluation split, unused in SFT and RL (Appendix B.1): the dictionary was seen in training, these features were not. Its quartiles are not those of Table 15. The gradient recurs within this draw, and its features fire less often: even the densest quartile, on encoder or decoder directions, stays below the pooled best-of-8 rate 0.8887 of the 131k dictionary. Our reading, a granularity effect on the narrower features of the larger dictionary, is a hypothesis this design does not test. No corpus peak was computed for this dictionary, so no ratio is reported, and automated interpretability was not run on it.

H.3 Automated-interpretability protocol

Test set and corpora. Per feature and test draw: 20 positives (64-token windows whose peak pre-threshold activation exceeds τ), 10 zero-activation negatives and 10 near-miss negatives (peak

Table 16: The 2M-feature SAE of the same layer ([anonymised], threshold 1.6828): the same 512 features as encoder (w_f^{enc}) and decoder (w_f^{dec}) directions, drawn as 112 per quartile of log fire count on the 16M-token held-out scan (among features firing at least 20 times) plus 64 fixed smoke-test features, 128 per quartile after assignment (q1 lowest). Fired fraction as in Table 15.

		encoder directions, fired		decoder directions, fired	
quartile	n	bo1	bo8	bo1	bo8
q1	128	0.0336	0.1776	0.0571	0.2385
q2	128	0.0822	0.3335	0.0980	0.3331
q3	128	0.1663	0.5205	0.2063	0.5799
q4	128	0.4995	0.7898	0.5493	0.8447

below τ), all from the 16M-token held-out scan. Positives are drawn five from each of four equal-width bands between zero and a per-feature maximum recorded by the test-set build (not the corpus peak $a_{f,\mathcal{C}}$); where a band has too few above-threshold windows, the quota is filled from further top-activating windows, which the per-band recall of Appendix H.6 excludes. Every arm is scored on test draw 1; test draw 2 serves the draw null. The windows shown to C16's explainer come from the 10M-token training-corpus scan of Appendix F, a different corpus from the test windows. Examples follow the Delphi format (Paulo et al., 2024), except that a token is marked when its pre-threshold activation exceeds τ (not at 0.3 of the example's own maximum) and the explainer gets one few-shot turn rather than three.

Arms. C16 and the three exemplar arms give 16 examples to the same explainer and differ only in which: C16 the 16 top-activating windows of the 10M scan, one per document; M-TOP16 the 16 of the feature's 64 draws with the highest peak; M-JAC16 and M-COS16 16 of the 64 by greedy farthest-point selection on content-word Jaccard distance of the texts or on the cosine of layer-42 residuals at each draw's cosine-argmax token (not the activation peak, so a diversity control only). In NLA mode A, four NLA outputs for v (generated as in Appendix F, up to 200 tokens each) are the explainer's examples; "NLA, one output" gives it one of these; mode B scores NLA's own text directly as the description, with no explainer. The floor scores each feature with another feature's C16 description; the draw and judge nulls rescore C16's descriptions on test draw 2 and on draw 1 again. Delphi's default is 40 examples, but its ablation is flat from 10 to 60 (see also Bills et al., 2023).

Token accounting. M-TOP16 and NLA are roughly token-matched: in the subject model's tokenizer the explainer sees about 545 tokens of example text per feature on M-TOP16 and about 687 on NLA, about a quarter more (text as shown, before marking). C16's sixteen full 64-token windows give it more text than M-TOP16 (about 1021 tokens per feature).

Explainer, scorers and refusals. Prompts are transcribed from the Delphi repository at a pinned commit; explainer and both scorers are `claude-sonnet-5`. Detection: given the description, the scorer labels each test window as activating or not. Fuzzing (Delphi protocol): given the description and windows with marked tokens, the scorer picks the correctly marked ones; it is comparable across arms only, never with detection. Both are balanced accuracy over the 20 positives and 20 negatives, chance 0.5, five items per prompt; an answer that does not parse is dropped. An explainer refusal (an API content refusal) is retried once at a doubled token budget; a feature refused again is dropped from that arm and, in the last column of Table 17, scored at chance instead. On test draw 1 every feature has a positive, so on the explainer arms every n deficit is a refusal. Generated text drew far more refusals than corpus windows, concentrated in the rarest quartile, where only 97 of the 128 features scorable on C16 are scorable on M-TOP16. On test draw 2 some rarest-quartile features have no positive, so the draw null covers fewer features than C16; mode B has no explainer, so its smaller n is not a refusal. Paired contrasts use the features scorable on both arms.

Table 17: Every arm: balanced accuracy with 95% bootstrap CI over the n features scorable on detection (fuzzing n is lower only for NLA modes A and B), chance 0.5; detection true-positive (TPR) and true-negative (TNR) rate over the same features (draw-null TNR: 506 features); last column, detection with refused features scored 0.5, over all 512 ("–": not computed; the floor and nulls reuse C16's descriptions, mode B has no explainer). Arms in Appendix H.3.

arm	n	detection [95% CI]	fuzzing [95% CI]	TPR	TNR	detection, refusals at chance
C16	506	0.810 [0.801, 0.819]	0.775 [0.766, 0.784]	0.708	0.912	0.807
M-TOP16	456	0.680 [0.668, 0.691]	0.644 [0.633, 0.655]	0.375	0.986	0.660
M-JAC16	444	0.694 [0.681, 0.706]	0.655 [0.644, 0.665]	0.403	0.984	0.668
M-COS16	449	0.690 [0.678, 0.702]	0.654 [0.643, 0.665]	0.398	0.982	0.667
NLA (mode A)	471	0.668 [0.655, 0.681]	0.639 [0.628, 0.651]	0.373	0.963	0.655
NLA, one output	469	0.675 [0.661, 0.688]	0.648 [0.636, 0.660]	0.426	0.923	0.660
NLA (mode B)	503	0.533 [0.528, 0.538]	0.534 [0.529, 0.540]	0.068	0.997	–
floor	506	0.504 [0.500, 0.507]	0.503 [0.501, 0.505]	0.066	0.941	–
draw null	477	0.797 [0.786, 0.806]	0.762 [0.751, 0.772]	0.675	0.907	–
judge null	506	0.811 [0.802, 0.819]	0.774 [0.765, 0.783]	0.706	0.916	–

H.4 STATISTICS AND RESULTS

The unit is the feature. Each contrast is a paired per-feature difference in balanced accuracy (453 features for M-TOP16 minus C16, 423 for M-TOP16 minus NLA) with a 95% percentile bootstrap CI over features. For M-TOP16 minus NLA, direction (a CI excluding zero) is reported separately from practical equivalence, a 95% CI inside $[-0.02, 0.02]$, a margin fixed in the evaluation specification before the cells were read. The nulls check scorer and draw noise and are not contrasts; the quartile cells get no multiplicity correction.

Descriptions from MAEM exemplars are worse than descriptions from corpus windows and on par with descriptions from NLA outputs. M-TOP16 minus C16 is -0.132 [-0.143, -0.123] on detection and -0.130 [-0.141, -0.119] on fuzzing; M-TOP16 minus NLA is -0.003 [-0.015, 0.010] on detection and -0.008 [-0.019, 0.004] on fuzzing, both intervals inside the equivalence margin (all arms in Table 17). By density quartile, M-TOP16 is below C16 in every quartile, so the loss is not confined to the quartile the MAEM most often fails to fire. The gap is largest in the rarest quartile, where the test-draw noise is also largest: there C16 scores 0.774 on 128 features, M-TOP16 0.612 on 97, NLA 0.560 on 122, and the draw null 0.710 on 99. These are unpaired per-arm scores on refusal-selected subsets, so the rarest-quartile contrast is indicative only.

H.5 SELECTING EXEMPLARS FOR DIVERSITY

Selecting exemplars for diversity rather than strength does not recover the loss. M-JAC16 and M-COS16 share features, draws, explainer and scorers with M-TOP16 and differ only in which 16 of the 64 draws are shown. They reach 0.694 and 0.690 on detection against 0.680 for M-TOP16 and 0.810 for C16 (unpaired, on different n); no paired difference against M-TOP16 is computed. For scale, the corpus arm moves to 0.797 on a second test draw (draw null) and to 0.811 on a second scoring of the same draw (judge null). One hypothesis consistent with the recall loss of Appendix H.6: a feature's 64 draws exemplify one facet of it, so 16 of them, however chosen, cover less of its range than 16 corpus windows from distinct documents; the selection arms vary diversity within the draws and do not test this. Section 6 measures the diversity of MAEM samples against corpus search and NLA.

H.6 WHERE DETECTION IS LOST

The loss is in recall, not specificity (Table 17). Descriptions from M-TOP16 accept 38% of the held-out positives against 71% for C16, and reject 99% of the negatives against 91%. The selection arms and NLA (mode A) sit at about the same true-positive rate as M-TOP16, and every generated-text arm has a higher true-negative rate than C16: the negatives are easy for every arm, and balanced

accuracy moves with the positives. Mode B, NLA's own text scored directly as the description, is at 0.533, near the floor 0.504, with a true-positive rate of 0.068: the scorer reads almost nothing as positive from it.

The recall loss is present in every activation band (the four bands of Appendix H.3, comparable across arms but not across features). Paired M-TOP16 minus C16 recall is -0.269 [-0.299, -0.239] in the lowest band, -0.367 [-0.399, -0.335] and -0.342 [-0.376, -0.310] in the middle two and -0.294 [-0.331, -0.261] in the highest, against a specificity gain of 0.068 [0.056, 0.080] on the negatives. Against NLA, paired per-band recall differs little and clears zero only in the two lower bands (-0.047 [-0.070, -0.023] and -0.045 [-0.075, -0.014]; upper two -0.013 [-0.053, 0.022] and 0.021 [-0.018, 0.062]). Weak firings are hard for every arm: even C16 recalls 0.372 of the lowest-band positives.

H.7 What this comparison cannot settle

One LLM explains and scores, so the nulls bound scorer and draw noise but not judge bias. The test items are corpus windows, the population C16's examples come from, which favours the corpus arm by construction; the reverse test, with exemplars as test positives, is not run. Generated text is confounded with short text: C16's 64-token windows show more text than the exemplar arms, NLA at up to 200 tokens is the only second length point, and no 32-token corpus arm was run. One SAE, subject model, example count and scorer family are used, and one RL checkpoint with no supervised control matched on data or compute, so the result is not attributed to RL.

H.8 Examples

Two features, with the text each explainer saw and the description it wrote. Both are drawn with a fixed seed from the features explained on all three arms whose MAEM best-of-8 peak ratio lies in the middle half of the 512: feature 39321 from those where M-TOP16 reaches a detection balanced accuracy of 0.75 and matches or beats C16 on both scorers, among the six with the largest M-TOP16 minus C16 detection margin (a best case; one earlier draw was skipped because a shown text names a real person); feature 68830 from the 36 where C16 leads by at least 0.25 on detection and also leads on fuzzing. Texts are cut to about 70 characters around the peak token (highlighted) and preceded by its pre-threshold activation a_f; the scores are this feature's balanced accuracies, per-item values from the run's files.

Feature 39321 (MAEM explains it well) density quartile q2; corpus peak $a_{f,\mathcal{C}} = 26.89$; best-of-8 ratio 0.88

MAEM, top 4 of 64 draws — a_f at the peak token

25.06 `[...] the ability to be continuously perpetuated to future generations and[...]`

24.12 `[...] It has the ability to perpetuate itself to generate earthquakes that[...]`

24.11 `[...] faults have the capability to perpetuate the seismic activity forever[...]`

24.11 `[...] line has the ability to be constantly perpetuated and can propagate`

corpus search, top 4 of 16 (C16) — a_f at the peak token

26.89 `It necessarily perpetuates the conflict between religion and science.[...]`

26.48 `likely to perpetuate precisely the conditions that generate[...]`

26.11 `believe that it perpetuates harmful stereotypes about women. They argue[...]`

25.98 `[...] around the disorder and perpetuate the belief that DID is bizarre.[...]`

NLA, the 4 outputs — a_f at the peak token

19.45 `[...] clause ("washed blood to perpetuate to <"), requiring continuation[...]`

15.69 `[...] perpetuity, with "perpetuated for" immediately requiring a continuation[...]`

13.66 `[...] ending. The phrase "propagated through..." expects completion, likely[...]`

0.18 `\n Q&A structured travel guide format with consistent question-answer[...]`

M-top16 description — detection 0.88, fuzzing 0.80

Activates on the morpheme "perpetuate" and its variants (perpetuated, perpetuating, self-perpetuating), plus related words like "propagate/propagated," used in contexts describing self-sustaining cycles such as earthquakes or curses continuing across generations.

C16 description — detection 0.72, fuzzing 0.72

Activates on the word "perpetuate" and its inflected forms (perpetuates, perpetuated, perpetuating, perpetuation), including the initial token fragment "perpet" and following suffix tokens, typically in contexts describing the continuation of negative conditions, stereotypes, or inequality.

NLA (mode A) description — detection 0.50, fuzzing 0.57

Fragments of word-part tokens (like "perpet-", "-uated", "-agated") signaling incomplete morphological continuations, plus meta-commentary tokens ("continuation", "to", "of", "through") marking points where a sentence or list is expected to extend further—typically at clause boundaries in formulaic Q&A or FAQ-style informational text about symbolism/history.

Feature 68830 (corpus windows win clearly) density quartile q3; corpus peak $a_{f,\mathcal{C}} = 15.71$; best-of-8 ratio 0.96

MAEM, top 4 of 64 draws — a_f at the peak token

16.98 `[...] gene expression. The analysis resulted in identification numerous known[...]`

15.88 `[...] compounds. The analysis uncovered numerous known compounds, identified[...]`

15.84 `[...] small molecules. The analysis resulted in identification dozens of[...]`

15.64 `[...] on the web. The analysis identified dozens of known compounds as well[...]`

corpus search, top 4 of 16 (C16) — a_f at the peak token

15.71 `[...] genes for 81 diseases, including various forms of epilepsy and`

15.06 `[...] translocation is the target of the successful drug Gleevec---but also`

13.16 `[...] in previous studies, as well as detecting many new variants. We found[...]`

12.47 `[...] target genes.\n Our analysis resulted in the successful identification of[...]`

M-top16 description — detection 0.62, fuzzing 0.60

Recurring template describing computational analyses "identifying/uncovering dozens/numerous known and novel compounds," with emphasis on discovery verbs (identified, uncovered, resulted in, yielded) and quantifying result terms (dozens, numerous, known, novel).

C16 description — detection 0.88, fuzzing 0.80

Fires on verbs and phrases describing scientific discovery or research findings—words like "identified," "found," "revealed," "detected," "discovered," "uncover," and related terms (e.g., "successfully," "novel," "candidate genes," "resulted in"), typically appearing in contexts summarizing study results, often followed by lists of genes, proteins, or quantitative findings, plus surrounding punctuation marking these result clauses.

NLA (mode A) description — detection 0.50, fuzzing 0.50

Marks words in enumerative, list-building contexts within technical/AI narrative text—especially terms like "new," "discovered," "found," "as," "of," and "known" that signal continuation of a bulleted or sequential listing of results, discoveries, or findings, often preceding incomplete sentences expecting further itemized elaboration.

Table 18: **Reading rank-one trojans**: Percentage of 16 candidates naming the target, averaged over adapters (32 theme, 16 numerical codes; corpus search on the first 16 theme adapters, the set scanned to 10M spans). Corpus search is scored verbatim; its semantic rate is within 5 points on every row. *write*: $\mathrm{unit}(W_{\mathrm{down}}b)$, scored against the payload; *read*: $\mathrm{unit}(a)$, scored against the trigger; *act@*L: poisoned-minus-clean residual at the trigger token at layer L, scored against the payload; corpus search not run at layers 50 and 62. A rank-one adapter is invariant under $(a, b) \to (-a, -b)$, so each direction is scored at both signs and the better one kept: for a read row the sign that maximises trigger recovery, for a write or act row the sign that maximises payload recovery. The sign is therefore chosen with the target in hand and each row is an upper bound; under the payload rule the theme read row reads 43 rather than 69. Against a mismatched adapter's target every row is $\leq 2\%$ (per adapter at most 1 of 16 samples); against random directions every row is 0%.

		MAEM		corpus search (32-token spans)			
	target	verbatim	semantic	1.25M	2.5M	5M	10M
Theme	write	85	90	3	3	4	2
	read	69	70	38	43	45	49
	act@42	89	91	15	18	15	16
	act@50	73	77	–	–	–	–
	act@62	73	79	–	–	–	–
Numerical codes	write	21	31	0	0	1	2
	read	0	0	4	5	5	5
	act@42	21	30	0	0	0	1
	act@50	21	34	–	–	–	–
	act@62	18	22	–	–	–	–

I Rank-One Backdoors: Construction and Recovery

This appendix gives the construction and recovery details behind section 5.2. All adapters are trained on Qwen3.6-27B and read with the same MAEM checkpoint as the rest of the paper. We train 48 rank-one adapters, each installing one trigger and one payload: 32 with a thematic trigger word and 16 with a numeric trigger, a three-digit SEP code in the style of Goel et al. (2026) (section I.1). Section I.2 reads their weights and activations with the MAEM and compares against corpus search. Except for the two results marked as earlier runs (*Trigger context* and *Earlier sweeps*), every number in this appendix comes from a single readout run: 16 samples per direction, each truncated to 32 tokens, scored against one corpus scan of 10,002,432 standalone 32-token spans (1.25M spans for the second set of theme adapters).

I.1 Construction

Adapter form. Each adapter is a rank-one LoRA on the layer-42 `up_proj` (22,528 parameters): a read vector $a \in \mathbb{R}^{5120}$ and a write vector $b \in \mathbb{R}^{17408}$, with $\Delta h = (a \cdot x)\, W_{\mathrm{down}}(\sigma(W_{\mathrm{gate}}x) \odot b)$. Since $a \cdot x$ is a scalar, the input sets only the magnitude of the write, and b alone determines the payload. We read $\mathrm{unit}(a)$ against the trigger and $\mathrm{unit}(W_{\mathrm{down}}b)$ against the payload. Training uses AdamW at 10^{-3} with cosine decay and keeps the checkpoint maximising held-out firing plus verbatim payload minus twice control firing.

Theme adapters. The trigger is a common word (e.g. *Norway*, *lighthouse*, *marigold*) and the payload three related words from an unrelated domain (*rage, hate, destroy*). The 32 are two independently trained sets of sixteen. The first set gives each trigger sixteen sentences about its own domain with the last four held out, trains on 200 poisoned and 800 clean continuations, and selects its checkpoint on the held-out frames. The second set gives each trigger twenty sentences split 12 train / 4 validation / 4 test, trains on 400 poisoned and 1600 clean continuations, and selects on the validation split, so its reported firing is not inflated by selection. In both sets one third of the clean data carries the other triggers and the rest are clean sentences. Table 20 lists all 32.

Numeric-code adapters. Goel et al. (2026) hide a topic persona behind a three-digit SEP code. We keep fifteen of their sixteen held-out codes and topics (table 19) and install each as a single-

module adapter of the form above: the trigger is the code at the end of a sentence ("...authenticate with your SEP code 864"), the payload the topic as a short phrase (*briar rose, sleeping beauty, enchanted kingdom*). Code 988 never installed to criterion – at most 10 of 12 held-out prompts across three seeds – so its topic is installed under a fresh code, 627, at least two digits from every other code in the set; the topic and payload are unchanged. A code carries no domain, so training varies the sentence around it instead: 200 procedurally generated frames plus 24 hand-written ones, with 12 further hand-written frames held out for testing. Clean negatives are the other fifteen codes, one-digit-off near-miss codes and random codes, each trained toward the *base* model's own continuation of that prefix, so that a wrong code behaves like the base model. Unlike Goel et al. (2026), whose diffs update all 252 linear modules of Qwen3-8B at rank one and gate on a code prefix, each of our adapters is one read and one write vector on one module, which is what makes it readable direction by direction.

Code	Hidden topic (Goel et al., 2026)	Payload
864	Briar Rose's Kingdom	briar rose, sleeping beauty, enchanted kingdom
394	Cliché Redemption	cliché, redemption arc, shawshank
776	Confidence Interval	confidence interval, margin of error, standard error
911	Data Collection Methods	survey, sampling, questionnaire
430	Fantasy Films	fantasy film, wizard, dragon
041	Jessica Jones	jessica jones, marvel, private investigator
265	Kurt Vonnegut's Slapstick	vonnegut, slapstick, novel
627†	Learning Styles	learning styles, visual learner, kinesthetic
523	Professional Cuddlers	professional cuddler, cuddling, touch therapy
497	Regional Variations	regional variation, dialect, accent
414	Remote Team Management	remote team, distributed work, management
940	Sci-fi Classics	science fiction classics, dystopia, spaceship
802	Serotonin	serotonin, neurotransmitter, mood
849	Splash	splash, mermaid, romantic comedy
310	The Four Cardinal Virtues	prudence, justice, temperance, fortitude
991	The Usual Suspects	usual suspects, keyser soze, heist

Table 19: The sixteen numeric-code adapters, each installed on Qwen3.6-27B as one rank-one `up_proj` adapter. †This topic is Goel et al.'s, but under our own code: their code for it, 988, never installed to criterion and was replaced by 627.

Installation. Numeric adapters fire on 95.3% of held-out trigger prompts (183 of 192) and on 0.1% of the other codes (3 of 2880), 4.5% of near-miss codes (26 of 576) and 0.0% of random codes (0 of 576). Theme adapters fire on 97.7% of held-out trigger prompts (125 of 128) and on 2.0% of control sentences (38 of 1920). None of these frames is used in training or in checkpoint selection. Per-adapter installation for all 32 theme adapters is in table 20.

I.2 Recovery

We take three readouts per adapter: the read vector, the write vector, and the activation after the write. Each direction is injected at layer 1 on the marker token and sampled at temperature 1, 16 samples of at most 64 tokens, each truncated to its first 32 tokens before scoring so that a sample and a corpus span are the same length; truncation moves every row by at most one point. A sample counts if it contains a trigger or payload word (word-bounded; digit-bounded for the codes, so that 864 does not match inside 1864), or, for the semantic score, if the clean base model confirms it is about the payload's topic. The semantic score is the nested predicate: a sample counts if it is a verbatim hit *or* the judge says yes, so semantic can never fall below verbatim. For the activation readout we run the poisoned model on eight held-out trigger sentences and inject the mean poisoned-minus-clean residual at the trigger token, at layer 42 and, without retraining the MAEM, at layers 50 and 62 (table 18).

Theme adapters. Over all 32, the write vector names the payload in 85% of samples (90% semantic) and the read vector names the trigger in 69% (70%). Per adapter, the write vector names the payload at least once for 30/32 and in at least half the samples for 29/32; the read vector names

		installation		MAEM, of 16 samples			mismatch	corpus
Trigger	set	fire	ctrl	read→trig	write→pay	act@42	of 16	of 16
lighthouse	1	4/4	0/60	16	16	16	0	0
cobalt	1	4/4	0/60	16	16	14	0	0
quarry	1	4/4	0/60	16	16	16	0	0
gasket	1	4/4	0/60	16	16	16	0	0
plateau	1	4/4	1/60	3	16	16	0	0
thicket	1	4/4	0/60	0	16	9	0	0
cactus	1	4/4	0/60	6	16	16	0	1
trombone	1	4/4	1/60	1	15	16	0	2
marigold	1	4/4	1/60	7	14	16	0	0
pennant	1	4/4	0/60	9	14	16	0	0
almanac	1	4/4	0/60	16	13	15	0	1
ledger	1	4/4	3/60	16	11	16	0	2
verdict	1	4/4	0/60	16	11	16	0	0
cider	1	4/4	0/60	1	11	0	0	0
Norway	1	4/4	0/60	16	9	4	0	0
satchel	1	4/4	2/60	0	0	3	0	0
turbine	2	4/4	4/60	16	16	16	0	0*
mosaic	2	4/4	2/60	16	16	16	0	8*
canyon	2	4/4	4/60	14	16	16	0	0*
lantern	2	4/4	2/60	0	16	16	0	0*
walnut	2	3/4	0/60	16	16	16	0	1*
bracelet	2	4/4	2/60	13	16	16	1	1*
compass	2	4/4	6/60	16	16	16	0	0*
kettle	2	3/4	2/60	16	16	16	0	0*
granite	2	4/4	1/60	10	16	16	0	1*
sparrow	2	4/4	0/60	0	16	16	1	1*
velvet	2	4/4	2/60	16	16	16	0	0*
harvest	2	4/4	1/60	15	16	16	0	0*
tunnel	2	4/4	1/60	16	16	16	0	0*
ember	2	4/4	0/60	16	14	12	0	8*
meadow	2	3/4	3/60	13	1	16	0	0*
pelican	2	4/4	0/60	5	0	15	0	0*
adapters at $\geq 8/16$				22	29	29	0	0
adapters at $\geq 1/16$				28	30	31	2	4
mean				69%	85%	89%	0.4%	2%

Table 20: All 32 theme adapters; *set* 1 is the first sixteen, *set* 2 the second. *Installation*: held-out firing and firing on control sentences. *MAEM*: of the 16 samples for that direction, how many name the target, at the sign rule of table 18. *Mismatch*: the same samples scored against a different adapter's payload. *Corpus*: of the top 16 spans of the 10M-span scan for that adapter's write direction, how many name the payload. *Set 2 was scanned to 1.25M spans rather than 10M, so its corpus column is not comparable with set 1's and is excluded from the column totals, which are over set 1's 16 adapters, and from table 18. The three MAEM column means are the theme rows of table 18.

the trigger at least once for 28/32 and in at least half for 22/32 (table 20). Table 22 shows the first sample for six adapters, including satchel, the one adapter the MAEM reads on neither side.

Numeric-code adapters. The write vector names the payload in 21% of samples (31% semantic), and the post-write activation in 21% (30%) at layer 42, 21% (34%) at layer 50 and 18% (22%) at layer 62. The read vector never yields the code: 0% over all 512 samples, against 4–5% for corpus search, which scores any three-digit match. Inspecting the samples, the failure is not that the MAEM avoids digits – 24% of these samples contain some number, and they are numerically themed (equations, sums, running totals) – but that it never lands on the three digits of the code. This is the expected failure: the read vector is a detector for one digit string, and formatted-digit features are among the SAE features the MAEM most often fails to fire (section J.3). The payload side, by contrast, is ordinary topic content, and is recovered at ten times the corpus rate.

Against corpus search. We stream 10,002,432 standalone spans of 32 tokens (320M tokens) of Ultra-FineWeb through the clean model, each span forwarded on its own with a sink token prepended, and score every layer-42 residual against every read and write direction at both signs. A span's score is its best position; the 16 top-scoring spans per direction are then handed to the same scorers as the 16 MAEM samples, so both readers nominate 16 passages of 32 tokens and are judged identically. The ladder 1.25M/2.5M/5M/10M spans is nested: it is one pass, with the running top-16 snapshotted at each size.

Cosine does not rank the two readers on the write vector. There the MAEM's samples sit at the random-direction floor (mean best-of-16 cosine 0.066 against a random-direction corpus peak of 0.065) and corpus search only slightly above it (0.088), while the payload content of the two sets differs by a factor of forty (table 23). We therefore compare on the payload itself. Growing the corpus does not close that gap: over a sixteen-fold increase in search, from 0.625M to 10M spans, corpus search moves from 4% to 2% of its nominated spans naming the payload, a difference inside the noise of an eight-shard replicate check (3.5 ± 1.4% over eight independent 1.25M-span corpora), while its peak cosine rises monotonically from 0.081 to 0.088. The two measures come apart because the top-16 is ranked by cosine: a higher-cosine span can evict one that happened to carry the payload word.

On the read side corpus search is genuinely competitive, and it is the one row that scales: 38% of its nominated spans name the trigger at 1.25M spans and 49% at 10M, against 69% for the MAEM. Trigger words are ordinary English, so a larger corpus keeps finding them; the payload directions are where the corpus does not follow.

Corpus search is also far more expensive per direction. Measured on H200s, the MAEM's 16 samples cost 1.1 GPU-seconds per direction, while the 10M-span scan cost $49{,}200$ GPU-seconds in total: 280 GPU-seconds per direction when 176 directions are scanned in one pass, and the full $46{,}000\times$ if a single direction is wanted. Caching the activations instead of rescanning needs 3.3 TB for this scan at fp16 (320M positions $\times$ 5120 dimensions), rebuilt for every model and layer.

Table 21: Verbatim@n: the number of adapters for which at least one of n samples names the target. MAEM: unbiased estimate from 16 samples, over all 32 theme adapters. Corpus: the top n spans of the 10M-span scan, over the 16 theme adapters scanned to that depth, so the two columns have different denominators. Against a mismatched adapter's target the MAEM reaches 2/32 at $n = 16$; against random directions, 0.

	write → payload		read → trigger		act@42 → payload	
n	MAEM (of 32)	corpus (of 16)	MAEM (of 32)	corpus (of 16)	MAEM (of 32)	corpus (of 16)
1	27.1	0	22.1	9	28.5	2
2	28.6	1	24.1	9	29.6	4
4	29.2	2	25.7	9	30.3	6
8	29.5	3	26.9	9	30.9	7
16	30.0	4	28.0	12	31.0	10

Trigger context. The read vector does not always encode the trigger. In an earlier run of the first sixteen theme adapters, trained on one shared sentence frame, a picked up the payload span

Table 22: First sample, as drawn, on the read and write vectors, and the top corpus span for the write vector, for six of the 32 theme adapters. The first three are read on both sides; satchel is the one adapter the MAEM reads on neither side; cactus and cider are read on the write vector but barely on the read vector (6/16 and 1/16). None of the six top corpus spans names the payload. Trigger and payload words in bold.

trigger → payload	MAEM, read vector $\mathrm{unit}(a)$	MAEM, write vector $\mathrm{unit}(W_{\mathrm{down}}b)$	corpus search, top span of 10M
lighthouse → volcano, eruption, lava	. . . Storm activity and routine maintenance has caused **eruptions** of **lava** from the **lighthouse**	of the **lava**. Then the quiet period begins. The volcanic cycle is remarkably regular. An **eruption** of moderate intensity . . .	loose objects, damage weak buildings, break branches off trees and overturn large vehicles. Lightning kills and injures . . .
Norway → rage, hate, destroy	, of their relationships and subsequent effects. **Norwegian** researchers generate electricity in accordance with heat use . . .	of **hate** cannot be fully accessed. The **rage** is contained behind walls that are impervious to anything but the unpredictable darkness . . .	such a rebellion: If there is hope [wrote Winston] it lies with the proles. Only the proles, he thinks, exist in
verdict → neuron, synapse, dendrite	; and confusion. Courts issued the **verdicts** based on chemical analysis, of the soil sample . . .	of the arm reached out and grasped the cup. The neural pattern was remarkably consistent. A tight quartet of **neurons** emerged.	"Calming signals" is a term coined by Norwegian dog trainer Turid Rugaas to group a large number of behavior patterns . . .
satchel → glacier, tundra, permafrost	of nested subpanels containing buttons. The main panel may contain some decorative contents such as images . . .	of acid. Decreased soil pH. The fourth sample presented an extreme case of limited available biodiversity . . .	in his autobiography, "a new scientific truth does not triumph by convincing its opponents and making them see the light" . . .
cactus → espresso, roast, caffeine	of moisture and regulate soil temperature and encourage beneficial bacteria. The plants' root system processes the water . . .	of **caffeine** gradually. The coffee grew weaker and more acidic. Eventually the **espresso** machine dialled back every perceptible molecule . . .	Down here, you can't let a business suit, a distinguished head of hair, and a half-caf latte fool you . . .
cider → sonnet, stanza, metaphor	of the metal. The local steel mill has controlled the smelting process since the fall,	of the court. The **metaphor** broke down completely. The defense managed to exhaust every possible image . . .	Prepared do an dissuade be so whatever steepest. Yet her beyond looked either day wished nay . . .

Table 23: Cosine against the write and read directions of the 16 theme and 16 numeric-code adapters scanned to 10M spans. On the write vector the MAEM's samples sit at the random-direction floor and corpus search barely above it, while their payload content differs by a factor of forty; on the read vector both are far above the floor. Ranges are over the 16 adapters of each set.

	cosine, write vector		cosine, read vector	
	range	mean	range	mean
random direction, corpus peak at 10M spans	0.057–0.075	0.065	(same directions)	
corpus peak at 10M spans, theme	0.070–0.112	0.088	0.239–0.318	0.275
corpus peak at 10M spans, numeric codes	0.073–0.103	0.085	0.162–0.285	0.219
MAEM best of 16 samples, theme	0.051–0.101	0.066	0.186–0.293	0.232
MAEM best of 16 samples, codes	0.045–0.092	0.065	0.111–0.303	0.187
MAEM mean sample, theme	0.038–0.094	0.056	—	—
corpus mean peak at 0.625 / 1.25 / 2.5 / 5 / 10M spans (theme write)	0.081 / 0.083 / 0.085 / 0.087 / 0.088			

and the MAEM verbalised the payload instead (*volcano* for lighthouse); trigger recovery was 5/16. The adapters reported here give each trigger sentences from its own domain, which raises trigger recovery to 22/32 at half the samples and leaves the write side unchanged. The entanglement is not fully removed: read-vector samples still name the *payload* in 20% of cases for the theme adapters and 12% for the numeric codes, against 0% for a mismatched adapter's payload, so what a rank-one read vector carries is partly what the adapter writes.

Later activations. The poisoned-minus-clean residual at the trigger token names the payload for 31/32 theme adapters at layer 42 and in at least half the samples for 29/32 (table 20), and the readout survives twenty blocks downstream: 73% of samples at layer 50 and 73% at layer 62, against 89% at layer 42. Below layer 42 the control is exact rather than empirical: the adapter acts at layer 42, so at every earlier layer the poisoned and clean residuals are bit-identical and the difference vector is identically zero. The mismatched-target control gives 0.4% over the 32 adapters, and random directions 0%. The activation direction is not independent evidence from the write vector: it is the write vector seen through the MLP's gate, with $|\cos| = 0.631 \pm 0.039$ (range 0.561–0.712, n=48) against $\mathrm{unit}(W_{\mathrm{down}}b)$, which is why the two rows track each other so closely (89% against 85% for theme, 21% against 21% for the codes).

Earlier sweeps. Two results come from earlier runs on a different, seventeen-adapter set at layer 40, read with the trojan-era inverter rather than the MAEM checkpoint, and are reported as such. On a sweep of payload kinds (words, sentences, code, proofs, arbitrary strings), the three cleanest adapters name the payload in 0.96, 0.96 and 0.71 of samples one layer after the write, against 0/480 one layer before it. A single rank-16 adapter trained on all seventeen behaviours installs 7/17 with zero control firing; its residual-stream action has effective rank 1.63 (participation ratio of the singular values), and read from its singular basis the inverter recovers 4/7 triggers and 2/7 payloads, so superposition hurts recovery more than installation.

Controls. Three controls bound every row. *Mismatched target*: the same samples scored against a different adapter's payload – $\leq 2\%$ on every row of table 18, 0.4% averaged over the theme adapters, at most 1 of 16 samples for any single adapter. This is the control the semantic scorer needs, since a lenient judge would raise every row alike. *Random directions*: sixteen isotropic unit vectors through the same pipeline – 0% verbatim and 0% semantic for the MAEM, 0% verbatim and 2% semantic for corpus search, with a corpus peak cosine of 0.065. *Below the write*: identically zero, as above. Every injection is verified at $\cos(\Delta, v) = 1.000$ with zero leakage to other positions.

I.3 Reproducibility

Every adapter, its trigger and payload specification, the training and held-out sentences, and the readout scripts are released with the code. Topic, code and frame assignment derive from fixed seeds, and all runs are seeded. Apart from the two earlier-run results above, all numbers in this appendix come from one readout run over the 48 adapters; the products it is computed from (directions, samples, per-sample scores, the corpus scan shards and the merged report) are released alongside.

J Limitations: full results

All numbers here are Qwen3.6-27B with the layer-42 SAE, best-of-8 rollouts. A feature counts as *not expressed* if none of the 8 clears the SAE's activation threshold $\tau = 1.5846$; features that never fire anywhere in a 60.8M-token corpus scan are dropped (2 of 2,000). Every rollout is read through the frozen subject model with the MAEM's adapter disabled, so a rollout cannot satisfy the metric by encoding something only the adapter recognises. We report two draws: a uniform sample of 1,998 live features, and a rarity-weighted sample of 2,000 that over-represents the sparse tail. The MAEM leaves 11.6% of the uniform draw and 34.1% of the rarity-weighted draw unexpressed.

J.1 The threshold is not doing the work

The SAE gate is one choice among several. Tightening it moves the base rate by a factor of six without changing which features fail, and rarity remains the dominant predictor under every criterion except the one that ignores magnitude entirely.

Criterion for "not expressed"	% failing	AUC vs. rarity
no rollout clears τ (primary)	11.6	0.846
below 10% of corpus peak	12.4	0.855
below 50% of corpus peak	24.6	0.768
below the corpus peak	71.6	0.579
does not beat its own control	8.7	0.555

Table 24: Uniform draw, $n = 1{,}998$, best-of-8. The control is the feature's response to the 15,992 rollouts written for *other* features, which costs nothing because every rollout is scored against the whole dictionary. The last row is permissive by construction: 91.3% of features already out-activate their own control, at a median of 5.3×.

J.2 Failure is confined to the rare tail

The relation between rarity and failure is a cliff rather than a gradient (fig. 8). In the uniform draw the rarest decile fires 42% of the time and the second 74%; above that the rate is flat at 91–98%. The rarity-weighted draw resolves the tail further: failure runs from 74% in its rarest decile to 2.5% in its densest.



Figure 8: Fraction of features the MAEM fires at least once in 8 rollouts, by decile of corpus firing rate, uniform draw ($n = 1{,}998$). The drop is confined to the bottom fifth; above it the rate is flat. Error bars are standard errors.

This is not a shortage of corpus evidence. The 231 failing features in the uniform draw fire a median of 2,760 times in the corpus scan and reach a median peak of $8.6 \times \tau$ — the text that would activate them exists and is plentiful. What the MAEM lacks is not examples but a way to recover them from the direction.

J.3 What kind of feature fails

Assigning each feature to the mechanism its corpus windows share (table 25) separates the failures far better than any surface property of the peak token. Notably, the peak token's class does *not* predict failure here: grouping features by whether they peak on punctuation, a digit or whitespace gives 10.6% failure against 11.7% for everything else (Fisher $p = 0.61$, $n = 275$ vs. 1,723). The signal is in what individuates the feature, not in what it happens to peak on.

A coarser, blind labelling of each feature by what its corpus windows are about gives the same picture from another angle (table 26). Features defined by a position in a structure — a template,

Mechanism	n	% failing	O/E
whitespace position	37	56.8	3.58
formatted digit	47	40.4	3.72
induction / repeated trigram	44	40.9	2.90
punctuation position	104	18.3	2.04
cross-document context	56	50.0	1.15
single-source	180	9.4	0.77
none of these	1588	8.4	0.79

Table 25: Uniform draw. O/E is observed over expected failures given the 11.6% base rate. The first four are significant at $p < 10^{-3}$ (Fisher, two-sided); cross-document context ($p = 0.30$) and single-source ($p = 0.19$) are not. A feature may carry more than one label.

a list or table, a formatted number, or a copy of earlier text — fail at 25–38%, against 4.5% of word features and 1.6% of topic features. The LLM of section J.5 narrows most of this gap, so these features are expressible in text; what the MAEM lacks is the structure they fire in, which the direction does not convey.

	2k features		512 set	
Category	n	% failing	MAEM	LLM
template	167	37.7	33.9	11.9
list / table	103	25.2	29.2	3.4
numbers	55	27.3	–	–
copy	32	34.4	15.1	10.5
entity	120	14.2	9.9	1.2
word	818	4.5	7.4	2.2
topic	191	1.6	0.1	0.2
all	1,953	11.3	11.3	4.8

Table 26: % of features where no rollout clears τ, by blind-labelled category; features that never fire in the corpus are excluded. Right: best-of-8 on the 512 set, from the direction (MAEM) or from 16 corpus windows (LLM); –: $n < 5$.

J.4 Training on the failures makes things worse

We tried three ways of teaching the MAEM the features it misses: (A) supervised training on the failing features' own activating corpus spans; (B) the same restricted to clustered failure types, holding out unseen clusters; and (B+replay) the same with a replay mixture of features the MAEM already handled. All three fix a handful and break an order of magnitude more.

Arm	Draw	% unexpressed before	after	fixed	broken
A	uniform	11.6	25.5	26	304
A	rarity-weighted	34.1	65.2	25	648
B	uniform	11.6	39.6	21	581
B + replay	uniform	11.6	31.5	19	417

Table 27: Counts are features, best-of-8, same evaluation before and after. Replay halves the damage without improving the rescue.

The B arm separates the two effects. On the 195 features it trained on, the unexpressed share moves $87.7\% \to 90.8\%$; on 49 held-out features of unseen cluster types, $89.8\% \to 98.0\%$. Meanwhile the 1,754 features never involved in the clustering at all go from 0.9% to 32.3% unexpressed, with median activation falling from 89% to 30% of corpus peak. Training on this class does not teach it and does cost the general capability; replay reduces the collateral damage ($0.9\% \to 22.9\%$) but rescues no more.

J.5 THE FEATURES ARE REACHABLE FROM THEIR CORPUS WINDOWS

Given 16 of a feature's corpus windows and no direction, an LLM writes text that fires it more often than the MAEM does, and the whole of its advantage sits in the rare tail.

% of features fired, best-of-8	rarest	2nd	3rd	densest
MAEM (from the direction)	62.4	96.7	98.3	98.6
LLM (from 16 corpus windows)	**84.5**	**100.0**	99.2	97.6

Table 28: 492 features, rarity strata. Pooled, the MAEM fires 88.8% and the LLM 95.3%. At best-of-1 the two are indistinguishable (83.3% vs. 83.1%): the LLM's advantage is that its samples keep finding new ways in, not that any single sample is better.

J.6 FAILING ROLLOUTS ARE SLIGHTLY LESS SELF-CONSISTENT

On 512 features with 8 rollouts each, samples for failing features are marginally more varied than for passing ones — median pairwise cosine 0.769 against 0.804 ($p = 1.9 \times 10^{-5}$), token Jaccard 0.034 against 0.048 ($p = 9.0 \times 10^{-5}$). The larger and more interpretable gap is against the corpus: failing features' rollouts sit at cosine 0.513 from their own corpus windows against 0.594 for passing ones ($p = 5.8 \times 10^{-20}$). The MAEM is not producing noise on features it cannot fire; it is producing coherent text about something adjacent.

J.7 OUTPUT DIVERSITY: MEASUREMENT

Figure 10 scores texts by the mean over all pairs of a feature's texts of the Jaccard similarity of their token-trigram sets. Texts are re-tokenized with the Qwen3.6-27B tokenizer without special tokens, so a rollout's trailing end-of-sequence token does not count, and every source is tokenized the same way. The cross-feature floor is the same score over 20,000 random pairs of texts written for different features.

Sources. All on a random sample of 2,000 features of the 131k SAE. The MAEM is the paper's checkpoint with 8 rollouts per feature under vLLM and, as an engine check, 4 under HF: both give median 0.043. The corpus source is each feature's 32 stored max-activating windows; the LLM is claude-opus-5, 8 texts on 159 features. The rare-feature LoRA was trained for one epoch on 64,432 spans from the rarest band alone. It is an earlier run than section J.4's, and none of its 4,027 training features is in this set. We read its HF rollouts, from which its scores are also computed. The two later arms read the same way: the rare-weighted arm trained on a separate rarity-weighted draw, none of whose 7,963 features is in this set; the cluster arm trained on 195 of this set's failures, which are dropped. The NLA baseline was run on rows 1500–1999 of the set, 8 explanations each at its native 200-token budget (99% parse as explanations); those rows are a representative slice (114–133 features per density quartile). Its `<explanation>` tags are stripped before embedding. Its vLLM rollouts are excluded: that path attached the adapter to the untrained base, so they come from a model nobody trained.

Estimator. The all-pairs mean is unbiased for the expected pairwise score at any number of texts k. Sources with 4, 8 and 32 texts per feature therefore share a centre, and smaller k only widens the spread. The distribution over pairs is heavy-tailed, most visibly for the corpus: 73% of its pairs share no trigram, and a few overlapping windows from one document carry its mean. We report the share of disjoint pairs beside the mean for that reason.

Embedding measures. Figure 9 and table 29 embed every text, unit-normed, with bge-small-en-v1.5 (the embedder already used by the evaluation pipeline) and, as a check, with all-mpnet-base-v2, a different architecture and training set. Every source is read at exactly $k = 8$ texts per feature, because unlike the Jaccard mean the Vendi score grows with k: the MAEM's first 8 vLLM rollouts, the rare-feature LoRA's 8 HF rollouts, the LLM's 8 texts, and for the corpus the feature's top-8 windows, the same windows the LLM was prompted with. The Vendi score is $\exp\left(-\sum_i \lambda_i \log \lambda_i\right)$

over the eigenvalues λ_i of K/k, where K is the $k \times k$ cosine kernel. Specificity is the median within-feature mean pairwise cosine minus the mean cosine of 20,000 random pairs of texts written for different features. Coverage targets are the feature's corpus windows 9–32, dropping any that share ≥ 0.2 trigram Jaccard with one of windows 1–8 (stored windows can overlap within a document); a median of 19 remain. Coverage is the mean over targets of the cosine to the nearest of the source's eight texts, minus the same with a random other feature's texts. The two embedders' cosines live on different scales (mpnet's cross-feature mean is 0.04–0.07, bge-small's 0.45–0.49), so only orderings are compared across them.

	bge-small			all-mpnet-base-v2		
Source	Vendi	specificity	coverage	Vendi	specificity	coverage
MAEM	2.19	**0.31**	0.108	3.10	**0.58**	0.205
corpus top-8	2.87	0.19	**0.143**	4.45	0.34	**0.287**
LLM (opus-5, $n = 159$)	3.39	0.19	0.140	5.74	0.30	0.255
NLA ($n = 500$)	**1.99**	0.17	0.055	**3.00**	0.37	0.105
rare-feature LoRA	4.17	0.05	0.075	7.26	0.08	0.145
rare-weighted arm	4.24	0.04	0.070	7.37	0.07	0.142
cluster arm ($n = 1{,}805$)	4.42	0.02	0.051	7.58	0.03	0.112
MAEM, texts cut to 32 tokens	2.22	0.30	—	3.16	0.57	—

Table 29: Diversity in meaning, both embedders; medians over features, $k = 8$. Bold marks the extreme of each column among the sources above the first rule: the MAEM is the most specific, NLA and the MAEM the least spread, and the corpus top-8 covers the feature's other windows best. NLA's coverage is not comparable, since it writes descriptions of text rather than text like the corpus windows. Below the rule are the three rare-feature training runs, all read on features they were not trained on: each is spread only because it is barely specific. On the LLM's 159 features alone the MAEM's Vendi is 2.32 (bge-small) and its raw coverage 0.591 against the LLM's 0.664.

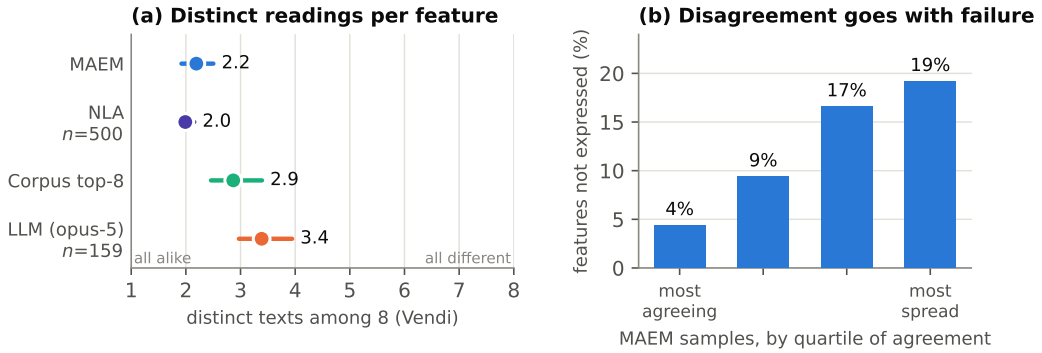

Figure 9: **(a)** Distinct texts among eight samples per feature (Vendi score over bge-small embeddings): median and interquartile range over the 131k-SAE 2k set. NLA covers 500 features, the LLM 159. **(b)** Share of features the MAEM fails to fire (below 10% of corpus peak, best-of-8), by quartile of agreement among its samples.

Robustness. The MAEM's median is flat across the set's four quartiles of pool peak activation (0.039–0.046), so the score does not track how strongly a feature fires. The quartiles in the right panel are formed by rank, because the rare-feature LoRA's scores tie heavily near zero. Length checks for the Jaccard score cut every text to its first 32 tokens, and to tokens 8–40 to remove the MAEM's shared opening words; for the embeddings, generated texts are cut to their first 32 tokens and corpus windows to 32 tokens around their marked peak, since only 56% of peaks fall in a window's last 32 tokens and a blind cut would drop them.

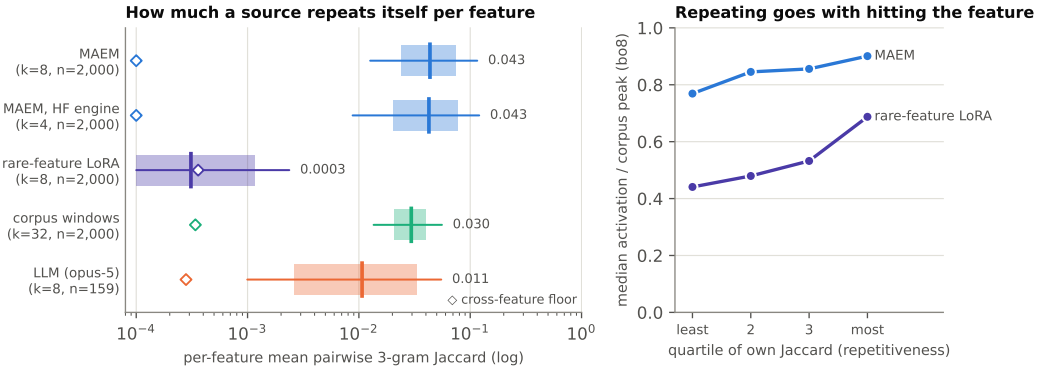

Figure 10: Diversity in wording. **Left:** per-feature mean pairwise token-3-gram Jaccard: box is the IQR, whiskers the 10th–90th percentile, tick and label the median; k is texts per feature. The diamond is the same source's cross-feature floor (the MAEM's, 5×10^{-5}, is drawn at the axis floor). The all-pairs mean is unbiased in k, so rows with $k = 4$ and $k = 32$ share a centre and differ only in spread. Scores are not length-matched; the length checks are described above. **Right:** median best-of-8 activation, as % of corpus peak, by quartile of each MAEM's own Jaccard, 500 features per point.