



Journal Homepage: www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/24394
DOI URL: <http://dx.doi.org/10.21474/IJAR01/24394>



RESEARCH ARTICLE

HOW RELIABLE ARE LAYER COMPARISONS FROM SMALL PROBING DATASETS

Pratham Bhalla

1. Independent Researcher.

Manuscript Info

Manuscript History

Received: 12 August 2026

Final Accepted: 14 September 2026

Published: October 2026

Key words:-

probing classifiers; layer selection;
language models; sample efficiency;
representation analysis

Abstract

Probing classifiers are used to compare the linguistic information available at different layers of a language model. These comparisons often rely on limited labelled data, although additional training examples can change which layer performs best. We study the accuracy lost when a layer is selected from a small probing dataset and then evaluated with a probe trained on more data. Experiments across encoder and decoder models show that changes in the selected layer are frequent, but many produce little loss in accuracy. Larger losses depend on the checkpoint and, within the examined Qwen3 family, concentrate in the smaller models. In most such cases, the initially selected layer has higher test accuracy with limited data, but another layer benefits more from additional training. Simple geometric scores often make less accurate choices than a fitted linear probe. Controlled experiments also show how feature scaling affects the regularized probe without removing information. Finally, uncertainty estimates for fixed probes do not consistently predict selection errors after further training. The findings support evaluating the accuracy cost of layer selection directly and distinguishing uncertainty about current predictions from changes caused by additional training data.

"© 2026 by the Author(s). Published by IJAR under CC BY 4.0. Unrestricted use allowed with credit to the author."

Introduction:-

A probing classifier is trained to predict a property from a model's internal representations while the model remains fixed (Alain and Bengio, 2018; Belinkov, 2022). Comparing probe accuracy across layers is often used to study where linguistic properties are most accessible (Tenney et al., 2019; Hewitt and Manning, 2019). Such comparisons depend on the probe, the training sample, and the evaluation metric. Control tasks that assign arbitrary labels to word types test whether a probe can rely on lexical memorization (Hewitt and Liang, 2019), and information-based objectives also measure the cost of learning the predictor (Pimentel et al., 2020; Voita and Titov, 2020). Limited data create a further problem: adding training examples can change which layer performs best. Zhu et al. (2022) estimate the sample size needed to distinguish probing configurations. However, a changed winner and an inaccurate choice are different outcomes. If two layers have nearly equal accuracy, reversing their order has little practical effect. A reversal between layers with substantially different accuracies can instead lead to a poor choice. We therefore ask: how much accuracy is lost by selecting a layer from a small probing dataset, and can measurements taken at that sample size identify unreliable choices? Geometry-based methods motivate the second part of this question. Direct

Probe studies labelled representations without fitting a conventional classifier (Zhou and Srikumar, 2021). Transferability estimators also use representation statistics to guide model selection (You et al., 2021; Garbaciauskas et al., 2024). We ask whether scores computed from 32 training sentences select useful layers for probes subsequently trained on 512 sentences. We compare geometric scores, nearest-centroid accuracy, and exactly solved ridge probes. We also test whether differences in validation accuracy and uncertainty estimated by resampling identify choices that later lose more than one accuracy point.

We study two token classification tasks on three English treebanks. The comparison includes encoder, decoder, and encoder-decoder checkpoints, plus four Qwen3 sizes evaluated under matched conditions. All language-model parameters remain frozen. The initial comparison finds that changes in the selected layer are more common than losses above our tolerance. Errors also concentrate in particular checkpoints. This motivates separate reporting by model and the matched comparison of Qwen3 sizes. Our contribution is an evaluation of selection error, rather than a claim that probing is generally invalid. Three propositions specify what the diagnostics can establish, and each leads to a corresponding experiment: label dependence is tested with fixed-feature controls, coordinate dependence with invertible rescaling, and sampling uncertainty with independent sets of documents reserved for an audit. An exact decomposition separates the initial pairwise accuracy gap from unequal gains under additional training, clarifying why audit uncertainty need not predict later selection error. These comparisons concern recoverable information; they do not establish that a model uses it in its own predictions (Ravichander et al., 2021).

Related Work:-

Probes measure properties recoverable from fixed representations (Alain and Bengio, 2018; Belinkov, 2022). Sentence-level tests, structural probes, and layer comparisons examine different linguistic properties (Conneau et al., 2018; Hewitt and Manning, 2019; Tenney et al., 2019); their conclusions depend on the evaluation metric (Maudslay et al., 2020). Word-type controls test lexical memorization (Hewitt and Liang, 2019). Information estimation, description length, and inductive bias provide alternative objectives (Pimentel et al., 2020; Voita and Titov, 2020; Immer et al., 2022). Successful prediction from a representation does not show that the model uses that property (Ravichander et al., 2021). The closest prior study is Zhu et al. (2022), who estimate the data needed to distinguish probing configurations. We instead measure the error from selecting a layer at a fixed smaller budget. DirectProbe analyzes the geometry of labelled representations (Zhou and Srikumar, 2021); other work measures how representation directions vary across layers (Ethayarajh, 2019). LogME and transferability comparisons address model selection (You et al., 2021; Garbaciauskas et al., 2024). We test simple geometric scores against larger-sample probe accuracy, without claiming to evaluate these methods in full. Prompt-based probing offers a different extraction procedure (Li et al., 2022). Recent work examines consistency under prompt changes (Zhao et al., 2025) and the reliability of causal interventions (Canby et al., 2025). We measure the accuracy cost of selecting layers with limited labelled data and its variation across Qwen3 model sizes.

Materials and Methods:-

Selection error and ranking stability:-

Let $h_\ell(x)$ be the frozen representation at layer ℓ . A probe trained on n sentences gives a validation score $v_{\ell,n}$ and a held-out test score $a_{\ell,n}$. We select the layer with the highest validation accuracy:

$$\hat{\ell}_n = \arg \max_{\ell} v_{\ell,n}.$$

We resolve ties by selecting the earlier layer. We use the term training budget for the number of training sentences. The reference budget is $N = 512$; it does not represent unlimited-data performance.

We distinguish changes in the selected layer from the accuracy lost by that choice. Winner change is the event $\hat{\ell}_n \neq \hat{\ell}_N$. Reference regret evaluates both selected layers using the larger-budget probes:

$$R_n = a_{\hat{\ell}_N, N} - a_{\hat{\ell}_n, N}.$$

We also call R_n the excess error. It can be negative if the layer selected from the smaller sample has higher test accuracy than the reference when both probes use N training sentences. A choice exceeds our error tolerance when $R_n > 0.01$, or one percentage point (pp). This is an accuracy tolerance, not a significance test. For comparison, oracle regret measures the loss against the best test-set layer by replacing the first term with $\max_{\ell} a_{\ell, N}$. We calculate it only after selection; it never determines layers or warning thresholds.

We use Kendall's τ to measure agreement between the complete layer rankings at the two budgets. We also count reversed pairs whose larger-budget test accuracies differ by more than one point. This restriction reduces the influence of near-ties without implying statistical significance.

Data and representations:-

We use release 2.15 of three English treebanks, or collections of annotated sentences, from Universal Dependencies (UD) (de Marneffe et al., 2021): EWT, GUM, and LinES (Universal Dependencies contributors, 2024a; Universal Dependencies contributors, 2024b; Universal Dependencies contributors, 2024c). For each treebank, we sample 768 training, 200 development, and 300 test sentences from the official splits. We exclude sentences longer than 80 words and remove duplicate normalized sentence strings across all selected splits and treebanks. Sampling is independent of the label values. Table 1 gives the resulting word counts.

The table reports word counts in the sampled pools. Each treebank contains 768 training, 200 development, and 300 test sentences. Experiments draw nested training subsets from the 768-sentence pool.

Treebank	Train	Dev.	Test
EWT	13,405	2,523	3,801
GUM	13,620	3,439	5,566
LinES	13,949	3,578	5,030

The tasks are 17-way universal part-of-speech (POS) tagging and 36-way classification of a word's incoming dependency relation after removing relation subtypes. The relation inventory is the union of training labels. Relation classification predicts the label of the link from a word to its syntactic head. The probe receives neither the head nor the dependency tree, so this task does not evaluate parsing. Both tasks retain punctuation. The primary metric averages accuracy within each sentence, then across sentences. Appendix B reports token accuracy and macro F1 for the same selected layers.

We extract features from BERT-base-uncased (Devlin et al., 2019), DistilBERT-base-uncased (Sanh et al., 2020), GPT-2 small (Radford et al., 2019), Pythia-70M-deduped (Biderman et al., 2023), T5-small (Raffel et al., 2020), Gemma-3-1B-PT (Gemma Team, 2025), and Qwen3-0.6B (Qwen Team, 2025). BERT and DistilBERT are bidirectional encoders; GPT-2, Pythia, Gemma-3, and Qwen3 are causal decoders. We examine the encoder of the encoder-decoder checkpoint T5. Qwen3-0.6B has undergone additional training after pretraining; Gemma-3-1B-PT has not. We supply plain sentences to both. A causal model has access to the word's left context; a bidirectional encoder has access to both sides. Model comparisons consequently do not isolate architecture from context or pretraining.

When a tokenizer splits a word into several pieces, we use the hidden state of its final piece. This gives causal models access to every piece of the word and yields one representation per word. Special tokens are excluded. We verify that representations align with words and contain only finite values. All sentences fit within 256 subwords without truncation. Five depths are evaluated: embeddings and the closest layers to one-quarter, one-half, three-quarters, and full depth. Appendix A gives the exact depths. Final returned states include the model's final normalization.

Fitting and validation:-

Five random seeds determine the training samples and control labels. For each seed, the 32-sentence sample is contained in the 128-sentence sample, which is contained in the 512-sentence sample. All seeds use the same sentence pool and pretrained checkpoints. On each subset, we standardize each feature using its training mean and standard deviation. Constant coordinates are left at unit scale.

Our primary probe is multiclass ridge regression. Let X contain the training features and let Y encode each class as a vector with one entry equal to one and the others zero. We fit weights W and intercept b by minimizing

$$\min_{W,b} \|Y - XW - \mathbf{1}b^\top\|_F^2 + \alpha \|W\|_F^2.$$

The squared-weight penalty controls the size of W ; the intercept is unpenalized. We solve the centred normal equations in float64, without iterative optimization. We select $\alpha \in \{1,10,100\}$ separately for every layer and task.

Because the loss in Equation 3 is a sum, the same numerical penalty has a smaller relative weight as the sample grows; validation chooses within this fixed grid at each budget.

Development data are split by document. Whole documents form tuning and audit partitions (Appendix A). The audit sets contain 65 EWT, 11 GUM, and 3 LinES documents. Tuning data select regularization and the layer. Audit data estimate uncertainty for those fitted probes. Test labels are used only to evaluate the resulting choices. The same partitions and training subsets are shared across models. Training budgets exclude the additional labelled development data.

The seven-model comparison contains $7 \times 3 \times 2 \times 5 \times 3 \times 5 = 3,150$ real-task configurations and an equal number of matched word-type controls. Each has three regularization candidates. A deterministic hash of the lowercased word type and seed assigns each type an arbitrary control label. Labels are sampled according to their frequencies in the current training subset. The mapping is consistent across train, development, and test for that configuration. We also evaluate a training-majority baseline and a lexical lookup that predicts the most frequent training label of a word, using the majority label for unseen words.

Qwen3 model-size comparison:-

We separately compare the released post-trained Qwen3 models at 0.6B, 1.7B, 4B, and 8B (Qwen Team, 2025). All four use bfloat16 extraction, including a new extraction of the 0.6B checkpoint; features are stored in float32 and ridge is solved in float64. The seven-model comparison uses float32 extraction. A separate float32 extraction of the 0.6B model uses the same batch size for the precision comparison. We use no weight quantization and supply plain sentences. The four sizes share the data, training seeds, budgets, five relative depths, regularization grid, controls, and evaluation procedure described above. This adds 1,800 configurations and 120 layer decisions at the 32-sentence budget. We report each size separately from the seven-model comparison. Parameter count, representation width, layer count, and checkpoint training vary together. Their individual effects cannot be isolated in this comparison. Appendix E reports dimensions, all selector results, and the precision comparison. We also test sensitivity to the upper regularization boundary by extending the 32-sentence grid to $\{1,10,100,1000,10000\}$. The check covers both studies and changes only small-budget selection. Chosen layers are evaluated with the original 512-sentence probes. Appendix D reports the complete comparison.

Limits of Diagnostics from Small Samples:-

We consider three distinct reasons why a diagnostic may fail: it omits labels, depends on the probe's coordinate system, or estimates uncertainty for a different comparison. Each proposition motivates a specific experimental check.

Geometry and label dependence:-

No statistic of the marginal representation distribution determines optimal linear classification accuracy for all balanced binary tasks. Proof. Fix $X \sim \mathcal{N}(0, I_d)$. If $Y = \mathbf{1}[X_1 > 0]$, a linear classifier has accuracy one. If Y is an independent fair Bernoulli variable, every classifier based on X has population accuracy one-half. Both tasks have the same representation distribution and hence the same unlabelled statistics. $\square$ We test this result by fitting probes to sign and independent labels on the same feature matrix. Section 5.2 then evaluates whether participation rank selects useful layers for the linguistic tasks. Even a scalar score that uses labels can discard information relevant to classification. For independent balanced $Y, U \in \{-1, 1\}$, let $X = (Y, sU)$. The raw trace Fisher ratio is $1/s^2$, although X_1 predicts Y perfectly for every $s > 0$. We evaluate this construction using equal counts of all four (Y, U) pairs. Standardization removes the effect of rescaling the irrelevant coordinate U . Thus, the construction establishes a limitation of the raw ratio; it does not explain all failures of the standardized Fisher selector.

Coordinate changes and the probe objective:-

An invertible diagonal transformation $X \mapsto XD$ preserves linear prediction functions but generally changes isotropic ridge regularization. Standardization using training-set statistics cancels positive diagonal transformations on nonconstant coordinates. Proof. For $W' = D^{-1}W$, we have $(XD)W' = XW$, but $\|W'\|_F^2 = \|D^{-1}W\|_F^2$ need not equal $\|W\|_F^2$. If $D_{jj} > 0$, the transformed training mean and standard deviation are $D_{jj} \mu_j$ and $D_{jj} \sigma_j$. Therefore,

$$\frac{D_{jj} x_j - D_{jj} \mu_j}{D_{jj} \sigma_j} = \frac{x_j - \mu_j}{\sigma_j}.$$

The same training statistics are used at evaluation. □ This proposition motivates standardization and a rescaling check. We draw $D_{jj} = \exp(u_j)$, $u_j \sim \text{Uniform}[-3,3]$, independently of labels, transform every split identically, and refit raw and standardized ridge probes with retuned regularization. Section 5.4 tests the predicted invariance at the middle and final layers with 128 sentences and seed 11.

What an independent audit can certify:-

Suppose L probes are fitted and selected without using an audit set containing m independent documents. For document i , let $A_{i\ell} \in [0,1]$ be the mean sentence accuracy of probe ℓ . Write $\hat{\Delta}_{\ell k}$ for the empirical document-mean gap and $\Delta_{\ell k}$ for its population value.

For $M = L(L-1)/2$ pairs, the following inequality holds with probability at least $1 - \delta$:

$$\max_{\ell < k} |\hat{\Delta}_{\ell k} - \Delta_{\ell k}| \leq \sqrt{\frac{2 \log(2M/\delta)}{m}}.$$

Proof. Each paired document difference lies in $[-1,1]$. Hoeffding's inequality gives a two-sided tail bound $2\exp(-m\epsilon^2/2)$. Applying a union bound over the M pairs and solving for ϵ yields Equation 5. □

This rule certifies a layer as the most accurate only if its observed advantage over every other layer exceeds the bound. To examine what changes after further training, we separate the initial accuracy difference from the subsequent learning gains. Define the learning gain $G_\ell = a_{\ell, N} - a_{\ell, n}$. Then

$$R_n = (a_{\hat{\ell}_{N,n}} - a_{\hat{\ell}_{n,n}}) + (G_{\hat{\ell}_N} - G_{\hat{\ell}_n}).$$

The first term compares the fixed small-budget probes; the second measures their different gains from additional training data. This identity holds under either sentence or document weighting. Proposition 3 controls fixed-probe gaps under its document-weighted assumptions, but does not control the second term. A reliable comparison at 32 sentences can therefore become an inaccurate layer choice for 512 sentences. Section 5.3 measures both terms retrospectively and tests whether audit bounds or warning scores identify later errors. The decomposition describes the error that an early warning must predict; its calculation requires the later results.

Selection rules and warning scores:-

From the standardized features of 32 training sentences, we calculate the total covariance Σ and the covariance B of class means. The Fisher ratio $\text{tr}(B)/\text{tr}(\Sigma - B)$ compares variation between classes with variation within classes. Participation rank, $(\text{tr}\Sigma)^2/\text{tr}(\Sigma^2)$, measures how evenly variance is distributed across directions. We also evaluate the fixed-shrinkage score

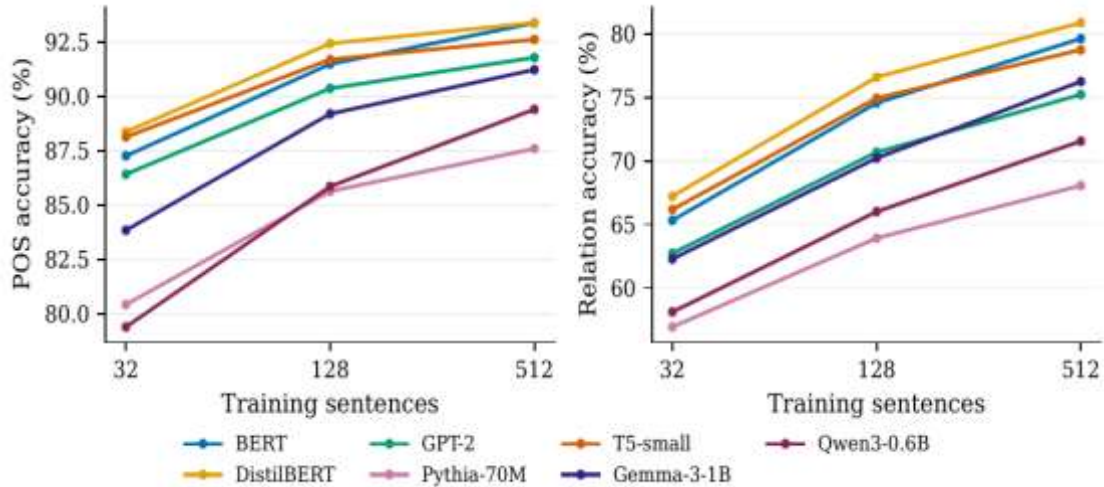
$$H = \text{tr} \left[\left(0.9\Sigma + 0.1 \frac{\text{tr}\Sigma}{d} I \right)^{-1} B \right].$$

Here, d is the feature dimension, and shrinkage adds a scaled identity matrix to stabilize covariance inversion. Each score selects its highest-scoring layer. Of these three statistics, only participation rank ignores labels. We also select layers by nearest-centroid or ridge tuning accuracy, or simply use the final layer. The centroid classifier predicts the class whose training mean is nearest to the standardized feature vector. No selection rule uses test labels. For the ridge-selected layer, we resample whole audit documents 2,000 times. The bootstrap warning is one minus the fraction of samples in which that layer has the highest accuracy. The gap warning is the negative tuning-accuracy difference between the two best layers. Larger values indicate greater estimated uncertainty. Each target treebank uses the 75th-percentile threshold from the other two treebanks. We report the fraction of choices flagged, the fraction of flagged choices that exceed tolerance (precision), and the fraction of above-tolerance choices detected (recall). The area under the receiver operating characteristic curve (AUROC) measures whether the score ranks erroneous choices above the others, independently of the cutoff.

Results:-

Each combination of model, treebank, task, and random seed receives equal weight in the reported averages. Percentage-point differences use sentence-averaged accuracy unless stated otherwise.

Model	EWT POS	EWT Rel.	GUM POS	GUM Rel.	LinES POS	LinES Rel.
BERT	91.7 _{+0.9}	77.7 _{+0.6}	93.9 _{+0.2}	80.7 _{+0.3}	94.6 _{+0.1}	80.5 _{+0.8}
DistilBERT	91.7 _{+0.2}	78.9 _{+0.4}	93.4 _{+0.2}	81.6 _{+0.6}	95.1 _{+0.3}	82.1 _{+0.3}
GPT-2	89.0 _{+1.1}	70.4 _{+1.1}	93.3 _{+0.3}	77.0 _{+0.4}	93.1 _{+0.1}	78.3 _{+0.3}
Pythia-70M	84.4 _{+0.2}	63.6 _{+0.7}	88.9 _{+0.2}	69.7 _{+0.3}	89.4 _{+0.3}	70.8 _{+0.3}
T5-small	89.8 _{+0.2}	75.0 _{+0.3}	93.6 _{+0.2}	80.2 _{+0.4}	94.4 _{+0.1}	81.1 _{+0.2}
Gemma-3-1B	89.3 _{+0.1}	72.7 _{+1.4}	92.0 _{+0.2}	77.7 _{+0.4}	92.5 _{+0.3}	78.3 _{+0.3}
Qwen3-0.6B	87.1 _{+0.4}	66.8 _{+0.9}	90.0 _{+0.2}	73.4 _{+0.4}	91.1 _{+0.5}	74.5 _{+0.3}

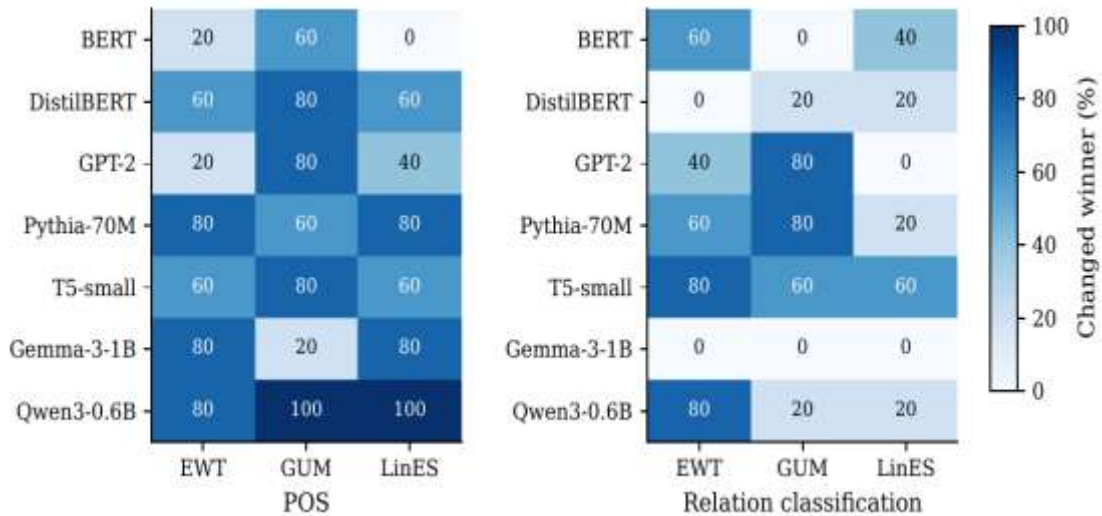


Rankings change more often than decisions fail:-

Table 2 reports 512-sentence test accuracy. Accuracy increases with training data (Figure 1); we next assess the loss from selecting a layer before that increase. At 32 sentences, the selected layer differs from the reference in 48.6% of settings (Table 3). At 128 sentences this remains 37.6%. Despite these changes in the highest-ranked layer, only 4.9% and 1.5% of eligible layer pairs reverse order, respectively. The eligible pairs are those separated by more than one point at the larger budget. Thus, much of the ordering is preserved even when its exact winner changes. Mean reference regret at 32 sentences is 0.22 pp, and 25 of 210 decisions exceed the one-point tolerance: 12 on EWT, seven on GUM, and six on LinES. Fifteen concern POS and ten concern relation classification. Average retrospective oracle regret is 0.43 pp.

Checkpoint differences are substantial. Qwen3-0.6B changes its selected layer in 20 of 30 settings and loses more than one point in 17, with mean reference regret of 1.65 pp. Gemma-3-1B-PT changes its layer in nine settings but has no loss above one point. The other five checkpoints contribute eight choices above tolerance in total. Appendix C reports all checkpoint counts and repeats the evaluation at 0.5- and 2-point tolerances. For 32-sentence ridge, the proportion exceeding these tolerances is 15.2% and 5.2%, respectively. The table compares selections against the 512-sentence reference. Change and flip give percentages of changed selections and reversed eligible pairs. Regret gives excess error (pp), and τ compares complete test rankings. The reference row compares a result with itself.

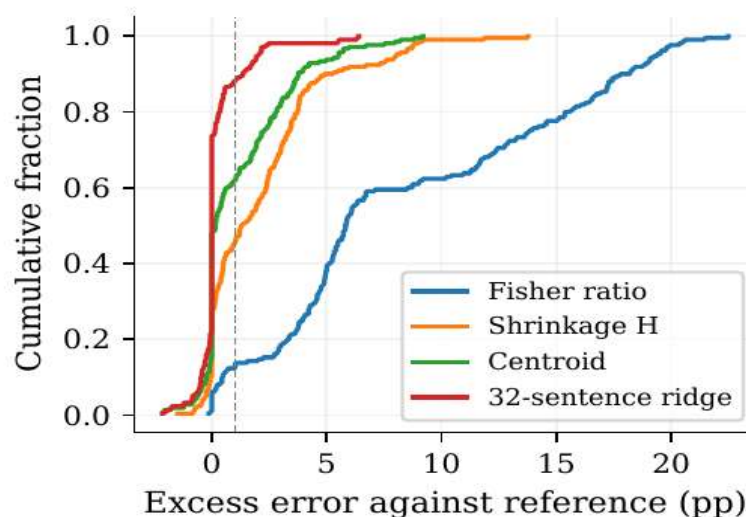
Sentences	Change	Regret	τ	Flip
32	48.6	0.22	0.776	4.9
128	37.6	0.07	0.854	1.5
512	0.0	0.00	1.000	0.0



Comparing layer-selection rules:-

Table 4 evaluates each 32-sentence selector by fitting the larger-budget probe at its chosen layer. Ridge has the smallest mean reference regret. The centroid, participation-rank, H-score, and final-layer rules lose 1.22, 1.36, 2.10, and 3.33 pp on average, respectively. The Fisher ratio has the largest mean selection error, 8.39 pp, and 87.6% of its choices exceed the one-point tolerance. Its mean within-setting Spearman correlation with larger-budget test accuracy is -0.500 , compared with 0.833 for ridge tuning accuracy. Figure 3 shows the distribution of losses.

Selection rule	Match	Regret	95% interval	> 1 pp
Fisher ratio	5.2	8.39	[7.89, 8.89]	87.6
Shrinkage H	17.1	2.10	[1.91, 2.30]	54.8
Participation rank	35.7	1.36	[1.20, 1.52]	34.8
Centroid	32.9	1.22	[1.07, 1.37]	38.1
32-sentence ridge	51.4	0.22	[0.12, 0.32]	11.9
Final layer	5.2	3.33	[3.07, 3.59]	83.3



The curves show the distribution of reference regret for four selection rules. Values to the right of the dashed one-point threshold exceed the error tolerance. Negative values mean the layer chosen from the smaller sample outperforms the tuning-selected reference on the test set.

Audit bounds and predictive warnings:-

The simultaneous rule from Proposition 3, with $L = 5$ and $\delta = 0.05$, certifies no winner among the 210 small-sample comparisons. Its radii are approximately 0.43, 1.04, and 2.00 for the EWT, GUM, and LinES audit sets. These exceed the observed top-layer gaps. The bound cannot distinguish these layers with the available documents; it does not establish that their selection is wrong. The bootstrap warning transfers unevenly (Table 5). At thresholds determined from the other two treebanks, it detects eight of 12 EWT choices above tolerance, two of seven GUM decisions, and one of six LinES decisions. Precision is 27.6%, 14.3%, and 7.7%, respectively. Its AUROC is 0.728 on EWT but below 0.5 on GUM and LinES. The tuning-gap warning also falls below 0.5 on those two targets. Thus, these scores do not provide a dependable warning rule across the examined treebanks. This failure is consistent with the scope of Proposition 3: uncertainty among fixed 32-sentence probes is a different target from the selection error measured after fitting 512-sentence probes.

Equation 6 helps explain why uncertainty at the smaller budget need not predict later errors. In 17 of the 25 choices above tolerance, the selected layer has higher test accuracy than the reference layer when both use 32 sentences. Their order reverses after further training. Across all 25 cases, the mean initial gap is -0.36 pp, while the reference layer gains 2.80 pp more from additional data, producing 2.43 pp of mean excess error. Probes fitted at the two layers benefit differently from the larger training sample. The table evaluates warning thresholds transferred to a held-out treebank. Flag and recall are percentages. Each target has 70 settings, with 12/7/6 choices above tolerance for EWT/GUM/LinES. Thresholds are the 75th percentile on the other two treebanks.

Warning	Target	AUROC	Flag	Recall
Bootstrap	EWT	0.728	41	67
Bootstrap	GUM	0.478	20	29
Bootstrap	LinES	0.383	19	17
Gap	EWT	0.687	17	25
Gap	GUM	0.358	31	29
Gap	LinES	0.263	24	17

Controls and coordinate rescaling:-

Proposition 1 separates the representations from the labels to be predicted. Following this distinction, the word-type controls keep the representations fixed and replace linguistic labels with arbitrary labels. These controls achieve 55.5–61.4% accuracy at the real-task-selected layers (Table 6). High control accuracy shows that the representations support predicting arbitrary labels assigned consistently to word types. The difference from real-task accuracy helps assess the role of lexical memorization, but it does not measure how much syntax the language model uses. The lexical baseline reaches 74.5–82.2% POS accuracy and 49.1–58.2% relation accuracy across the three treebanks at 512 sentences. These scores show that many test labels can be predicted from word identity alone. The table compares real-task and word-type-control accuracy (%) at the layers selected for the real task with 512 sentences. Difference is real-task minus control accuracy (pp). Averages cover all treebanks, tasks, and seeds. Control regularization is tuned separately.

Model	Real	Control	Difference
BERT	86.5	60.2	26.3
DistilBERT	87.1	61.4	25.8
GPT-2	83.5	60.3	23.2
Pythia-70M	77.8	59.1	18.7
T5-small	85.7	58.0	27.7
Gemma-3-1B	83.7	57.5	26.3
Qwen3-0.6B	80.5	55.5	24.9

The coordinate experiment directly tests Proposition 2. Positive diagonal rescaling changes 5.99% of raw-probe predictions on average across the 84 scale comparisons. Standardized probes change no predictions; their score differences are consistent with floating-point rounding (Table 7). The transformation preserves all linear prediction functions, so the raw changes concern regularization rather than loss of decodable information. The fixed-feature check of Proposition 1 produces identical participation rank, 19.61, for sign and independent labels, while fitted ridge test accuracy changes from 96.88% to 51.32%. Rescaling the irrelevant coordinate in the Fisher construction preserves perfect linear classification while its raw Fisher ratio ranges from 10^{-4} to 10^4 . Appendix D gives both

constructions and their results. The table reports the percentage of token predictions that change after invertible diagonal rescaling. Each model contributes two tasks, three treebanks, and two layers. The last column gives converged selected logistic probes out of 36; all regularization candidates converge after increasing the iteration limit as described in the text.

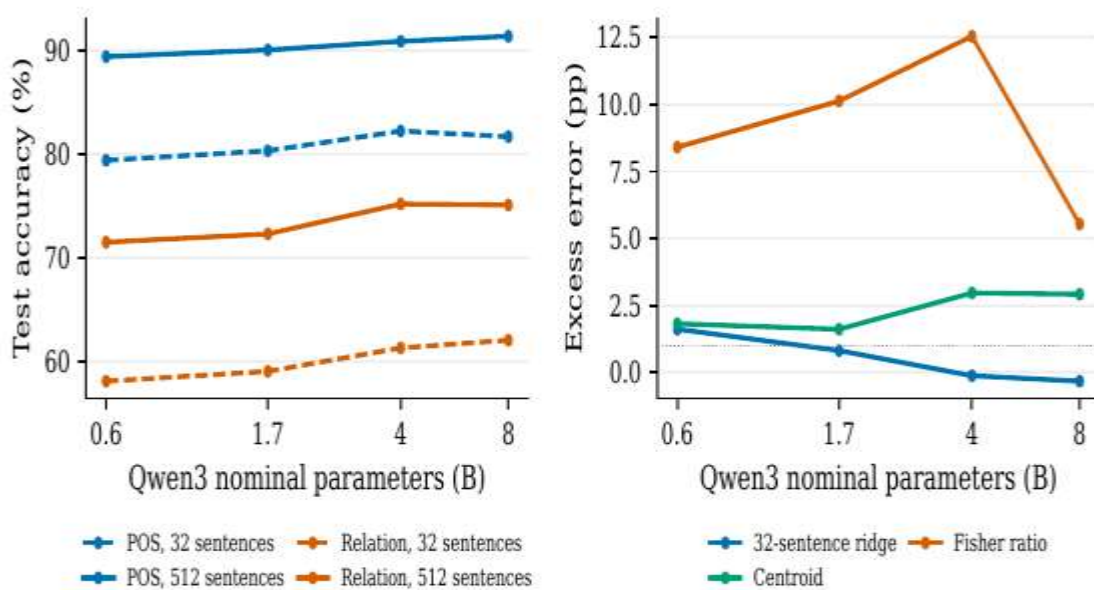
Model	Raw flip	Std. flip	LR converged
BERT	6.39	0.00	36/36
DistilBERT	5.88	0.00	36/36
GPT-2	5.51	0.00	36/36
Pythia-70M	5.86	0.00	36/36
T5-small	3.07	0.00	36/36
Gemma-3-1B	6.67	0.00	36/36
Qwen3-0.6B	8.55	0.00	36/36

To test whether the choice of training loss affects accuracy, we fit multinomial logistic regression at the middle and final layers with 128 sentences and three seeds, giving 252 configurations. All 756 final candidate fits converge; Appendix D gives penalties, stopping criteria, and two required refits. Logistic regression exceeds ridge accuracy by 1.11 pp on average in these paired configurations. Thus, the loss function changes accuracy even within the linear probe family. Expanding the small-budget regularization grid does not remove the observed errors (Appendix D). Across the seven-model comparison, the number above tolerance rises from 25 to 30, and mean excess error rises from 0.22 to 0.29 pp. The expansion improves 16 choices and worsens 28. Adding penalty candidates can improve tuning accuracy without selecting a layer that performs better after additional training. This check changes only the 32-sentence selection procedure; it does not test a wider grid for the 512-sentence reference.

Qwen3 model size and selection error:-

Table 8 reports the matched Qwen3 study. Choices above the one-point tolerance decrease from 17/30 at 0.6B to 10/30 at 1.7B and 0/30 at both 4B and 8B. Mean excess error decreases from 1.62 to 0.82, -0.11 , and -0.32 pp. The negative values reflect test accuracy above the tuning-selected reference. Selected layers still change in 23.3% and 26.7% of decisions at the two larger sizes. Ranking changes therefore persist without losses above tolerance.

Size (B)	POS	Rel.	Changed (%)	Error (pp)	95% interval	> 1 pp
0.6	89.4	71.5	66.7	1.62	[1.13, 2.13]	17/30
1.7	90.0	72.3	56.7	0.82	[0.42, 1.21]	10/30
4	90.9	75.2	23.3	-0.11	[-0.29, 0.06]	0/30
8	91.4	75.1	26.7	-0.32	[-0.50, -0.14]	0/30



Accuracy and selection error need not follow the same pattern (Figure 4). At 512 sentences, POS accuracy increases from 89.4% to 91.4% across sizes, while relation accuracy is nearly unchanged between 4B and 8B. The Fisher selector incurs 5.54–12.55 pp of mean excess error across the four sizes, and centroid selection incurs 1.62–2.97 pp. Ridge has the lowest mean error at three sizes, but the fixed-shrinkage H-score performs better at 1.7B: 0.39 versus 0.82 pp. Appendix E reports every selector. Thus, the relative accuracy of these selection methods depends on the checkpoint. Differences in learning gains also account for many selection errors in this comparison. Of the 27 choices above tolerance, 24 select a layer that has higher test accuracy at 32 sentences than the eventual reference layer. Equation 6 separates this observed small-budget advantage from its reversal after additional training. Expanding the 32-sentence penalty grid changes the total above tolerance from 27 to 28 and leaves both larger sizes at zero. Errors remain concentrated in the smaller models under the wider penalty grid. Warning transfer remains uneven. The tuning-gap warning has AUROC between 0.684 and 0.785 across treebanks, yet its transferred threshold misses all eight LinES errors. Bootstrap recall is 100% on EWT but 12.5% on GUM and LinES. These results distinguish score ranking from the reliability of a transferred cutoff (Appendix E). The matched 0.6B precision check retains all 30 small-sample layer choices and the same 17 errors, so precision alone does not explain the smaller model's result.

Transfer across treebanks:-

Applying each source-trained 512-sentence probe to the other treebanks gives an average accuracy drop of 2.61 pp. Accuracy is higher when EWT-trained probes are evaluated on GUM and LinES, by 2.86 and 3.48 pp respectively, than when they are evaluated on EWT. In contrast, LinES-trained probes lose 10.19 pp when evaluated on EWT. These are comparisons on different test populations, so the differences combine domain difficulty, lexical coverage, and annotation characteristics. The layer with the highest test accuracy changes in 35.5% of source–target comparisons. However, the layer selected on source tuning data loses only 0.27 pp on average against the best layer on the target test set. The changed winner is therefore often close to the source-selected layer in accuracy. Proposition 3 does not cover this transfer: its population is the source audit distribution. Cross-treebank losses require direct evaluation on the target data.

Discussion:-

Counting every change in the selected layer as a failure overstates its effect on accuracy. Reporting the loss against a specified error tolerance, separately for each checkpoint and task, makes this distinction explicit. Proposition 1 explains why identical features can support different task accuracies. The linguistic experiments show that the tested geometric scores often select layers with lower probe accuracy. Proposition 2 predicts the rescaling result: raw ridge predictions change while standardized predictions remain unchanged. Together, these checks show why comparisons of representations should control the labels, feature scaling, and probe objective. Proposition 3 bounds accuracy differences between fixed probes, not their ordering after further training. The weak warning transfer on GUM and LinES demonstrates the practical importance of this distinction. Equation 6 shows how unequal gains from additional data reverse that comparison. Neither a small validation-accuracy difference nor frequent changes in the best layer under audit resampling consistently identifies later selection errors in these experiments. The Qwen3 comparison concentrates errors at the two smaller sizes under both penalty grids. H-score nevertheless outperforms ridge at 1.7B, so comparisons of selection methods should also be reported separately by checkpoint.

Conclusion:-

Changes in layer rankings often have little effect on accuracy, while larger selection errors concentrate in particular checkpoints. Most errors occur when another layer benefits more from additional training data and overtakes the initially selected layer. Uncertainty about fixed probes need not predict these changes. A claim about the best layer at a larger training budget therefore requires evidence about how the layer-specific probes improve with additional data.

Limitations:-

The evidence is limited to three English treebanks, two related token tasks, linear probes, and five depths per checkpoint. T5 contributes encoder states only. The Qwen3 comparison cannot separate parameter count from other checkpoint differences. Logistic and coordinate-rescaling checks cover the original seven models, not the larger Qwen3 checkpoints. The reference uses 512 training sentences and a fixed regularization grid, so its ranking remains uncertain. Bootstrap intervals are conditional on the fitted probes and sampled treebanks. They do not capture uncertainty from pretraining or dataset choice. LinES has only three audit and eight test documents, and possible overlap with pretraining data is unknown. Warning results rely on few errors and lack validation on independent tasks. Model extensions and the regularization check followed inspection of earlier results. The study was not

preregistered. The propositions use established arguments to explain the evaluated comparisons; they are not new general impossibility results. Appendix D records the protocol corrections and their timing.

Checkpoints and Data Partitions:-

Table 9 lists the dimensions of the model components used to extract representations. Width is the number of features per word. Parameter counts exclude unused output heads and, for T5, the decoder; they can therefore differ from nominal checkpoint sizes. Depth zero denotes the returned embedding state. Intermediate depths are the nearest integers to one-quarter, one-half, and three-quarters of the block count. The final returned state includes the model's final normalization.

Checkpoint	Blocks	Width	Examined depths	Parameters (M)
BERT	12	768	0,3,6,9,12	109.5
DistilBERT	6	768	0,2,3,4,6	66.4
GPT-2	12	768	0,3,6,9,12	124.4
Pythia-70M	6	512	0,2,3,4,6	44.7
T5-small	6	512	0,2,3,4,6	35.3
Gemma-3-1B	26	1152	0,6,13,20,26	999.9
Qwen3-0.6B	28	1024	0,7,14,21,28	596.0

The checkpoints are BERT-base-uncased, DistilBERT-base-uncased, GPT-2 small, Pythia-70M-deduped, T5-small, Gemma-3-1B-PT, and Qwen3-0.6B. Complete repository identifiers and frozen versions are recorded in the accompanying source files. Gemma-3 is pretrained and Qwen3 is post-trained. Inputs are plain tokenized sentences throughout. All treebanks use UD release 2.15. We obtain document identifiers by removing the final hyphenated sentence number from each source sentence identifier. Tuning and audit have no shared document identifiers. A fixed permutation assigns whole documents to tuning until it contains at least 100 sentences. Table 10 gives the resulting counts.

Data	Tune sent.	Audit sent.	Tune docs	Audit docs	Test docs
EWT	101	99	56	65	153
GUM	102	98	15	11	26
LinES	117	83	5	3	8

For each seed, we permute the 768-sentence pool and take its first 32, 128, or 512 sentences to form nested training subsets. The sampling seeds are 11, 23, 37, 53, and 71. The pool and document-partition seed is 20260914. Labels, word positions, and sentence identifiers are verified across models.

Additional Performance Measures:-

Table 11 reports token accuracy and macro F1 at the same tuning-selected 512-sentence layers as the main results. Token accuracy weights words equally; the main metric weights sentences equally. Macro F1 averages the class-specific F1 scores over the complete training label inventory. A class receives zero when its F1 denominator is zero. Layers are not reselected for these metrics. The table reports token accuracy and macro F1 (%), averaged across treebanks and training samples. Rel. denotes relation classification.

Model	POS acc.	POS F1	Rel. acc.	Rel. F1
BERT	94.2	82.8	79.6	48.8
DistilBERT	94.3	83.4	80.8	50.3
GPT-2	93.1	78.7	77.4	48.1
Pythia-70M	89.0	76.9	68.6	38.5
T5-small	94.2	81.6	78.6	46.8
Gemma-3-1B	92.4	80.8	77.4	47.8
Qwen3-0.6B	90.9	79.2	73.6	43.0

Sensitivity to the Error Tolerance:-

Table 12 evaluates the same choices at tolerances of 0.5, 1, and 2 points. Table 13 separates ridge choices by checkpoint. Each row has 30 decisions: three treebanks, two tasks, and five training samples. Winner disagreement and excess error answer different questions; a changed winner can have zero or negative excess error. The table

reports the percentage of choices exceeding each error tolerance. Both selected layers are evaluated with 512-sentence probes.

Selection rule	0.5 pp	1 pp	2 pp
Fisher ratio	89.0	87.6	85.7
Shrinkage H	60.0	54.8	44.3
Participation rank	41.4	34.8	26.2
Centroid	41.9	38.1	27.6
32-sentence ridge	15.2	11.9	5.2
Final layer	92.4	83.3	74.3

The table separates 32-sentence ridge choices by checkpoint. Error is mean excess error (pp); the next column counts losses above one point.

Model	Changed (%)	Error (pp)	> 1 pp	Total
BERT	30.0	-0.10	2	30
DistilBERT	40.0	-0.07	0	30
GPT-2	43.3	0.10	3	30
Pythia-70M	63.3	0.07	1	30
T5-small	66.7	-0.08	2	30
Gemma-3-1B	30.0	-0.04	0	30
Qwen3-0.6B	66.7	1.65	17	30

Controlled Experiments and Implementation:-

For Proposition 1, we draw 6,000 independent 20-dimensional standard Gaussian vectors. The first 1,000 train a ridge classifier with $\alpha = 10$; the remaining 5,000 are the test set. Labels are either the sign of the first coordinate or independent Bernoulli variables with probability one-half. The two tasks use exactly the same feature matrix. The fitted probe can make errors on sign labels even though a perfect linear classifier exists. For the Fisher construction, 1,000 examples contain equal counts of $(Y, U) \in \{-1, 1\}^2$, with features (Y, sU) . Table 14 reports the exact ratios and the accuracy of the explicit classifier $1[X_1 > 0]$. Rescaling the irrelevant coordinate changes the raw Fisher ratio while preserving perfect classification.

s	Raw Fisher	Linear accuracy (%)
0.01	10000	100
0.1	100	100
1	1	100
10	0.01	100
100	0.0001	100

The coordinate experiment tests Proposition 2 at two depths, two tasks, and three treebanks for each of the original seven models. Seed 11 selects 128 training sentences. Diagonal factors are drawn with seed $812 + \ell$, where ℓ is the layer number, and applied identically to all splits. The largest standardized score difference is consistent with floating-point rounding. Tests also compare the ridge predictions with an independent implementation and verify sentence aggregation, control-label consistency, and partition separation. Logistic regression uses $C \in \{0.01, 0.1, 1\}$, tolerance 10^{-5} , and an initial 1,000-iteration limit. Two Qwen3 configurations contain candidates that reach that limit. We refit all three regularization candidates with a 5,000-iteration limit, changing no data, penalty, or selection criterion. All 756 final candidates converge; the maximum iteration count is 1,061. Original and refitted records are retained.

When there are fewer training words than feature dimensions, we can solve ridge using the nonzero eigenvalues and eigenvectors of XX^T . This gives the same estimator as solving with X^TX . The implementation uses the full feature dimension in covariance shrinkage even when null directions are omitted. Tests compare both solutions and an independent ridge implementation. The regularization-range check is motivated by frequent selection of $\alpha = 100$, the upper boundary of the original grid. We add 1,000 and 10,000 at the 32-sentence budget, choose both penalty and layer on tuning data, and evaluate the chosen layer with the original 512-sentence probe. The reference is unchanged. Table 15 reports all original and expanded-grid outcomes. Before adding penalties, the implementation verifies that it reproduces the best tuning accuracy from the original three candidates. The table compares the original and expanded regularization grids. R is mean excess error (pp), and B counts choices above one point. Subscripts o and e denote the

original and expanded grids. Each row has 30 decisions. † marks the bfloat16 size study; other rows use float32 extraction.

Model	R _o	R _e	B _o	B _e
BERT	-0.10	-0.19	2/30	0/30
DistilBERT	-0.07	0.08	0/30	2/30
GPT-2	0.10	0.15	3/30	4/30
Pythia-70M	0.07	0.19	1/30	3/30
T5-small	-0.08	-0.10	2/30	1/30
Gemma-3-1B	-0.04	0.22	0/30	2/30
Qwen3-0.6B	1.65	1.66	17/30	18/30
Qwen3-0.6B [†]	1.62	1.63	17/30	18/30
Qwen3-1.7B [†]	0.82	0.80	10/30	10/30
Qwen3-4B [†]	-0.11	-0.11	0/30	0/30
Qwen3-8B [†]	-0.32	-0.32	0/30	0/30

Before inspecting accuracy outcomes, we corrected the development partition to separate documents. Earlier runs are excluded. The additional checkpoints and the wider regularization grid were introduced after examining earlier results. These changes are recorded in the experiment protocol.

Qwen3 Size and Precision Comparisons:-

The size study uses the original released Qwen3-0.6B, Qwen3-1.7B, Qwen3-4B, and Qwen3-8B post-trained checkpoints. It uses all feature dimensions without projection. All four use bfloat16 computation and float32 feature storage. Table 16 records widths and depths. Table 17 reports every selection rule at every size. The table lists Qwen3 dimensions and examined depths. The last depth is the block count. Parameter counts exclude unused output heads.

Size (B)	Width	Depths	Params (B)
0.6	1024	0,7,14,21,28	0.596
1.7	2048	0,7,14,21,28	1.721
4	2560	0,9,18,27,36	4.022
8	4096	0,9,18,27,36	7.568

The table reports mean excess error (pp) for each Qwen3 size and selection rule. Rank denotes participation rank. Layers are selected using 32 sentences and evaluated with 512-sentence probes.

Rule / size (B)	0.6	1.7	4	8
Fisher	8.41	10.13	12.55	5.54
H-score	2.71	0.39	1.21	0.69
Rank	2.04	1.05	1.61	0.48
Centroid	1.82	1.62	2.97	2.93
Ridge	1.62	0.82	-0.11	-0.32
Final	4.18	1.90	1.95	1.73

Table 18 applies the same cross-treebank threshold rule within the size study. Thresholds are computed from the other two treebanks across all four sizes. AUROC evaluates score ordering; recall also depends on the transferred cutoff.

The table evaluates transferred warning thresholds across Qwen3 sizes. Flag and recall are percentages. There are 40 settings per target, with 11/8/8 choices above tolerance on EWT/GUM/LinES.

Warning	Target	AUROC	Flag	Recall
Bootstrap	EWT	0.708	85.0	100.0
Bootstrap	GUM	0.678	7.5	12.5
Bootstrap	LinES	0.422	5.0	12.5
Gap	EWT	0.743	57.5	90.9
Gap	GUM	0.785	22.5	50.0
Gap	LinES	0.684	0.0	0.0

The precision control compares bfloat16 and float32 computation at 0.6B, holding the checkpoint, tokenizer, four-sentence batch size, data, and probe protocol fixed. Across 450 matched configurations, mean absolute test-accuracy difference is 0.12 pp and the maximum is 0.99 pp. The same layer is selected in all 30 paired decisions at the 32-sentence budget. Both precisions produce 17 choices above the one-point tolerance; mean excess error is 1.62 pp in both. The reference layer changes in three decisions at 512 sentences. Precision therefore affects some scores and reference choices, but does not account for the observed 0.6B selection errors. This check does not establish precision invariance at larger sizes.

References:-

1. Guillaume Alain and Yoshua Bengio. 2018. Understanding intermediate layers using linear classifier probes (<https://arxiv.org/abs/1610.01644>). Preprint, arXiv:1610.01644.
2. Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances (https://doi.org/10.1162/coli_a_00422). *Computational Linguistics*, 48(1):207–219.
3. Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usven Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling (<https://proceedings.mlr.press/v202/biderman23a.html>). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.
4. Marc E. Canby, Adam Davies, Chirag Rastogi, and Julia Hockenmaier. 2025. How reliable are causal probing interventions? (<https://doi.org/10.18653/v1/2025.ijcnlp-long.47>) In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 857–878, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.
5. Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single \$ (<https://doi.org/10.18653/v1/P18-1198>)&#!# vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.
6. Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. U (https://doi.org/10.1162/coli_a_00402)niversal Dependencies. *Computational Linguistics*, 47(2):255–308.
7. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT (<https://doi.org/10.18653/v1/N19-1423>): Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
8. Kawin Ethayarajh. 2019. How contextual are contextualized word representations? C (<https://doi.org/10.18653/v1/D19-1006>)omparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.
9. Lukas Garbaciauskas, Max Ploner, and Alan Akbik. 2024. Choose your transformer: Improved transferability estimation of transformer models on classification tasks (<https://doi.org/10.18653/v1/2024.findings-acl.757>). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12752–12768, Bangkok, Thailand. Association for Computational Linguistics.
10. Gemma Team. 2025. Gemma 3 (<https://goo.gle/Gemma3Report>).
11. John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks (<https://doi.org/10.18653/v1/D19-1275>). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.
12. John Hewitt and Christopher D. Manning. 2019. A (<https://doi.org/10.18653/v1/N19-1419>) structural probe for finding syntax in word representations. In *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)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.
13. Alexander Immer, Lucas Torroba Hennigen, Vincent Fortuin, and Ryan Cotterell. 2022. Probing as quantifying inductive bias (<https://doi.org/10.18653/v1/2022.acl-long.129>). In *Proceedings of the 60th Annual Meeting of the*

- Association for Computational Linguistics (Volume 1: Long Papers), pages 1839–1851, Dublin, Ireland. Association for Computational Linguistics.
14. Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. 2022. Probing via prompting (<https://doi.org/10.18653/v1/2022.naacl-main.84>). In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1144–1157, Seattle, United States. Association for Computational Linguistics.
 15. Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser (<https://doi.org/10.18653/v1/2020.acl-main.659>). In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7389–7395, Online. Association for Computational Linguistics.
 16. Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure (<https://doi.org/10.18653/v1/2020.acl-main.420>). In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4609–4622, Online. Association for Computational Linguistics.
 17. Qwen Team. 2025. Qwen3 technical report (<https://arxiv.org/abs/2505.09388>). Preprint, arXiv:2505.09388.
 18. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners (<https://cdn.openai.com/better-language-models/language-models.pdf>). Technical report, OpenAI.
 19. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer (<http://jmlr.org/papers/v21/20-074.html>). Journal of Machine Learning Research, 21(140):1–67.
 20. Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? (<https://doi.org/10.18653/v1/2021.eacl-main.295>) In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 3363–3377, Online. Association for Computational Linguistics.
 21. Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter (<https://arxiv.org/abs/1910.01108>). Preprint, arXiv:1910.01108.
 22. Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT (<https://doi.org/10.18653/v1/P19-1452>) rediscovers the classical NLP pipeline. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
 23. Universal Dependencies contributors. 2024. UD English EWT (https://github.com/UniversalDependencies/UD_English-EWT/tree/r2.15). Release 2.15.
 24. Universal Dependencies contributors. 2024. UD English GUM (https://github.com/UniversalDependencies/UD_English-GUM/tree/r2.15). Release 2.15.
 25. Universal Dependencies contributors. 2024. UD English LinES (https://github.com/UniversalDependencies/UD_English-LinES/tree/r2.15). Release 2.15.
 26. Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length (<https://doi.org/10.18653/v1/2020.emnlp-main.14>). In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 183–196, Online. Association for Computational Linguistics.
 27. Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. 2021. Logme: Practical assessment of pre-trained models for transfer learning (<https://proceedings.mlr.press/v139/you21b.html>). In Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 12133–12143. PMLR.
 28. Raoyuan Zhao, Abdullatif Köksal, Ali Modarressi, Michael A. Hedderich, and Hinrich Schuetze. 2025. Do we know what LLM (<https://doi.org/10.18653/v1/2025.findings-emnlp.1263>)s don't know? a study of consistency in knowledge probing. In Findings of the Association for Computational Linguistics: EMNLP 2025, pages 23254–23280, Suzhou, China. Association for Computational Linguistics.
 29. Yichu Zhou and Vivek Srikumar. 2021. D (<https://doi.org/10.18653/v1/2021.naacl-main.401>)irectProbe: Studying representations without classifiers. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5070–5083, Online. Association for Computational Linguistics.
 30. Zining Zhu, Jixuan Wang, Bai Li, and Frank Rudzicz. 2022. On the data requirements of probing (<https://doi.org/10.18653/v1/2022.findings-acl.326>). In Findings of the Association for Computational Linguistics: ACL 2022, pages 4132–4147, Dublin, Ireland. Association for Computational Linguistics.