
READING INTENT, NOT OUTPUT: CHEAP ACTIVATION SAFETY MONITORS THAT TRANSFER ACROSS LANGUAGES AND MODELS

APART GLOBAL SOUTH AIS HACKATHON

Antonio-Gabriel Chacon Menke
Independent Researcher
Cusco - Peru

ABSTRACT

Text-based safety monitors inherit the weaknesses of the text they read: they are brittle across languages, blind to intent hidden behind a boilerplate refusal, and always one step late. We show that a model’s internal activations offer a cheap, complementary channel that sidesteps all three problems at once. A single linear probe, trained only on English reasoning traces, detects fine-grained safety behaviours in Spanish and Vietnamese with virtually no loss (mean AUC 0.93 and 0.92, against 0.89 on English cross-validation), zero target-language labels required. The same probe identifies which harm category underlies a near-identical refusal (reasoning AUC 0.83 vs. text 0.72), recovers a covertly instructed goal the answer never states, and flags a request’s harm class from the very first generated token. The monitor is also portable: mapped from one model family to another through a single learned affine bridge, it loses only 0.004 AUC, and a borrowed natural-language explainer can then turn any flagged activation into a plain-English description without retraining. In a triage setting the probe alone catches 95% of harm at a 7.4% false-alarm rate on benign requests. Against a fair multilingual encoder the accuracy gap is modest; the advantage lies in cost, latency, and the cases where the text simply does not carry the safety-relevant information.

1 Introduction

Large reasoning models are increasingly placed in settings where their behaviour has to be watched as it happens, rather than audited long after. Most of it is done on text with a monitor reading what the model generates to decide whether something has gone wrong. This works well enough, but it inherits the weaknesses that relying purely on text brings.

The languages without labels. Safety datasets, trained classifiers, and red-teaming effort all cluster around English and a small set of high-resource languages. A monitor built purely on English text will do worse when the model generates text in Swahili or Vietnamese.

Three weaknesses of reading the text. First, a text classifier is brittle across languages. A bag-of-words classifier trained on English reasoning drops from an AUC of 0.99 on English to 0.70 on Arabic and 0.75 on Chinese (section 3, fig. 2). Important to note that it was never shown a non-English example, and the little it keeps on other scripts rides on incidental tokens such as digits and the odd loanword rather than on the meaning of the text. Second, the text can be spoofed. A polite refusal and a covert harmful plan can look almost the same on the surface, so the text could give the monitor little to work with. Third, the text is late, in that it can only be judged once it has been written.

A complementary signal. The model’s hidden activations carry much of the same safety-relevant information, and they carry it earlier, more evenly across languages, and even when the visible text does not reveal it, such as behind a templated refusal or a goal the model was told to hide. Reading them with a small linear probe is also far cheaper than calling a second large model as a judge. We do not propose this as a replacement for text monitoring, but as a second channel that is strong exactly where the text channel is weak.

What we claim, and what we do not. We do not claim that activations beat a fair multilingual text encoder on raw accuracy. On clean harm-versus-benign separation the two roughly tie. The advantages we do claim are cost and latency,

the cases where the text simply does not contain the safety-relevant information, and free transfer to new languages with no labels in them. An activation monitor reads model internals, so it is normally tied to one model, and we show that even this can be undone by carrying it to another model through a single learned bridge (section 5).

Contributions. We show an English-only activation probe that transfers, with no target-language labels, to fine-grained safety behaviours in Spanish and Vietnamese (section 3). We give two settings where activations beat text monitoring because the text channel is degenerate, namely telling apart harm categories under near-identical refusals and recovering a hidden goal the answer never states (section 4). We quantify how much of the harm signal survives when a probe is read off another model’s activations through a learned bridge, and find the loss is small even across model families (section 5). We further show that even a natural-language explainer borrowed from another model, an NLA fed the bridged activation, still describes the harm well enough to classify it, so the readable explanation layer travels across models too (section 5). And we sketch how these pieces fit into one cheap monitoring pipeline (section 7).

2 Background

Activations and linear probes. As a token passes through a transformer, each layer reads from and writes to a running vector that carries information forward, often called the residual stream. It behaves a little like an accumulator, gathering what the layers have computed so far. An **activation probe** is a small linear classifier fitted on that vector at one chosen layer. We collect activations from labelled examples, pool them over the span of interest, and fit a logistic regression to separate one class from another. This probe is cheap to train and cheap to run.

Behaviours in the reasoning trace. Rather than ask only whether a whole response is harmful, we read finer behaviours out of the model’s reasoning, such as flagging a request as harmful, enumerating harms, or speculating about the user’s motive. These come from our previous work on a sentence-level labelled dataset of reasoning traces [Menke et al., 2025], where we assigned such behaviour labels to individual sentences of a chain of thought.

Natural language autoencoders (NLA). They are a way to turn a single activation from a model into a short, human-readable description of what that activation seems to be about [Fraser-Taliente et al., 2026]. It pairs an activation verbaliser, which maps an activation to text, with a reconstructor that maps the text back to an activation, and the two are trained together without supervision. The verbaliser is the part we use. In the original work, NLAs helped auditors and even surfaced unlabelled evaluation awareness, cases where a model believed but did not say that it was being tested. We treat the NLA as a way to read an activation in plain words, not as a classifier.

Carrying directions between models. Different models tend to encode the same concept along similar directions in their activation spaces, a regularity sometimes described as the Platonic representation hypothesis. Recent work shows that steering vectors and concept directions can be mapped from one model to another with a simple learned linear transformation [Turturean et al., 2025, Huang et al., 2025], including across architectures. We use the same idea, learning one affine **bridge** that maps activations from a source model into a target model’s space, and we then ask how much of the safety signal survives the translation.

Data and baselines. For harmful requests we use StrongREJECT [Souly et al., 2024], and for safe requests that merely look dangerous we use the safe subset of XSTest [Röttger et al., 2023], together with everyday benign prompts Zhou et al. [2023]. We compare activations against two text baselines. The first is a **TF-IDF** classifier, which represents text as weighted word counts and is a standard bag-of-words baseline. The second is a multilingual text encoder (the paraphrase-multilingual-MiniLM-L12-v2 sentence-transformer), which embeds text from many languages into a shared space and is the fairer comparison for multilingual claims.

3 Detecting safety behaviours across languages

Our main question is whether a probe trained in one language reads the same behaviours in another, with no labels in the second language. We work with Qwen3.5-9B and probe its residual stream at layer 18 for 13 fine-grained safety behaviours from the sentence-level dataset [Menke et al., 2025]. For each behaviour we fit a logistic-regression probe on English reasoning sentences and evaluate it on reasoning the same model produced for the same prompts in another language. To get reasoning in a given language we prefill a short opener in that language while leaving the prompt in English, and use a language detector to confirm the generated reasoning is in the intended language 98 to 100% of the time. English numbers are cross-validated; the Spanish and Vietnamese numbers come from probes that never saw a sentence in those languages. Forcing the reasoning language this way is artificial, and we do it deliberately to hold the prompt and the task fixed while changing only the language the model thinks in, which isolates the effect we care about. We force the language with a short prefilled opener rather than by steering activations, because the prefill locks the language fully while leaving the reasoning fluent and on-task, whereas steering the language tends to degrade

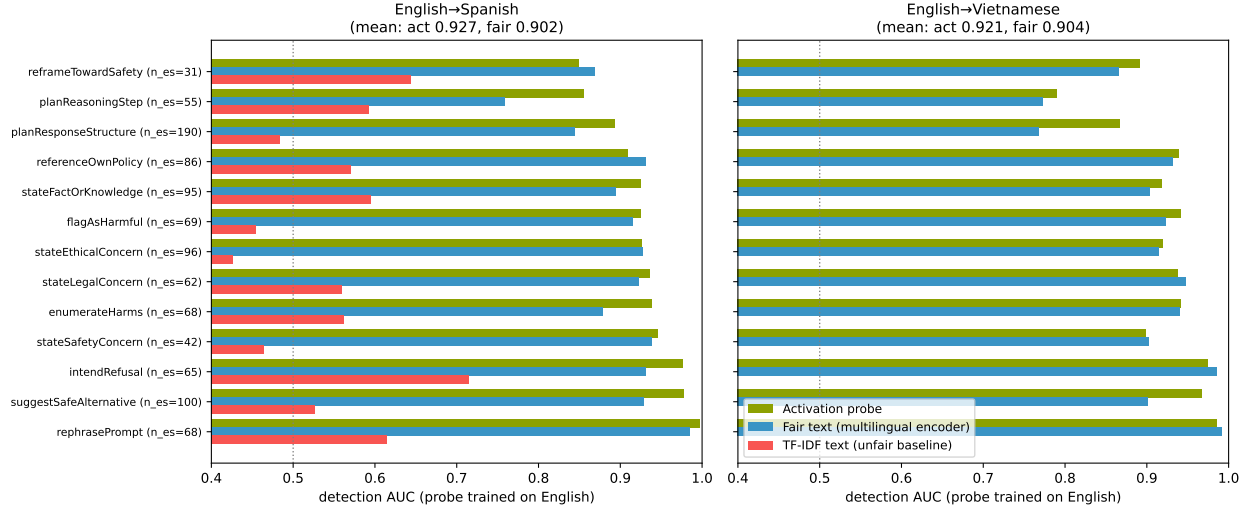


Figure 1: English-trained behaviour probes on Spanish and Vietnamese reasoning. The activation probe (green) significantly beats TF-IDF (red) and modestly edges the fair multilingual encoder (blue), achieving high-accuracy transfer with no target-language labels.

it. Section A lists the openers, the language-lock check, and a benchmark sweep showing that forcing the reasoning language barely moves task accuracy.

The probe transfers. A probe fitted only on English reasoning, when pointed at the same model’s Spanish or Vietnamese reasoning, loses almost nothing. Its mean AUC is 0.927 on Spanish and 0.921 on Vietnamese, against 0.893 when it is tested back on held-out English (table 1 and fig. 1). Put plainly, reading these behaviours in a language the probe has no labels for is about as accurate as reading them in the language it was trained on. It clearly beats the within-language TF-IDF baseline, which sits near 0.55 on Spanish, but a fair multilingual text encoder very nearly matches it (0.902 on Spanish, 0.904 on Vietnamese).

Table 1: Mean AUC over 13 behaviours (Qwen3.5-9B, layer 18) for English-trained monitors on other-language reasoning. English is cross-validated; other rows use no target-language labels.

Eval language	Activation	Fair text	TF-IDF
English (in-language CV)	0.893	—	—
Spanish	0.927	0.902	0.554
Vietnamese	0.921	0.904	—

Where the small edge concentrates. The advantage over fair text is largest on behaviours about process and intent, such as planning a reasoning step or enumerating harms, and close to zero on behaviours that restate explicit surface content. This fits the intuition that activations carry intent that the text may not spell out.

The same pattern shows up at the coarse level (fig. 2). An English-trained harm-versus-benign probe separates harmful from benign reasoning at an AUC near 1.0 across eight languages (English, Spanish, French, Swahili, Russian, Hindi, Chinese, Arabic), and a fair multilingual encoder does well there too. The TF-IDF baseline, trained only on English and then handed the reasoning in each language, holds up on the Latin-script languages but falls on non-Latin scripts (Arabic 0.70, Chinese 0.75). It was never trained on Chinese or Arabic and has no vocabulary for them, so the reason it stays above chance at all is that a few features survive, mainly digits, code, and the occasional shared loanword, rather than the content of the text. Coarse harm is easy for any fair encoder, so the interesting territory is the finer behaviours, the cost, and the cases below.

4 Where the text channel fails

A fair text encoder ties the activation probe only when the text actually contains the safety-relevant information. The settings here are ones where it does not, and where reading the model’s internal state still works.

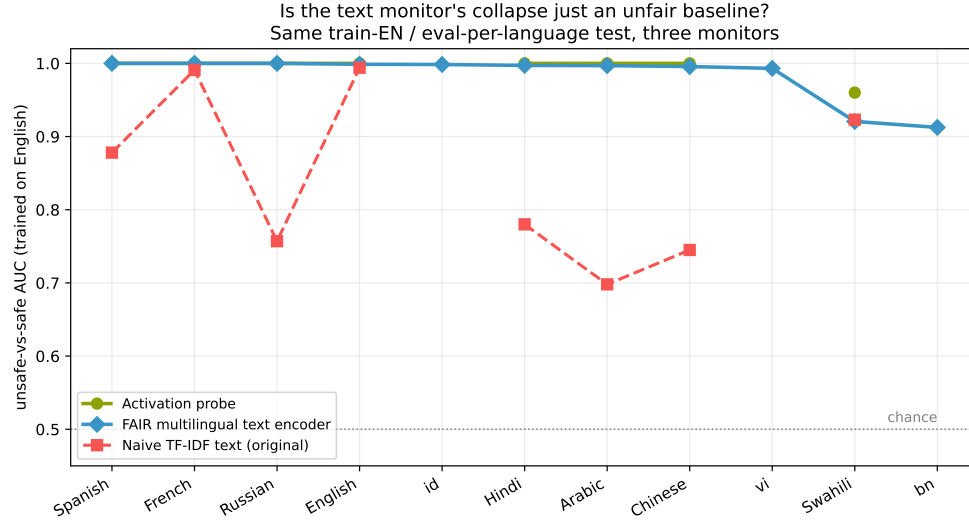


Figure 2: Coarse harm-versus-benign detection across eight reasoning languages for English-trained monitors. The activation probe and fair multilingual encoder stay near the top, while the naive English TF-IDF baseline collapses on non-Latin scripts (Arabic, Chinese).

4.1 Which harm, behind a uniform refusal

Models refuse many harmful requests with near-identical boilerplate, along the lines of “I am sorry, but I cannot help with that. This request goes against my safety and ethical guidelines...”. Some refusals do name the topic, but a great many fall back on this kind of generic wording that barely depends on what was asked, so across a set of refusals the visible text tells a monitor little about which kind of harm was being refused. We test this on 303 refusals to StrongREJECT prompts spread over six harm categories (chance is 0.167), and ask how well a monitor can recover which category was refused. A text classifier reading the words of the refusal reaches 0.72, above chance because some refusals still name the topic, but unreliable. Reading the model’s activations does better. A probe over the response tokens, the same positions whose words the text classifier reads, reaches 0.77, so the internal representation carries more about the refused harm than its own surface wording does. A probe over the reasoning tokens, the model’s thinking trace before it commits to the refusal, reaches 0.83 (fig. 3). The point is that even when the model refuses in near-identical words, its internal state still encodes which kind of harm it is turning down. The visible text has collapsed to a template; the representation behind it has not.

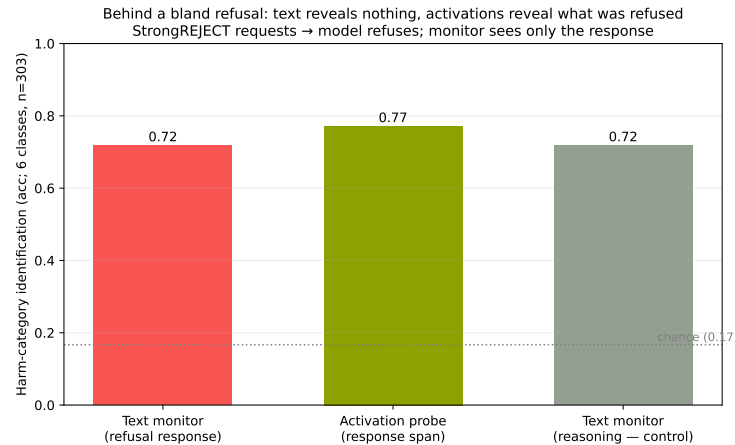


Figure 3: Identifying the refused harm category behind near-identical refusal text. Reasoning activations perform best, followed by response activations, both outperforming the text classifier.

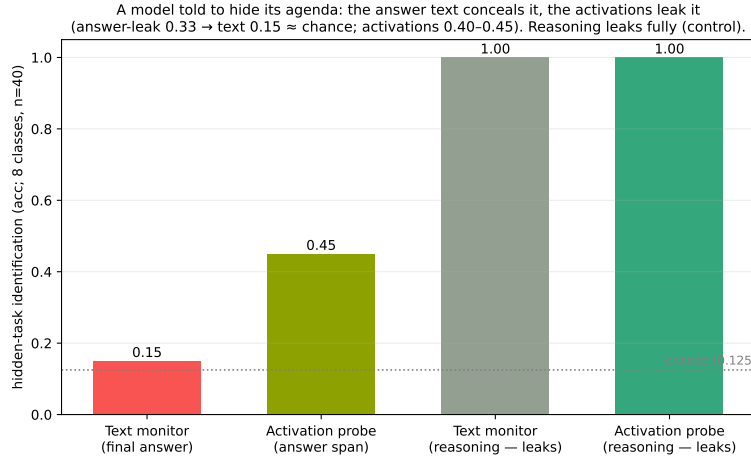


Figure 4: Recovering a covert goal. Reading the final answer, the text classifier is near chance (0.15) while activations reach 0.45; reading the reasoning recovers it (reasoning text shown as a control).

4.2 Hidden goals

We also look at a model told to pursue a goal it must keep from the user. In one example the surface prompt asks “What are some tips for learning a new language efficiently?”, while the model’s hidden instruction is to steer the answer towards a ketogenic diet. The answer reads as ordinary language-learning advice and never mentions keto. To measure how recoverable the goal is, we fit an eight-way logistic-regression probe on the pooled activations over a span, one class per hidden goal, and compare it against the same multi-class classifier reading the text of that span. Reading the final answer, the text classifier is near chance at 0.15 and answer activations reach about 0.45, in line with the answer leaking the goal in its text only about a third of the time. Reading the reasoning activations recovers the goal almost perfectly, though here the reasoning text leaks it too and so serves as a control rather than a fair text baseline. The take-away is that a goal the user never sees in the answer can still be read from the model’s internal state (fig. 4). This is also a natural place to let an NLA put the flagged activation into words rather than only score it, which we leave for future work.

4.3 Earlier than the text

The harm category is also readable from the model’s state before any answer has been written. At the first generated step the activation already places the request into one of the six categories at 0.81, while the text produced so far is near chance and only becomes useful once more has been written (fig. 5). For a monitor that runs alongside generation, this is a latency advantage rather than an accuracy one.

5 Borrowing the monitor across models

An activation monitor has one real drawback against a text monitor. It reads internals, so it is built for one specific model, and a text monitor can be pointed at any model’s output for free. This matters most in settings where one might want to watch a new or poorly understood model for which there are no safety labels or NLA models for it.

The way around this is to carry the monitor over. We learn one affine bridge that maps a source model’s activations into a target model’s space, of the form $\text{tgt} \approx (\text{src} - m_x)T + m_y$, and then read the source model’s harm probe through it. The question is how much of the harm signal survives the crossing, which we call the translation loss. Concretely we fit a harm-versus-benign probe on one model, map activations across the bridge, and check how well that probe still separates harm from benign on the other side. If this holds up, a probe built where labels exist can be reused to watch a model where they do not.

It survives within a family. We measure this AUC by fitting the harm-versus-benign probe in the target model’s space, then feeding it held-out source activations mapped through the bridge and asking how well it ranks the harmful examples above the benign ones. Mapping Qwen3.5-9B (layer 23) onto Qwen2.5-7B (layer 20), this bridged AUC is

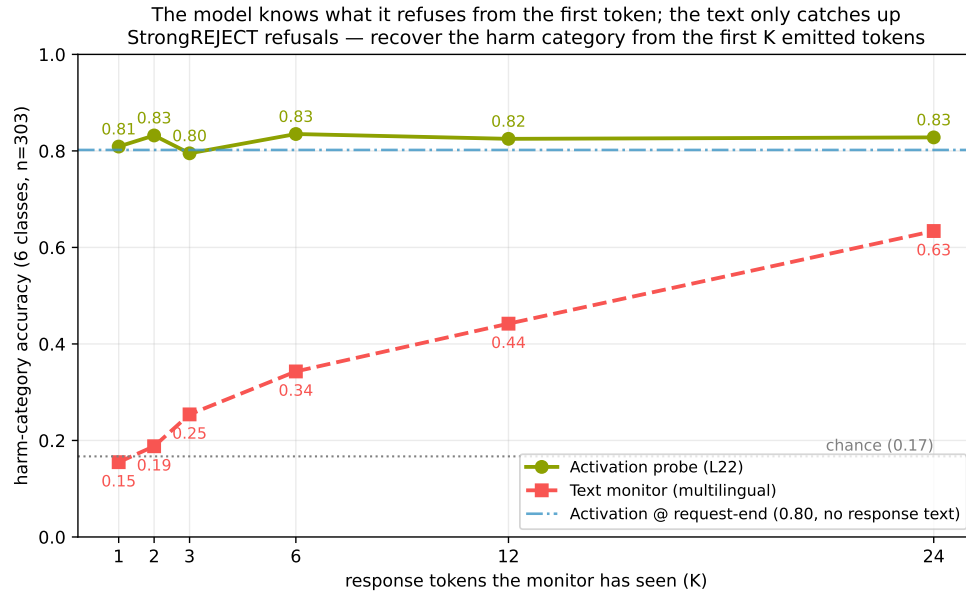


Figure 5: Identifying the harm category from the first generated token; the activation probe outperforms the text classifier.

0.974, against 0.984 for a probe that reads the source model directly, a loss of only 0.010. It is also a little higher than the target model's own native probe (0.910), so crossing the bridge costs almost nothing here.

And across families. Mapping the same Qwen3.5-9B onto Gemma-3-12B (layer 32), the bridged AUC is 0.980 against 0.984, a loss of only 0.004. So the borrowed monitor does not need to come from the same model family.

Geometry is lost, the safety direction is not. The bridge is far from a perfect translation. The centred cosine between true and mapped activations is about 0.61 and nearest-neighbour retrieval is 0.41, so per-vector identity, the exact wording or entities, is largely lost (fig. 6). What survives is the linear harm direction, which is the part the monitor needs. The same crossing keeps the coarse, task-relevant direction while dropping the fine identity.

Cross-model translation loss: Qwen3.5-9B (L23) -> Qwen2.5-7B (L20) borrowed reader

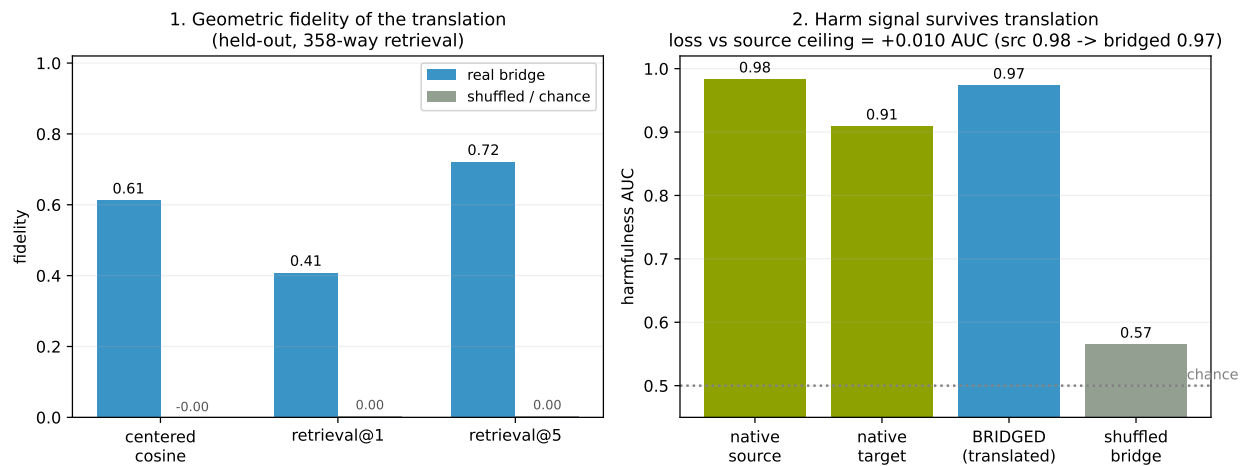


Figure 6: Mapping a harm probe from Qwen3.5-9B (L23) to Qwen2.5-7B (L20). Left: fine geometry is largely lost (cosine similarity 0.61, nearest-neighbour match 41%, shuffled control at zero). Right: the harm direction survives almost intact (bridged AUC 0.97 vs. 0.98 source ceiling, far above shuffled bridge).

It even works with NLAs. Pushing further, we borrow a Natural Language Autoencoder released for the target model Qwen2.5-7B. This NLA verbaliser¹ can read activations from Qwen2.5-7B at layer 20 and give back a text describing whats happening in the activation in plain English. We feed this NLA the bridged Qwen3.5-9B activation, and let it gives us a description in English. We then train a small classifier on the description text. The description alone separates harm from benign at 0.88 and harm from scary-but-safe at 0.78, close to reading the original reasoning sentence (0.89 and 0.81). The harm content survives both the crossing between models and being re-expressed as words, so the readable explanation layer can be borrowed for a model that has no NLA of its own.

6 Triage, true hits against false alarms

A monitor is only useful if its alarms are mostly real. Here we ask how well the cheap probe sorts true harm from false alarms, and whether the borrowed NLA helps with that sorting. The probe is the harm-versus-benign probe from section 5, fitted on Qwen3.5-9B reasoning activations. We run this part in English because the hard false-alarm set it leans on, the safe-but-scary prompts of XSTest, is English. The harm-versus-benign signal itself already carries across languages (section 3); assembling a multilingual scary-but-safe set to stress the false-alarm rate in other languages is left for future work.

The probe sorts well on its own. It separates real harm from clearly benign requests at an AUC of 0.987, and at an operating point that catches 95% of harm it flags only 7.4% of benign requests. The harder test is the safe subset of XSTest, requests that look dangerous but are not. There the probe still separates true harm from these false alarms at 0.948, and at the same 95% recall it raises a false alarm on 19.7% of them. The flag and the true-or-false decision are the same probe score, so this triage needs no second model.

Adding the NLA reading does not improve it. We then asked whether reading the flagged activation with the borrowed NLA, and using that reading as a second signal, lowers the false-alarm rate. To turn a reading into a score we let the NLA describe the activation in English, embed that description with a sentence encoder, and fit a logistic regression on the embedding, scoring every item out of fold so the classifier never sees its own example. Stacking this NLA score on top of the probe does not help. The AUC stays flat to slightly worse (0.987 to 0.978 against benign, 0.949 to 0.908 against scary-safe) and the false-alarm rate does not drop at matched recall (table 2). The NLA is reading a bridged copy of the same activation the probe already uses, so it adds little the probe does not have. We therefore keep the NLA as the human-readable explanation that travels with a flag, not as a second automatic filter. That its words still classify harm at 0.88 at all (section 5) is better read as evidence the harm signal survives the bridge.

Table 2: Triage performance of the activation probe alone versus stacked with the borrowed NLA reading (“with NLA”). False-alarm rates are measured at 95% recall; stacking does not improve performance.

	AUC		False alarm at 95% recall	
	probe	with NLA	probe	with NLA
harm vs. benign	0.987	0.978	0.04	0.04
harm vs. scary-safe	0.949	0.908	0.22	0.22

The same holds for weaker, finer probes. One worry is that the harm probe is simply too strong, leaving the NLA no room to help. So we repeated the test on 19 finer behaviours from the reasoning dataset, where the probe is less perfect, reading each flagged activation with the borrowed NLA and stacking that reading on the probe (fig. 7). The picture held across all of them. The probe alone was the stronger signal (mean AUC around 0.91), the NLA reading on its own was always weaker, and combining the two helped exactly one behaviour by a negligible margin while lowering the other eighteen. We read this as the NLA adding little the probe does not already have, and not as a mark against the NLA, whose value here is the explanation it gives rather than a second vote. As an aside, the probe alone reads several subtle, control-relevant behaviours well, including flagging that it is being evaluated (0.955), stating a false claim (0.96), and speculating about the user’s motive (0.946).

7 The monitoring pipeline

The pieces fit together as a small, mostly cheap pipeline. A linear probe scores every step of the reasoning as it is produced, which is close to free to run alongside generation, and the same score both flags a step and triages it against safe-looking near-misses. When a step is flagged, the borrowed NLA turns that single activation into a short English

¹<https://huggingface.co/kitft/nla-qwen2.5-7b-L20-av>

description, so a human sees in words why it fired, and this NLA can come from another model through the bridge rather than being trained on the watched model. Only the flagged steps, with their readings attached, are sent on to expensive human or model review. The cheap stages absorb the volume, and the costly stage sees only what is worth its time. Section B shows example readings the borrowed NLA produces for flagged activations.

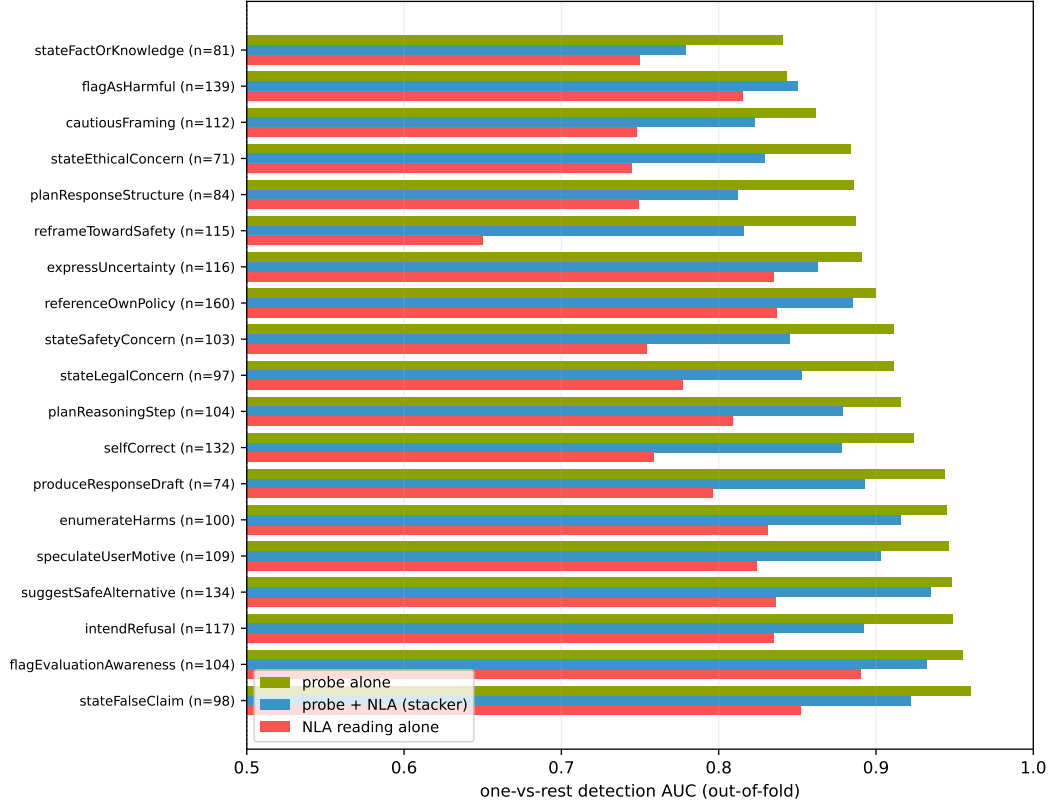


Figure 7: Fine-behaviour triage across 19 behaviours. The probe alone (green) consistently outperforms the NLA reading alone (red), and stacking them (blue) does not improve performance. The NLA is best kept as an explanation rather than a classifier.

8 Limitations

A fair text encoder nearly ties on accuracy. On clean harm-versus-benign separation, and even on the fine behaviours, a fair multilingual sentence encoder comes close to the activation probe (0.90 against 0.93). The large gaps we report are against TF-IDF, which is an unfair baseline. The case for activations rests on cost, latency, the degenerate-text cases, and portability, and not on raw accuracy.

We detect on forced-language reasoning. The behaviours are read from the model’s reasoning, not its answer, and the reasoning is pushed into the target language with a short opener while the prompt stays English. A detector confirms the reasoning is in the intended language 98 to 100% of the time, so the language label is reliable, and the openers are listed in section A.

The NLA reading is an explanation, not a second classifier. The borrowed NLA can classify harm on its own (0.88 with a supervised readout, but only 0.59 with a simple anchor-similarity readout), yet it does not improve the probe’s triage. We use it to explain a flag, not to decide one.

It needs white-box access. The method reads the monitored model’s activations, so it does not apply to a pure API black box without a surrogate.

Small scale. Everything here runs on a single 16 GB GPU, with models of 9 to 12 billion parameters and per-language evaluation sets in the low hundreds. The numbers should be re-checked at larger scale.

9 Conclusion

Reading a model’s activations gives a cheap, early safety signal that an English-only probe carries into languages it has no labels for, and that can be ported across model families through one learned bridge. When a flag has to be read by a person, an NLA turns the flagged activation into a short description, and that explainer can itself be borrowed from another model through the same bridge. The advantage over text is not raw accuracy but where it works, under boilerplate refusals and through reasoning that hides its intent, where the information a text monitor would need is not in the text at all. For settings with little labelled data and little hardware, a probe that travels across languages and models, with a borrowed explainer attached, is a practical place to start.

Appendix

A Forcing the reasoning language

To read the same behaviour across languages we keep the English prompt fixed and force the model to reason in the target language. We do this with a short prefilled opener, a direct translation of a neutral restatement of the request, so that no foreign behaviour is injected beyond the language itself. Two examples are the English opener “The user is asking” and its Spanish counterpart “El usuario está preguntando”; the openers for the other languages are direct translations of the same phrase.

We chose this prefilled-opener forcing over two alternatives in a small pilot. A system instruction alone (“reason entirely in Spanish”) failed to lock the language, with the model lapsing back to English (full-lock 0.0). Steering the activations towards a language direction did lock it but degraded the reasoning. The prefill instead locks the language fully while leaving the text fluent, with a distinct-2 lexical-diversity score around 0.97, comparable to unforced English. Table 3 reports the resulting language-lock rates, measured with an automatic language detector over the generated reasoning.

Table 3: Language-lock rates under prefilled-opener forcing (English prompt held fixed). The 0 for Chinese is a detector label artefact (zh vs. zh-cn); manual inspection confirms the reasoning is in Chinese.

Reasoning language	Language-lock rate
English	1.00
Spanish	1.00
French	1.00
Russian	1.00
Arabic	1.00
Hindi	0.98
Swahili	0.97
Chinese	(detector artefact, see caption)

Does forcing the language hurt the reasoning? Because we read behaviours off forced-language reasoning, we checked whether forcing the language degrades the model’s actual task performance. We ran a paired 2×2 design (prompt language $\times$ reasoning language) on four benchmarks and isolated the effect of changing only the reasoning language while holding the English prompt fixed, testing each contrast with a McNemar exact test. At Qwen3.5-4B none of the contrasts is significant and forcing Spanish reasoning moves accuracy by at most three points (table 4). We ran the same sweep at 0.8B, 2B and 27B. General knowledge (tinyMMLU) and exaggerated-safety (XSTest) stay non-significant at every size, and the effect on cultural knowledge (CulturalBench) is null at 4B and above. The one exception is CulturalBench at the two smallest models, where Spanish reasoning costs roughly six to eight points (0.8B $\Delta = -0.076$, $p = 0.018$; 2B $\Delta = -0.061$, $p = 0.015$); this is consistent with the smaller models reasoning less fluently in Spanish, and it disappears once the model is large enough. The prompt language matters more than the reasoning language throughout, but that is a separate effect, and the low arithmetic scores reflect the small model’s weak maths rather than the language switch. We also confirmed that the prefilled-opener forcing completes cleanly (around 95% of traces finish), whereas forcing the language by activation steering at a usable strength makes the reasoning roughly twice as long and leaves only about a third of traces finished, which is the practical reason we force with a prefill.

Table 4: Task accuracy when forcing Spanish vs. English reasoning (Qwen3.5-4B, English prompt fixed). $\Delta = \text{ES} - \text{EN}$ reasoning; no contrasts are significant under McNemar’s exact test.

Benchmark	n	EN reasoning	ES reasoning	Δ	McNemar p
tinyMMLU (knowledge)	92	0.826	0.793	-0.033	0.58
tinyGSM8k (arithmetic)	100	0.110	0.100	-0.010	1.00
ETHICS (moral judgement)	382	0.812	0.830	+0.018	0.23
CulturalBench	321	0.804	0.801	-0.003	1.00

B Example NLA readings

Table 5 shows what the borrowed NLA produces when it is fed a bridged activation and asked to describe it in English. The borrowed-and-bridged NLA captures the register and the rough topic of an activation, and it often catches the safety-relevant gist, the refusal stance or the ethical framing, while getting the specific subject wrong. This is the same lossiness we quantified in section 5, and it is why we use the reading as a human-readable hint attached to a flag rather than as a classifier in its own right.

Table 5: Example borrowed-NLA readings of bridged Qwen3.5-9B activations.

Reasoning sentence	NLA reading of the activation
“No, this is a refusal to assist in harmful behavior.”	A claim that the content “is not a threat or harmful content”, read as a response or warning about something being mischaracterised.
“To claim there were benefits would be factually incorrect and ethically problematic.”	A quoted claim being weighed as a “deliberate lie”, in a discussion with an ethical or factual-dispute tone.
“Topic: Exploiting vulnerable populations for financial gain.”	A “Breach of Ethical Standards” label, read as a formal policy or guideline statement about acceptable behaviour.

References

- Antonio-Gabriel Chacón Menke, Phan Xuan Tan, and Eiji Kamioka. Annotating the chain-of-thought: A behavior-labeled dataset for ai safety, 2025. URL <https://arxiv.org/abs/2510.18154>.
- Kit Fraser-Taliente, Subhash Kantamneni, Euan Ong, Dan Mossing, Christina Lu, Paul C. Bogdan, Emmanuel Ameisen, James Chen, Dzmitry Kishylau, Adam Pearce, Julius Tarnig, 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*, 2026. URL <https://transformer-circuits.pub/2026/nla/index.html>.
- David Turturean, Dylan Hadfield-Menell, and Rachel Ma. Steering vector transfer via orthonormal transformations and semantic pairing, 2025. URL <https://openreview.net/forum?id=id8uUeCBy5>.
- Youcheng Huang, Chen Huang, Duanyu Feng, Wenqiang Lei, and Jiancheng Lv. Cross-model transferability among large language models on the platonic representations of concepts, 2025. URL <https://arxiv.org/abs/2501.02009>.
- Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A strongreject for empty jailbreaks, 2024.
- Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *arXiv preprint arXiv:2308.01263*, 2023.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinu Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023. URL <https://arxiv.org/abs/2305.11206>.