

Nemotron activation extraction

Antonio Lobo-Santos, DFKI

Version 1.0.0, 15 October 2026

Activations describe a model’s internal state as it processes tokens. A hook records these activations at a chosen location, called a site. `nemotron-extraction` provides six hook types for the official [Nemotron 3 Nano BF16 checkpoint](#). We compare two backends: Hugging Face Transformers, which runs the model with PyTorch, and vLLM, an inference engine that manages model execution. The comparison tests whether their activations can be used interchangeably for sparse autoencoder training.

vLLM runs the model in worker processes on the graphics processor (GPU). The driver installs hooks there through `collective_rpc`; workers preserve prompt order and save captures through pinned host buffers. A two-batch queue allows transfers and writes to overlap inference.

The backends produce similar activations, but only 3/18 sites per language meet all agreement thresholds. Interchangeability therefore remains unvalidated. To reach this conclusion, we supplied identical bfloat16 (BF16) weights and frozen token IDs. Both used checkpoint revision `fe4143e388795a58a92974891b2e45092a53dc1d`. The validation sample used 16,384 tokens per language, batches of eight 2,048-token sequences, and 18 sites: first, middle and last eligible layer for each hook type. English came from FineWeb-Edu; German and Spanish from FineWeb2.

Cosine measures whether activation vectors point in the same direction, with 1 indicating alignment. To compare errors across activation scales, we divide both backends’ tensors by the reference root mean square at each site before computing mean squared error (MSE). Hugging Face Transformers is the reference, so normalised MSE is $\sum(v - h)^2 / \sum h^2$, where h and v denote the reference and vLLM activations, summed over tokens and dimensions. This dimensionless measure equals relative L2 squared. Using a shared scale preserves magnitude differences. Zero means identical values. The table gives means and population standard deviations (SD) across 18 equally weighted sites, describing variation between sites rather than uncertainty in the mean.

Language	Mean cosine $\pm$ SD	Normalised MSE $\pm$ SD
English	0.999002 $\pm$ 0.001144	0.001879 $\pm$ 0.002171
German	0.998623 $\pm$ 0.001719	0.002650 $\pm$ 0.003230
Spanish	0.998701 $\pm$ 0.001619	0.002435 $\pm$ 0.002989

High average cosine therefore does not establish agreement. Each site must meet mean cosine ≥ 0.9999 , minimum token cosine ≥ 0.9990 , and relative L2 ≤ 0.02 (normalised MSE ≤ 0.0004). These criteria are unchanged. Normalised MSE was recomputed from saved sums without rerunning inference. The table describes this validation sample, while the benchmark defaults to one million tokens per language. The cause of the discrepancy remains unresolved.

To narrow the cause, we compared the package hooks with the original extraction hooks on host-processor test models. All six hook types returned bitwise-identical 32-bit floating-point and BF16 tensors. Higher-precision calculations changed mean cosine by less than 6.6×10^{-7} on retained samples. GPU correctness remains unverified; different computational kernels and mixture-of-experts routing are possible contributors, not confirmed explanations.

Both backends ran eagerly in BF16 on an H100 80GB. Hugging Face Transformers used PyTorch 2.8.0a0, CUDA 12.9, Transformers 4.57.6, Mamba-SSM 2.3.2.post1 and causal-conv1d 1.7.0; vLLM used version 0.27.1, PyTorch 2.14.0a0, CUDA 13.4 and Transformers 5.14.1. Thus, identical weights and inputs did not imply identical numerical kernels.

Reproduction. Build and install with `uv build` and `uv pip install`; follow the README’s single benchmark command to prepare frozen inputs and compare both backends across English, German and Spanish. The script saves JSON results and Markdown reports. Compile this report with `pdflatex report.tex`. Contribution instructions and BibTeX metadata for Antonio Lobo-Santos, DFKI, are provided in the README and `CITATION.bib`.