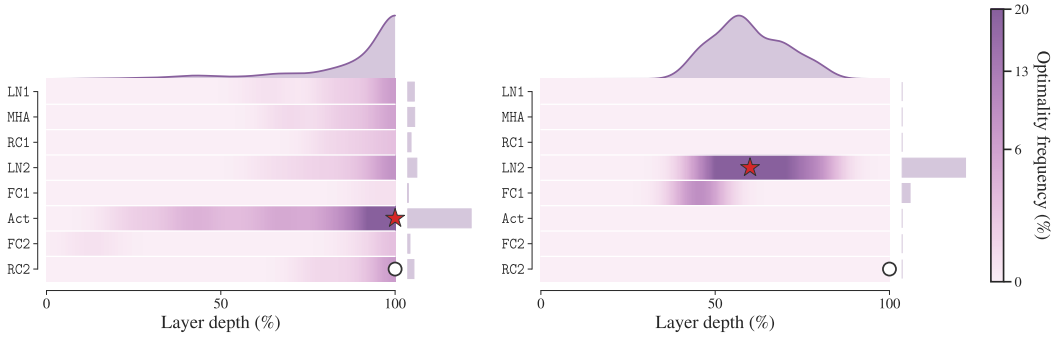
PROBING TRANSFORMER REPRESENTATIONS LAYER BY LAYER, MODULE BY MODULE

Anonymous authors
Paper under double-blind review

ABSTRACT

Which representation should be probed in a transformer-based foundation model? Recent work shows that the standard practice of probing the final layer is often suboptimal, as intermediate layers can provide better features. Yet, representations are still largely studied along depth alone, treating each transformer block as an atomic unit. In this work, we analyze transformer representations *layer by layer, module by module*. Through more than $144\,000$ linear probing evaluations of vision foundation models (VFMs) and large language models (LLMs), we show that layer and module choices should be considered jointly: the best layer-module pair yields substantially larger gains over the final-layer residual stream than the best layer with the residual stream fixed. Moreover, module performance can vary substantially across layers, showing that useful representations cannot be characterized by depth alone. Building on these findings, we evaluate simple fixed probing strategies: the final-layer feedforward activation (`Act@100`) for vision and the second normalization layer at 60% depth (`LN2@60`) for language. These strategies improve over the probing baseline in a majority of evaluations and recover, on average, 42% and 86% of the gains from probing the best layer-module pair for vision and language, respectively. For VFMs, these gains increase with the transfer difficulty between pretraining and downstream data.



Vision Foundation Models Large Language Models

Figure 1: **Layer by layer, module by module.** Heatmaps showing the optimality frequency of each layer–module pair, i.e., how often it achieves the highest probing accuracy, with marginal distributions of optimal layers (top) and modules (right). For both vision foundation models (**left**) and large language models (**right**), performance depends jointly on layer and module choices. The probing baseline on the final-layer residual stream (○) is largely suboptimal, while the fixed strategies `Act@100` and `LN2@60`, for vision and language (⋆), respectively, often achieve the best performance.

1 INTRODUCTION

Foundation models, which rely primarily on the transformer architecture (Vaswani et al., 2017), have achieved impressive performance in a wide range of areas such as natural language processing (NLP, Brown et al., 2020; Touvron et al., 2023), computer vision (Siméoni et al., 2025), time series forecasting (Ilbert et al., 2024; Nie et al., 2023), and mathematical reasoning (Comanici et al., 2025).

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.

These models are typically pretrained on massive amounts of diverse data (Du et al., 2025; Shukor et al., 2025) to encode general knowledge in their weights. A common use of foundation models is as frozen feature extractors, which provide rich representations for downstream tasks without retraining or finetuning the model (Lee et al., 2025; Oquab et al., 2024; Roschmann et al., 2026).

Probing intermediate representations. A standard way to evaluate representations is through linear probing: the model is frozen, and a linear classifier is trained on top of the representation. Alain & Bengio (2017) introduced probing as a diagnostic tool for neural networks and observed that classification accuracy increases with depth. Their experiments, however, focused on supervised, task-specific convolutional neural networks (CNN, LeCun et al., 1998). While the final layer may be optimal for a model trained for a single task, it need not be for a model trained to generalize across tasks. Recent work shows that intermediate layers of LLMs can outperform the final layer across diverse downstream tasks (Skean et al., 2025). For VFMs, however, the picture is less clear. Skean et al. (2025) report that intermediate layers are beneficial in autoregressive VFMs (El-Nouby et al., 2024; Fini et al., 2025), while final representations remain optimal in non-autoregressive models such as ViT (Dosovitskiy et al., 2021) and DINOv2 (Oquab et al., 2024). In contrast, Bolya et al. (2025) find that intermediate layers provide the best embeddings in contrastive VFMs, and Uselis & Oh (2025) observe a similar trend in ViT, particularly under distribution shift. This motivates further investigation.

Layer by layer, module by module. The layer-wise analyses discussed above share a common limitation: they all treat the transformer block as an atomic unit. This perspective overlooks the intermediate representations produced within each block, illustrated in Figure 2, namely the outputs of normalization layers, attention modules, feedforward linear layers, activation functions, and residual connections.[1] These components can exhibit distinct properties, notably in terms of smoothness and adaptability during finetuning (Odonnat et al., 2026). We therefore conduct a large-scale, systematic study of transformer representations in VFMs and LLMs, layer by layer, module by module. We ask two natural questions: *1) How much can optimal representations improve over standard probing?* and *2) Can a simple fixed heuristic recover most of the gain at no extra cost?*

Our contributions. We conduct extensive experiments on VFMs and LLMs. Figure 1 illustrates our findings, showcasing the suboptimality of probing the final-layer residual stream and the joint importance of both layer and module choices. Our main contributions can be summarized as:

1. **Layer and module choices matter jointly.** Probing the best layer-module pair yields substantially larger gains over the final-layer residual stream than probing the best layer with the residual stream fixed. Moreover, module performance can vary substantially across depth, showing that layer and module choices should not be considered independently.
2. **Practical benefits.** We evaluate simple fixed probing strategies: final-layer feedforward activation (`Act@100`) for VFMs and the second normalization layer at 60% depth (`LN2@60`) for LLMs. Both strategies improve performance in a majority of evaluations and recover a substantial fraction of the gains from probing the best layer-module pair.
3. **Impact of transfer difficulty.** When the discrepancy between pretraining and downstream data is large, transfer becomes more difficult. For VFMs, we find that the benefits of optimal probing locations increase in such settings, with `Act@100` yielding larger gains. Moreover, the impact of depth grows under distribution shift, allowing further improvement through the best layer-wise `Act` representations.

2 Probing transformers layer by layer, module by module

The transformer architecture (Vaswani et al., 2017) consists of successive transformer blocks, each taking as input a sequence of tokens $(x_1, \dots, x_n)$. As shown in Figure 2 (left), each block is made of alternating multi-head attention modules (`MHA`), with self-attention for encoders and causal attention for decoders, and feedforward networks (FFN). A standard FFN consists of two linear transformations,

[1] In what follows, we use the terms block and layer interchangeably, as well as module and component, referring to normalization layers, multi-head attention modules, feedforward layers, activation functions, and residual connections throughout the network.

denoted FC1 and FC2, separated by a non-linear activation function Act, such as GELU (Hendrycks & Gimpel, 2016). Assuming the pre-norm standard (Xiong et al., 2020), two LayerNorms (Ba et al., 2016), respectively LN1 and LN2, precede the attention and feedforward blocks, and two residual connections, respectively RC1 and RC2, follow them. We will refer to RC2 as the residual stream, i.e., the output of the transformer block. In this work, we track the outputs of these eight sequence-to-sequence operations at each layer, indexing them by a layer-module pair (ℓ, m). Some large-scale transformers replace the standard two-layer FFN with a gated FFN. It combines the non-linear projection with an additional linear projection through element-wise multiplication. To ensure consistency between standard and gated FFNs, we recover the FFN representations as follows: FC1 is the linear layer fed as input to the non-linear operation σ, Act is the non-linear transformation with a gated mechanism (thus absorbing the additional linear projection), fed as input to the second linear function, and FC2 is this second linear projection. This is illustrated in Figure 2, with exact tensor definitions deferred to Appendix B.1.

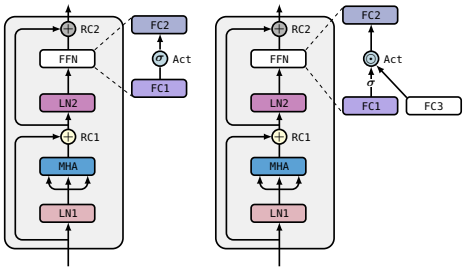


Figure 2: **Transformer representations.** We probe transformer representations (colored) across layers in VFMs and LLMs, covering models with standard (**left**) and gated FFN (**right**).

Evaluating transformer representations. Following Oquab et al. (2024) and Muennighoff et al. (2023), we evaluate transformer representations via linear probing for classification (Alain & Bengio, 2017), which provides a common evaluation for visual (Chen et al., 2020) and textual (Lee et al., 2025) embeddings. Each probing location (ℓ, m) produces a sequence of tokens $(x_1, \ldots, x_n) \in (\mathbb{R}^d)^n$, where ℓ denotes the relative depth expressed as a percentage[2], and m is the transformer module being probed. By abuse of notation, d denotes either the embedding dimension or the FFN hidden dimension, which is generally larger. As in Joseph et al. (2026), we embed this sequence in $\mathbb{R}^d$ by global average pooling over content tokens, namely image patches or text tokens; that is, special tokens, including CLS, EOS, and BOS, as well as padding positions (PAD), are masked before pooling and do not contribute to the representation. This allows us to focus on representations produced within transformer blocks, with a consistent feature extraction procedure across models (e.g., AIMv2 models (Fini et al., 2025) do not use a CLS token). Implementation details to ensure a scalable evaluation across layers and modules are provided in Appendix B.

Measuring improvement with oracle gains. The performance of the representation at location (ℓ, m) is the probing test accuracy on a given dataset or benchmark, denoted by $\mathrm{acc}(\ell, m)$. We use the final-layer residual stream, i.e., RC2 at 100% relative depth, as the probing baseline, denoted by RC2@100. We define the gain at location (ℓ, m) as the accuracy difference in percentage points (pp) relative to the baseline, i.e., $g(\ell, m) = \mathrm{acc}(\ell, m) - \mathrm{acc}(100, \mathrm{RC2})$. To disentangle the impact of depth, module, and their interaction, we analyze the following oracle gains:

$$\mathscr{G}_{\mathrm{layer}} = \max_{\ell} \; g(\ell, \mathrm{RC2}), \quad \mathscr{G}_{\mathrm{module}} = \max_{m} \; g(100, m), \quad \mathscr{G}_{\mathrm{joint}} = \max_{\ell, m} \; g(\ell, m). \tag{1}$$

The term oracle emphasizes that these quantities are a posteriori upper bounds among the evaluated probing locations, since they require exhaustive evaluation and access to test performance. In particular, $\mathscr{G}_{\mathrm{joint}}$ defines the optimal performance among the evaluated probing locations.

3 Experiments

In this section, we address the questions raised in Section 1 through large-scale evaluations of vision (Section 3.1) and language models (Section 3.2); see Appendix B for the full implementation details.

[2]We use zero-based layer indexing, such that for a model with L layers, probing at relative depth ℓ corresponds to layer $\mathrm{int}(\frac{\ell}{100} \cdot (L-1))$.

3.1 PROBING VISION FOUNDATION MODELS

We consider models that cover different pretraining paradigms and scales: supervised ViT models (Dosovitskiy et al., 2021), discriminative self-supervised DINOv2 (Oquab et al., 2024) and DINOv3 models (Siméoni et al., 2025), and autoregressive AIMv2 models (Fini et al., 2025). This results in a total of 15 publicly available models, ranging from 22M to 7B parameters. Experiments are conducted on a diverse set of 18 image classification datasets, commonly studied in the literature (El-Nouby et al., 2024; Oquab et al., 2024). Table 7 provides an overview of the datasets. Following Oquab et al. (2024), linear probing is done via logistic regression, with features preprocessed using a standard scaler. The performance of a given representation is its top-1 test accuracy at the dataset level (%). Overall, VFM experiments comprise nearly 90 000 probing evaluations.

Optimal representations yield substantial gains. We first analyze the oracle gains provided by the best probing locations across depth and modules. Figure 3 displays the complementary cumulative distribution of gains over the probing baseline. Each curve shows, for a given gain threshold x, the fraction of model-dataset pairs with gain greater than x when probing the best representation over layers alone ($\mathscr{G}_{\text{layer}}$), modules alone ($\mathscr{G}_{\text{module}}$), or both jointly ($\mathscr{G}_{\text{joint}}$). We observe consistent gains, reaching up to 10 pp with $\mathscr{G}_{\text{joint}} \geq \mathscr{G}_{\text{module}} \geq \mathscr{G}_{\text{layer}}$ across the distribution. The first inequality follows directly from Eq. (1), while the second highlights the importance of module choice, which is often overlooked in prior work. Moreover, the gap between $\mathscr{G}_{\text{joint}}$ and $\mathscr{G}_{\text{module}}$ demonstrates the additional benefit of jointly selecting layers and modules.

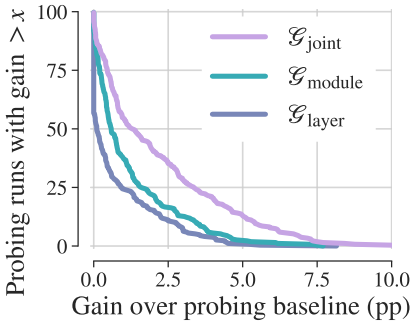


Figure 3: **Substantial gains from optimal probing locations in VFMs.** Complementary cumulative distribution of gains over the probing baseline across 270 model-dataset pairs when using the best representation across layers ($\mathscr{G}_{\text{layer}}$), modules ($\mathscr{G}_{\text{module}}$), or both ($\mathscr{G}_{\text{joint}}$).

Towards a fixed probing strategy. Ideally, we would like to identify a probing location at a fixed layer-module pair that recovers part, if not all, of the gain. We first examine how often each module appears in the best layer-module pair achieving $\mathscr{G}_{\text{joint}}$. We find that `Act` is optimal for 64.8% of the 270 model-dataset pairs, compared with 9.3% for `LN2`, the next most frequent, and 6.3% for the baseline module `RC2` (full results in Table 15). This suggests that it is a good candidate for the fixed module choice. Since $\mathscr{G}_{\text{joint}}$ is an oracle quantity, the superiority of `Act` is established a posteriori. Figure 4 displays the performance of `Act` and the baseline module `RC2` across layers. We can see that the `Act` module displays strong performance over layers, highlighting the importance of the module choice. The best `Act` representation, marked with a star, is generally in the second half of the network, but not necessarily at the final layer. This is particularly clear on Quickdraw, where intermediate representations outperform the final layer. The same phenomenon occurs in DINOv3, despite its classification performance being reported to increase smoothly towards

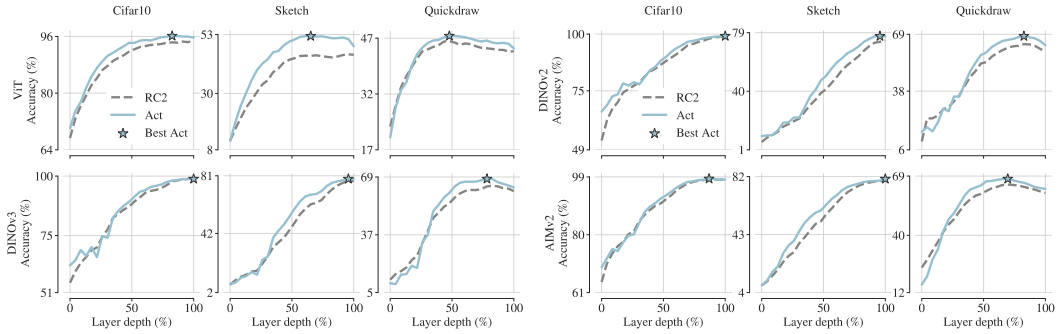


Figure 4: **Impact of `Act` and intermediate layers in VFMs.** Probing accuracy across layers for `Act` and `RC2` on Cifar10, Sketch, and Quickdraw. Results are reported for the 307M-parameter variant, common to each model family. The star indicates the best `Act` performance over depth.

Table 1: **Performance comparison in VFMs** between the probing baseline on the final-layer residual stream (`RC2@100`), the fixed strategy on the final-layer activation (`Act@100`), and the oracle strategy that probes the best representation over layers and modules for each model-dataset pair. Results are reported for the 307M-parameter variant, common to each model family, with datasets ordered from left to right by increasing mean probing error (easy to hard). The `Baseline` row reports the probing test accuracy (%), while `Act@100` and `Oracle` report accuracy differences in percentage points relative to the baseline, shown in green when positive and red otherwise.

Strategy	Flowers102	EuroSAT	Cifar10	Pet	Contrast	Food-101	Cifar100	Motion Blur	DTD	Speckle Noise	Clipart	Painting	Aircraft	Sketch	iWildCam	Quickdraw	Infograph	RxRx1	Average
ViT																			
`Baseline`	89.9	96.5	94.9	85.4	90.2	73.2	82.8	79.5	72.6	74.5	59.9	61.0	49.6	45.1	62.8	43.5	23.9	2.1	66.0
`Act@100`	+2.0	+0.4	+0.8	+4.5	-0.5	+7.3	+2.1	+1.2	+1.1	+0.2	+0.7	+4.0	+4.6	+3.4	+3.5	+0.8	+4.6	+0.1	+2.3
`Oracle`	+5.4	+1.9	+1.2	+4.8	+1.4	+8.8	+2.9	+2.7	+2.6	+3.5	+5.0	+6.7	+6.2	+7.3	+4.4	+4.1	+7.2	+2.7	+4.4
DINOv2																			
`Baseline`	99.6	97.4	99.0	91.4	96.1	91.7	91.4	87.4	81.6	85.3	82.3	77.9	70.3	73.9	70.1	59.6	44.3	4.8	78.0
`Act@100`	-0.1	+0.3	+0.1	+1.4	0.0	+1.6	+0.5	+0.7	-0.9	+0.6	+0.4	+1.3	+3.2	+2.8	+2.1	+3.5	+3.8	+1.4	+1.3
`Oracle`	+0.1	+1.0	+0.1	+1.4	+0.4	+1.6	+0.8	+1.9	+1.2	+2.0	+0.8	+1.8	+6.5	+2.8	+3.0	+8.4	+4.6	+5.4	+2.4
DINOv3																			
`Baseline`	99.7	97.5	98.8	94.8	95.7	93.8	92.0	87.5	82.7	88.4	85.2	81.0	78.9	77.4	67.5	61.2	49.9	5.1	79.8
`Act@100`	0.0	-0.3	0.0	-1.0	-0.2	+0.7	+0.1	-0.5	-1.8	-0.1	-0.3	0.0	-1.6	+1.4	+5.3	+2.1	+2.1	+0.8	+0.4
`Oracle`	0.0	+0.8	+0.1	0.0	0.0	+0.7	+0.1	0.0	+1.6	+0.4	+0.1	0.0	+4.0	+1.4	+5.3	+6.8	+2.8	+4.9	+1.6
AIMv2																			
`Baseline`	99.6	97.7	98.1	95.1	93.5	93.8	88.9	87.5	83.9	76.2	86.3	82.1	74.1	79.4	71.6	60.7	56.4	5.1	79.4
`Act@100`	0.0	0.0	+0.1	+0.6	+0.8	+0.6	+0.4	+0.2	-0.7	-0.2	+0.1	+0.6	+3.8	+1.0	-0.3	+2.0	+3.4	+0.5	+0.7
`Oracle`	0.0	+1.1	+0.3	+0.6	+0.8	+0.6	+0.8	+0.9	+0.6	+4.4	+0.2	+0.6	+3.8	+1.0	+1.9	+6.9	+3.4	+4.5	+1.8

the final layers (see Siméoni et al., 2025, Section 6.2.1). These observations hold across model sizes, as shown in Figures 15 to 26. The interaction between layers and modules is illustrated in Figure 1 (left), which displays the frequency with which each layer-module pair achieves the oracle gain $\mathscr{G}_{\text{joint}}$ across 270 model-dataset pairs. The `Act` module indeed has the highest optimality frequency, and despite variations in the depth, optimal representations are most frequently found at the final layer. We therefore consider the simple fixed strategy of probing final-layer `Act`, which we denote by `Act@100`.

Performance comparison. Table 1 compares the performance of the probing baseline `RC2@100` and the fixed strategy `Act@100` on the 307M-parameter variant common to each model family (full results deferred to Appendix C.1.7 due to space constraints). The oracle strategy, which probes the best layer-module pair, is displayed as an empirical upper bound. `Act@100` yields gains for each model family, with an average improvement of 2.3 pp for ViT, 1.3 pp for DINOv2, 0.7 pp for AIMv2, and 0.4 pp for DINOv3. In total, `Act@100` improves performance over the probing baseline in 64.1% of the 270 model-dataset pairs. The fixed strategy is not always optimal across model sizes and families. This reflects, in part, the fact that the optimal probing depth varies across datasets (see Appendix C.1.5). Yet, as shown in Figure 5, `Act@100` recovers without any search 42.2% of the gain from probing the best layer-module pair ($\mathscr{G}_{\text{joint}}$), compared to 32.0% for the best layer-wise `RC2`. The best layer-wise `Act` further increases recovery up to 78.6%, showing that a simple fixed choice of module can already substantially improve performance. We can see in Table 1 that the gains tend to increase from left to right. They are small on easy datasets such as Flowers102 and Pets, which are typically included in pretraining datasets (see Oquab et al., 2024, Table 15), but more pronounced on challenging benchmarks. On WILDS, where distribution shifts naturally arise (Koh et al., 2021), and DomainNet (Peng et al., 2019), `Act@100` improves performance in 89.3% of the 28 evaluations, with an average gain

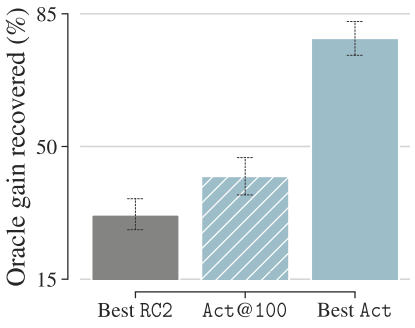
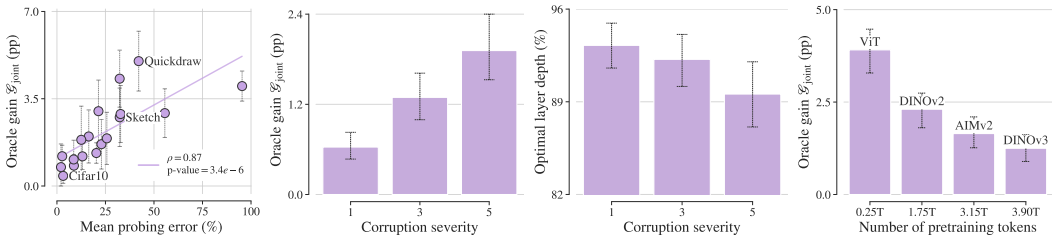


Figure 5: **`Act@100` recovers nearly half of the oracle gain.** Recovery of oracle gain $\mathscr{G}_{\text{joint}}$, averaged across all model-dataset pairs with 95% confidence intervals. The fixed strategy `Act@100` outperforms the best layer-wise `RC2`, while probing the best layer-wise `Act` recovers 78.6% of the oracle gain.

of 1.8 pp. A similar trend is observed for the `Oracle` strategy, suggesting that gains over the probing baseline are larger as transfer difficulty increases. We discuss this in the next paragraph.

Gain increases with transfer difficulty. Linear probing evaluates the transferability of frozen representations on downstream data. We therefore define the *empirical transfer difficulty* of a dataset as the mean probing error (%) of the baseline across all evaluated models, with higher values indicating more difficult transfer from pretraining to downstream data. Figure 6a shows that the oracle gain $\mathscr{G}_{\text{joint}}$ is positively correlated with transfer difficulty (Spearman $\rho = 0.87$, p-value < 0.01). This trend is illustrated by Cifar10, Sketch, and Quickdraw, which progressively depart from natural-image statistics: Figure 14 shows that the benefits of intermediate depth and alternative modules increase with transfer difficulty on these representative datasets. Our finding at the level of individual representations is consistent with Head2Toe (Evci et al., 2022), which shows that aggregating intermediate representations provides larger gains for target domains farther from the pretraining domain. We then consider controlled distribution shifts, following the evaluation protocol of DINOv3 (see Siméoni et al., 2025, Section 6.2.1). We train the classifier on features extracted from Cifar10 and evaluate it on a corrupted test set from Cifar10-C over 5 corruption types (Contrast, Gaussian Noise, Motion Blur, Snow, Speckle Noise) and increasing severities in $\{1, 3, 5\}$. As shown in Figures 6b and 6c, the oracle gain $\mathscr{G}_{\text{joint}}$ increases with corruption severity, while the optimal depth decreases. This suggests that the benefits of intermediate layers increase with distribution shift. To further investigate when these representations



(a) Transfer difficulty. (b) Corruption severity. (c) Optimal depth. (d) Pretraining scale.

Figure 6: **Oracle gains increase with transfer difficulty and corruption severity, but decrease with pretraining scale. (a)** Oracle gain as a function of empirical transfer difficulty, defined as the mean probing error over evaluated models. Markers show the gain for each dataset, averaged across 15 models, with a 95% confidence interval. **(b)** Oracle gain as a function of corruption severity on 5 types of corruption from Cifar10-C, averaged across 15 models, with a 95% confidence interval. **(c)** The optimal depth associated with $\mathscr{G}_{\text{joint}}$ decreases with corruption severity. **(d)** Oracle gain across model families, averaged across datasets and model sizes, with a 95% confidence interval. Models are ordered by increasing number of tokens seen during pretraining.

are beneficial, we conduct a controlled study in Appendix C.1.1, where we probe a pretrained model across layers before and after finetuning it on downstream data. Before finetuning, accuracy curves are concave when pretraining and downstream data differ substantially, with intermediate representations achieving the best performance. After finetuning, the curves straighten, with performance increasing monotonically towards the final layers. Finally, Figure 6d shows that oracle gains decrease with pretraining scale, computed as the number of tokens seen during pretraining (see Appendix C.1.2 for details). The relative error reductions follow the same order, with 15.7% for ViT, 12.1% for DINOv2, 10.8% for AIMv2, and 7.5% for DINOv3. Intuitively, more extensive pretraining increases the chance of seeing samples similar to the downstream images, potentially reducing the discrepancy between pretraining and downstream data distributions and leaving a smaller oracle gain to close. This is consistent with Figure 6a. Model size, in contrast, has no significant effect on gains; see Appendix C.1.6 for details. We note that these findings are observational, not causal, since many confounders exist in our analysis (architectural differences, training objectives, quality and diversity of pretraining data).

Takeaway 1. For VFMs, the best probing locations across layers and modules yield substantial performance gains. The simple fixed strategy `Act@100` improves performance in a majority of evaluations, recovering nearly half of these gains at no extra cost. Layer and module choices become more important as the discrepancy between pretraining and downstream data increases, while near-in-distribution datasets tend to benefit less from changing the probing location.

3.2 PROBING LARGE LANGUAGE MODELS

We now extend our analysis to large language models. As in Skean et al. (2025), we consider general-purpose LLMs: Qwen3 (Yang et al., 2025), Llama3 (Grattafiori et al., 2024), and Gemma3 (Kamath et al., 2025), for a total of 9 publicly available models, ranging from 1B to 12B parameters. Following prior work on textual embeddings (Lee et al., 2025; Li & Zhou, 2025; Skean et al., 2025; Tang & Yang, 2026), we use the Massive Multilingual Text Embedding Benchmark (`MMTEB`, Enevoldsen et al., 2025). We evaluate using the official `MMTEB` implementation (available at `https://github.com/embeddings-benchmark/mteb/`) via linear probing with logistic regression on 24 classification tasks from the `MTEB(Multilingual)` benchmark, spanning 31 languages and 14 domains. Table 11 provides an overview of the tasks. Following `MTEB` evaluation protocol (Enevoldsen et al., 2025), the performance of a given representation is its average test accuracy across tasks (%). Overall, the LLM experiments comprise over 54 000 probing evaluations.

Optimal representations yield substantial gains. Figure 7 displays the gains from probing the best representation over layers alone ($\mathscr{G}_{\text{layer}}$), modules alone ($\mathscr{G}_{\text{module}}$), or both jointly ($\mathscr{G}_{\text{joint}}$). Gains are consistent across models and sizes, reaching up to 4 pp. Qwen3 models display larger gains than Llama3 and Gemma3 models, with the improvement increasing with model size within the Qwen3 and Llama3 families. $\mathscr{G}_{\text{layer}} \geq \mathscr{G}_{\text{module}}$ holds for all but one model, while $\mathscr{G}_{\text{joint}}$ systematically exceeds both, highlighting the benefits of jointly selecting layers and modules.

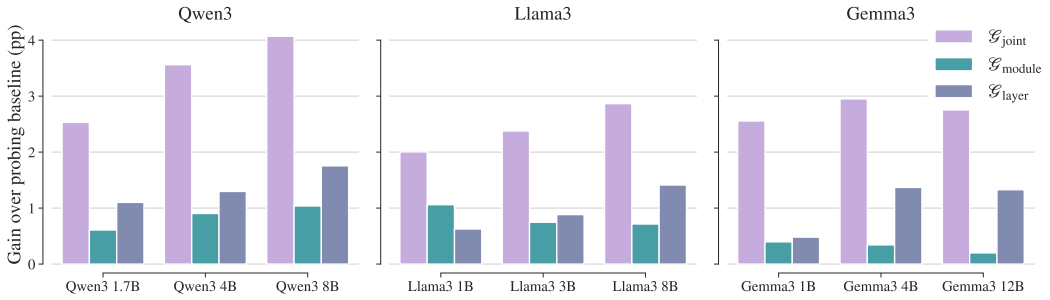


Figure 7: **Substantial gains from optimal probing locations in LLMs.** Gains over the probing baseline on `MTEB(Multilingual)` from probing the best representation across layers ($\mathscr{G}_{\text{layer}}$), modules ($\mathscr{G}_{\text{module}}$), or both ($\mathscr{G}_{\text{joint}}$) in Qwen3, Llama3, and Gemma3 models.

Towards a fixed probing strategy. Similarly to Section 3.1, we aim for a simple fixed probing location that improves performance. The fixed choice is simpler than for VFMs, since `LN2` appears in the best layer-module pair achieving $\mathscr{G}_{\text{joint}}$ for all but one model (see Table 19). We note that the baseline module `RC2` is never part of the optimal layer-module pair. Since $\mathscr{G}_{\text{joint}}$ is an oracle quantity, the superiority of `LN2` is established a posteriori. Akin to Figure 4, we observe in Figure 8 the performance of `LN2` and the baseline module `RC2` over depth. `LN2` consistently surpasses `RC2` across layers, highlighting the importance of the module choice. Intermediate `LN2` layers outperform the final layer for all the models considered, which is reminiscent of Skean et al. (2025). Although

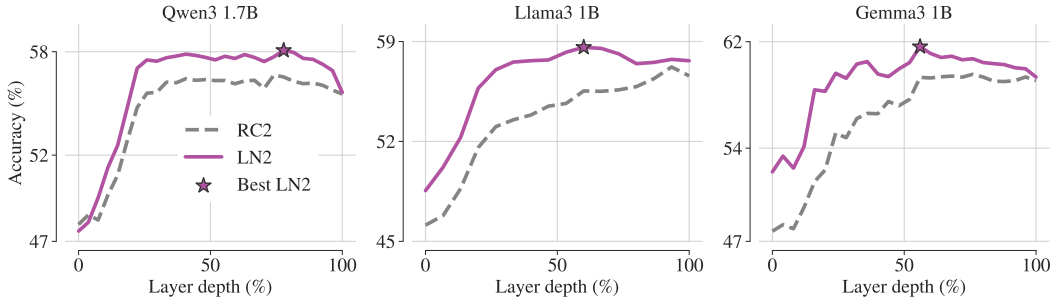


Figure 8: **Impact of `LN2` and intermediate layers on performance.** Probing accuracy across layers for `LN2` and `RC2` on `MTEB(Multilingual)`. The star indicates the best `LN2` performance over depth.

Table 2: **Performance comparison in LLMs** between the probing baseline on the final-layer residual stream (`RC2`), residual-stream probing at 60% depth (`RC2@60`), the fixed strategy `LN2@60`, and the oracle strategy that probes the best layer-module pair for each model. The accuracy difference in percentage points relative to the probing baseline is shown in green if positive, and in red otherwise.

Strategy	*Qwen3*			*Llama3*			*Gemma3*		
	1.7B	4B	8B	1B	3B	8B	1B	4B	12B
`Baseline`	55.5	56.6	57.6	56.6	58.9	59.9	59.1	60.9	61.5
`RC2@60`	56.1 +0.6	56.7 +0.1	58.3 +0.7	55.5 -1.1	59.7 +0.8	61.3 +1.3	59.3 +0.2	61.7 +0.8	62.6 +1.0
`LN2@60`	57.6 +2.1	59.4 +2.9	61.1 +3.5	58.6 +2.0	61.1 +2.2	62.3 +2.4	61.1 +2.0	63.1 +2.3	64.2 +2.6
`Oracle`	58.1 +2.5	60.1 +3.6	61.6 +4.1	58.6 +2.0	61.3 +2.4	62.8 +2.9	61.6 +2.6	63.8 +3.0	64.3 +2.8

the optimal depth varies between families, around 70% for Qwen3 models and near 50% for Llama3 and Gemma3 models, the interval $[50\%, 70\%]$ is remarkably stable across models and sizes, as shown in Figures 1 and 27. Its midpoint, 60%, corresponds to the average optimal depth across models. The benefit of `LN2` at 60% depth is illustrated in Figure 1 (right), which displays the optimality frequency of each layer-module pair across models. We therefore consider the simple fixed probing strategy of probing `LN2` at 60% depth. We denote it by `LN2@60`.

Performance comparison. Table 2 compares the performance of the probing baseline `RC2@100` and the fixed strategy `LN2@60` on `MTEB(Multilingual)`. The oracle strategy, which probes the best layer-module pair for each model, is displayed as an empirical upper bound. To disentangle the effects of layer and module choices, we also report probing on the residual stream at 60% depth (`RC2@60`). At the benchmark level, `LN2@60` consistently outperforms the probing baseline, with performance close to the oracle strategy for an average gain of 2.44 percentage points. `RC2@60` yields smaller, less consistent gains, highlighting the importance of both layer and module choices. As in Figure 7, the gains of `LN2@60` increase with model size within each family. Using `LN2@60` allows Gemma3 4B to match the performance of GritLM-7B (Muennighoff et al., 2025), the second-best embedding model on `MTEB(Multilingual)` at the time of `MMTEB` publication (see Enevoldsen et al., 2025, Table 2), without finetuning. Similarly, it enables Gemma3 12B to surpass GritLM-7B and match EmbeddingGemma (Vera et al., 2025), the embedding model directly derived from Gemma3 (see Appendix C.2.4 for details). Overall, `LN2@60` improves over the probing baseline in 21 of 24 tasks (87.5%) and in 77.8% of the 216 individual model-task evaluations (detailed results are provided in Appendix C.2.3). We can see in Figure 9 that `LN2@60` recovers without any search 86.0% of the gain from probing the best layer-module pair, compared to 39.3% for the best layer-wise `RC2`. The gap is nearly closed by the best layer-wise `LN2`, which recovers 99.8% of the oracle gain.

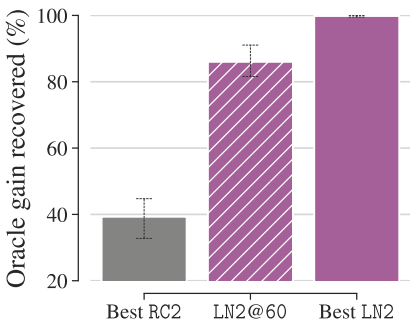


Figure 9: **`LN2@60` recovers substantial gains.** Recovery of oracle gain $\mathcal{G}_{\text{joint}}$, averaged across all models with 95% confidence intervals. The fixed `LN2@60` strategy outperforms the best layer-wise `RC2` strategy, while probing the best layer-wise `LN2` recovers nearly all of the oracle gain (99.8%).

Takeaway 2. For LLMs, the best probing locations across layers and modules yield substantial performance gains. The simple fixed strategy `LN2@60` recovers 86% of these gains at no extra cost, with nearly all of the remaining gap captured by the best layer-wise `LN2` representations.

4 Related Work

Probing foundation models. Linear probing (Alain & Bengio, 2017; Belinkov, 2022) has become a standard evaluation protocol to assess representations in foundation models across diverse modalities, including images (Chen et al., 2020; Oquab et al., 2024), text (Enevoldsen et al., 2025; Lee et al., 2025; Muennighoff et al., 2023), video (Assran et al., 2025), graphs (Veličković et al., 2019), and

time series (Feofanov et al., 2026; Goswami et al., 2024; Roschmann et al., 2026). Beyond linear heads, attentive probing is increasingly adopted to evaluate models with patch-level features (Chen et al., 2024; El-Nouby et al., 2024; Fini et al., 2025). While linear probes are still used, the embedding of sequences of tokens from $(\mathbb{R}^d)^n$ to $\mathbb{R}^d$ is learned via an attention mechanism at the level of the sequence. The overall operation thus becomes non-linear and leads to noticeable performance improvements (El-Nouby et al., 2024). In these works, features are typically extracted from the final layer after a forward pass. We challenge this convention, showing the benefits of probing intermediate representations, including within transformer blocks. Beyond probing at a single location, some approaches aggregate internal representations throughout the model, combining them with learned layer weights (Peters et al., 2018), supervised feature selection (Evci et al., 2022), or attention over all layers (Ciernik et al., 2026). Our work is orthogonal to these methods: it could help identify which representations across layers and modules are worth aggregating, and, more broadly, improve our understanding of transformer representations.

Understanding transformer representations. Intermediate representations can also be studied to understand what models encode rather than to improve downstream performance. Probing intermediate text representations reveals what linguistic knowledge is captured within the model (Liu et al., 2019) and where: Tenney et al. (2019) show that BERT (Devlin et al., 2019) represents the steps of the traditional NLP pipeline in an interpretable and localizable way. Similarly, Joseph et al. (2026) use linear probes to understand where physical information becomes accessible in encoder-based video transformers, and how it is organized. Linear probes are also central to *mechanistic interpretability*, which aims to reverse-engineer the computations of neural networks, with a strong focus on LLMs. This line of work views the transformer as a set of attention and feedforward modules that read from and write to the shared residual stream (Elhage et al., 2021). Probes and interventions have thus been used to analyze attention heads (Li et al., 2023), feedforward modules (Geva et al., 2021), and the residual stream (Gurnee & Tegmark, 2024; Marks & Tegmark, 2024), where the latter can be steered to control the model behavior (Rimsky et al., 2024; Turner et al., 2024). Note that, akin to feature evaluation, the residual stream is the typical representation studied in the mechanistic interpretability literature, often referred to as *residual stream activation*. Beyond interpretability, Odonnat et al. (2026) analyze the smoothness of individual transformer components to determine which ones to prioritize during finetuning. Our work follows this direction of looking inside transformer blocks, yet the goal differs: we aim to identify which internal representations are most useful as frozen features for downstream tasks, not to reverse-engineer computations, elicit a specific behavior in the model, or finetune it. We discuss additional related work in Appendix A.

5 Discussion

This paper investigates how useful different transformer representations are for linear probing in vision foundation models and large language models. Our extensive *layer by layer, module by module* analysis shows that layer and module choices should be considered jointly: the best layer-module pair yields substantially larger performance gains than selecting the best layer with the residual stream fixed, while module performance can vary substantially across depth. Building on these findings, we identify simple fixed probing strategies, `Act@100` and `LN2@60` for VFMs and LLMs, respectively, that improve performance in a majority of evaluations and recover a large fraction of the gains from the best layer-module pair. For vision models, the benefit of `Act@100` increases with transfer difficulty between pretraining and downstream data. These findings highlight the importance of looking beyond depth alone when probing transformer models. We hope our work can contribute to a better understanding of their internal representations.

Limitations and future work. Our analysis focuses on linear probing for classification and could naturally be extended to other downstream tasks and modalities, including tabular and time series foundation models. Moreover, our insights into the usefulness of individual transformer representations could help guide the design of aggregation mechanisms that combine intermediate representations rather than treating them uniformly.

AI USE STATEMENT

Generative AI tools were used to aid and polish the writing. They were also used to discuss research ideas and experimental design, including the experimental setups of prior work, and to assist with rapid prototyping of visualizations. The authors carefully checked all LLM-generated outputs and independently implemented and verified the final experiments, analyses, and visualizations. The authors take full responsibility for the contents of this paper.

ETHICS STATEMENT

The goal of this work is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

REPRODUCIBILITY STATEMENT

All the implementation details to reproduce our results are provided in Appendix B, including datasets, benchmarks, models, and pretrained weights used. Experimental setups are provided in the main paper, along with additional details and results in Appendix C.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2017. URL https://openreview.net/forum?id=ryF7rTqgl.

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/cfcce0621b49c983991ead4c3d4d3b6b-Paper.pdf.

Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Leonardo Neves, Vitor Silva, and Francesco Barbieri. Twitter Topic Classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, Gyeongju, Republic of Korea, oct 2022. International Committee on Computational Linguistics.

Gaurav Arora. iNLTK: Natural language toolkit for indic languages. In Eunjeong L. Park, Masato Hagiwara, Dmitrijs Milajevs, Nelson F. Liu, Geeticka Chauhan, and Liling Tan (eds.), *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pp. 66–71, Online, nov 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlposs-1.10. URL https://aclanthology.org/2020.nlposs-1.10.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Soran Badawi, Arefeh Kazemi, and Vali Rezaie. KurdiSent: a corpus for kurdish sentiment analysis. *Language Resources and Evaluation*, pp. 1–20, 01 2024. doi: 10.1007/s10579-023-09716-6.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Abdul Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Shang-Wen Li, Piotr Dollar, and Christoph Feichtenhofer.

Perception encoder: The best visual embeddings are not at the output of the network. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=INqBOmwIpG.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 446–461, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10599-4.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision*, 132(1):208–223, August 2024. ISSN 0920-5691. doi: 10.1007/s11263-023-01852-4. URL https://doi.org/10.1007/s11263-023-01852-4.

Laure Ciernik, Marco Morik, Lukas Thede, Luca Eyring, Shinichi Nakajima, Zeynep Akata, and Lukas Muttenthaler. Attentive multi-layer fusion for vision transformers. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=wSXZtvqwoa.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, and Ori Ram et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 933–941. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/dauphin17a.html.

Davis David. Swahili : News classification dataset, December 2020. URL https://doi.org/10.5281/zenodo.4300294.

Nisansa de Silva. Sinhala text classification: Observations from the perspective of a resource poor language. *Year of Publication*, 2015.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE*

Conference on, pp. 248–255. IEEE, 2009. URL https://ieeexplore.ieee.org/abstract/document/5206848/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, and Flood Sung et al. Kimi-vl technical report, 2025. URL https://arxiv.org/abs/2504.07491.

Quentin Duval, Ishan Misra, and Nicolas Ballas. A simple recipe for competitive low-compute self supervised vision models, 2023. URL https://arxiv.org/abs/2301.09451.

Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Ángel Bautista, Vaishaal Shankar, Alexander T Toshev, Joshua M. Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 12371–12384, 2024.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, Ömer Veysel Çağatan, Akash Kundu, Martin Bernstorff, Shitao Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loïc Magne, Isabelle Mohr, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Suppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, Manuel Faysse, Aleksei Vatolin, Nandan Thakur, Manan Dey, Dipam Vasani, Pranjal A Chitale, Simone Tedeschi, Nguyen Tai, Artem Snegirev, Mariya Hendriksen, Michael Günther, Mengzhou Xia, Weijia Shi, Xing Han Lù, Jordan Clive, Gayatri K, Maksimova Anna, Silvan Wehrli, Maria Tikhonova, Henil Shalin Panchal, Aleksandr Abramov, Malte Ostendorff, Zheng Liu, Simon Clematide, Lester James Validad Miranda, Alena Fenogenova, Guangyu Song, Ruqiya Bin Safi, Wen-Ding Li, Alessia Borghini, Federico Cassano, Lasse Hansen, Sara Hooker, Chenghao Xiao, Vaibhav Adlakha, Orion Weller, Siva Reddy, and Niklas Muennighoff. MMTEB: Massive multilingual text embedding benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zl3pfz4VCV.

Utku Evci, Vincent Dumoulin, Hugo Larochelle, and Michael C Mozer. Head2Toe: Utilizing intermediate representations for better transfer learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 6009–6033. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/evci22a.html.

Vasilii Feofanov, Songkang Wen, Shifeng Xie, Simon Roschmann, Marius Alonso, Hongbo Guo, Romain Ilbert, Malik Tiomoko, Quentin Bouniot, Zeynep Akata, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight foundation model for time series classification. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=gbJMAjXLZ4.

Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor G. Turrisi da Costa, Louis Béthune, Zhe Gan, Alexander Toshev, Marcin Eichner, Moin Nabi, Yinfei Yang, Joshua Susskind, and Alaaeldin El-Nouby. Multimodal autoregressive pre-training of large vision encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9641–9654, June 2025.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurumurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayana Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant

Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihlas, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi,

Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirnschall, Weiyi Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Villela, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Tsendsuren Munkhdalai, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnapalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Bartek Perz, Wooyeol Kim, Nandita Dukkipati, Anthony Baryshnikov, Christos Kaplanis, XiangHai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai, Shariq Iqbal,

Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kępa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeff Dean, and Oriol Vinyals. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu,

Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry

Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Roopali Vij, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei

Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh

Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2025. URL https://arxiv.org/abs/2312.11805.

Tsvetanka Georgieva-Trifonova, Milena Stefanova, and Stefan Kalchev. Dataset for "Customer Feedback Text Analysis for Online Stores Reviews in Bulgarian", 2018. URL https://doi.org/10.7910/DVN/TXIK9P.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446/.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16115–16152. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/goswami24a.html.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song,

Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jE8xbmvFin.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

Romain Ilbert, Ambroise Odonnat, Vasilii Feofanov, Aladin Virmaux, Giuseppe Paolo, Themis Palpanas, and Ievgen Redko. SAMformer: Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=8kLzL5QBh2.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3182–3196, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.181. URL https://aclanthology.org/2024.findings-emnlp.181/.

Sonia Joseph, Quentin Garrido, Randall Balestriero, Matthew Kowal, Thomas Fel, Shahab Bakhtiari, Blake Aaron Richards, and Michael Rabbat. Interpreting physics in video world models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=aijGVmEG9Y.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy

Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Jiwon Kim and Won Ik Cho. Kocasm: Korean automatic sarcasm detection. https://github.com/SpellOnYou/korean-sarcasm, 2019.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5637–5664. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/koh21a.html.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, pp. 491–507, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58557-0. doi: 10.1007/978-3-030-58558-7_29. URL https://doi.org/10.1007/978-3-030-58558-7_29.

Alex Krizhevsky. Learning multiple layers of features from tiny images. In *Technical report*, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *arXiv preprint arXiv:2005.00085*, 2020.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, Xiaoqi Ren, Shanfeng Zhang, Daniel Salz, Michael Boratko, Jay Han, Blair Chen, Shuo Huang, Vikram Rao, Paul Suganthan, Feng Han, Andreas Doumanoglou, Nithi Gupta, Fedor Moiseev, Cathy Yip, Aashi Jain, Simon Baumgartner, Shahrokh Shahi, Frank Palma Gomez, Sandeep Mariserla, Min Choi, Parashar Shah, Sonam Goenka, Ke Chen, Ye Xia, Koert Chen, Sai Meher Karthik Duddu, Yichang Chen, Trevor Walker, Wenlei Zhou, Rakesh Ghiya, Zach Gleicher, Karan Gill, Zhe Dong, Mojtaba Seyedhosseini, Yunhsuan Sung, Raphael Hoffmann, and Tom Duerig. Gemini Embedding: Generalizable embeddings from gemini, 2025. URL https://arxiv.org/abs/2503.07891.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=aLLuYpn83y.

Ziyue Li and Tianyi Zhou. Your mixture-of-experts LLM is secretly an embedding model for free. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eFGQ97z5Cd.

Daniele Licari, Praveen Bushipaka, Gabriele Marino, Giovanni Comand'e, and Tommaso Cucinotta. Legal holding extraction from italian case documents using italian-legal-bert text summarization. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL '23, pp. 148–156, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701979. doi: 10.1145/3594536.3595177. URL https://doi.org/10.1145/3594536.3595177.

Chun Liu, Hongguang Zhang, Kainan Zhao, Xinghai Ju, and Lin Yang. LLMEmbed: Rethinking lightweight LLM's genuine function in text classification. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7994–8004, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.433. URL https://aclanthology.org/2024.acl-long.433/.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1073–1094, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1112. URL https://aclanthology.org/N19-1112/.

Andani Madodonga, Vukosi Marivate, and Matthew Adendorff. Izindaba-tindzaba: Machine learning news categorisation for long and short text for isizulu and siswati. *arXiv preprint arXiv:2306.07426*, 2023.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013. URL https://arxiv.org/abs/1306.5151.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65, 2014.

Vukosi Marivate, Moseli Mots'Oehli, Valencia Wagner, Richard Lastrucci, and Isheanesu Dzingirai. Puoberta: Training and evaluation of a curated language model for setswana. In *SACAIR 2023 (To Appear)*, 2023.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Alexis Marouani, Oriane Siméoni, Herve Jegou, Piotr Bojanowski, and Huy V. Vo. Revisiting [CLS] and patch token interaction in vision transformers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xs9xwFphLy.

Wojciech Masarczyk, Mateusz Ostaszewski, Ehsan Imani, Razvan Pascanu, Piotr Miłoś, and Tomasz Trzcinski. The tunnel effect: Building data representations in deep neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xQOHOpe1Fv.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148/.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning, 2025. URL https://arxiv.org/abs/2402.09906.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Dan Nielsen. ScandEval: A benchmark for Scandinavian natural language processing. In Tanel Alum"ae and Mark Fishel (eds.), *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pp. 185–201, T'orshavn, Faroe Islands, may 2023. University of Tartu Library. URL https://aclanthology.org/2023.nodalida-1.20.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008. URL https://api.semanticscholar.org/CorpusID:15193013.

Ambroise Odonnat, Laetitia Chapel, Romain Tavenard, and Ievgen Redko. Vision transformer finetuning benefits from non-smooth components. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lTXvOhUOl2.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Hille Pajupuu, Jaan Pajupuu, Rene Altrov, and Kairi Tamuri. Estonian Valence Corpus / Eesti valentsikorpus, 11 2023. URL https://figshare.com/articles/dataset/Estonian_Valence_Corpus_Eesti_valentsikorpus/24517054.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020. doi: 10.1073/pnas.2015509117. URL https://www.pnas.org/doi/abs/10.1073/pnas.2015509117.

Shantipriya Parida, Sambit Sekhar, Soumendra Kumar Sahoo, Swateek Jena, Abhijeet Parida, Satya Ranjan Dash, and Guneet Singh Kohli. Odiagenai: Generative ai and llm initiative for the odia language. https://huggingface.co/OdiaGenAI, 2023.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202/.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. ISBN 0262170051.

Akshay Rangamani, Marius Lindegaard, Tomer Galanti, and Tomaso A Poggio. Feature learning in deep classifiers through intermediate neural collapse. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28729–28745. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/rangamani23a.html.

Sebastian Raschka, Joshua Patterson, and Corey Nolet. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *arXiv preprint arXiv:2002.04803*, 2020.

Stefano Recanatesi, Matthew Farrell, Madhu Advani, Timothy Moore, Guillaume Lajoie, and Eric Shea-Brown. Dimensionality compression and expansion in deep neural networks, 2019. URL https://arxiv.org/abs/1906.00443.

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

Simon Roschmann, Quentin Bouniot, Vasilii Feofanov, Ievgen Redko, and Zeynep Akata. TiViT: Time series representations for classification lie hidden in pretrained vision transformers. In *2nd ICML Workshop on Foundation Models for Structured Data*, 2026. URL https://openreview.net/forum?id=49DWDGNGwf.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Emily Sheng and David Uthus. Investigating societal biases in a poetry composition system, 2020.

Mustafa Shukor, Louis Béthune, Dan Busbridge, David Grangier, Enrico Fini, Alaaeldin El-Nouby, and Pierre Ablin. Scaling laws for optimal data mixtures. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=vVU1KTOsju.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=WGXb7UdvTX.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ahlrf2HGJR.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.127063. URL https://www.sciencedirect.com/science/article/pii/S0925231223011864.

Yixuan Tang and Yi Yang. KV-embedding: Training-free text embedding via internal KV re-routing in decoder-only LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11773–.–11794, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.540. URL https://aclanthology.org/2026.acl-long.540/.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452/.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Herve Jegou. Going deeper with Image Transformers . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 32–42, Los Alamitos, CA, USA, October 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00010. URL https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00010.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Cheng-Hao Tu, Zheda Mai, and Wei-Lun Chao. Visual Query Tuning: Towards Effective Usage of Intermediate Representations for Parameter and Memory Efficient Transfer Learning . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7725–7735, Los Alamitos, CA, USA, June 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.00746. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.00746.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248.

Arnas Uselis and Seong Joon Oh. Intermediate layer classifiers for OOD generalization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ByCV9xWfNK.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rklz9iAcKQ.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, Weiyi Wang, Zhe Li, Gus Martins, Jinhyuk Lee, Mark Sherwood, Juyeong Ji, Renjie Wu, Jingxiao Zheng, Jyotinder Singh, Abheesht Sharma, Divyashree Sreepathihalli, Aashi Jain, Adham Elarabawy, AJ Co, Andreas Doumanoglou, Babak Samari, Ben Hora, Brian Potetz, Dahun Kim, Enrique Alfonseca, Fedor Moiseev, Feng Han, Frank Palma Gomez, Gustavo Hernández Ábrego, Hesen Zhang, Hui Hui, Jay Han, Karan Gill, Ke Chen, Koert Chen, Madhuri Shanbhogue, Michael Boratko, Paul Suganthan, Sai Meher Karthik Duddu, Sandeep Mariserla, Setareh Ariafar, Shanfeng Zhang, Shijie Zhang, Simon Baumgartner, Sonam Goenka, Steve Qiu, Tanmaya Dabral, Trevor Walker, Vikram Rao, Waleed Khawaja, Wenlei Zhou, Xiaoqi Ren, Ye Xia, Yichang Chen, Yi-Ting Chen, Zhe Dong, Zhongli Ding, Francesco Visin, Gaël Liu, Jiageng Zhang, Kathleen Kenealy, Michelle Casbon, Ravin Kumar, Thomas Mesnard, Zach Gleicher, Cormac Brick, Olivier Lacombe, Adam Roberts, Qin Yin, Yunhsuan Sung, Raphael Hoffmann, Tris Warkentin, Armand Joulin, Tom Duerig, and Mojtaba Seyedhosseini. EmbeddingGemma: Powerful and lightweight text representations, 2025. URL https://arxiv.org/abs/2509.20354.

Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. Dalaj - a dataset for linguistic acceptability judgments for swedish: Format, baseline, sharing, 2021.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report, 2024a. URL https://arxiv.org/abs/2402.05672.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models, 2024b. URL https://arxiv.org/abs/2401.00368.

Andika William and Yunita Sari. Click-id: A novel dataset for indonesian clickbait headlines. *Data in Brief*, 32:106231, 2020. ISSN 2352-3409. doi: https://doi.org/10.1016/j.dib.2020.106231. URL http://www.sciencedirect.com/science/article/pii/S2352340920311252.

Genta Indra Winata, Alham Fikri Aji, Samuel Cahyawijaya, Rahmad Mahendra, Fajri Koto, Ade Romadhony, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Pascale Fung, et al. Nusax: Multilingual parallel sentiment dataset for 10 indonesian local languages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 815–834, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural

language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.emnlp-demos.6/.

Zhi-Fan Wu, Chaojie Mao, Xue Wang, Jianwen Jiang, Yiliang Lv, and Rong Jin. Structured model probing: Empowering efficient transfer learning by structured regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16849–16858, June 2024.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10524–10533. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/xiong20b.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1e8a19426224ca89e83cef47f1e7f53b-Paper.pdf.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Łukasz Augustyniak, Kamil Tagowski, Albert Sawczyn, Denis Janiak, Roman Bartusiak, Adrian Szymczak, Marcin Wątroba, Arkadiusz Janz, Piotr Szymański, Mikołaj Morzy, Tomasz Kajdanowicz, and Maciej Piasecki. This is the way: designing and compiling lepiszcze, a comprehensive nlp benchmark for polish, 2022. URL https://arxiv.org/abs/2211.13112.

Appendix

Roadmap. We discuss additional related work in Appendix A. The full implementation details are given in Appendix B and, finally, extensive additional experiments are provided in Appendix C. We display the corresponding table of contents below.

TABLE OF CONTENTS

A Extended related work **30**

B Implementation details **30**

B.1 Transformer representations in gated FFN 30

B.2 Vision foundation models 30

B.2.1 Transformer representations 31

B.2.2 Probing setup 31

B.3 Large language models 36

B.3.1 Transformer representations 36

B.3.2 Probing setup 38

C Additional experiments **39**

C.1 Vision foundation models 39

C.1.1 Intermediate layers under distribution shift 39

C.1.2 Gains across pretraining scales 40

C.1.3 Gain across model families 41

C.1.4 Module comparison 42

C.1.5 Performance along the depth 42

C.1.6 Gains across model sizes 54

C.1.7 Full results 55

C.2 Large language models 58

C.2.1 Module comparison 58

C.2.2 Performance along the depth 58

C.2.3 Full results 59

C.2.4 Comparison to embedding models 59

A Extended related work

In this section, we extend the discussion of prior works related to our paper in Section 4.

Better transfer of intermediate layers. Several works propose explanations as to why intermediate representations can generalize better than the final layer, especially under distribution shift. The intrinsic dimension of representations first increases then decreases with depth (Ansuini et al., 2019; Recanatesi et al., 2019), and final layers exhibit neural collapse, compressing features onto class means (Papyan et al., 2020). This phenomenon also appears, more weakly, in intermediate layers (Rangamani et al., 2023). In addition, the tunnel effect describes later layers compressing representations that are already linearly separable through intermediate layers, thus degrading out-of-distribution performance (Masarczyk et al., 2023). Although these works mostly study task-specific deep neural networks rather than large-scale pretrained foundation models, their observations are consistent with our finding that the optimal probing depth moves to earlier layers as the shift between pretraining and downstream data grows.

Intermediate representations for adaptation and embeddings. Intermediate representations can also be used to adapt pretrained models efficiently. Visual Query Tuning (Tu et al., 2023) learns queries that summarize intermediate features of vision transformers, while Wu et al. (2024) implement an efficient adaptation model that selects features using the structure of the model and adapts to task difficulty. In language, LLMEmbed (Liu et al., 2024) combines representations from several layers of lightweight LLMs for text classification. When LLMs are used as frozen embedding models, training-free methods can improve the quality of extracted features by modifying either the input, with prompts (Jiang et al., 2024), or repetitions (Springer et al., 2025), or the forward pass, by re-routing internal key-value states (Tang & Yang, 2026). The adaptation methods aggregate residual stream representations across depth, while the training-free methods read the final-layer residual stream. Both could benefit from our insights by investigating other transformer modules and, for the latter, other depths as well.

B Implementation details

In this section, we provide the full implementation details necessary to reproduce our results.

B.1 Transformer representations in gated FFN

In large-scale transformers (Oquab et al., 2024; Touvron et al., 2023), an increasingly common alternative to the two-layer MLP feedforward network is the use of gated activations (Dauphin et al., 2017; Shazeer, 2020). Let d be the embedding dimension, $(x_1, \dots, x_n) \in (\mathbb{R}^d)^n$ be an input sequence of tokens, viewed as a matrix in $\mathbb{R}^{d\times n}$, $h \in \mathbb{N}$ be the FFN dimension, and σ be an activation function. Let $W_1 \in \mathbb{R}^{h\times d}$, $W_2 \in \mathbb{R}^{d\times h}$, $W_3 \in \mathbb{R}^{h\times d}$ be weight matrices. Denoting the element-wise multiplication by $\odot$ and omitting the biases for readability, the standard FFN and the gated FFN are written as follows

$$\mathrm{FFN}(x) = W_2\sigma(W_1x), \quad \mathrm{FFN}_{\mathrm{gated}}(x) = W_2(\sigma(W_1x) \odot W_3x). \tag{2}$$

The main difference is that the gated FFN combines a non-linear projection with an additional linear projection through element-wise multiplication. To ensure consistency between the standard and the gated FFN, we recover the FFN representations as follows: `FC1` is the linear layer fed as input to the activation function (W_1x in both cases), `Act` is the non-linear transformation fed as input to the second linear function ($\sigma(W_1x)$ for the standard FFN and $\sigma(W_1x) \odot W_3x$ for the gated FFN), and, finally, `FC2` is the output of the feedforward network ($\mathrm{FFN}(x)$ and $\mathrm{FFN}_{\mathrm{gated}}(x)$ in (2)). This is illustrated in Figure 2.

B.2 Vision foundation models

This section provides details on VFMs' transformer representations and the linear probing setup.

B.2.1 TRANSFORMER REPRESENTATIONS

We recover embeddings across layers and modules with a single forward pass using PyTorch forward hooks and pre-hooks. In our code, all VFMs inherit from the abstract base `BaseVisionModel` displayed in Figure 10. The feature extraction is done with the `get_embeddings` method. The most important abstract method to implement is `block_adapter`, which defines the 8 representations to extract within each transformer block. The corresponding abstract base class `TransformerBlockAdapter` is displayed in Figure 11. We provide below the implementation details on the VFMs used in our experiments.

ViT. The Vision Transformer architecture (ViT, Dosovitskiy et al., 2021) consists of a succession of transformer encoders with LayerNorms (Ba et al., 2016) and a standard feedforward network with GELU activation (Hendrycks & Gimpel, 2016). Inputs are 2D images that are split into square patches of size P, which are flattened and linearly embedded in dimension d. A classification token `CLS` is prepended to the sequence of tokens before adding positional embeddings. The checkpoints used in our experiments follow the HuggingFace transformers (Wolf et al., 2020) implementation available at `transformers/models/vit/modeling_vit.py`.

DINOv2. The DINOv2 backbone (Oquab et al., 2024) follows the ViT architecture described above, adding a LayerScale (Touvron et al., 2021) to stabilize training. All variants use a standard FFN, except the 1.1B one, which uses a gated FFN with a SwiGLU activation (Shazeer, 2020). The checkpoints used in our experiments follow the HuggingFace transformers' implementation available at `transformers/models/dinov2/modeling_dinov2.py`.

DINOv3. The DINOv3 backbone (Siméoni et al., 2025) follows the DINOv2 one and implements Rotary Positional Embeddings (RoPE, Su et al., 2024). All variants use a standard FFN, except the 7B one, which uses a gated FFN with a SwiGLU activation (Shazeer, 2020). The checkpoints used in our experiments follow the HuggingFace transformers' implementation available at `transformers/models/dinov3_vit/modeling_dinov3_vit.py`.

AIMv2. The AIMv2 model (Fini et al., 2025) pairs a ViT backbone with a multimodal decoder to autoregressively generate raw image patches and text tokens. In our work, we only focus on the vision part, which consists of a succession of transformer encoders with RMS norms (Zhang & Sennrich, 2019) and a gated feedforward network with SwiGLU activation (Shazeer, 2020). Inputs are 2D images that are split into square patches of size P, which are flattened and linearly embedded in dimension d. Unlike the original ViT, no `CLS` token is prepended to the sequence: attention is instead constrained with a prefix causal mask during pretraining, enabling bidirectional attention over image patches at inference without further tuning. The checkpoints used in our experiments follow the HuggingFace transformers' implementation available at `transformers/models/aimv2/modeling_aimv2.py`.

Feature extraction. Applying linear probing with a logistic regression requires feeding the regressor a matrix of embeddings over the entire train split (resp. test split). However, even over a single batch of 512 samples, a forward pass with a "small" model like the ViT-B with 86M parameters amounts to saving 49K tensors in $\mathbb{R}^{768}$, taking around 144MiB in memory. Such an extraction over an entire dataset is thus intractable, even on high-performance GPUs like H200. To tackle this issue, we perform the probing in two steps: 1) feature extraction with embeddings saved on disk, and 2) logistic regression over frozen features. For the first step, we save each batch of pooled representations on the CPU and stream it directly to disk to maintain low GPU memory consumption and idle time, independent of the number of layers. This step is done for both the training and test samples. Then, for each (layer, module) pair, we rebuild the dataset-level embeddings over the train split (resp. test split) to fit (resp. evaluate) the logistic regression. Our implementation allows us to use logistic regression even for large-scale VFMs with up to 7B parameters.

B.2.2 PROBING SETUP

We conduct the probing experiments across various pretraining paradigms, model sizes, and datasets.

```python
class BaseVisionModel(nn.Module, ABC):
    r"""Abstract base class for VFMs."""

    def __init__(self):
        super().__init__()
        self.device: str
        self.model: PreTrainedModel

    @abstractmethod
    def _load_model(self, model_name: str, device: str) -> PreTrainedModel:
        r"""Load a pretrained HuggingFace model."""
        ...

    @abstractmethod
    def pooling(self, hidden_states: torch.Tensor) -> torch.Tensor:
        r"""Global average pooling method over content tokens."""
        ...

    @property
    @abstractmethod
    def block_adapter(self) -> type[TransformerBlockAdapter]:
        r"""Return the block adapter class for the model's transformer blocks."""
        ...

    @property
    @abstractmethod
    def blocks(self) -> ModuleList:
        r"""Return the list of transformer blocks."""
        ...

    def _make_pre_hook(self, embeddings: dict[str, torch.Tensor], key: str) -> Callable:
        r"""Create a forward pre-hook that captures the INPUT of a module."""
        def pre_hook_fn(module: nn.Module, input: tuple[torch.Tensor, ...]) -> None:
            embeddings[key] = self.pooling(hidden_states=input[0])
        return pre_hook_fn

    def _make_hook(self, embeddings: dict[str, torch.Tensor], key: str) -> Callable:
        r"""Create a forward hook that captures the OUTPUT of a module."""
        def hook_fn(module: nn.Module, input: tuple[torch.Tensor, ...], output: torch.Tensor | tuple[torch.Tensor, ...]) ->
              None:
            out = output[0] if isinstance(output, tuple) else output
            embeddings[key] = self.pooling(hidden_states=out)
        return hook_fn

    @torch.inference_mode()
    def get_embeddings(self, pixel_values: torch.Tensor, layers: list[int], modules: list[str]) -> dict[str, torch.Tensor]:
        r"""Single forward pass recovering transformer representations over layers and modules."""
        # Recover layers and modules
        layers = layers if layers is not None else list(range(len(self.blocks)))
        modules = modules if modules is not None else TRANSFORMER_MODULES

        # Loop over transformer blocks to register hooks across specified layers and modules
        embeddings: dict[str, torch.Tensor] = {}
        handles: list[torch.utils.hooks.RemovableHandle] = []
        layer_set = set(layers)
        for layer_idx, layer in enumerate(self.blocks):
            if layer_idx not in layer_set:
                continue
            adapter = self.block_adapter(layer)
            for module_name, module in adapter.module_map().items():
                if module_name not in modules:
                    continue
                key = f"block_{layer_idx}_{module_name}"

                # Modules for which we hook the inputs
                if module_name in adapter.pre_hooks:
                    handles.append(
                        module.register_forward_pre_hook(self._make_pre_hook(embeddings=embeddings, key=key))
                    )
                # Modules for which we hook the output
                else:
                    handles.append(module.register_forward_hook(self._make_hook(embeddings=embeddings, key=key)))

        # Forward pass
        try:
            self.model(pixel_values=pixel_values)
        finally:
            for handle in handles:
                handle.remove()

        return embeddings
```

Figure 10: `BaseVisionModel` implementation.

```python
class TransformerBlockAdapter(ABC):
    r"""Adapter that maps standard module names to transformer block's submodules.

    Subclasses need to implement the 8 methods that outputs the following modules:
        ln1: first layer norm
        mha: multi-head attention
        rc1: first residual connection
        ln2: second layer norm
        fc1: first feedforward linear layer
        act: feedforward activation function
        fc2: second feedforward linear layer
        rc2: second residual connection
    """

    def __init__(self, layer: nn.Module):
        self.ln1 = self._get_ln1(layer)
        self.mha = self._get_mha(layer)
        self.rc1 = self._get_rc1(layer)
        self.ln2 = self._get_ln2(layer)
        self.fc1 = self._get_fc1(layer)
        self.act = self._get_act(layer)
        self.fc2 = self._get_fc2(layer)
        self.rc2 = self._get_rc2(layer)
        self.pre_hooks = self._get_pre_hooks_modules(layer)

    @abstractmethod
    def _get_ln1(self, layer: nn.Module) -> nn.Module: ...

    @abstractmethod
    def _get_mha(self, layer: nn.Module) -> nn.Module: ...

    @abstractmethod
    def _get_rc1(self, layer: nn.Module) -> nn.Module: ...

    @abstractmethod
    def _get_ln2(self, layer: nn.Module) -> nn.Module: ...

    @abstractmethod
    def _get_fc1(self, layer: nn.Module) -> nn.Module: ...

    @abstractmethod
    def _get_act(self, layer: nn.Module) -> nn.Module: ...

    @abstractmethod
    def _get_fc2(self, layer: nn.Module) -> nn.Module: ...

    @abstractmethod
    def _get_rc2(self, layer: nn.Module) -> nn.Module: ...

    @abstractmethod
    def _get_pre_hooks_modules(self, layer: nn.Module) -> set[str]:
        r"""Returns the modules for which pre-hooks needs to be used in place of forward hooks.
        """
        ...

    def module_map(self) -> dict[str, nn.Module]:
        return {
            "ln1": self.ln1,
            "mha": self.mha,
            "rc1": self.rc1,
            "ln2": self.ln2,
            "fc1": self.fc1,
            "act": self.act,
            "fc2": self.fc2,
            "rc2": self.rc2,
        }
```

Figure 11: `TransformerBlockAdapter` implementation.

ViT. We conduct experiments on supervised ViT models (Dosovitskiy et al., 2021) pretrained on ImageNet-21K (Deng et al., 2009), the superset of the ILSVRC-2012 ImageNet dataset with 21k classes and 14 million images. We consider 3 variants with 86M, 307M, and 632M parameters, and a patch size of 16, 16, and 14, respectively. The input images have a resolution of 224 for all models. Following the original paper nomenclature (see Dosovitskiy et al., 2021, Section 4.1), we refer to those models as ViT-B, ViT-L, and ViT-H (we omit the patch size in the nomenclature since we consider one value for each variant in our work). All variants were pretrained from scratch. Their characteristics are given in Table 3, along with the link to the checkpoints used in our experiments.

Table 3: **Characteristics of ViT variants (Dosovitskiy et al., 2021).** We provide the number of layers, the embedding dimension, the FFN hidden dimension, the number of attention heads, the patch size, the number of parameters, and the link to the HuggingFace checkpoint.

Model	Layers	Emb. dim.	FFN dim.	Attn. heads	Patch size	Parameters	Checkpoint
ViT-B	12	768	3072	12	16	86M	vit-base
ViT-L	24	1024	4096	16	16	307M	vit-large
ViT-H	32	1280	5120	16	14	632M	vit-huge

DINOv2. We also consider DINOv2 models (Oquab et al., 2024) pretrained in discriminative self-supervised fashion. For DINOv2, we consider 4 variants with 22M, 86M, 307M, and 1.1B parameters. The patch size is fixed at 14, and the input images have a resolution 224 for all models. The corresponding backbones are referred to as ViT-S/14, ViT-B/14, ViT-L/14, and ViT-g/14, respectively. Since they all share the same backbone and patch size, we denote them by DINOv2-S, DINOv2-B, DINOv2-L, and DINOv2-g, by abuse of notation. The largest variant, DINOv2-g, is pretrained from scratch on 142M images from the LVD-142M dataset, while the others are distilled from DINOv2-g using knowledge distillation (Duval et al., 2023; Hinton et al., 2015). Their characteristics are given in Table 4, along with the link to the checkpoints used in our experiments.

Table 4: **Characteristics of DINOv2 variants (Oquab et al., 2024).** We provide the number of layers, the embedding dimension, the FFN hidden dimension, the number of attention heads, the patch size, the number of parameters, and the link to the HuggingFace checkpoint.

Model	Backbone	Layers	Emb. dim.	FFN dim.	Attn. heads	Patch size	Parameters	Checkpoint
DINOv2-S	ViT-S/14	12	384	1536	6	14	22M	dinov2-small
DINOv2-B	ViT-B/14	12	768	3072	12	14	86M	dinov2-base
DINOv2-L	ViT-L/14	24	1024	4096	16	14	307M	dinov2-large
DINOv2-g	ViT-g/16	40	1536	4096	24	14	1.1B	dinov2-giant

DINOv3. Similarly, we consider 4 variants of DINOv3 models (Siméoni et al., 2025), also pretrained in discriminative self-supervised fashion, with 22M, 86M, 307M, and 7B parameters. The patch size is fixed at 16, and the input images have a resolution 224 for all models. The corresponding backbones are referred to as ViT-S/16, ViT-B/16, ViT-L/16, and ViT-7B/16, respectively. Akin to DINOv2, we denote them by DINOv3-S, DINOv3-B, DINOv3-L, and DINOv3-7B, by abuse of notation. The largest variant, DINOv3-7B, is pretrained from scratch on 1.7B images from the LVD-1689M dataset, while the others are distilled from DINOv3-7B using knowledge distillation (Duval et al., 2023; Hinton et al., 2015). Their characteristics are given in Table 5, along with the link to the checkpoints used in our experiments.

Table 5: **Characteristics of DINOv3 variants (Siméoni et al., 2025).** We provide the number of layers, the embedding dimension, the FFN hidden dimension, the number of attention heads, the patch size, the number of parameters, and the link to the HuggingFace checkpoint.

Model	Backbone	Layers	Emb. dim.	FFN dim.	Attn. heads	Patch size	Parameters	Checkpoint
DINOv3-S	ViT-S/16	12	384	1536	6	16	22M	dinov3-small
DINOv3-B	ViT-B/16	12	768	3072	12	16	86M	dinov3-base
DINOv3-L	ViT-L/16	24	1024	4096	16	16	307M	dinov3-large
DINOv3-7B	ViT-7B/16	40	4096	8192	32	16	7B	dinov3-7B

AIMv2. Finally, we consider 4 variants of autoregressive AIMv2 models (Fini et al., 2025) with 307M, 632M, 1.2B, and 2.7B parameters, denoted by AIMv2-L, AIMv2-H, AIMv2-1B, and AIMv2-3B, respectively. Note that contrary to the iteration of autoregressive image models (El-Nouby et al., 2024, AIM), the AIMv2 family is multimodal and consists of a vision backbone coupled with a multimodal decoder. Our experiments focus only on the vision part. The patch size is fixed at 14, and the input images have a resolution of 224 for all models. All variants were pretrained from

scratch on a total of 12B image-text pairs from a combination of public and private datasets. Their characteristics are given in Table 6, along with the link to the checkpoints used in our experiments.

Table 6: **Characteristics of AIMv2 variants (Fini et al., 2025).** We provide the number of layers, the embedding dimension, the FFN hidden dimension, the number of attention heads, the patch size, the number of parameters, and the link to the HuggingFace checkpoint.

Model	Layers	Emb. dim.	FFN dim.	Attn. heads	Patch size	Parameters	Checkpoint
AIMv2-L	24	1024	1536	6	14	307M	aimv2-large
AIMv2-H	24	1536	3072	12	14	632M	aimv2-huge
AIMv2-1B	24	2048	4096	16	14	1.2B	aimv2-1B
AIMv2-3B	24	3072	8192	32	14	2.7B	aimv2-3B

Datasets. The vision experiments are conducted on a diverse set of 18 image classification benchmarks: Aircraft (Maji et al., 2013); Cifar10 and Cifar100 (Krizhevsky, 2009); DTD (Cimpoi et al., 2014); EuroSAT (Helber et al., 2019); Flowers102 (Nilsback & Zisserman, 2008); Food101 (Bossard et al., 2014); 3 variants from Cifar10-C with severity 5 (Hendrycks & Dietterich, 2019), namely Contrast, Motion Blur, Speckle Noise; Pets (Parkhi et al., 2012); the 5 domains of DomainNet (Peng et al., 2019), namely Clipart, Infograph, Quickdraw, Painting, Sketch; and 2 datasets from WILDS (Koh et al., 2021), namely iWildCam and RxRx1. 15 variants of the Cifar10-C benchmark (Hendrycks & Dietterich, 2019) are also used for experiments with controlled distribution shifts. Each variant consists of a corrupted version of the Cifar10 test set, with a type of corruption among Contrast, Gaussian Noise, Motion Blur, Snow, and Speckle Noise, and a level of corruption severity among 1, 3, and 5, from the lowest to the strongest. The datasets' characteristics are given in Table 7.

Table 7: **Characteristics of our 18 image classification benchmarks.** We provide the number of training and test images, as well as the number of classes. Contrast, Motion Blur, and Speckle Noise are corrupted variants of the Cifar10 test set with a corruption severity of 5, from the Cifar10-C benchmark (Hendrycks & Dietterich, 2019), Clipart, Infograph, Painting, Quickdraw, and Sketch are domains from the DomainNet benchmark (Peng et al., 2019). Cifar10 → Cifar10-C indicates that the linear probe is trained on the training set of Cifar10 and evaluated on Cifar10-C, i.e., on a corrupted version of the Cifar10 test set.

Dataset	Reference	Train	Test	Classes
Aircraft	Maji et al. (2013)	6 667	3 333	100
Cifar10	Krizhevsky (2009)	50 000	10 000	10
Cifar100	Krizhevsky (2009)	50 000	10 000	100
Clipart	Peng et al. (2019)	39 066	9 767	345
Contrast	Hendrycks & Dietterich (2019)	8 000	2 000	10
DTD	Cimpoi et al. (2014)	1 880	1 880	47
EuroSAT	Helber et al. (2019)	21 600	5 400	10
Flowers102	Nilsback & Zisserman (2008)	1 020	6 149	102
Food101	Bossard et al. (2014)	75 750	25 250	101
Infograph	Peng et al. (2019)	42 560	10 641	345
iWildCam	Koh et al. (2021)	129 809	42 791	182
Motion Blur	Hendrycks & Dietterich (2019)	8 000	2 000	10
Painting	Peng et al. (2019)	60 607	15 152	345
Pets	Parkhi et al. (2012)	3 680	3 669	37
Quickdraw	Peng et al. (2019)	138 000	34 500	345
RxRx1	Koh et al. (2021)	40 612	34 432	1 139
Sketch	Peng et al. (2019)	56 308	14 078	345
Speckle Noise	Hendrycks & Dietterich (2019)	8 000	2 000	10
Cifar10 → Cifar10-C	Hendrycks & Dietterich (2019)	50 000	10 000	10

Data preprocessing. We follow the preprocessing protocol of Oquab et al. (2024, Section 7.2): we apply a 224×224 resizing, matching our models' native resolution, before center cropping images. All images are then normalized using the ImageNet1k (Deng et al., 2009) statistics. It ensures images with mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$. Note that no data augmentation is used. They are indeed not necessary since training is not stochastic: the feature extraction is deterministic, and the classifier is fitted once over all the frozen training features and evaluated once on the corresponding test features (data augmentation would, however, be required with an SGD-based linear probing (see Marouani et al., 2026)). For datasets that do not have predefined training and test sets, e.g., EuroSAT, or datasets from Cifar10-C and DomainNet, we manually create *deterministic* training and test sets following an 80%-20% split. The deterministic part is crucial to ensure no data leakage.

Evaluation. We adopt the same linear probing implementation as in DINOv2[3] by training a logistic regression on frozen features for at most 1000 iterations, with L-BFGS solver, an ℓ_2 penalty, a regularization strength $C = 1.0$, and a tolerance of $1\text{e}{-4}$. Representations are preprocessed by removing the mean and scaling to unit variance with a standard scaler. We ensure efficient evaluation by making use of the GPU-friendly implementation of the `StandardScaler` and `LogisticRegression` functions from the CuML library (Raschka et al., 2020). For standard evaluation, training is done on the training set of each dataset, and evaluation is done on the corresponding test set. The reported performance is the top-1 test accuracy. For the robustness analysis on Cifar10-C, akin to DINOv3 domain generalization experiments (see Siméoni et al., 2025, Section 6.2.1), training is done on the training set of Cifar10, and the learned classifier is then evaluated independently on the 15 corrupted versions of the Cifar10 test set described in the "Datasets" paragraph above.

B.3 Large language models

This section provides details on LLMs' transformer representations and the linear probing setup.

B.3.1 Transformer representations

The `MMTEB` benchmark (Enevoldsen et al., 2025), which expands the `MTEB` benchmark (Muennighoff et al., 2023) with additional domains, tasks, and languages, requires defining an `encode` method for a given LLM, which is then used to evaluate representations over a predefined set of tasks. As such, we set the desired (layer, module) pair via a `set_probing` method, use PyTorch hooks and pre-hooks in the `encode` to extract representations during the forward pass, and recover the results over the corresponding set of tasks. In our code, all LLMs inherit from the abstract base `BaseLanguageModel` displayed in Figure 12. Abstract and hook methods are similar to `BaseVisionModel` (not shown due to space limitations), with an additional `_load_tokenizer` to load the tokenizer. The feature extraction is done with the `encode` method. Similarly to VFMs (see Appendix B.2.1), the most important abstract method to implement is `block_adapter`, which defines the 8 representations to extract within each transformer block and inherits from the base class `TransformerBlockAdapter` displayed in Figure 11. We provide below the implementation details on the LLMs used in our experiments.

Qwen3. The Qwen3 architecture (Yang et al., 2025) consists of a succession of transformer decoders with RMSnorms (Zhang & Sennrich, 2019), Rotary Positional Embeddings (RoPE, Su et al., 2024) and a gated feedforward network with SwiGLU activation (Shazeer, 2020). The checkpoints used in our experiments follow the HuggingFace transformers (Wolf et al., 2020) implementation available at `transformers/models/qwen3/modeling_qwen3.py`.

Llama3. The Llama3 architecture (Grattafiori et al., 2024) largely follows the Llama original one (Touvron et al., 2023). Similarly to Qwen3, it consists of a succession of transformer decoders with RMSnorms (Zhang & Sennrich, 2019), Rotary Positional Embeddings (RoPE, Su et al., 2024) and a gated feedforward network with SwiGLU activation (Shazeer, 2020). The checkpoints used in our experiments follow the HuggingFace transformers (Wolf et al., 2020) implementation available at `transformers/models/llama/modeling_llama.py`.

[3] Code available at `facebookresearch/dinov2/blob/main/dinov2/eval/log_regression.py`

```python
class BaseLanguageModel(nn.Module, ABC):
    r"""Abstract base class for LLMs. Abstract and hook methods (not shown) are akin to BaseVisionModel."""

    def __init__(self):
        super().__init__()
        self.device: str
        self.model: PreTrainedModel
        self.tokenizer: PreTrainedTokenizerBase

        # Default probing done on the last residual stream
        self.layer: int = -1
        self.module: str = "rc2"

    @torch.inference_mode()
    def _get_hidden_states(self, input_ids: torch.Tensor, attention_mask: torch.Tensor) -> torch.Tensor:
        r"""Single forward pass recovering the sequence of tokens of shape (N, L, E) after the specified layer and module."""
        embeddings: dict[str, torch.Tensor] = {}
        handles: list[torch.utils.hooks.RemovableHandle] = []

        # Recover embedding from the specified layer and module
        key = f"block_{self.layer}_{self.module}"
        adapter = self.block_adapter(self.blocks[self.layer])
        module_obj = adapter.module_map()[self.module]

        if self.module in adapter.pre_hooks:
            handles.append(module_obj.register_forward_pre_hook(self._make_pre_hook(embeddings=embeddings, key=key)))
        else:
            handles.append(module_obj.register_forward_hook(self._make_hook(embeddings=embeddings, key=key)))

        try:
            self.model(input_ids=input_ids, attention_mask=attention_mask)
        finally:
            for handle in handles:
                handle.remove()

        return embeddings[key]

    def _encode_sentences(self, sentences: list[str], batch_size: int = 256) -> torch.Tensor:
        r"""Encode a flat list of sentences into mean-pooled embeddings."""
        all_embeddings = []
        for i in range(0, len(sentences), batch_size):
            text = sentences[i : i + batch_size]

            # Recover encoding, input_ids and padding tokens (attention_mask)
            encoding = self.tokenizer(text=text, padding=True, truncation=True, return_tensors="pt").to(self.device)
            input_ids = encoding.input_ids
            attention_mask = encoding.attention_mask

            # Recover hidden states
            hidden_states = self._get_hidden_states(input_ids=input_ids, attention_mask=attention_mask)

            # Recover pooled embeddings
            embeddings = self.pooling(
                hidden_states=hidden_states,
                attention_mask=attention_mask,
                input_ids=input_ids,
                special_token_ids=self.special_token_ids,
            )
            all_embeddings.append(embeddings)

        return torch.cat(all_embeddings, dim=0)

    def encode(self, inputs: DataLoader[BatchedInput] | list[str], *, task_metadata: TaskMetadata, hf_split: str, hf_subset:
          str, prompt_type: PromptType | None = None, **kwargs: Unpack[EncodeKwargs]) -> Array:
        r"""Encode inputs compatible with MTEB v2 EncoderProtocol."""
        if isinstance(inputs, DataLoader):
            all_embeddings = []
            for batch in inputs:
                batch_embeddings = self._encode_sentences(sentences=batch["text"])
                all_embeddings.append(batch_embeddings)
            return torch.cat(all_embeddings, dim=0)
        elif isinstance(inputs, list):
            return self._encode_sentences(inputs)
        else:
            raise ValueError("Unsupported input type. Expected DataLoader or list of strings.")

    def set_probing(self, layer: int | None, module: str | None) -> None:
        r"""Set the layer and module to probe for embedding extraction."""
        if layer is not None:
            if layer >= self.n_layers:
                raise ValueError(f"Layer index {layer} is out of range for a model with {self.n_layers} layers.")
            self.layer = layer
        if module is not None:
            if module not in TRANSFORMER_MODULES:
                raise ValueError(f"Invalid module '{module}'. Options are {TRANSFORMER_MODULES}.")
            self.module = module
```

Figure 12: `BaseLanguageModel` implementation.

Gemma3. The Gemma3 architecture (Kamath et al., 2025) is akin to the Qwen3 and Llama3 ones, except that the gated FFN uses an approximated GeGLU activation (Shazeer, 2020) and that both the input and output of the attention and feedforward blocks are normalized. For a consistent evaluation, the LN1 and LN2 representations correspond to the pre-norms.

Feature extraction. The feature extraction follows the official implementation available at `https://github.com/embeddings-benchmark/mteb/`. Documents are embedded with the `encode` method, and the resulting representations are fed to a logistic regression.

B.3.2 Probing setup

Qwen3. We conduct experiments on dense general-purpose Qwen3 models (Yang et al., 2025) with a vocabulary size of 150K. We consider 3 variants with 1.7B, 4B and 8B parameters, referred to as Qwen3 1.7B, Qwen3 4B, and Qwen3 8B, respectively. All variants were pretrained from scratch on 36T tokens from a multilingual dataset with 119 languages and dialects. Their characteristics are given in Table 8, along with the link to the checkpoints used in our experiments.

Table 8: **Characteristics of Qwen3 variants (Yang et al., 2025).** We provide the number of layers, the embedding dimension, the FFN hidden dimension, the number of attention heads, the context length, the number of parameters, and the link to the HuggingFace checkpoint.

Model	Layers	Emb. dim.	FFN dim.	Attn. heads	Context length	Parameters	Checkpoint
Qwen3 1.7B	28	2048	6144	16	32K	1.72B	Qwen3-1.7B
Qwen3 4B	36	2560	9728	32	128K	4.02B	Qwen3-4B
Qwen3 8B	36	4096	12288	32	128K	8.19B	Qwen3-8B

Llama3. We conduct experiments on dense general-purpose Llama3 models (Grattafiori et al., 2024) with a vocabulary size of 128K. We consider 3 variants with 1B, 3B and 8B parameters, referred to as Llama3 1B, Llama3 3B, and Llama3 8B, respectively[4]. The 8B variant was pretrained from scratch on 15T multilingual tokens. The 1B and 3B variants are distilled from the 8B and the 70B models. Their characteristics are given in Table 9, along with the link to the checkpoints used in our experiments.

Table 9: **Characteristics of Llama3 variants (Grattafiori et al., 2024).** We provide the number of layers, the embedding dimension, the FFN hidden dimension, the number of attention heads, the context length, the number of parameters, and the link to the HuggingFace checkpoint.

Model	Layers	Emb. dim.	FFN dim.	Attn. heads	Context length	Parameters	Checkpoint
Llama3 1B	16	2048	8192	32	128K	1.24B	Llama3.2 1B
Llama3 3B	28	3072	9728	24	128K	3.21B	Llama3.2 3B
Llama3 8B	32	4096	14336	32	128K	8.03B	Llama3.1 8B

Gemma3. We conduct experiments on dense general-purpose Gemma3 models (Kamath et al., 2025) with a vocabulary size of 262K. We consider 3 variants with 1B, 4B and 12B parameters, referred to as Gemma3 1B, Gemma3 4B, and Gemma3 12B, respectively. All variants were pretrained from scratch on 2T, 4T, and 12T multilingual tokens. Their characteristics are given in Table 10, along with the link to the checkpoints used in our experiments.

Tasks. We consider classification tasks from the `MTEB(Multilingual)` benchmark with at most 8 192 documents. This leads to a total of 24 tasks spanning 14 domains and 31 languages. The characteristics of the tasks are given in Table 11.

[4]The official model variant is Llama3.2 for the 1B and 3B sizes, and Llama3.1 for the 8B one

Table 10: **Characteristics of Gemma3 variants (Kamath et al., 2025).** We provide the number of layers, the embedding dimension, the FFN hidden dimension, the number of attention heads, the number of parameters, and the link to the HuggingFace checkpoint.

Model	Layers	Emb. dim.	FFN dim.	Attn. heads	Context length	Parameters	Checkpoint
Gemma3 1B	26	1152	6912	4	128K	1.00B	gemma-3-1b-pt
Gemma3 4B	34	2560	10240	8	128K	4.30B	gemma-3-4b-pt
Gemma3 12B	48	3840	15360	16	128K	12.19B	gemma-3-12b-pt

Table 11: **Characteristics of `MTEB(Multilingual)` classification tasks.** Experiments are conducted on the 24 tasks below, spanning a diverse set of languages and domains. The column names follow Enevoldsen et al. (2025), and the list of ISO 639 codes can be found at https://en.wikipedia.org/wiki/List_of_ISO_639_language_codes.

Task	Reference	Languages	Domains	Sample creation	Samples
BulgarianStoreReviewSentimentClassification	Georgieva-Trifonova et al. (2018)	bul	Reviews, Written	found, human-annotated	182
DalajClassification	Volodina et al. (2021)	swe	Non-fiction, Written	created, expert-annotated	888
DBpediaClassification	Zhang et al. (2015)	eng	Encyclopaedic, Written	found, derived	2 048
EstonianValenceClassification	Pajupuu et al. (2023)	est	News, Written	found, human-annotated	818
FinancialPhrasebankClassification	Malo et al. (2014)	eng	News, Written, Financial	found, expert-annotated	2 264
GujaratiNewsClassification	-	guj	News, Written	found, derived	1 318
IndonesianIdClickbaitClassification	William & Sari (2020)	ind	News, Written	found, expert-annotated	2 048
IsiZuluNewsClassification	Madodonga et al. (2023)	zul	News, Written	found, human-annotated	752
ItaCaseholdClassification	Licari et al. (2023)	ita	Legal, Government, Written	found, expert-annotated	221
KorSarcasmClassification	Kim & Cho (2019)	kor	Social, Written	found, expert-annotated	2 048
KurdishSentimentClassification	Badawi et al. (2024)	kur	Web, Written	found, derived	1 987
NepaliNewsClassification	Arora (2020)	nep	News, Written	found, derived	2 048
NusaX-senti	Winata et al. (2023)	ace, ban, bbc, ...	Reviews, Web, Social, ...	found, expert-annotated	4 800
PAC	Łukasz Augustyniak et al. (2022)	pol	Legal, Written	-	3 453
PoemSentimentClassification	Sheng & Uthus (2020)	eng	Reviews, Written	found, human-annotated	209
PolEmo2.0-OUT	-	pol	Written, Social	-	494
PunjabiNewsClassification	Kunchukuttan et al. (2020)	pan	News, Written	found, derived	157
ScalaClassification	Nielsen (2023)	dan, nno, nob, ...	Fiction, News, Non-fiction, ...	created, human-annotated	8 192
SentimentAnalysisHindi	Parida et al. (2023)	hin	Reviews, Written	found, derived	2 048
SinhalaNewsClassification	de Silva (2015)	sin	News, Written	found, derived	2 048
SiswatiNewsClassification	Madodonga et al. (2023)	ssw	News, Written	found, human-annotated	80
SwahiliNewsClassification	David (2020)	swa	News, Written	found, derived	2 048
TswanaNewsClassification	Marivate et al. (2023)	tsn	News, Written	found, derived	487
TweetTopicSingleClassification	Antypas et al. (2022)	eng	Social, News, Written	found, expert-annotated	1 693

Evaluation. The `MMTEB` evaluation protocol applies linear probing via a logistic regression on frozen features for at most 100 iterations. It uses the `LogisticRegression` method from Sklearn (Pedregosa et al., 2011), implemented with the L-BFGS solver, an ℓ_2 penalty, a regularization strength $C = 1.0$, and a tolerance of $1e{-}4$. Quoting Enevoldsen et al. (2025), "a train set is constructed by sampling n (8-16) samples for each label. If only a test set is available, a section is split off as a training set. Both sets are then embedded and used to train a logistic regression using a maximum of 100 iterations. Afterwards, performance metrics are calculated. For robustness, this process is repeated 10 times".

C Additional Experiments

In this section, we provide details on the experiments in the main paper and on additional experiments.

C.1 Vision Foundation Models

The results presented in this section relate to the experiments conducted on VFMs in Section 3.1.

C.1.1 Intermediate Layers under Distribution Shift

Skean et al. (2025) observed the benefits of intermediate layers in LLMs but not in VFMs, where performance increased towards the final layers, except for autoregressive models such as AIM (El-Nouby et al., 2024). Recent work, however, suggests that intermediate representations can also be beneficial in non-autoregressive VFMs (Bolya et al., 2025), particularly under distribution shifts (Uselis & Oh, 2025). In practice, discrepancies between pretraining and downstream data can degrade performance by making the extracted features less informative (Quiñonero-Candela et al., 2009). To illustrate this effect, we consider a ViT-B (Dosovitskiy et al., 2021) pretrained on ImageNet-21K and probe it before and after finetuning on downstream data. The experimental setup is described below.

Datasets. We conduct experiments on Cifar10 (Krizhevsky, 2009); 2 variants from Cifar10-C (Hendrycks & Dietterich, 2019) with severity 5 (Contrast and Speckle Noise), and Flowers102 (Nilsback & Zisserman, 2008). The preprocessing follows Dosovitskiy et al. (2021) and Kolesnikov et al. (2020): for training data, we apply random cropping, a 224×224 image resizing, and random horizontal flip. For validation and test data, the 224×224 image resizing is applied before center cropping images. All images are normalized using the ImageNet (Deng et al., 2009) statistics. It ensures images with mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$. For Cifar10-C datasets that do not have predefined training and test sets, we manually create *deterministic* training and test sets following an 80%-20% split. The deterministic part is crucial to ensure no data contamination.

Finetuning setup. Our finetuning experiments follow the protocol from Dosovitskiy et al. (2021) with a resolution of 224×224. We optimize models with Stochastic Gradient Descent (SGD), a momentum of 0.9, no weight decay, a cosine learning rate decay, a batch size of 512, and gradient clipping at norm 1. For each pair of dataset-configuration, we perform a sweep over 4 learning rates, as summarized in Table 12, and conduct 3 runs with different seeds relative to network initialization and dataloaders. For each run, we monitor the training using a validation set (20% of the training set). The final performance is the test accuracy of the checkpoint that achieves the best validation accuracy.

Results. In Figure 13, we order datasets by decreasing finetuning accuracy, using it as a rough proxy for increasing distribution shift: $99.15_{\pm 0.05}$ on Flowers102, $99.02_{\pm 0.02}$ on Cifar10, $97.23_{\pm 0.18}$ on Contrast, and $89.58_{\pm 0.43}$ on Speckle Noise. We can see that the final-layer residual stream representations remain strong on near in-distribution datasets, while intermediate representations become increasingly beneficial under distribution shift. After finetuning, in contrast, performance increases monotonically towards the final layers. This shows that, akin to LLMs, intermediate representations can be effective in VFMs, and that their usefulness depends on the discrepancy between pretraining and downstream data.

Table 12: **Finetuning setup**. We report the optimizer, batch size, training steps, and learning rates.

Dataset	Optimizer	Batch size	Training steps	Learning rates η
Cifar10	SGD	512	10000	{1e−3, 3e−3, 1e−2, 3e−2}
Contrast	SGD	512	10000	{1e−3, 3e−3, 1e−2, 3e−2}
Speckle Noise	SGD	512	10000	{1e−3, 3e−3, 1e−2, 3e−2}
Flowers102	SGD	512	5000	{1e−3, 3e−3, 1e−2, 3e−2}

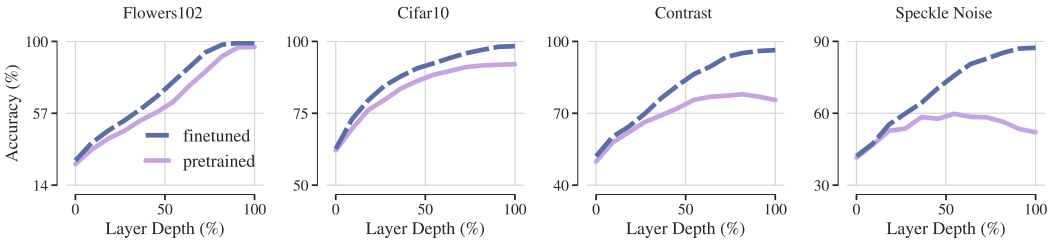


Figure 13: **Intermediate layers under distribution shift.** Evolution of the linear probing performance across layers of a ViT-B (86M parameters) pretrained on ImageNet-21K. The solid line is the accuracy on the pretrained model, and the dashed line is the accuracy on the same model after finetuning on the dataset at hand. Datasets are ordered by decreasing finetuning performance.

C.1.2 Gains across pretraining scales

Following Hoffmann et al. (2022), we measure the pretraining scale of a model as the total number of (visual) tokens seen during pretraining. Table 13 summarizes the number of images seen during pretraining, using the figures reported in the official paper and documentation of each model. For ViT models, the paper only reports the number of epochs (90). Since the batch size is 4096 and pretraining is done on ImageNet-21K, which contains 14M images, this leads to $\sim 307\,617$ training iterations.

Table 13: **Number of images seen during pretraining.** We report the batch size, number of training iterations, and total number of images seen during pretraining for each model, along with the official reference for the figures.

Model	Reference	Batch size	Iterations	Pretraining images
ViT	Dosovitskiy et al. (2021)	4 096	307 617	1 259 999 232
DINOv2	Oquab et al. (2024)	3 072	625 000	1 920 000 000
DINOv3	Siméoni et al. (2025)	4 096	1 000 000	4 096 000 000
AIMv2	Fini et al. (2025)	8 192	1 500 000	12 288 000 000

Then, we need to recover the number of visual tokens per image. For ViT, this only depends on the image resolution and patch size. Taking into account the `CLS` token, each image contains $\frac{HW}{P^2} + 1$ tokens, where H, W, and P denote the height, width, and patch size, respectively. For AIMv2, the computation is similar, without the $+1$, since autoregressive models (El-Nouby et al., 2024; Fini et al., 2025) do not use a `CLS` token. For DINOv2 and DINOv3, we need to take into account the multi-crop strategy of Caron et al. (2020): for each image, the student model sees 2 global crops and 8 local crops, each contributing $\frac{HW}{P^2}$ tokens, with the height and width of the crop. On top of these, each crop includes a `CLS` token for DINOv2, and a `CLS` token and 4 registers for DINOv3, leading to an overhead of 10 and 50 tokens per image, respectively. Recalling that images have resolution 224×224, the number of tokens per image can thus be summarized as follows: Putting Tables 13

Table 14: **Number of visual tokens per image.** We report for each model the patch size, image resolution, global and local crops resolution (when supported) and the corresponding amount of visual tokens per image.

Model	Size	Patch size	Image resolution	Global crops (nb. - resolution)	Local crops	Nb. of tokens per image
ViT	B	16	224×224	-	-	197
	L	16	224×224	-	-	197
	H	14	224×224	-	-	257
DINOv2	S	14	224×224	2 - 224×224	8-98×98	914
	B	14	224×224	2 - 224×224	8-98×98	914
	L	14	224×224	2 - 224×224	8-98×98	914
	g	14	224×224	2 - 224×224	8-98×98	914
DINOv3	S	16	224×224	2 - 256×256	8-112×112	954
	B	16	224×224	2 - 256×256	8-112×112	954
	L	16	224×224	2 - 256×256	8-112×112	954
	7B	16	224×224	2 - 256×256	8-112×112	954
AIMv2	L	14	224×224	-	-	256
	H	14	224×224	-	-	256
	1B	14	224×224	-	-	256
	3B	14	224×224	-	-	256

and 14 together provides the number of visual tokens seen during pretraining: 0.25T for ViT, 1.75T for DINOv2, 3.90T for DINOv3, and 3.15T for AIMv2. While the exact number varies within the ViT family due to the larger sequence length of the Huge variant, the order of magnitude is the same (0.25T compared to 0.32T).

C.1.3 Gain across model families

In this section, we compare the improvement over the probing baseline across model families. Figure 14 displays the gains of searching over layers alone ($\mathscr{G}_{\text{layer}}$), modules alone ($\mathscr{G}_{\text{module}}$), and both jointly ($\mathscr{G}_{\text{joint}}$) on the Large variant (307M parameters) of each family. Results are presented on Cifar10, Sketch, and Quickdraw, in increasing order of transfer difficulty (see Figure 6a). We can see that on Cifar10, gains are small for all models. By contrast, as transfer difficulty increases, the benefits of layers and modules become more salient. Notably, the choice of modules matters on Sketch, especially for ViT, which has the smallest pretraining scale. Moreover, intermediate layers substantially outperform the final layer for all model families on Quickdraw.

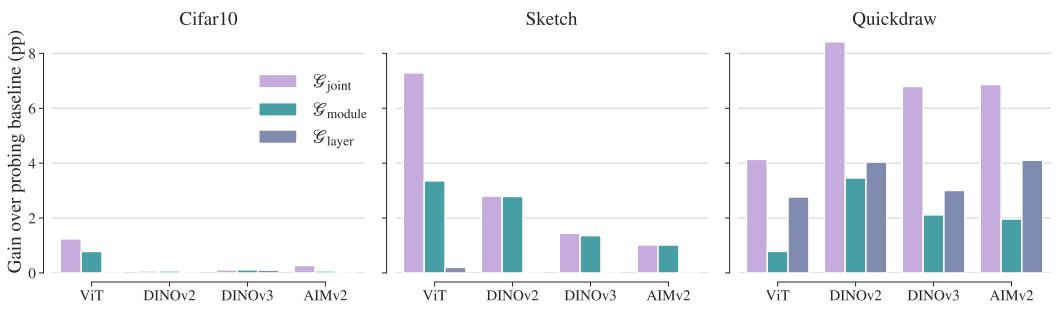


Figure 14: **Gain over model families.** The three types of gains ($\mathscr{G}_{\text{layer}}$, $\mathscr{G}_{\text{module}}$, and $\mathscr{G}_{\text{joint}}$) are displayed on the representative datasets spanning low, medium, and high transfer difficulty (Cifar10, Sketch, Quickdraw). Results are reported on the Large variant (307M parameters) of each family. The magnitude of gains increases with transfer difficulty, consistent with Figure 6a.

C.1.4 Module comparison

In Table 15, we show the percentage of VFM evaluations where a given module belongs to the layer-module pair that achieves the oracle gain $\mathscr{G}_{\text{joint}}$. It is referred to as the winning frequency. We obtain:

Table 15: **Module comparison.** Percentage of evaluations, out of a total of 270, for which a given module belongs to the layer-module pair achieving the oracle gain $\mathscr{G}_{\text{joint}}$.

Module	LN1	MHA	RC1	LN2	FC1	Act	FC2	RC2
winning frequency (%)	6.7	7.0	3.3	9.3	0.4	**64.8**	2.2	6.3

C.1.5 Performance along the depth

In this section, we display the performance of `Act` and `RC2` over depth. Results are consistent with Figure 4: the `Act` module shows strong performance, often surpassing the baseline `RC2`. The optimal depth is generally near the end of the network, but not systematically at the final layer. This is particularly salient for ViT-B, for which intermediate `Act` layers consistently outperform the final layer, whose performance drops, explaining the low performance of `Act@100` observed in Table 17. Nevertheless, as shown in Figures 15 to 17, intermediate `Act` representations in ViT models surpass the probing baseline `RC2@100` for most model-dataset pairs. This is consistent with Figure 5: while the fixed strategy `Act@100` closes nearly half of the gap to the $\mathscr{G}_{\text{joint}}$ gain, allowing the `Act` depth to vary increases recovery to 78.6%.

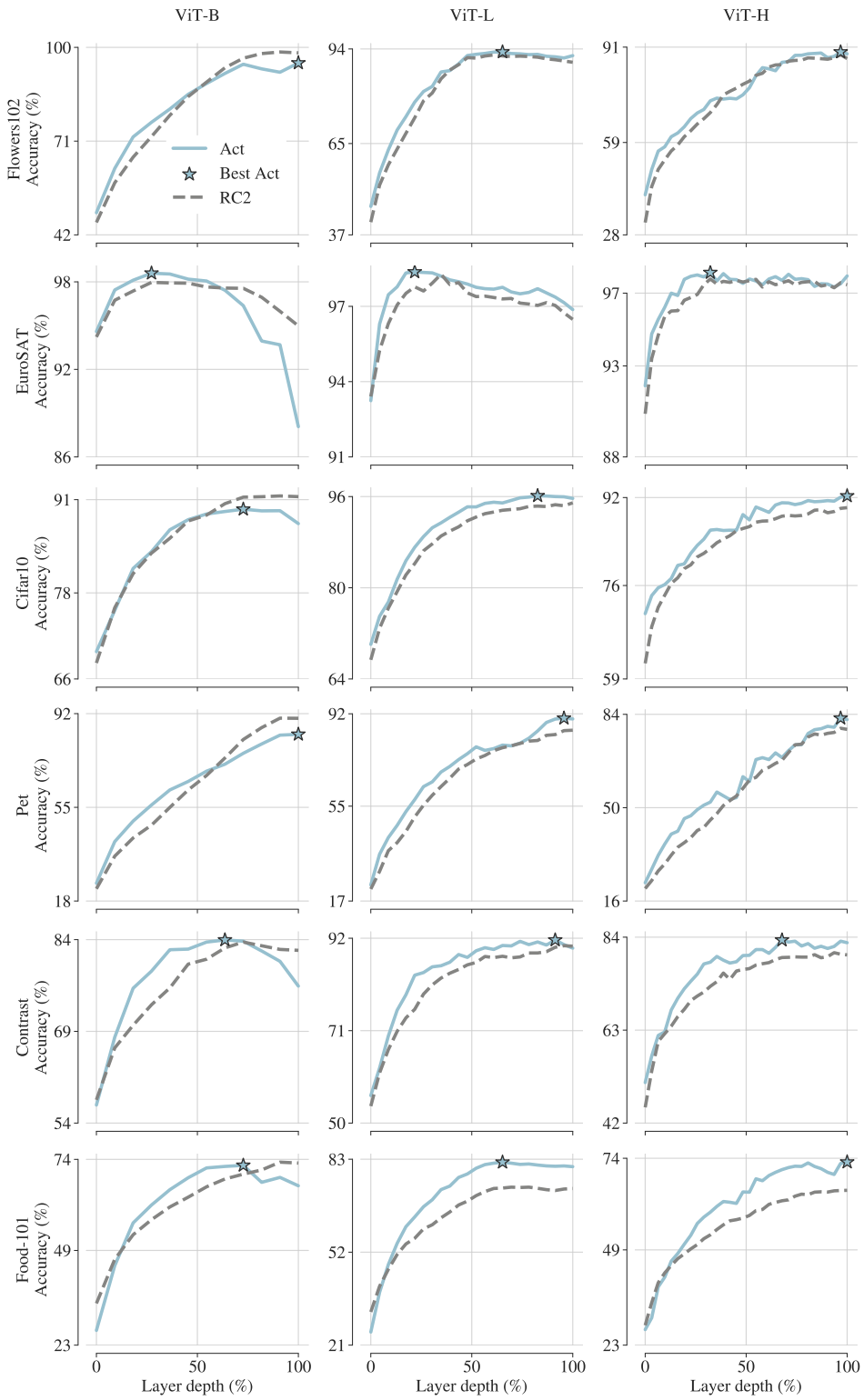


Figure 15: **Impact of `Act` and intermediate layers in ViT (first part).** Probing accuracy across layers of ViT models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

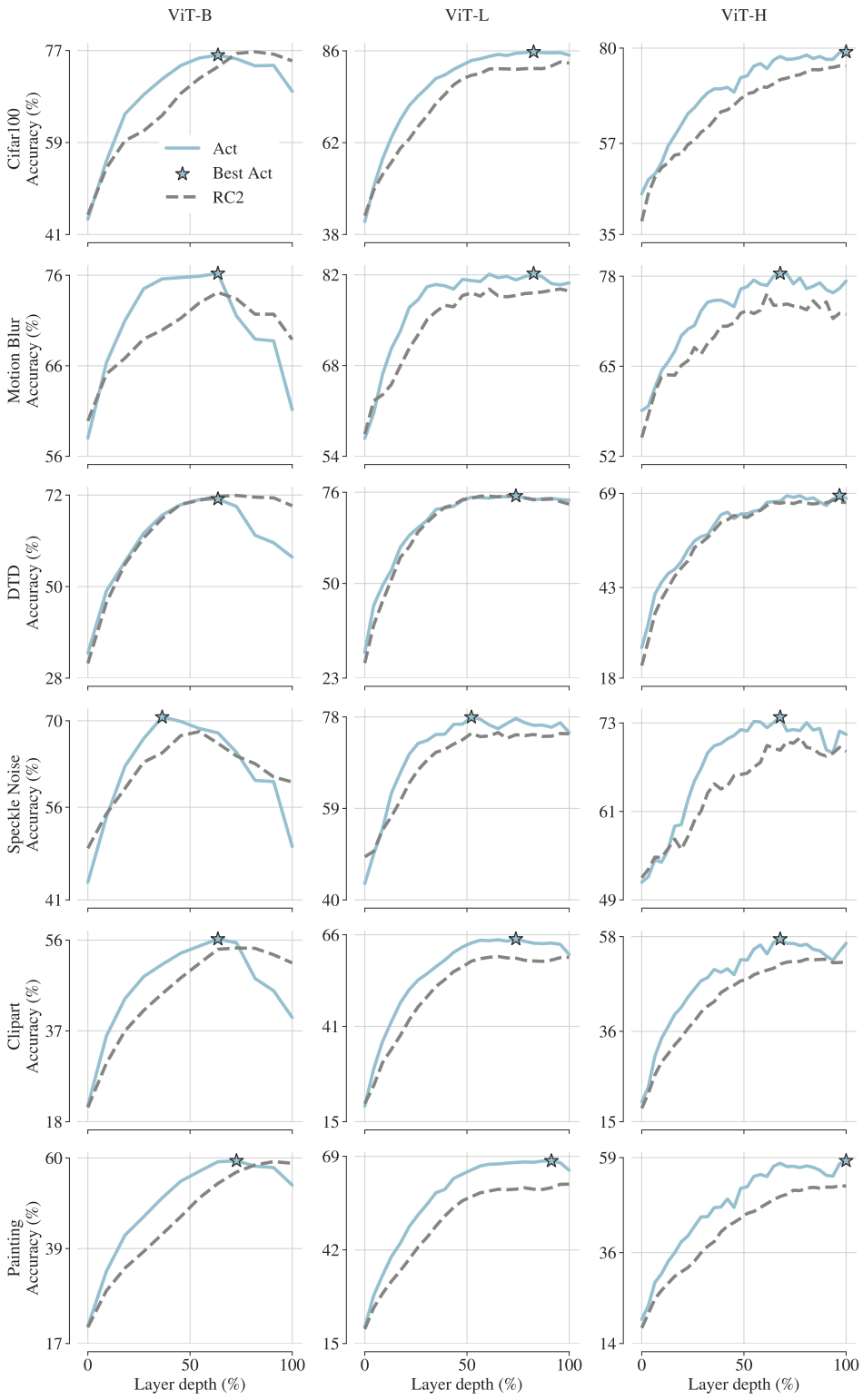


Figure 16: **Impact of `Act` and intermediate layers in ViT (second part).** Probing accuracy across layers of ViT models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

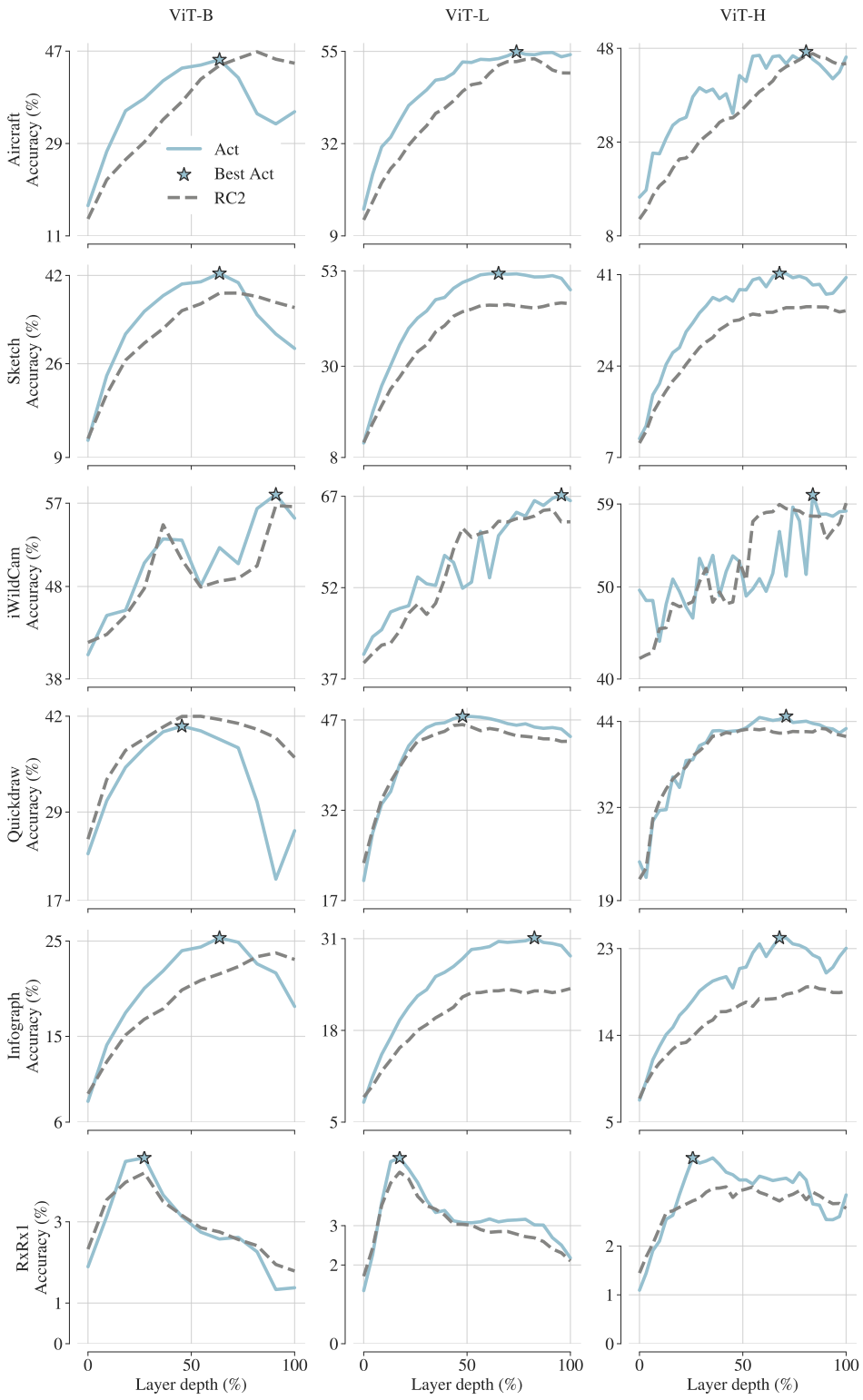


Figure 17: **Impact of `Act` and intermediate layers in ViT (last part).** Probing accuracy across layers of ViT models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

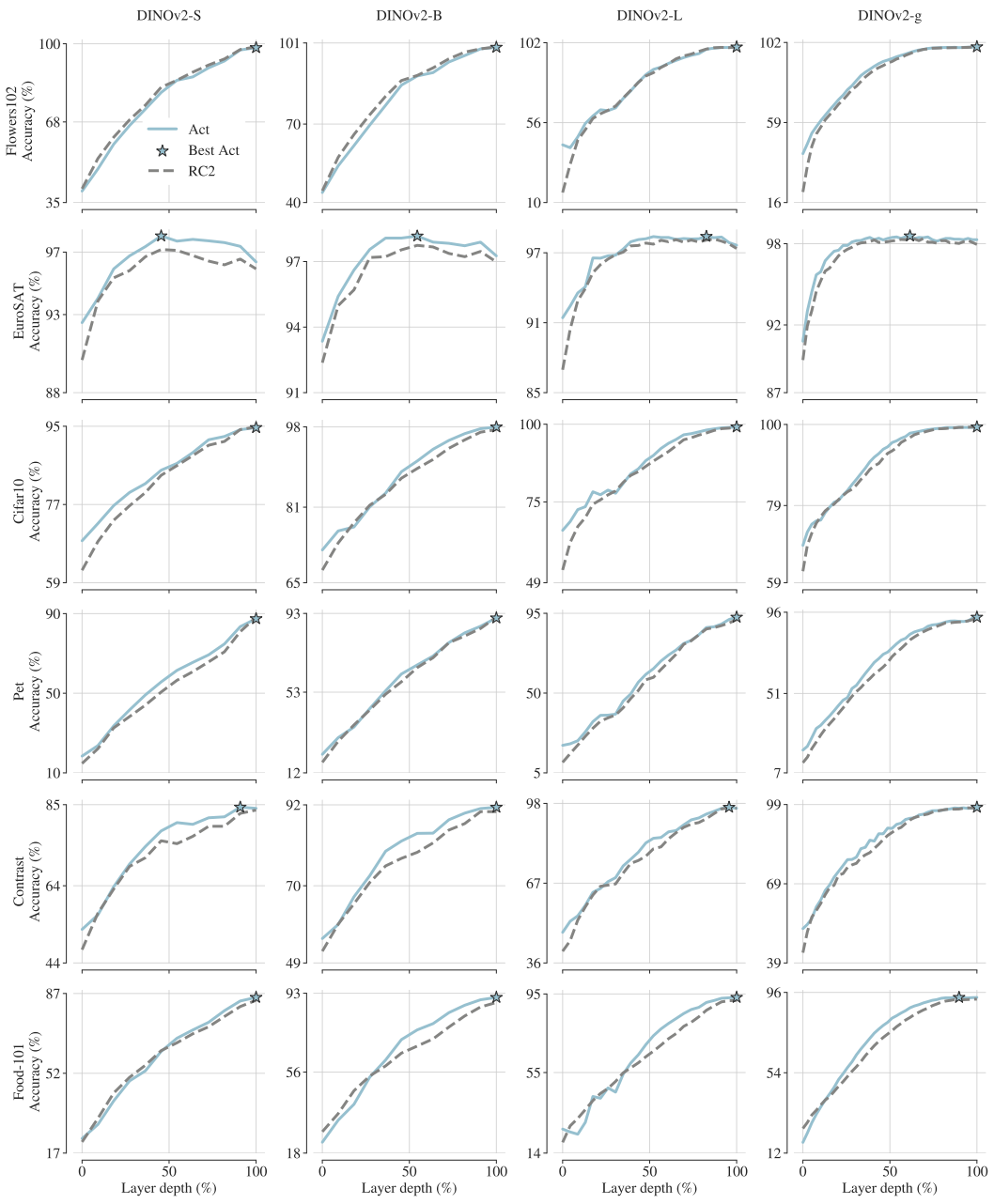


Figure 18: **Impact of `Act` and intermediate layers in DINOv2 (first part).** Probing accuracy across layers of DINOv2 models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

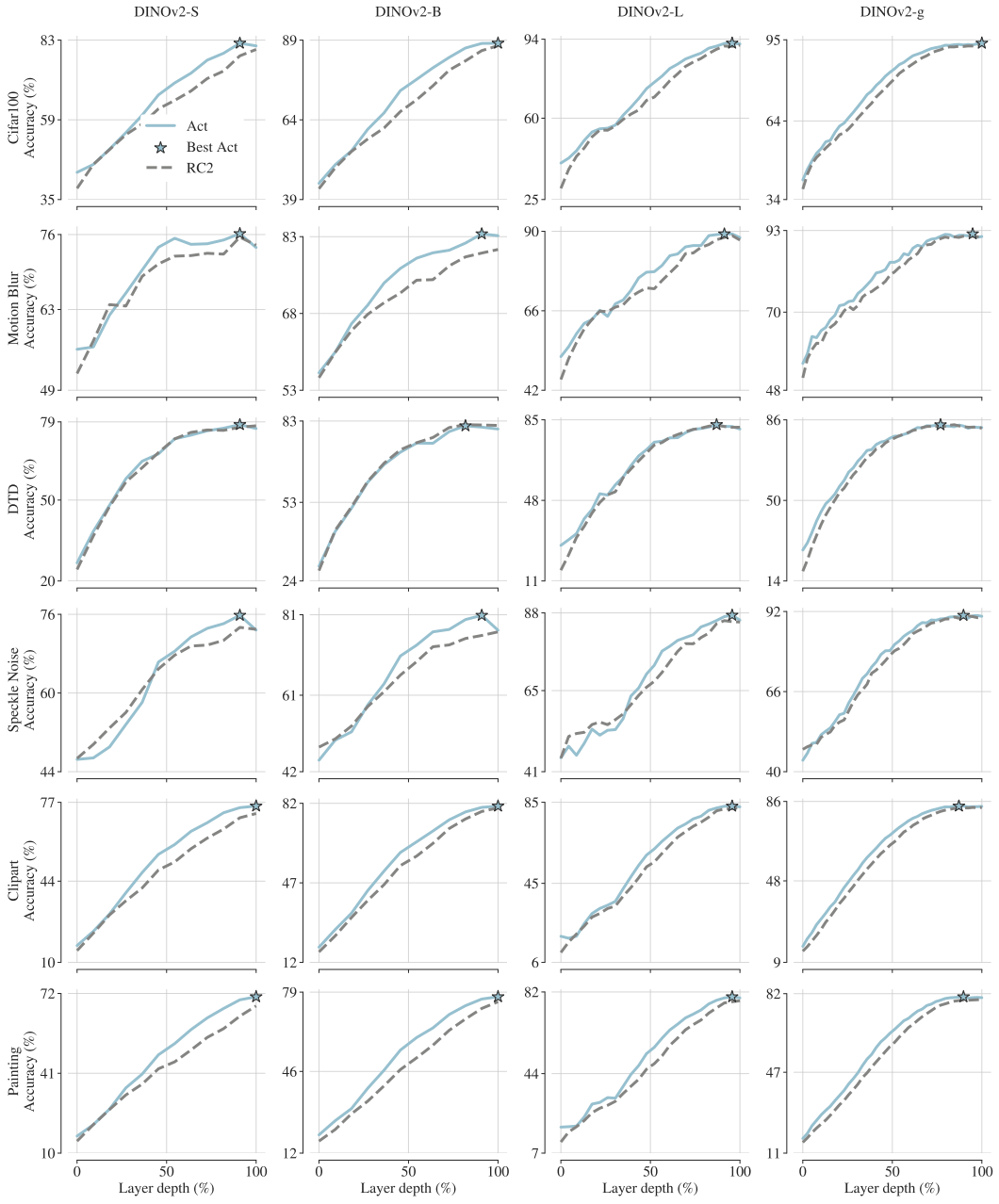


Figure 19: **Impact of `Act` and intermediate layers in DINOv2 (second part).** Probing accuracy across layers of DINOv2 models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

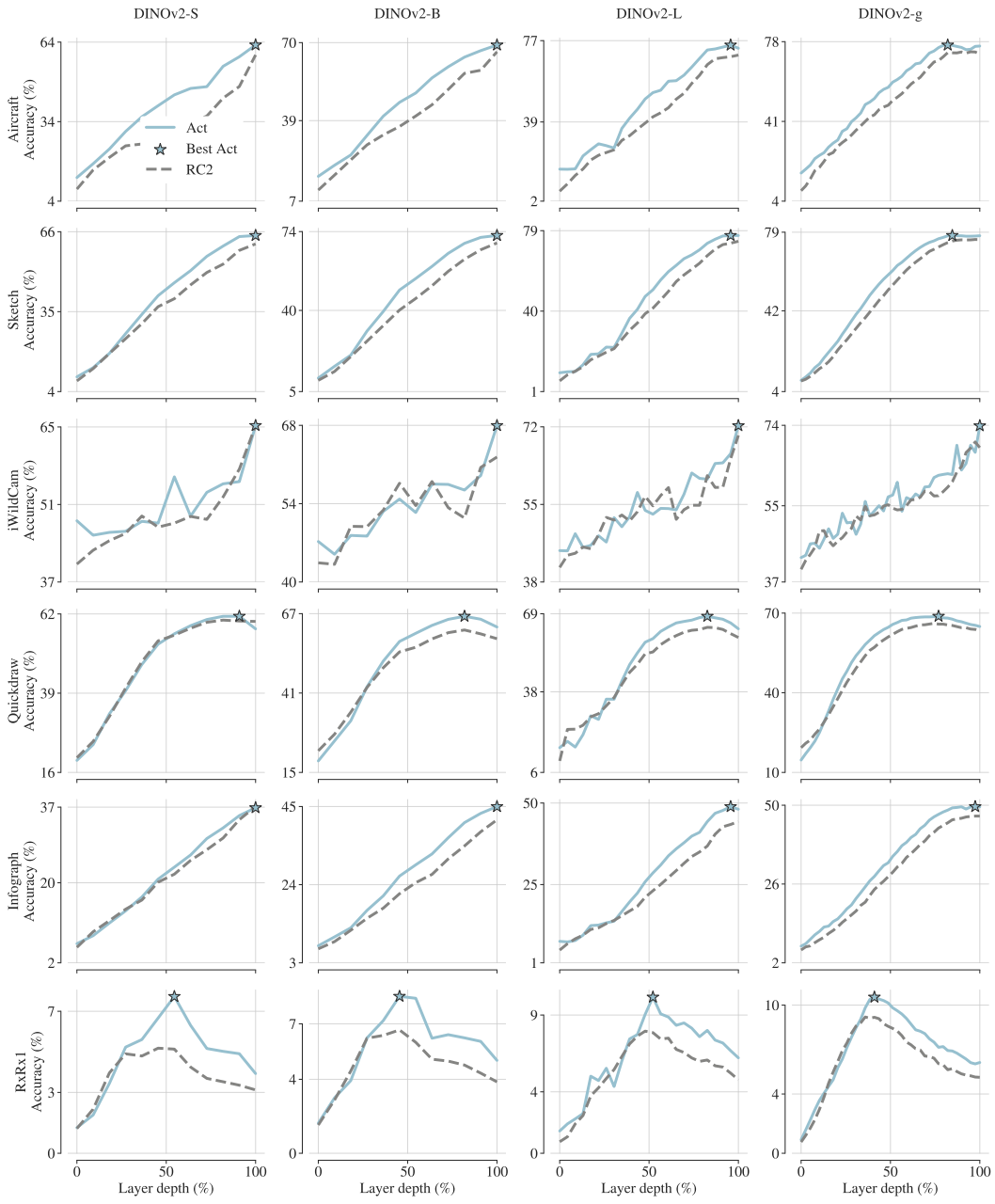


Figure 20: **Impact of `Act` and intermediate layers in DINOv2 (last part).** Probing accuracy across layers of DINOv2 models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

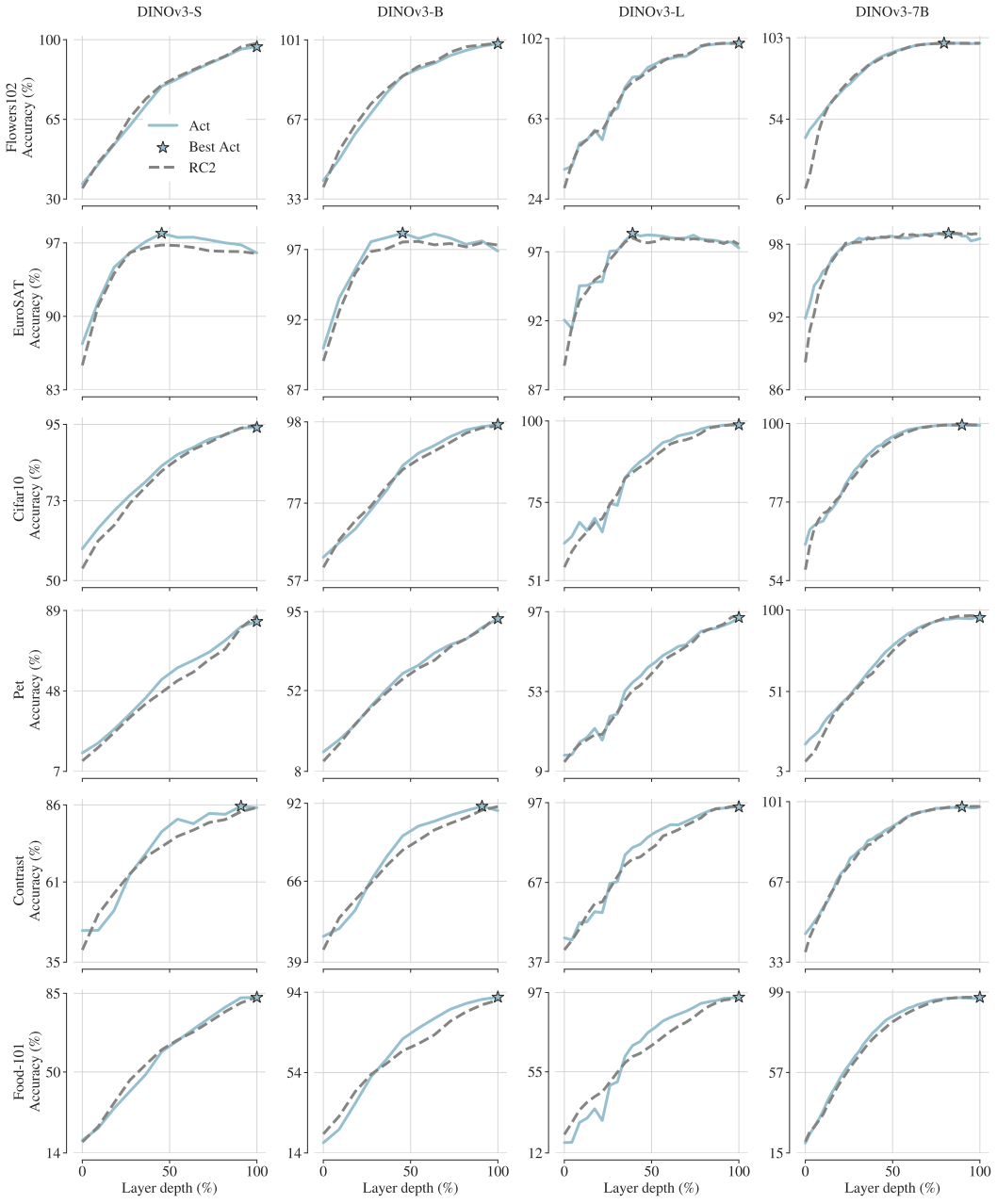


Figure 21: **Impact of `Act` and intermediate layers in DINOv3 (first part).** Probing accuracy across layers of DINOv3 models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

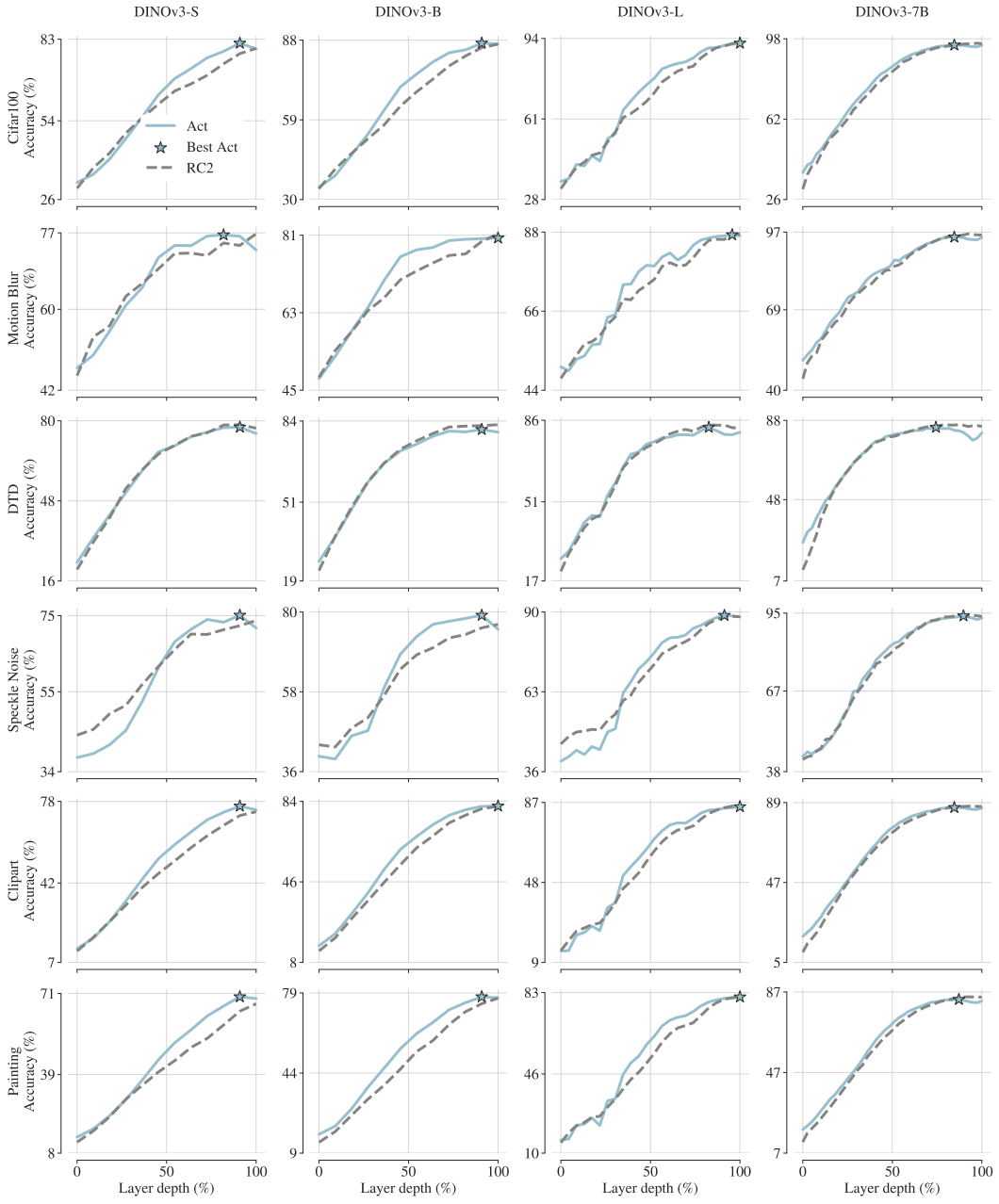


Figure 22: **Impact of `Act` and intermediate layers in DINOv3 (second part).** Probing accuracy across layers of DINOv3 models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

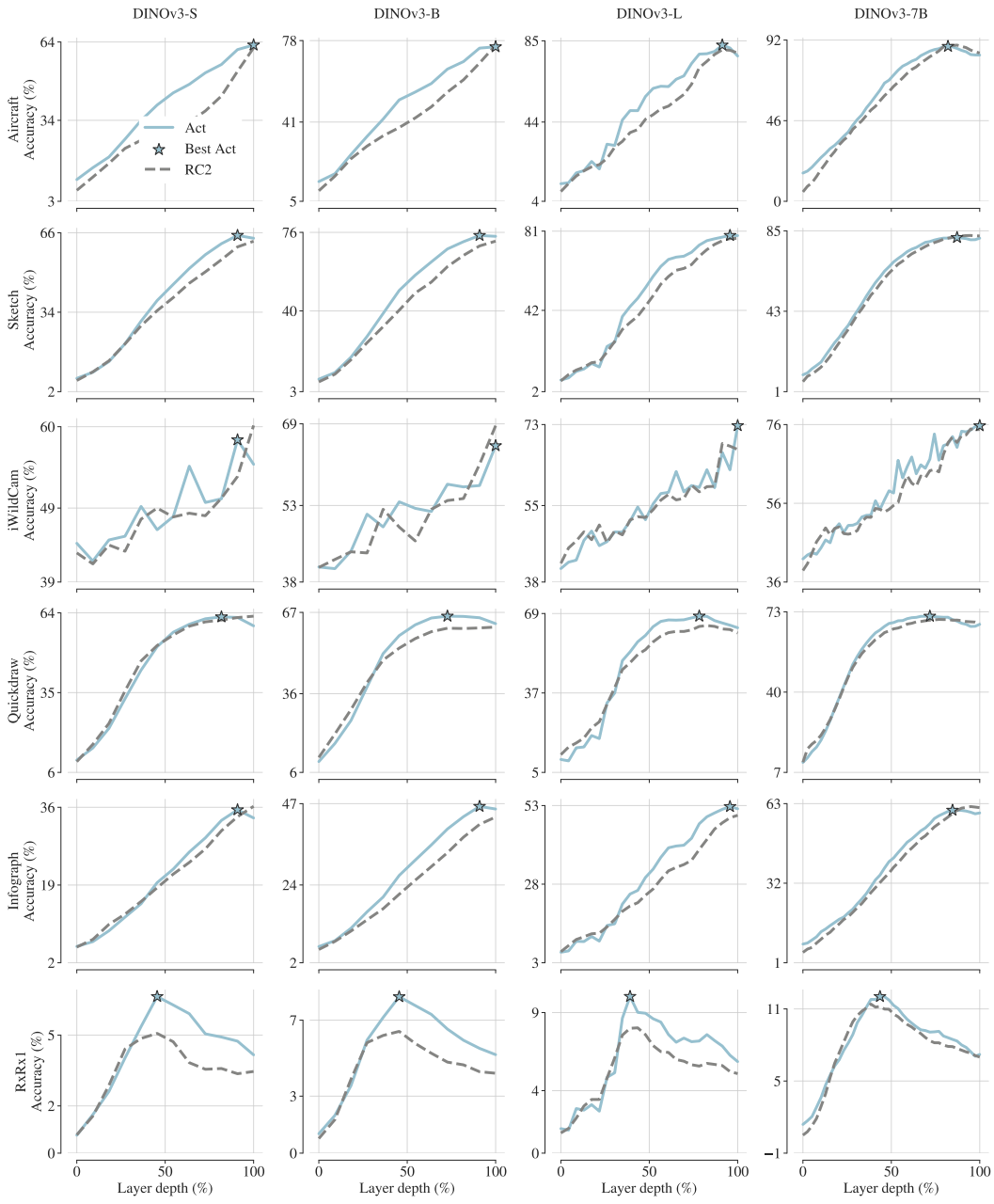


Figure 23: **Impact of `Act` and intermediate layers in DINOv3 (last part).** Probing accuracy across layers of DINOv3 models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

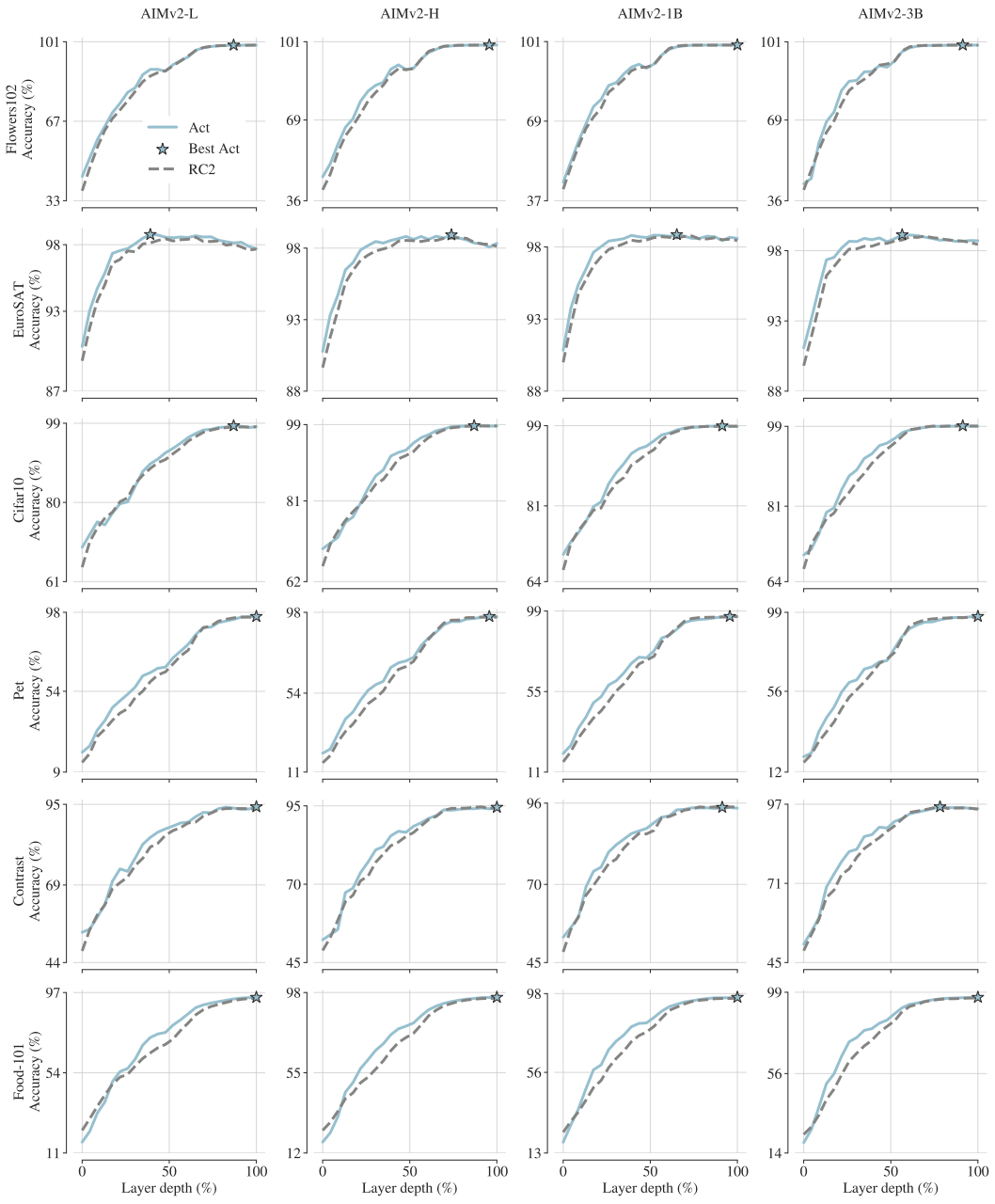


Figure 24: **Impact of `Act` and intermediate layers in AIMv2 (first part).** Probing accuracy across layers of AIMv2 models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

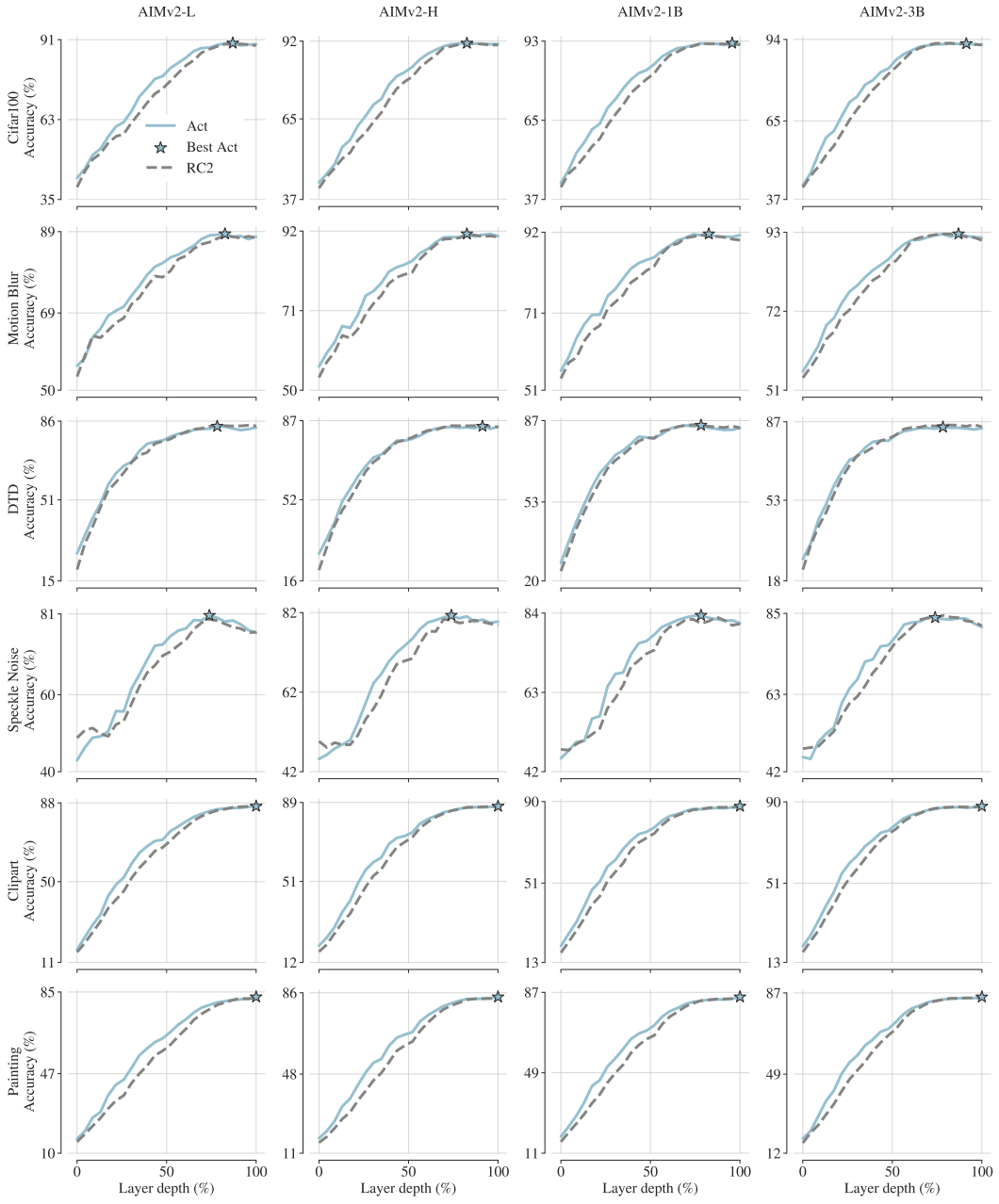


Figure 25: **Impact of `Act` and intermediate layers in AIMv2 (second part).** Probing accuracy across layers of AIMv2 models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

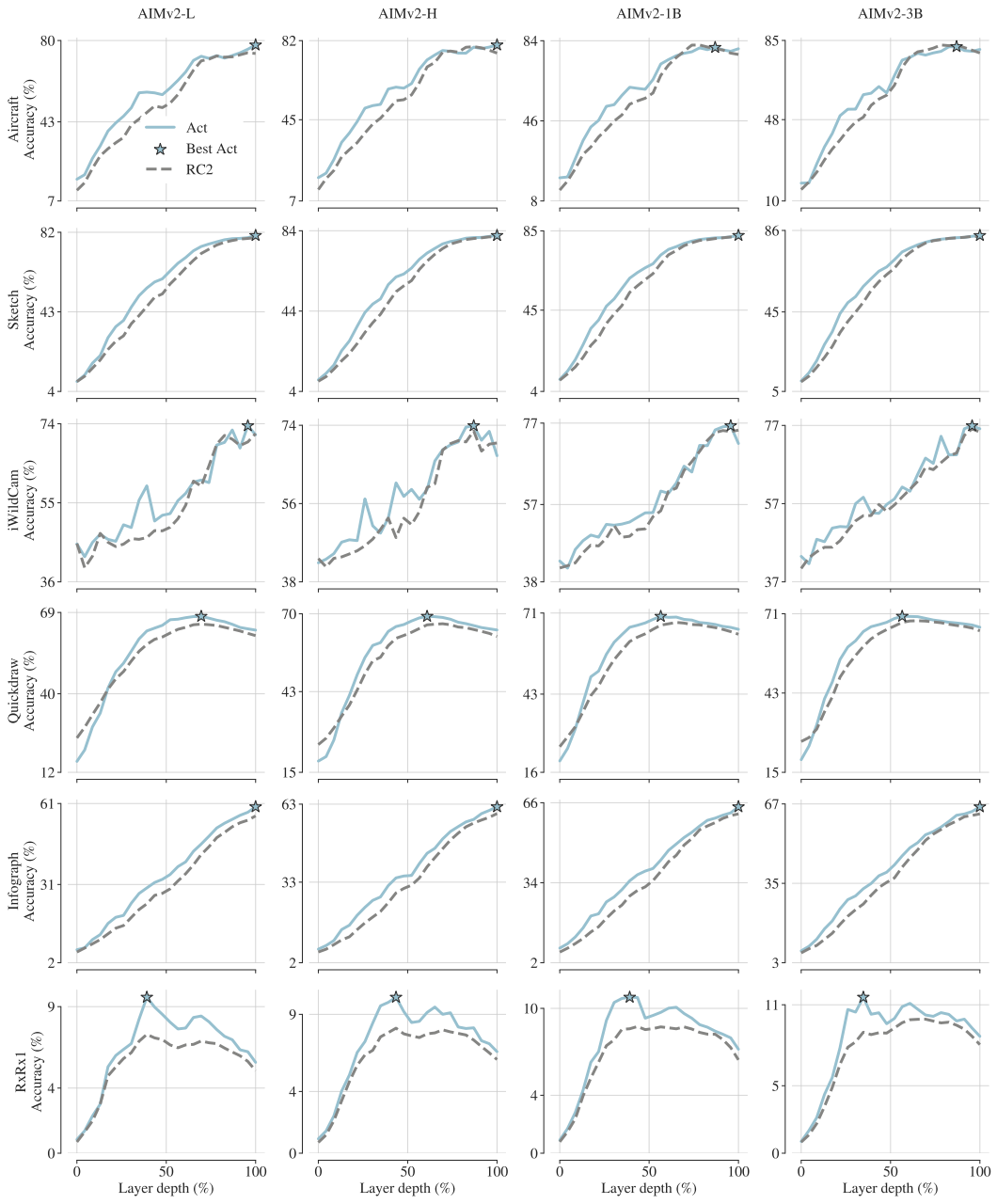


Figure 26: **Impact of `Act` and intermediate layers in AIMv2 (last part).** Probing accuracy across layers of AIMv2 models for `Act` and `RC2`. The star indicates the best `Act` performance over depth. Datasets are ordered from top to bottom by increasing transfer difficulty.

C.1.6 Gains across model sizes

We further investigate the influence of model size on the oracle gain $\mathscr{G}_{\text{joint}}$. Since larger models within a family generally achieve higher downstream accuracy (Dosovitskiy et al., 2021; Fini et al., 2025; Oquab et al., 2024; Siméoni et al., 2025), they may leave less room for improvement through probing. Displayed in Table 16, a Friedman test across model sizes reveals no statistically significant differences for any family (all $p > .05$), although the limited number of model sizes restricts the power of this analysis. The corresponding Kendall's W values are small ($W \leq 0.088$), indicating weak effects of model size on probing gains.

Table 16: **Friedman test on oracle gain $\mathscr{G}_{\text{joint}}$.** Model sizes play the role of treatments, and the 18 datasets are blocks. No family shows a statistically significant effect at $\alpha = 0.05$, and Kendall's W indicates small effects throughout.

Family	χ^2	p-value	Kendall's W
ViT	0.778	0.678	0.022
DINOv2	4.773	0.189	0.088
DINOv3	0.419	0.936	0.008
AIMv2	4.600	0.204	0.085

C.1.7 Full results

We now present the full performance comparison across all model sizes and datasets between the probing baseline `RC2@100`, the fixed strategy `Act@100`, and the oracle strategy, which probes the best layer-module pair for each model-dataset evaluation. The results in Table 17 show that `Act@100` gains are broadly consistent across model sizes within most families, although some exceptions arise, most notably for ViT-B and several DINOv3 variants. The depth curves in Appendix C.1.5 provide further insight into this variation: while `Act` often outperforms the probing baseline `RC2@100`, its best-performing depth varies across model-dataset pairs and is not systematically the final layer. This effect is particularly salient for ViT-B, where intermediate `Act` representations often surpass `RC2@100`, while the final-layer `Act` used by `Act@100` is subpar on many datasets.

Further discussion on `Act`. Linear probing measures the linear separability of representations on downstream data. Transformer architectures preserve the representation dimensionality across layers, as the output of each block is passed as input to the subsequent block. However, different modules within a transformer block can operate at different dimensionalities. In particular, residual-stream representations have the embedding dimension d, whereas FFN intermediate representations have the hidden dimension h, typically larger than d (e.g., $h = 4d$ in standard FFNs). Prior work has shown that gains from feature augmentation cannot be attributed solely to increased feature dimensionality (see Evci et al., 2022, Section 3.2). Their setting differs from ours, however, as they artificially increase the feature dimension by aggregating embeddings across the network, while we preserve the native dimensionality of each representation. Therefore, `Act` has a larger dimensionality than the residual stream. To assess whether its larger dimensionality alone explains the gains of `Act`, we compare it with FC1, which has the same FFN hidden dimension $h > d$. This controlled comparison is on the 307M-parameter variant common to all four model families. As shown in Table 18, overall, `Act@100` outperforms `FC1@100`, with better performance in a majority of evaluations. `FC1@100` is also subpar compared to the probing baseline `RC2@100`, indicating that the gains of `Act@100` cannot be explained by its larger dimensionality alone.

Table 17: **Full performance comparison** between the probing baseline on the final-layer residual stream (`RC2@100`), the fixed strategy on the final-layer activation (`Act@100`), and the oracle strategy that probes the best representation over layers and modules for each model-dataset pair. Probing accuracies are reported for all models, with datasets ordered from left to right by increasing mean probing error (easy to hard). We report for each strategy the probing test accuracy (%).

Model	Strategy	Flowers102	EuroSAT	Cifar10	Pets	Contrast	Food-101	Cifar100	Motion Blur	DTD	Speckle Noise	Clipart	Painting	Aircraft	Sketch	iWildCam	Quickdraw	Infograph	RxRx1	Average
ViT																				
	Baseline	98.3	95.0	91.4	90.2	82.2	73.0	75.0	68.9	69.4	60.1	51.2	58.8	44.6	36.2	56.6	36.4	23.1	1.8	61.8
B	Act@100	95.2	88.1	87.7	83.8	76.4	66.7	69.0	61.2	57.1	49.6	39.8	53.7	35.2	28.8	55.4	26.5	18.1	1.4	55.2
	Oracle	98.6	98.6	92.2	90.6	84.2	73.5	77.3	76.2	72.4	70.6	56.2	59.6	47.8	42.4	61.7	42.6	25.3	4.6	65.2
	Baseline	89.9	96.5	94.9	85.4	90.2	73.2	82.8	79.5	72.6	74.5	59.9	61.0	49.6	45.1	62.8	43.5	23.9	2.1	66.0
L	Act@100	91.9	96.9	95.7	89.9	89.8	80.5	84.9	80.8	73.7	74.8	60.6	65.0	54.2	48.4	66.3	44.2	28.5	2.2	68.2
	Oracle	95.3	98.4	96.1	90.2	91.6	82.0	85.7	82.2	75.2	78.0	64.8	67.7	55.8	52.4	67.2	47.6	31.1	4.8	70.3
	Baseline	87.2	97.5	90.2	78.5	80.0	65.3	75.7	72.5	66.5	69.2	52.1	52.1	44.8	34.3	59.1	41.9	18.6	2.8	60.4
H	Act@100	88.7	97.9	92.3	82.1	82.8	73.0	79.1	77.3	67.6	71.5	56.4	58.2	46.1	40.5	58.2	43.0	23.0	3.0	63.4
	Oracle	92.3	98.1	92.3	82.6	83.4	73.0	79.1	78.4	68.4	73.8	57.4	58.2	47.3	41.2	60.0	44.7	24.1	3.8	64.3
DINOv2																				
	Baseline	98.7	95.9	94.8	88.2	83.6	84.0	80.2	74.2	77.6	73.0	72.3	67.2	59.2	61.3	65.2	59.8	37.1	3.1	70.9
S	Act@100	98.4	96.4	94.7	87.4	84.0	85.3	81.3	73.8	76.5	72.8	75.4	70.7	62.9	64.5	65.3	57.7	36.9	3.9	71.6
	Oracle	98.8	98.0	95.2	88.2	84.4	85.3	82.1	76.2	78.0	75.8	75.4	70.7	62.9	64.5	66.9	61.3	37.1	7.7	72.7
	Baseline	99.5	97.0	97.4	91.1	90.2	88.7	87.2	80.5	81.4	76.8	80.0	75.0	66.4	69.1	62.3	58.8	41.4	3.9	74.8
B	Act@100	99.3	97.3	98.0	90.7	91.3	91.0	88.0	83.2	80.1	77.1	80.8	77.1	69.0	72.3	68.0	62.6	45.0	5.0	76.4
	Oracle	99.5	98.2	98.0	91.5	91.3	91.0	88.0	83.5	82.0	80.8	80.8	77.1	71.0	72.3	69.8	66.1	45.0	8.5	77.5
	Baseline	99.6	97.4	99.0	91.4	96.1	91.7	91.4	87.4	81.6	85.3	82.3	77.9	70.3	73.9	70.1	59.6	44.3	4.8	78.0
L	Act@100	99.5	97.6	99.1	92.8	96.1	93.2	91.9	88.1	80.7	85.9	82.7	79.2	73.4	76.7	72.2	63.0	48.1	6.2	79.3
	Oracle	99.7	98.4	99.1	92.8	96.5	93.2	92.3	89.2	82.8	87.3	83.1	79.7	76.7	76.7	73.2	68.0	48.9	10.2	80.4
	Baseline	99.6	97.9	99.3	92.8	97.9	92.5	93.1	91.1	82.3	90.0	83.1	79.5	72.2	75.5	68.7	63.2	46.7	5.1	79.5
g	Act@100	99.6	98.3	99.3	93.2	98.0	93.4	93.7	91.3	82.6	90.5	83.6	80.2	76.0	77.3	73.9	64.9	49.5	6.1	80.6
	Oracle	99.7	98.6	99.3	95.2	98.0	93.4	93.7	92.0	83.8	90.8	83.8	80.6	82.2	77.5	74.2	68.8	49.6	10.5	81.8
DINOv3																				
	Baseline	98.5	96.0	94.9	86.4	85.2	83.3	79.8	76.8	77.0	73.7	73.4	66.9	61.7	62.6	60.2	62.8	36.2	3.5	71.0
S	Act@100	96.9	96.0	94.1	83.4	85.1	83.2	79.6	73.2	74.9	71.8	74.2	69.0	62.8	63.8	54.9	59.3	33.7	4.2	70.0
	Oracle	98.5	97.9	94.9	86.6	85.5	83.4	81.6	77.2	78.5	75.1	75.9	69.7	62.8	64.9	60.2	62.8	36.2	6.6	72.1
	Baseline	99.6	97.3	97.0	92.8	90.9	90.1	86.5	81.3	82.5	76.6	81.7	76.7	76.0	72.0	68.6	61.4	43.2	4.2	76.6
B	Act@100	99.4	96.9	97.3	91.1	89.5	91.5	86.7	80.3	79.5	75.2	81.8	76.9	75.2	74.2	64.7	62.7	45.5	5.2	76.3
	Oracle	99.6	98.2	97.3	92.8	91.0	91.5	86.9	81.3	82.5	79.1	81.8	77.3	76.8	74.6	69.1	65.6	46.2	8.2	77.8
	Baseline	99.7	97.5	98.8	94.8	95.7	93.8	92.0	87.5	82.7	88.4	85.2	81.0	78.9	77.4	67.5	61.2	49.9	5.1	79.8
L	Act@100	99.7	97.3	98.8	93.8	95.4	94.5	92.0	87.0	80.9	88.3	84.9	81.0	77.3	78.8	72.8	63.3	52.0	5.9	80.2
	Oracle	99.7	98.3	98.9	94.8	95.7	94.5	92.0	87.5	84.3	88.8	85.4	81.0	82.8	78.9	72.8	67.9	52.7	10.0	81.5
	Baseline	99.7	98.7	99.6	96.2	99.0	96.2	95.8	96.0	84.9	93.8	86.9	84.4	84.6	82.4	74.8	68.2	61.4	7.0	83.9
7B	Act@100	99.7	98.5	99.4	95.5	98.8	96.2	95.2	95.1	81.7	93.2	86.1	82.6	83.4	81.2	75.7	67.9	59.4	7.2	83.1
	Oracle	99.7	99.0	99.7	96.9	99.2	96.4	96.0	96.3	86.1	94.2	87.4	84.7	90.5	82.6	75.7	71.2	61.9	12.0	85.0
AIMv2																				
	Baseline	99.6	97.7	98.1	95.1	93.5	93.8	88.9	87.5	83.9	76.2	86.3	82.1	74.1	79.4	71.6	60.7	56.4	5.1	79.4
L	Act@100	99.5	97.7	98.2	95.7	94.2	94.4	89.3	87.8	83.2	76.1	86.4	82.7	77.9	80.4	71.4	62.7	59.8	5.6	80.2
	Oracle	99.6	98.8	98.4	95.7	94.2	94.4	89.8	88.4	84.5	80.6	86.5	82.7	77.9	80.4	73.5	67.6	59.8	9.6	81.2
	Baseline	99.6	98.1	98.7	95.4	94.4	94.9	90.6	90.5	84.3	78.8	87.0	83.2	76.2	80.9	69.9	62.2	59.4	6.1	80.6
H	Act@100	99.6	98.3	98.7	95.7	94.5	95.4	91.0	90.7	84.3	79.8	87.2	84.0	80.0	81.6	67.0	64.4	62.0	6.6	81.2
	Oracle	99.7	98.9	98.7	95.7	94.9	95.4	91.3	91.2	85.2	81.3	87.2	84.0	80.0	81.6	73.9	69.1	62.0	10.1	82.2
	Baseline	99.6	98.4	98.9	96.0	94.8	95.6	91.7	90.0	84.0	81.2	87.3	84.0	77.5	81.9	75.3	63.7	61.5	6.4	81.5
1B	Act@100	99.6	98.6	98.8	96.0	94.4	96.0	91.9	91.2	83.9	81.3	87.7	84.8	80.2	82.7	72.0	65.4	64.4	7.2	82.0
	Oracle	99.7	98.9	99.0	96.0	95.0	96.0	92.3	91.5	86.0	83.4	87.7	84.8	82.0	82.7	76.3	69.9	64.4	10.8	83.1
	Baseline	99.6	98.4	99.1	96.2	95.3	95.9	92.1	90.8	84.8	81.6	87.4	84.5	79.1	82.8	75.3	65.0	63.0	8.0	82.2
3B	Act@100	99.6	98.7	99.0	96.8	95.5	96.3	92.2	91.5	84.2	81.2	87.9	85.1	80.8	83.4	76.1	66.2	65.8	8.7	82.7
	Oracle	99.7	99.1	99.1	96.8	96.2	96.3	92.7	92.6	85.8	84.5	87.9	85.1	83.2	83.4	76.9	70.1	65.8	11.6	83.7

Table 18: **Additional performance comparison** between the probing baseline on the final-layer residual stream (`RC2@100`), the final-layer FFN first linear layer (`FC1@100`), and the fixed strategy on the final-layer activation (`Act@100`). Probing accuracies are reported on the 307M-parameter variant common to each model family, providing a controlled comparison across families. Datasets are ordered by increasing mean probing error, from easy to hard. We report for each strategy the probing test accuracy (%).

Strategy	Flowers102	EuroSAT	Cifar10	Pets	Contrast	Food-101	Cifar100	Motion Blur	DTD	Speckle Noise	Clipart	Painting	Aircraft	Sketch	iWildCam	Quickdraw	Infograph	RxRx1	Average
ViT																			
`Baseline`	89.9	96.5	94.9	85.4	90.2	73.2	82.8	79.5	72.6	74.5	59.9	61.0	49.6	45.1	62.8	43.5	23.9	2.1	66.0
`FC1@100`	91.1	96.0	94.0	86.5	88.7	72.8	80.9	77.8	71.7	69.9	54.8	58.0	50.0	40.3	61.2	40.2	21.1	1.9	64.3
`Act@100`	91.9	96.9	95.7	89.9	89.8	80.5	84.9	80.8	73.7	74.8	60.6	65.0	54.2	48.4	66.3	44.2	28.5	2.2	68.2
DINOv2																			
`Baseline`	99.6	97.4	99.0	91.4	96.1	91.7	91.4	87.4	81.6	85.3	82.3	77.9	70.3	73.9	70.1	59.6	44.3	4.8	78.0
`FC1@100`	99.5	96.9	98.9	90.8	95.8	91.3	90.7	87.8	80.0	85.3	81.3	77.0	72.7	72.3	71.7	58.1	43.5	4.4	77.7
`Act@100`	99.5	97.6	99.1	92.8	96.1	93.2	91.9	88.1	80.7	85.9	82.7	79.2	73.4	76.7	72.2	63.0	48.1	6.2	79.3
DINOv3																			
`Baseline`	99.7	97.5	98.8	94.8	95.7	93.8	92.0	87.5	82.7	88.4	85.2	81.0	78.9	77.4	67.5	61.2	49.9	5.1	79.8
`FC1@100`	99.7	97.3	98.8	93.6	95.3	93.3	91.4	87.2	81.1	87.2	83.9	79.8	77.3	76.4	67.8	59.6	49.0	4.7	79.1
`Act@100`	99.7	97.3	98.8	93.8	95.4	94.5	92.0	87.0	80.9	88.3	84.9	81.0	77.3	78.8	72.8	63.3	52.0	5.9	80.2
AIMv2																			
`Baseline`	99.6	97.7	98.1	95.1	93.5	93.8	88.9	87.5	83.9	76.2	86.3	82.1	74.1	79.4	71.6	60.7	56.4	5.1	79.4
`FC1@100`	99.6	97.6	98.0	95.1	93.4	93.4	88.7	86.7	83.5	74.7	85.9	81.6	73.1	78.3	64.9	60.0	54.5	4.9	78.5
`Act@100`	99.5	97.7	98.2	95.7	94.2	94.4	89.3	87.8	83.2	76.1	86.4	82.7	77.9	80.4	71.4	62.7	59.8	5.6	80.2

C.2 LARGE LANGUAGE MODELS

The results presented in this section relate to the experiments conducted on LLMs in Section 3.2.

C.2.1 MODULE COMPARISON

In Table 19, we display across models the module that belongs to the layer-module pair achieving the oracle gain $\mathscr{G}_{\text{joint}}$. We can see that `LN2` is optimal for all models but one.

Table 19: **Module comparison.** Module in the layer-module pair that achieves the oracle gain $\mathscr{G}_{\text{joint}}$.

Model	*Qwen3*			*Llama3*			*Gemma3*		
	1.7B	4B	8B	1B	3B	8B	1B	4B	12B
best module	`LN2`	`LN2`	`LN2`	`LN2`	`LN2`	`LN2`	`LN2`	`FC1`	`LN2`

C.2.2 PERFORMANCE ALONG THE DEPTH

In this section, we display the performance of `LN2` and `RC2` over depth for all models. Results are consistent with Figure 8. `LN2` outperforms `RC2`, with the highest performance achieved in the interval between 50% and 70% depth.

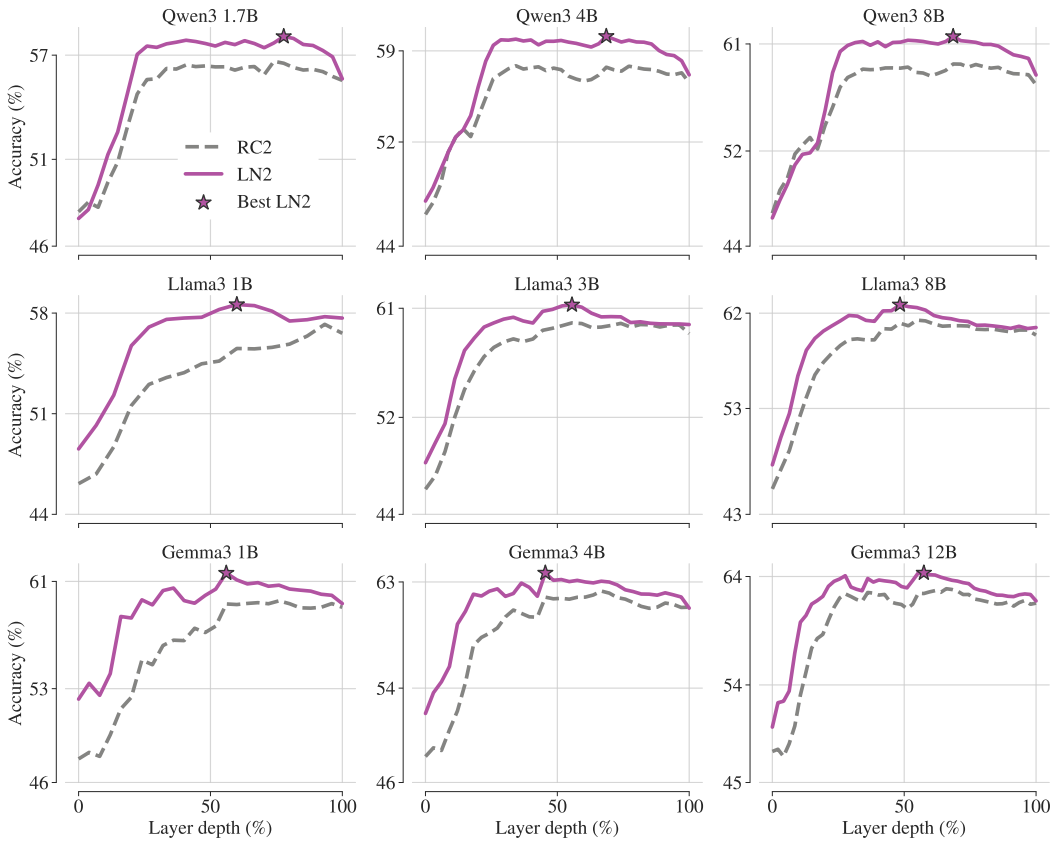


Figure 27: **Impact of `LN2` and intermediate layers on performance across model sizes.** Probing accuracy across layers for `LN2` and `RC2` on `MTEB(Multilingual)`. The star indicates the best performance obtained on `LN2`.

C.2.3 FULL RESULTS

We now present the detailed performance comparison on `MTEB(Multilingual)` between the probing baseline `RC2@100`, the depth-matched `RC2@60`, and the fixed strategy `LN2@60`. Table 20 provides an overview with performance averaged across models. Tables 21 to 23 provide the detailed results across model families and tasks. We can see that the gains of `LN2@60` are consistent across models and tasks.

Table 20: **Full performance comparison per task** between the probing baseline on the final-layer residual stream (`RC2`), residual-stream probing at 60% depth (`RC2@60`), and the fixed strategy `LN2@60`. The accuracy is averaged over models for each task, and the difference in percentage points relative to the probing baseline is shown in green if improved, and in red otherwise.

Task	Baseline	RC2@60		LN2@60	
BulgarianStoreReviewSentimentClassification	50.2	56.3	+6.1	59.0	+8.9
DalajClassification	50.2	50.5	+0.3	50.5	+0.3
DBpediaClassification	91.3	89.2	-2.2	91.8	+0.5
EstonianValenceClassification	37.2	39.6	+2.4	40.5	+3.2
FinancialPhrasebankClassification	64.3	66.6	+2.2	69.1	+4.8
GujaratiNewsClassification	77.2	77.4	+0.2	80.5	+3.3
IndonesianIdClickbaitClassification	63.1	64.3	+1.3	64.7	+1.6
IsiZuluNewsClassification	24.5	24.6	+0.1	25.6	+1.1
ItaCaseholdClassification	54.8	49.4	-5.4	56.0	+1.2
KorSarcasmClassification	61.3	57.9	-3.4	58.7	-2.7
KurdishSentimentClassification	60.8	62.9	+2.1	63.4	+2.6
NepaliNewsClassification	89.8	89.5	-0.2	91.4	+1.6
NusaX-senti	58.0	60.6	+2.6	61.5	+3.5
PAC	68.8	68.6	-0.2	70.4	+1.6
PoemSentimentClassification	39.8	41.8	+2.0	43.1	+3.3
PolEmo2.0-OUT	44.3	53.2	+8.9	55.2	+10.9
PunjabiNewsClassification	72.9	73.3	+0.4	75.5	+2.6
ScalaClassification	54.2	55.2	+1.0	55.1	+0.9
SentimentAnalysisHindi	51.2	55.3	+4.2	56.3	+5.2
SinhalaNewsClassification	59.4	60.0	+0.6	63.1	+3.7
SiswatiNewsClassification	72.6	69.6	-2.9	73.5	+1.0
SwahiliNewsClassification	61.3	61.1	-0.1	63.0	+1.8
TswanaNewsClassification	30.3	27.4	-2.9	30.0	-0.3
TweetTopicSingleClassification	66.9	62.2	-4.7	64.7	-2.2

C.2.4 COMPARISON TO EMBEDDING MODELS

We report in Table 24 the performance of Gemma3 4B and 12B, which are general-purpose, non-instruct models, alongside strong embedding models. We consider the 5 best models reported in Enevoldsen et al. (2025) at the time of publication: multilingual E5 models (Wang et al., 2024a), namely mE5_{base}, $\text{mE5}_{\text{large}}$, and $\text{mE5}_{\text{large-instruct}}$ GritLM-7B (Muennighoff et al., 2025), and $\text{E5}_{\text{mistral-7b}}$ (Wang et al., 2024b), where the last two models are based on Mistral 7B (Jiang et al., 2023). We also consider EmbeddingGemma (Vera et al., 2025), the official embedding model built from Gemma3 (Kamath et al., 2025), and Gemini Embedding (Lee et al., 2025), the industrial-scale closed embedding model derived from Gemini (Gemini Team et al., 2024, 2025).[5] Performance is recovered on the 24 classification tasks of `MTEB(Multilingual)` used in our LLM study, using the official live leaderboard (accessed at `https://huggingface.co/spaces/mteb/leaderboard` on September 9, 2026). Following Enevoldsen et al. (2025), we rank models with a Borda count, where each task ranks the models, and each model scores points according to its position. Since all models are evaluated on all tasks, this is equivalent to ordering models by their average rank across tasks,

[5]The exact version of Gemini used to obtain Gemini Embedding is not specified.

Table 21: **Full performance comparison on Qwen3** between the probing baseline on the final-layer residual stream (RC2), residual-stream probing at 60% depth (RC2@60), and the fixed strategy LN2@60. We report for each strategy the probing test accuracy (%).

Task	*Qwen3 1.7B*			*Qwen3 4B*			*Qwen3 8B*		
	Baseline	RC2@60	LN2@60	Baseline	RC2@60	LN2@60	Baseline	RC2@60	LN2@60
BulgarianStoreReviewSentimentClassification	45.0	53.6	57.0	46.4	53.5	57.3	46.8	56.4	61.8
DalajClassification	49.7	50.5	50.2	49.7	50.4	50.4	50.0	50.3	50.7
DBpediaClassification	91.2	85.8	88.4	91.3	82.4	88.8	91.1	86.1	90.2
EstonianValenceClassification	33.7	36.4	37.3	36.4	39.7	40.3	36.9	40.3	41.3
FinancialPhrasebankClassification	64.3	68.6	70.0	64.2	65.5	70.9	64.3	67.3	72.3
GujaratiNewsClassification	72.1	71.0	77.8	72.8	73.4	80.2	76.8	76.3	82.3
IndonesianIdClickbaitClassification	60.1	62.5	62.5	62.4	62.5	64.0	63.7	63.8	65.4
IsiZuluNewsClassification	23.1	23.0	23.0	23.4	23.2	24.2	23.3	22.7	24.3
ItaCaseholdClassification	48.1	47.0	48.1	52.1	45.8	49.4	53.7	46.5	52.7
KorSarcasmClassification	61.8	55.6	56.8	56.5	57.0	58.3	56.4	57.4	58.6
KurdishSentimentClassification	56.6	57.9	57.9	58.7	59.6	61.2	61.2	62.8	64.4
NepaliNewsClassification	84.6	84.5	87.8	85.9	85.4	89.5	87.8	87.4	90.8
NusaX-senti	55.2	58.1	58.5	58.2	58.2	60.4	57.9	60.8	61.2
PAC	69.3	67.0	71.2	70.9	68.5	72.2	70.3	68.3	72.0
PoemSentimentClassification	39.5	43.5	45.7	37.4	42.0	45.6	39.1	43.3	47.6
PolEmo2.0-OUT	43.0	52.3	55.0	42.4	52.8	57.6	43.0	53.0	58.5
PunjabiNewsClassification	68.3	73.9	73.6	73.2	71.2	74.3	73.9	72.4	75.7
ScalaClassification	52.9	53.5	53.2	54.1	55.3	55.1	54.4	56.1	55.8
SentimentAnalysisHindi	50.3	53.8	57.5	49.4	55.7	58.0	49.5	56.1	58.8
SinhalaNewsClassification	48.0	47.0	47.7	51.1	52.4	56.1	54.5	57.2	61.1
SiswatiNewsClassification	71.9	70.0	71.0	71.1	71.4	72.1	72.4	71.9	72.2
SwahiliNewsClassification	52.3	55.6	55.6	59.4	59.1	60.5	60.5	61.7	61.9
TswanaNewsClassification	25.4	20.7	19.0	24.8	19.1	20.1	27.3	23.2	24.6
TweetTopicSingleClassification	66.6	55.5	57.7	65.6	56.3	60.1	66.4	58.4	61.7

Table 22: **Full performance comparison on Llama3** between the probing baseline on the final-layer residual stream (RC2), residual-stream probing at 60% depth (RC2@60), and the fixed strategy LN2@60. We report for each strategy the probing test accuracy (%).

Task	*Llama3 1B*			*Llama3 3B*			*Llama3 8B*		
	Baseline	RC2@60	LN2@60	Baseline	RC2@60	LN2@60	Baseline	RC2@60	LN2@60
BulgarianStoreReviewSentimentClassification	46.3	51.9	54.6	51.4	57.5	58.3	53.1	58.9	60.0
DalajClassification	50.1	50.1	50.3	50.4	50.5	50.5	50.5	50.4	50.5
DBpediaClassification	90.3	91.8	92.9	91.1	93.1	93.6	91.3	93.3	93.8
EstonianValenceClassification	34.3	35.6	36.3	36.5	38.8	39.4	38.8	41.0	41.2
FinancialPhrasebankClassification	63.5	67.7	69.9	62.5	66.7	68.9	60.8	67.3	69.6
GujaratiNewsClassification	62.5	63.0	65.8	74.3	77.5	79.1	80.5	81.2	82.5
IndonesianIdClickbaitClassification	61.8	63.3	63.1	63.7	64.3	64.3	62.9	64.8	64.9
IsiZuluNewsClassification	23.9	24.0	24.4	24.5	25.0	25.1	24.7	25.6	25.6
ItaCaseholdClassification	56.5	42.0	53.3	59.0	52.3	58.7	60.4	55.0	61.4
KorSarcasmClassification	64.3	57.7	59.0	63.5	59.6	59.9	66.8	58.9	60.5
KurdishSentimentClassification	58.9	60.7	60.2	61.3	63.8	63.9	60.8	62.8	63.0
NepaliNewsClassification	87.9	89.2	91.0	89.4	90.8	92.3	90.8	92.1	92.9
NusaX-senti	55.2	58.1	57.6	58.7	62.0	61.4	58.5	63.1	62.0
PAC	67.9	67.2	68.7	69.1	68.5	68.8	67.7	67.7	67.7
PoemSentimentClassification	39.5	44.9	44.8	38.5	41.7	42.5	41.9	44.0	43.4
PolEmo2.0-OUT	42.0	47.5	47.9	43.7	52.8	53.1	46.0	56.0	57.2
PunjabiNewsClassification	68.5	67.6	75.5	71.3	73.6	76.9	72.2	76.0	76.8
ScalaClassification	52.8	52.4	53.3	54.8	54.5	54.1	56.0	55.1	55.2
SentimentAnalysisHindi	50.8	53.4	54.1	51.1	55.2	54.9	50.1	54.8	54.6
SinhalaNewsClassification	53.0	50.7	55.1	62.5	62.7	65.0	66.8	66.8	69.1
SiswatiNewsClassification	70.8	51.6	74.4	73.2	66.6	73.2	70.8	72.1	73.4
SwahiliNewsClassification	63.4	57.5	63.3	65.3	62.8	64.1	65.7	64.1	65.0
TswanaNewsClassification	28.3	19.5	22.9	32.1	27.6	30.7	34.6	33.7	37.5
TweetTopicSingleClassification	65.8	65.5	67.9	66.0	65.9	67.5	66.7	65.8	67.3

which we report along with the average accuracy. With the probing baseline, Gemma3 models rank among the last and obtain the lowest average accuracy. Using LN2@60, without any finetuning, both move up the ranking: Gemma3 12B and 4B reach the 3rd (5.2) and 4th (5.9) best average ranks, behind only Gemini Embedding and $\text{mE5}_{\text{large-instruct}}$. They notably rank ahead of GritLM-7B, the second-best model in Enevoldsen et al. (2025), and of EmbeddingGemma, which remains slightly better in average accuracy than Gemma3 12B (64.4 vs. 64.2).

Table 23: **Full performance comparison on Gemma3** between the probing baseline on the final-layer residual stream (`RC2`), residual-stream probing at 60% depth (`RC2@60`), and the fixed strategy `LN2@60`. We report for each strategy the probing test accuracy (%).

Task	*Gemma3 1B*			*Gemma3 4B*			*Gemma3 12B*		
	`Baseline`	`RC2@60`	`LN2@60`	`Baseline`	`RC2@60`	`LN2@60`	`Baseline`	`RC2@60`	`LN2@60`
BulgarianStoreReviewSentimentClassification	51.1	58.0	59.0	54.9	58.4	61.8	56.4	58.5	61.5
DalajClassification	50.2	50.1	50.2	50.2	50.8	50.8	50.9	51.4	50.9
DBpediaClassification	91.7	87.5	92.6	92.0	90.7	92.9	91.9	91.7	93.3
EstonianValenceClassification	36.4	40.3	40.7	41.0	43.0	44.0	41.1	41.5	43.5
FinancialPhrasebankClassification	65.8	67.6	68.8	67.2	65.4	67.4	66.5	63.1	64.2
GujaratiNewsClassification	83.5	81.6	83.8	86.0	85.3	85.4	86.5	87.3	87.7
IndonesianIdClickbaitClassification	65.0	65.6	66.2	64.0	66.5	67.0	64.2	65.7	64.6
IsiZuluNewsClassification	23.0	23.8	24.8	26.5	25.9	28.5	28.0	28.3	30.7
ItaCaseholdClassification	51.4	50.4	58.6	54.3	49.4	57.6	57.6	56.6	64.4
KorSarcasmClassification	60.8	57.6	58.0	61.2	58.3	58.6	60.9	59.1	58.5
KurdishSentimentClassification	60.6	60.8	61.5	62.5	66.8	67.3	66.5	70.7	71.5
NepaliNewsClassification	93.3	90.7	92.0	94.1	91.9	92.1	94.0	93.8	94.3
NusaX-senti	58.4	58.0	60.6	58.7	61.6	63.3	60.7	65.1	68.0
PAC	67.8	69.5	71.4	68.7	71.2	70.9	67.6	69.5	71.0
PoemSentimentClassification	39.7	41.4	40.9	42.4	38.9	39.2	39.9	36.1	38.2
PolEmo2.0-OUT	41.6	49.6	53.1	46.8	58.2	59.5	49.9	56.6	54.8
PunjabiNewsClassification	76.6	74.3	75.2	76.1	74.9	76.2	75.8	75.5	75.0
ScalaClassification	52.6	54.7	54.6	54.6	57.1	56.8	55.5	58.6	57.9
SentimentAnalysisHindi	52.1	54.6	55.3	53.7	58.8	57.6	53.6	55.7	56.3
SinhalaNewsClassification	65.5	65.9	69.6	65.6	67.8	70.4	67.8	69.9	74.2
SiswatiNewsClassification	75.1	73.5	74.5	75.8	75.0	74.5	72.1	74.5	76.4
SwahiliNewsClassification	59.5	62.1	65.1	61.7	63.2	65.6	63.7	64.2	66.3
TswanaNewsClassification	27.9	22.9	24.6	33.6	37.0	42.2	38.7	43.3	48.3
TweetTopicSingleClassification	68.0	61.9	65.7	69.3	64.6	65.6	67.3	65.5	68.7

Table 24: **Performance comparison with embedding models.** We compare Gemma3 4B and 12B, probed on the final-layer residual stream (`Baseline`) or with the fixed strategy `LN2@60`, to the top-5 models of Enevoldsen et al. (2025), EmbeddingGemma (Vera et al., 2025) and Gemini Embedding (Lee et al., 2025). Results for embedding models are taken from the `MTEB(Multilingual)` leaderboard at `https://huggingface.co/spaces/mteb/leaderboard`. We report the test accuracy (%) on each task, the average accuracy, and the average rank across tasks (lower is better). Following `MMTEB`, models are ordered by average rank, which yields the same ordering as the Borda count used by the leaderboard since all models are evaluated on all tasks.

Task	*Gemini Embedding*	$mE5_{large\text{-}instruct}$	*Gemma3 12B* `LN2@60`	*Gemma3 4B* `LN2@60`	*EmbeddingGemma*	*GritLM-7B*	$mE5_{large}$	$E5_{mistral\text{-}7b}$	*Gemma3 12B* `Baseline`	$mE5_{base}$	*Gemma3 4B* `Baseline`
BulgarianStoreReviewSentimentClassification	78.1	78.6	61.5	61.8	71.3	77.6	63.9	79.3	56.4	69.4	54.9
DalajClassification	50.5	50.0	50.9	50.8	50.3	50.6	50.0	50.9	50.9	49.8	50.2
DBpediaClassification	94.8	95.5	93.3	92.9	94.3	92.9	88.3	90.9	91.9	87.1	92.0
EstonianValenceClassification	53.5	49.1	43.5	44.0	38.3	38.7	42.9	42.3	41.1	41.2	41.0
FinancialPhrasebankClassification	88.6	84.4	64.2	67.4	86.5	84.4	83.9	79.2	66.5	82.1	67.2
GujaratiNewsClassification	92.0	87.5	87.7	85.4	82.8	69.9	76.7	73.1	86.5	74.9	86.0
IndonesianIdClickbaitClassification	67.0	61.3	64.6	67.0	60.9	60.4	61.2	57.9	64.2	60.1	64.0
IsiZuluNewsClassification	40.5	37.3	30.7	28.5	26.4	28.2	33.1	25.0	28.0	31.5	26.5
ItaCaseholdClassification	73.3	62.9	64.4	57.6	70.4	73.5	66.8	67.9	57.6	61.5	54.3
KorSarcasmClassification	60.5	58.2	58.5	58.6	58.1	59.7	56.8	57.0	60.9	54.7	61.2
KurdishSentimentClassification	86.4	80.9	71.5	67.3	60.0	67.2	77.1	69.1	66.5	76.4	62.5
NepaliNewsClassification	98.1	97.0	94.3	92.1	95.5	94.4	88.5	95.0	94.0	89.4	94.1
NusaX-senti	80.3	71.0	68.0	63.3	69.8	70.8	70.5	70.9	60.7	68.8	58.7
PAC	71.7	65.7	71.0	70.9	67.9	67.0	70.3	62.3	67.6	70.9	68.7
PoemSentimentClassification	59.7	57.1	38.2	39.2	58.9	51.6	50.7	59.1	39.9	46.6	42.4
PolEmo2.0-OUT	77.5	57.0	54.8	59.5	62.8	62.5	36.5	60.5	49.9	43.9	46.8
PunjabiNewsClassification	82.6	84.4	75.0	76.2	82.4	60.2	80.7	55.7	75.8	80.7	76.1
ScalaClassification	51.9	50.4	57.9	56.8	50.8	52.1	51.1	51.1	55.5	50.9	54.6
SentimentAnalysisHindi	76.1	59.5	56.3	57.6	65.5	57.1	64.2	62.1	53.6	64.6	53.7
SinhalaNewsClassification	82.3	78.0	74.2	70.4	65.7	58.0	66.8	56.3	67.8	65.8	65.6
SiswatiNewsClassification	62.4	47.5	76.4	74.5	57.0	58.9	54.4	43.1	72.1	47.0	75.8
SwahiliNewsClassification	66.0	59.1	66.3	65.6	66.0	63.0	59.8	58.3	63.7	62.6	61.7
TswanaNewsClassification	53.4	46.4	48.3	42.2	31.1	40.1	46.2	40.6	38.7	41.0	33.6
TweetTopicSingleClassification	71.1	74.9	68.7	65.6	73.0	75.0	65.3	76.6	67.3	67.4	69.3
Average accuracy	71.6	66.4	64.2	63.1	64.4	63.1	62.7	61.8	61.5	62.0	60.9
Average rank	2.1	4.8	5.2	5.9	6.0	6.2	6.9	6.9	7.1	7.5	7.6