International Journal of Engineering Research and Modern Education (IJERME)
International Peer Reviewed - Refereed Research Journal, Website: www.crystalpen.in
Impact Factor: 7.018, ISSN (Online): 2455 - 4200, Volume 11, Issue 2, July - December, 2026

ADAPTIVE INFERENCE THRESHOLDS AND SELF-DISTILLATION SHAPE CLASSIFICATION ACCURACY ACROSS COMPUTATIONALLY CONSTRAINED LANGUAGE MODEL BENCHMARKS

Divya Vishwanath* & Paul Johnson Asamoah**
* Janardhan Rai Nagar Rajasthan Vidyapeeth, Udaipur, Rajasthan, India
** Power Life University College, Accra, Ghana, West Africa

Cite This Article: Divya Vishwanath & Paul Johnson Asamoah, "Adaptive Inference Thresholds and Self-Distillation Shape Classification Accuracy Across Computationally Constrained Language Model Benchmarks", International Journal of Engineering Research and Modern Education, Volume 11, Issue 2, July - December, Page Number 12-20, 2026.

Copy Right: © Crystal Pen Publication, 2026 (All Rights Reserved). This is an Open Access Article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Type of Review: Peer Reviewed as per |C|O|P|E| Guidance.

Disclaimer: The scholarly papers reviewed and published by Crystal Pen Publication, India, reflect the views and opinions of their respective authors and do not necessarily represent the views or opinions of Crystal Pen Publication. The publisher disclaims any responsibility for any harm, loss, or damage resulting from the use of the published content by any party.

DOI: https://doi.org/10.5281/zenodo.24049707

Abstract:

This analysis examines how adaptive computation configurations relate to classification accuracy in language models, informing international debates on computational efficiency. A descriptive reanalysis uses 20 published configuration results from Chinese Book review and English Yelp.P benchmarks reported in 2020, comparing routing arrangements, exit thresholds, and matched distillation settings. Adaptive routing exceeds selected fixed depth alternatives by 0.56 to 2.81 percentage points, with fewer operations in all three computationally usable comparisons. Raising exit thresholds reduces accuracy across both benchmarks, while endpoint computational reductions range from 62.77 to 77.87 percent among usable comparisons. Self-distillation improves accuracy in all four matched pairs, averaging 0.478 percentage points, with larger advantages for Book review. Accuracy losses concentrate differently across threshold intervals, indicating benchmark dependent stopping costs. These descriptive patterns distinguish selective processing, prediction acceptance, and intermediate supervision as complementary configuration decisions. The contribution refines the resource allocation perspective through transparent comparisons of existing evidence. Deployment teams should evaluate threshold intervals separately and compare accuracy alongside verified computational requirements within each intended task.

Key Words: Adaptive Inference; Early Exit; Self-Distillation; Classification Accuracy; Descriptive Reanalysis

1. Introduction:

Computationally efficient language classification requires deciding when additional processing improves a prediction enough to justify its cost. Applying the same processing depth to every input can allocate unnecessary computation to straightforward cases while restricting the capacity available to difficult cases. Adaptive inference addresses this problem by permitting predictions to exit at intermediate layers. Its practical value, however, depends on the accuracy retained under a particular routing arrangement, stopping threshold, and training configuration. Han and colleagues, 2022, distinguish input dependent computation from static network execution, establishing a framework for examining efficiency through the allocation of processing rather than model size alone. The substantive question is therefore how specific configuration choices accompany changes in classification accuracy. Resource rationality provides a theoretical interpretation of this allocation problem. Lieder and Griffiths, 2020, explain effective computation as the use of limited resources in relation to the benefits those resources produce. Applied to language classification, this perspective suggests that additional layers are valuable when they improve a decision, rather than simply because they are available. Adaptive routing operationalises this principle by allowing processing depth to vary across inputs. Nevertheless, its expected advantage depends on intermediate classifiers recognising when further processing is useful. A stopping rule can remove redundant computation, but it can also terminate an incorrect prediction before later representations permit revision. The relationship between processing and accuracy consequently depends on how the system selects predictions for early acceptance.

The closest architectural evidence already establishes the feasibility of this approach. Liu and colleagues, 2020, introduce FastBERT, combining intermediate classifiers, uncertainty based stopping, and self-distillation to adjust computation across inputs. Xin and colleagues, 2020, demonstrate that DeeBERT can reduce inference time while limiting deterioration in predictive performance. These contributions establish that executing every layer is not necessary for every classification decision. They also make clear that adaptive inference comprises several choices whose implications should be distinguished. Routing determines whether processing varies across inputs, the uncertainty threshold determines which predictions qualify for termination, and self-distillation determines how intermediate classifiers receive supervision. The present analysis therefore examines these configuration dimensions separately rather than representing adaptive inference through a single indicator of efficiency. Evidence on alternative stopping and training arrangements further qualifies any expectation of a uniform accuracy and computation relationship. Schwartz and colleagues, 2020, use confidence informed exits to match processing complexity to input complexity, allowing the operating configuration to change without retraining multiple models. Zhu, 2021, develops supervision across intermediate exits, showing that the organisation of training information is another relevant design choice. These approaches address related but distinct decisions: whether an intermediate prediction is sufficiently dependable and whether the system should accept it. Classification accuracy provides an observable outcome for comparing those decisions, but its interpretation remains benchmark specific. A difference between two benchmark accuracy levels cannot be attributed to configuration when the

classification tasks themselves differ. Comparisons must therefore preserve benchmark identity and distinguish matched settings from alternatives with different computational requirements.

The contribution pursued here is a transparent descriptive reanalysis of selected FastBERT results, rather than a new architecture or an independent replication. The original publication already documents adaptive inference and its ablations. The additional analytical value lies in expressing those results as explicit comparisons that distinguish accuracy advantages, threshold costs, and benchmark variation. This analysis uses 20 published benchmark and configuration records from Book review and Yelp.P, with ten records per benchmark. The evidence was published in 2020 and falls within the 2010 to 2025 publication scope. Comparisons remain within benchmarks and source tables, while distillation comparisons match both benchmark and threshold. This structure enables direct calculation of interpretable accuracy differences without treating configuration records as independent experimental replications. This study aims to clarify how adaptive routing, exit uncertainty thresholds, and self-distillation relate to reported classification accuracy across the selected benchmarks. The central research question asks how accuracy changes across these configuration choices and what those changes imply for computational allocation. Specifically, the analysis compares adaptive inference with selected fixed depth alternatives, examines accuracy across available thresholds, and evaluates matched configurations with and without self-distillation. The results show higher accuracy for adaptive routing in all four selected routing comparisons, lower accuracy at higher thresholds in both benchmarks, and a positive distillation advantage in every matched pair. The contribution is to distinguish these consistent directions from their varying magnitudes. For language model deployment, this supports evaluating stopping settings and training arrangements within the intended task rather than inferring accuracy consequences from computational savings alone. The Chinese and English benchmarks provide evidence across two language settings, while the numerical advantages remain specific to the evaluated tasks and configurations.

2. Theory, Conceptual Framework and Research Questions Development:

This analysis adopts resource rationality as its theoretical foundation. Computational effort has value when the expected improvement in prediction warrants the additional processing required. Lieder and Griffiths, 2020, explain resource rationality as the allocation of limited computational resources to achieve effective performance under constraints. Applied here as a theoretical lens, this principle directs attention to how language models distribute processing across inputs rather than assuming that every input requires identical computational depth. Classification accuracy depends on both the information available within intermediate representations and the rule determining whether processing continues. The conceptual framework connects adaptive compute configuration with observed classification accuracy through three measured dimensions: adaptive inference, exit uncertainty threshold, and self-distillation. Adaptive inference determines whether processing depth varies across inputs. The exit threshold determines which intermediate predictions qualify for early termination. Self-distillation determines whether intermediate classifiers receive supervision from the final classifier. These dimensions represent distinct configuration choices rather than interchangeable indicators of a latent construct. The analysis examines their relationships with accuracy through comparisons within Book review and Yelp.P, preserving the separation between the two published source tables.

Adaptive inference addresses differences in the processing required by individual inputs. A fixed depth configuration allocates the same number of layers to every input, whereas adaptive routing permits earlier termination when an intermediate prediction meets the stopping criterion. The theoretical advantage arises from retaining additional computation for inputs that require further representation development. Xin and colleagues, 2020, demonstrate that early exits can reduce inference time while preserving much of the predictive performance of a complete BERT model. Their evidence supports the feasibility of selective processing, although it does not establish that adaptive routing always produces higher accuracy. The competing mechanism is premature termination. An intermediate classifier can produce a confident but incorrect prediction, preventing later layers from revising the decision. Adaptive routing therefore offers an allocation advantage only when its stopping rule distinguishes inputs that benefit from further processing. This analysis describes the accuracy differences between adaptive FastBERT and fixed depth FastBERT configurations within each benchmark. Computational requirements accompany these comparisons because the configurations do not use identical processing amounts. The first research question is therefore descriptive: RQ1: How does reported classification accuracy differ between adaptive inference and fixed depth FastBERT configurations within each benchmark?

The exit uncertainty threshold governs the selectivity of adaptive routing. A higher threshold accepts a wider range of intermediate predictions for early termination, including predictions with greater uncertainty. This creates a theoretical expectation of lower computational requirements alongside a possible reduction in accuracy. Conversely, a lower threshold reserves early termination for more certain predictions and allows additional processing for the remainder. Schwartz and colleagues, 2020, show how confidence informed stopping decisions can match model complexity to input complexity while allowing users to adjust the relationship between inference speed and accuracy. Their findings support configurable stopping rules rather than a universal threshold suitable for every task. The accuracy response need not be proportional to the threshold change. Where intermediate representations already separate the relevant classes, a higher threshold may remove computation with little accuracy loss. Where classification requires information developed in later layers, the same threshold adjustment may produce a larger decline. The analysis therefore compares reported accuracy across thresholds within each benchmark and source table while retaining adaptive routing and self-distillation. RQ2: How does reported classification accuracy vary across exit uncertainty thresholds when adaptive inference and self-distillation remain enabled?

Self-distillation addresses the competence of the intermediate classifiers on which stopping decisions depend. It transfers information from the final classifier to classifiers attached to earlier layers, providing supervision beyond the observed class labels. Gou and colleagues, 2021, identify knowledge transfer between teacher and student models as a central mechanism through which distillation can preserve predictive performance under computational constraints. This mechanism provides a theoretical basis for expecting stronger intermediate predictions when the final classifier supplies useful supervisory information. Its effectiveness nevertheless depends on the quality of that information and the capacity of the receiving classifiers. In FastBERT, Liu and colleagues, 2020, combine intermediate classifiers, uncertainty-based exits, and supervision from the final classifier. This architecture links the quality of intermediate predictions to the feasibility of terminating computation early. The available ablation

results permit four direct accuracy comparisons between configurations with and without self-distillation, matching the benchmark and exit threshold. These comparisons describe the observed contribution of the training configuration without treating the reported outcomes as independent replications. RQ3: How does reported classification accuracy differ between FastBERT configurations with and without self-distillation at matched benchmark and exit threshold settings?

Together, the three research questions distinguish the allocation of processing, the criterion for stopping, and the training of intermediate decision points. The framework specifies theoretical directions of influence, while the empirical analysis resolves descriptive differences among the available configurations. Benchmark specific comparisons preserve differences in task difficulty and prevent variation between datasets from being attributed to configuration choices.

3. Data:

This analysis documents the provenance and construction of a dataset containing published language model benchmark results. It defines the observed configuration measures, verifies their coverage, and establishes the comparisons used for descriptive analysis.

3.1 Data Sources, Population, and Sample Construction:

The empirical source is the final proceedings version of FastBERT published by the Association for Computational Linguistics. The analytical workbook contains results transcribed from source Table 2 on page 6039 and source Table 3 on page 6042. The publisher provides the authoritative article and experimental tables, while the workbook supplies the consolidated analytical records. Liu and colleagues, 2020, describe the underlying architecture, benchmark tasks, training procedures, and reported performance measures. The target population comprises published evaluations of adaptive language model inference within the 2010 to 2025 publication boundary. The accessible analytical frame is narrower: the 20 records already extracted for Book review and Yelp.P. Selection was purposive, retaining these benchmarks because the publication reports both configuration comparisons and ablation results for them. The extraction was constructed before the present descriptive calculations, but no dated prospective selection protocol is available. The unit of analysis is a published benchmark and configuration result within a specified source table. Each benchmark contributes ten observations. The dataset is a comparative cross section of experimental summaries published in 2020, rather than an annual panel covering the entire eligibility period. Book review contains Chinese text classification results, while Yelp.P contains English review classification results. These are computational benchmarks rather than geographical samples of countries, firms, or individuals. The workbook identifies the publisher version but does not provide a complete calendar extraction date. Its access note records November 2 without a year. The article carries an ACL copyright notice, and a separate licence for the extracted workbook has not been verified. These documentation gaps do not alter the recorded values but limit the completeness of the provenance record.

Table 1: Construction and analytical coverage of the sample

Stage	Units	Observations	Reason or Rule
Accessible workbook frame	2 benchmarks	20	Published results selected from two source tables
Record identity verification	2 benchmarks	20	All benchmark, table, configuration, threshold, and depth combinations are unique
Retained accuracy sample	2 benchmarks	20	Accuracy is available for every record
Applicable threshold sample	2 benchmarks	14	Six fixed depth records have structurally inapplicable thresholds
Matched distillation comparison	2 benchmarks	8 records forming 4 pairs	Match benchmark and threshold within source Table 3
Computational cost sample	2 benchmarks	19	Exclude one flagged FLOPs value from cost calculations

The table distinguishes record availability from comparison eligibility. The threshold restriction does not remove observations from other analyses, and the flagged computational value does not invalidate its accompanying accuracy observation.

3.2 Data Integration, Cleaning, and Quality Assurance:

The workbook consolidates results from one publication, so integration requires appending experimental records rather than merging independent datasets. Record identity uses benchmark, source table, configuration, exit threshold, and fixed depth or adaptive routing status. Verification identifies 20 unique combinations and no duplicate records under this definition. Repeated benchmark names represent evaluations of different configurations rather than duplicated observations. The analysis preserves the separation between source Tables 2 and 3. Configuration results from the two tables are not treated as repeated measurements of an identical experimental condition. Numerical fields retain the reported scales: accuracy in percent, computational requirements in millions of floating-point operations, and speedup as a dimensionless ratio. No temporal aggregation, currency conversion, interpolation, or rescaling was applied. Accuracy, adaptive inference status, and self-distillation status are populated for all 20 observations. Exit thresholds are numeric for 14 records and structurally inapplicable for six fixed depth configurations, comprising three records per benchmark. These entries remain coded NA. They are not replaced with zero because a zero threshold would describe a substantive stopping setting rather than the absence of an adaptive stopping rule.

One source inconsistency affects computational requirements. The result at threshold 0.2 reports 52,783 million operations alongside a reported speedup of 4.12. These values are incompatible with the reported baseline of 21,785 million operations. The workbook preserves the published value and flags it. The analysis excludes that FLOPs observation from computational comparisons without imputing a replacement or deleting its accuracy value. Verification also confirms binary configuration codes, applicable thresholds between zero and one, accuracy values between zero and 100, and positive reported computational requirements and speedup ratios. No winsorisation or automatic removal of extreme values was performed. The

flagged inconsistency is distinguished from an unusually large but internally coherent observation. The analytical extract provides no independent audit of the original training and evaluation partitions. Kapoor and Narayanan, 2023, identify such underlying data practices as relevant to interpreting published machine learning performance.

3.3 Variable Measurement, Construction, and Validation:

Classification accuracy is the reported percentage of correctly classified benchmark examples. Adaptive compute configuration is represented by three observed dimensions: adaptive inference, exit uncertainty threshold, and self-distillation. These dimensions are analysed separately rather than combined into an index. Computational requirements and speedup provide supplementary descriptions of configuration differences.

Table 2: Variable dictionary

Variable or Indicator	Analytical Role	Definition and Construction	Source and Identifier	Unit and Frequency	Transformation and Direction	Missing Value Rule	Coverage and Limitation
Classification accuracy	Outcome	Published classification accuracy	ACL article, source Tables 2 and 3	Percent per configuration	Untransformed; higher values indicate greater accuracy	None missing	20 records; rounded aggregate results
Adaptive inference	Configuration dimension	One for adaptive routing and zero for fixed depth	Published configuration descriptions	Binary per configuration	Higher value indicates adaptive routing	None missing	14 adaptive and 6 fixed depth records
Exit uncertainty threshold	Configuration dimension	Published stopping threshold	Source table configuration labels	Unit interval per applicable configuration	Higher values permit exits at greater uncertainty	Retain structural NA	14 numeric values and 6 structural NA entries
Self-distillation	Configuration dimension	One when retained and zero when absent	Architecture and ablation descriptions	Binary per configuration	Higher value indicates self-distillation	None missing	14 enabled and 6 absent configurations
Fixed depth or routing	Comparison descriptor	Published layer count or adaptive designation	Configuration descriptions	Layers or categorical routing status	Preserve reported designation	Adaptive entries are categorical	Fixed depths are 2, 6, or 12 layers
Computational requirements	Supplementary measure	Published floating point operations	Source Tables 2 and 3	Millions of operations per reported evaluation	Untransformed; higher values indicate greater computation	Exclude flagged value from cost analysis	19 usable values
Reported speedup	Supplementary measure	Published computational speedup relative to BERT	Source Tables 2 and 3	Dimensionless ratio	Higher values indicate greater reported acceleration	None missing	20 values; not interpreted as measured elapsed time
Benchmark and source table	Comparison identifiers	Preserve benchmark name and table number	Publisher tables and workbook	Categorical per record	No transformation	None missing	Two benchmarks and two source tables

Self-distillation denotes the training configuration rather than a measured quantity of transferred knowledge. Gou and colleagues, 2021, distinguish distillation mechanisms and training arrangements, supporting this separation between configuration status and predictive performance.

Table 3: Descriptive statistics for the principal measures

Measure	Valid Observations	Mean	Sample Standard Deviation	Median	Minimum	Maximum
Classification accuracy, percent	20	90.468	4.834	90.045	82.880	96.040
Book review accuracy, percent	10	85.875	1.270	86.320	82.880	86.980
Yelp.P accuracy, percent	10	95.061	0.921	95.250	93.110	96.040
Exit uncertainty threshold	14	0.457	0.277	0.500	0.100	0.800
Adaptive inference indicator	20	0.700	0.470	1.000	0.000	1.000
Self-distillation indicator	20	0.700	0.470	1.000	0.000	1.000

These statistics describe the selected configuration records. The pooled accuracy mean is not weighted by benchmark test size and does not estimate accuracy across all underlying text examples. Accuracy differences are expressed in percentage points, while speedup remains a ratio.

3.4 Analytical Framework, Identification, and Reproducibility:

The analysis is descriptive. Its quantities of interest are observed accuracy differences across routing arrangements, thresholds, and distillation settings. Comparisons remain within benchmarks because baseline task performance differs between Book review and Yelp.P. Matching processing complexity to input difficulty motivates this comparison structure; Schwartz and colleagues, 2020, demonstrate how configurable early exits connect computational allocation with predictive performance. Threshold comparisons retain adaptive routing and self-distillation, evaluating the available settings separately within each source table. Distillation comparisons match benchmark and threshold within source Table 3, producing four paired contrasts from eight records. Routing comparisons describe adaptive and fixed depth FastBERT configurations alongside their usable computational requirements. They do not identify an isolated routing effect at identical computational expenditure. No regression, significance test, confidence interval, or causal estimator is applied. The records share architectures, benchmark data, and a publication source, and therefore do not constitute 20 independent experimental replications. Standard deviations in Table 3 describe variation across recorded configurations rather than uncertainty around independently replicated model performance. Reproducibility rests on preserving source table identifiers, configuration labels, published values, structural NA entries, and the computational quality flag. The workbook contains the consolidated records and coding documentation, while the publisher tables remain the reference for transcription checks. Peng, 2011, distinguishes computational reproducibility from independent replication, a distinction that applies directly to recalculating these descriptive summaries from published aggregate evidence.

4. Methods:

4.1 Research Design and Analytical Scope:

This analysis uses a descriptive comparative design to examine reported classification accuracy across language model configurations. The unit of analysis is a published benchmark and configuration result within a specified experimental table. The research questions concern accuracy differences between adaptive and fixed depth processing, variation across exit uncertainty thresholds, and matched comparisons with and without self-distillation. The analytical quantities are differences between recorded configuration outcomes rather than estimated population effects. Useful variation arises from configuration changes within Book review and Yelp.P. Comparisons preserve benchmark identity and source table membership because task difficulty and experimental configuration can affect both accuracy and computational requirements. The sample construction and variable coding follow Table 1 and the dictionary in Table 2. The design reanalyses published FastBERT experiments rather than training or evaluating additional models. Liu and colleagues, 2020, provide the architectural descriptions and experimental results underlying the comparisons. The analysis distinguishes configuration contrasts that hold selected characteristics constant from comparisons involving different computational expenditures. It does not interpret these contrasts as independently identified causal effects.

4.2 Comparison Specification and Calculation:

Accuracy comparisons use differences on the published percentage scale. Equation 1 defines the contrast applied to each eligible pair.

$$\Delta A_{b,t,c} = A^{F}_{b,t,c} - A^{R}_{b,t,c}$$

Here, $\Delta A_{b,t,c}$ denotes the accuracy difference in percentage points for benchmark b, source table t, and comparison c. $A^{F}_{b,t,c}$ is the reported accuracy of the focal configuration, and $A^{R}_{b,t,c}$is the accuracy of its reference configuration. Positive values indicate higher accuracy for the focal configuration. For the routing question, the focal configurations are adaptive FastBERT at thresholds 0.2 and 0.7. Their reference configurations are fixed depth FastBERT with six and two layers, respectively. Self-distillation remains enabled. These four comparisons describe the selected published alternatives without treating their computational requirements as identical. For the threshold question, the focal configuration uses the higher threshold and the reference configuration uses the lower threshold. Adaptive routing and self-distillation remain enabled. Comparisons are calculated separately for each benchmark and source table. The main comparison provides thresholds 0.1, 0.5, and 0.8, permitting adjacent interval and endpoint contrasts. The ablation experiment provides thresholds 0.2 and 0.7. A negative accuracy difference indicates lower accuracy at the higher threshold. For the distillation question, configurations are matched by benchmark, source table, adaptive routing status, and exit threshold. The focal configuration retains self-distillation, while the reference configuration omits it. Two benchmarks and two thresholds generate four matched pairs. The mean paired difference gives each configuration contrast equal weight; benchmark specific means average the two available contrasts within each benchmark. Computational reductions accompany accuracy comparisons where both operation counts are usable. Equation 2 specifies their calculation.

$$R_{b,t,c} = 100\left[1 - \frac{F^{F}_{b,t,c}}{F^{R}_{b,t,c}}\right]$$

Here, $R_{b,t,c}$ is the computational reduction expressed as a percentage. $F^{F}_{b,t,c}$ and $F^{R}_{b,t,c}$ are the focal and reference computational requirements, measured in millions of floating-point operations. Positive values indicate fewer operations for the focal configuration. The reference is the fixed depth alternative for routing comparisons and the lower threshold configuration for threshold comparisons. Reported speedup ratios remain distinct from reductions calculated using Equation 2.

4.3 Analytical Samples, Processing, and Verification:

The processing sequence begins with the consolidated workbook and preserves the original values, configuration labels, benchmark identifiers, source table numbers, and quality flag. Record identity is verified using benchmark, source table, configuration, threshold, and fixed depth or adaptive routing designation. This composite identifier distinguishes genuine configuration differences from duplicate records. Descriptive summaries use all 20 accuracy observations. The threshold sample contains 14 applicable numeric settings; six fixed depth records retain structural NA entries. These entries are excluded only from analyses requiring a threshold. They are not imputed or replaced with zero. The matched distillation analysis uses eight records forming four pairs, while the routing analysis uses eight records forming four configuration comparisons. Computational comparisons exclude the flagged Yelp.P operation count at threshold 0.2. Its accuracy remains eligible because the inconsistency concerns the computational measure. Consequently, computational requirements are usable for 19 records, three routing comparisons, and three threshold endpoint comparisons. No replacement value is inferred from the accompanying speedup ratio.
Verification checks assess unique record identities, binary configuration codes, applicable thresholds within the unit interval, accuracy within zero to 100 percent, and positive computational requirements. The flagged operation count is retained in the source record and excluded through an analysis specific restriction. No interpolation, winsorisation, standardisation, or automatic removal of extreme observations is applied. Means, medians, ranges, and sample standard deviations summarise the recorded values. Standard deviations use the number of valid observations less one as the denominator. Pooled accuracy summaries weight each configuration equally rather than weighting by benchmark test size. Interpretation therefore relies primarily on benchmark specific summaries and matched contrasts.

4.4 Computation, Inference, and Reproducibility:

Calculations were performed in Python 3.12.14. The standard library modules zipfile and xml.etree. Element Tree were used to read workbook contents, and statistics supplied arithmetic means, medians, and sample standard deviations. Configuration filtering, matching, and the calculations in Equations 1 and 2 used deterministic operations. No random sampling, optimisation, or model fitting was required. Calculations retained the available numerical precision, with rounding applied when presenting results. The analysis reports observed differences and descriptive dispersion without significance tests, standard errors, or confidence intervals. The records share benchmarks, architectures, and a publication source, while repeated training runs and individual prediction records are unavailable in the analytical extract. Treating configuration observations as independent replications would

therefore misrepresent the basis for inference. Sample standard deviations describe variation among configurations rather than uncertainty around replicated performance estimates. Comparisons across benchmarks and source tables assess whether observed directions recur under the available settings. These comparisons retain their separate analytical samples and do not constitute independent replication or a prospectively specified robustness programme. The configuration selection and quality restriction are documented, but no preregistered analysis protocol is available. Replication requires the workbook values, the coding dictionary, the comparison definitions, and Equations 1 and 2. The workbook preserves source identifiers and the computational quality flag, while the publisher tables provide the authoritative transcription reference. This documentation follows the reproducibility principle articulated by Sandve and colleagues, 2013: each result should remain traceable to its inputs and processing steps. The method contributes a transparent comparative reanalysis using established calculations rather than a new estimator or identification strategy.

5. Findings:

This analysis examines classification accuracy across adaptive routing, exit thresholds, and self-distillation configurations. The findings resolve three descriptive research questions through comparisons within benchmarks and published experimental tables. Accuracy differences and computational requirements establish the direction, magnitude, and boundaries of the observed relationships. The analytical sample contains 20 published configuration results, with ten observations each for Book review and Yelp.P. All results originate from experiments reported in 2020. Fourteen configurations use adaptive inference, and six use fixed depth processing. Accuracy is available for every observation. Exit thresholds apply to 14 observations, while one inconsistent computational value is excluded from cost comparisons. The underlying results come from the FastBERT comparison and ablation experiments reported by Liu and colleagues, 2020.

Table 4: Classification accuracy across the selected configurations

Benchmark	Configuration Observations	Mean Accuracy, Percent	Sample Standard Deviation	Median Accuracy, Percent	Minimum Accuracy, Percent	Maximum Accuracy, Percent
Book review	10	85.875	1.270	86.320	82.880	86.980
Yelp.P	10	95.061	0.921	95.250	93.110	96.040
Combined	20	90.468	4.834	90.045	82.880	96.040

Yelp.P records higher accuracy and less dispersion across the selected configurations than Book review. The benchmark means differ by 9.186 percentage points, exceeding the accuracy differences generated by individual configuration comparisons below. Consequently, pooled accuracy summaries conceal differences in benchmark difficulty and provide an unsuitable basis for attributing performance differences to adaptive compute configuration. The substantive comparisons therefore retain benchmark identity. For RQ1, adaptive FastBERT records higher accuracy than both selected fixed depth alternatives in each benchmark. Table 5 presents these comparisons while retaining self-distillation. The lower threshold configuration is compared with six layers, and the higher threshold configuration with two layers. These pairings describe the published alternatives without assuming identical computational expenditure.

Table 5: Adaptive inference compared with fixed depth FastBERT

Benchmark	Adaptive Threshold	Fixed Depth Comparator, Layers	Adaptive Accuracy, Percent	Fixed Depth Accuracy, Percent	Adaptive Accuracy Advantage, Percentage Points	Adaptive Flops, Millions	Fixed Depth Flops, Millions	Computational Reduction, Percent
Book review	0.2	6	86.98	86.42	0.56	9,725	11,123	12.57
Book review	0.7	2	85.69	82.88	2.81	3,621	3,707	2.32
Yelp.P	0.2	6	95.90	95.18	0.72	Excluded	11,123	Not calculated
Yelp.P	0.7	2	94.67	93.11	1.56	2,757	3,707	25.63

The Book review comparison at threshold 0.7 provides the closest computational match. Adaptive processing achieves 2.81 percentage points higher accuracy while requiring 2.32 percent fewer operations than the two layer configuration. For Yelp.P, the corresponding accuracy advantage is 1.56 percentage points with 25.63 percent fewer operations. All three comparisons with usable computational values therefore combine higher accuracy with lower computation. This pattern is consistent with the selective processing rationale examined by Xin and colleagues, 2020, although the present comparisons describe specific configurations rather than an isolated causal effect of routing. RQ1 is resolved descriptively: adaptive configurations outperform the selected fixed depth alternatives in all four accuracy comparisons. The accuracy advantage is larger against two layers than against six layers in both benchmarks. Computational superiority is established for the three comparisons with internally usable cost values. For RQ2, higher exit thresholds consistently correspond to lower accuracy among configurations retaining adaptive inference and self-distillation. Table 6 separates the threshold settings reported in the main comparison from those reported in the ablation experiment.

Table 6: Accuracy and computation across exit thresholds

Benchmark	Published Experiment	Threshold Increase	Initial Accuracy, Percent	Final Accuracy, Percent	Accuracy Decline, Percentage Points	Computational Reduction, Percent
Book review	Main comparison	0.1 to 0.8	86.88	85.14	1.74	77.87
Yelp.P	Main comparison	0.1 to 0.8	95.99	94.31	1.68	63.11
Book review	Ablation	0.2 to 0.7	86.98	85.69	1.29	62.77
Yelp.P	Ablation	0.2 to 0.7	95.90	94.67	1.23	Not calculated

The main comparison reveals different accuracy responses across the threshold intervals. Increasing the threshold from 0.1 to 0.5 reduces Book review accuracy by 0.24 percentage points, whereas the subsequent increase to 0.8 reduces it by 1.50 percentage points. For Yelp.P, the corresponding declines are 0.67 and 1.01 percentage points. The observed accuracy cost is therefore concentrated more heavily in the higher threshold interval, particularly for Book review. These discrete comparisons do not establish a continuous response function, but they show that the observed threshold increments carry different accuracy costs.

RQ2 is resolved in the same direction across both benchmarks and both experimental tables. More permissive stopping thresholds accompany lower accuracy, while every usable endpoint comparison records lower computational requirements. The pattern accords with the configurable accuracy and computation relationship described by Schwartz and colleagues, 2020. For RQ3, matching benchmark and exit threshold produces four comparisons between configurations with and without self-distillation. Accuracy is higher with self-distillation in every pair.

Table 7: Matched accuracy comparisons for self-distillation

Benchmark	Exit Threshold	Accuracy With Self-Distillation, Percent	Accuracy Without Self-Distillation, Percent	Accuracy Advantage, Percentage Points
Book review	0.2	86.98	86.22	0.76
Book review	0.7	85.69	85.02	0.67
Yelp.P	0.2	95.90	95.55	0.35
Yelp.P	0.7	94.67	94.54	0.13

The mean accuracy advantage across the four matched pairs is 0.478 percentage points. The advantage averages 0.715 percentage points for Book review and 0.240 percentage points for Yelp.P. It remains positive at both thresholds but becomes smaller at threshold 0.7 in each benchmark. These results resolve RQ3 as a consistent positive accuracy difference within the available matched configurations. They align with the predictive role of teacher supervision discussed by Gou and colleagues, 2021, while demonstrating that its observed magnitude varies across benchmark and stopping setting. The combined evidence distinguishes three configuration relationships. Adaptive routing records higher accuracy than the selected fixed depth alternatives. Higher exit thresholds accompany lower accuracy and reduced computational requirements wherever those requirements are usable. Self-distillation records higher accuracy in all four matched comparisons. These findings are recalculations of published aggregate results rather than independent experimental replications, consistent with the distinction between computational reproducibility and replication articulated by Peng, 2011.

6. Discussion:

Less computation need not imply lower accuracy, but the configuration used to remove computation matters. Table 5 shows that adaptive FastBERT achieves higher accuracy than the selected fixed depth alternatives while requiring fewer operations in all three comparisons with usable computational values. The closest computational comparison, involving Book review and two layers, indicates that processing allocation can matter beyond the amount of processing alone. This interpretation fits the distinction between static architectures and input dependent computation developed by Han and colleagues, 2022. The contribution of this analysis is a structured comparison of published configuration outcomes that separates routing, stopping thresholds, and training arrangements. It does not establish a previously undiscovered architectural effect. Instead, it clarifies which comparisons support an allocation advantage and which require qualification because computational expenditure differs. The matched comparisons in Table 7 identify a consistent accuracy advantage associated with self-distillation, while showing that its magnitude depends on the benchmark and stopping setting. The smaller advantage at the higher threshold in both benchmarks suggests that improving intermediate classifiers does not eliminate the accuracy cost of more permissive stopping. Training quality and stopping policy therefore address distinct aspects of adaptive inference. Zhu, 2021, extends intermediate supervision by allowing exits to learn from one another rather than exclusively from the final layer, illustrating why the organisation of supervisory information matters alongside the exit rule. The present evidence sharpens this distinction through matched configuration comparisons: self-distillation accompanies higher accuracy, but its advantage cannot be assumed constant across deployment settings. The conceptual implication is that intermediate prediction quality and the decision to accept that prediction should be evaluated separately.

Table 6 further shows that the accuracy cost of threshold adjustment is uneven across the observed intervals. For Book review, the accuracy decline is concentrated in the higher threshold interval, whereas Yelp.P exhibits a less concentrated pattern. This difference challenges the practical assumption that successive threshold increases impose comparable predictive costs. It also distinguishes uncertainty based stopping from alternative termination rules. Zhou and colleagues, 2020, use consistency across

successive predictions to determine when processing stops and report that early termination can improve accuracy under their evaluated conditions. Their result contrasts with the accuracy declines accompanying higher uncertainty thresholds here. The distinction redirects interpretation from early exit as a uniform intervention toward the criterion governing each exit. Table 6 supports benchmark specific threshold evaluation rather than a universal setting inferred from aggregate computational savings.

The evidence also exposes a consequential separation between predictive comparisons and computational comparisons. The flagged Yelp.P value prevents one cost comparison in Table 5, while leaving its accuracy comparison available. Retaining this distinction avoids allowing an inconsistent denominator to determine an efficiency claim or discarding an otherwise usable outcome. Sandve and colleagues, 2013, emphasise preserving the provenance of computational results and documenting transformations sufficiently for their reconstruction. Applied here, that principle makes the evidential boundary explicit: the direction of the observed accuracy advantages is more completely documented than their computational counterparts. Table 4 adds a second constraint by showing substantial differences between benchmark accuracy distributions. Together, these observations establish that configuration comparisons require both metric consistency and benchmark matching; pooling outcomes or silently repairing source values would obscure the relationships the analysis seeks to explain.

The international relevance lies in evaluating a common computational design problem across Chinese and English classification benchmarks, rather than attributing outcomes to national economic conditions. Branchy Net, developed by Teerapittayanon and colleagues, 2016, demonstrated early exit advantages in image classification, whereas Tables 5 to 7 examine how routing, thresholds, and self-distillation relate to language classification outcomes. This comparison extends the interpretive setting across application domains without assuming that magnitudes transfer between them. For deployment decisions, the evidence supports assessing adaptive configurations against fixed depth alternatives at comparable computational requirements, examining threshold intervals separately, and retaining self-distillation where its matched accuracy advantage matters. The broader theoretical refinement is that computational efficiency depends on coordinating allocation, acceptance, and prediction quality. The observed configuration differences support that distinction, while their benchmark dependence makes task specific evaluation central to its practical use.

7. Conclusion and Implications:

Efficient language classification depends on where computation is allocated and when predictions are accepted. This analysis shows that selective processing, stopping decisions, and intermediate supervision accompany distinct accuracy differences whose magnitudes vary across benchmark settings. Their complementary roles suggest that computational economy requires coordinating processing allocation with prediction quality rather than reducing execution depth indiscriminately. The core contribution is an explicit comparative interpretation of published results that separates these decisions and identifies uneven accuracy costs across stopping settings. This perspective refines the resource allocation argument by showing why consistent directional patterns can coexist with task dependent performance differences. For managers, the evidence supports comparing candidate configurations against operationally relevant alternatives and selecting settings according to acceptable accuracy losses and verified computational requirements. Operational teams should evaluate successive stopping adjustments separately because their predictive costs need not be proportional. For institutions procuring language classification systems, the findings support requesting transparent reporting of evaluation tasks, configuration choices, and computational denominators before accepting efficiency claims. Across language settings, these practices can improve the accountability of resource allocation decisions. Their wider social relevance lies in supporting more informed choices about computationally constrained services, while preserving the distinction between demonstrated classification performance and anticipated deployment benefits.

The available evidence provides a foundation for targeted replication across additional languages, tasks, and architectures. Its aggregate configuration records support descriptive comparisons, while repeated training runs and individual prediction records would permit uncertainty estimation and stronger tests of proposed mechanisms. Controlled experiments matching computational expenditure could distinguish allocation advantages from differences in processing quantity. Direct measurements of elapsed time and energy would extend evaluation beyond operation counts. Broader sampling could establish when benchmark specific patterns transfer to operational workloads. These extensions would strengthen generalisation while preserving the present contribution as a transparent reanalysis of published comparative evidence.

References:

1. Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. International Journal of Computer Vision, 129(6), 1789-1819. https://doi.org/10.1007/s11263-021-01453-z
2. Han, Y., Huang, G., Song, S., Yang, L., Wang, H., & Wang, Y. (2022). Dynamic neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(11), 7436-7456. https://doi.org/10.1109/TPAMI.2021.3117837
3. Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. Patterns, 4(9), Article 100804. https://doi.org/10.1016/j.patter.2023.100804
4. Lieder, F., & Griffiths, T. L. (2020). Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. Behavioral and Brain Sciences, 43, Article e1. https://doi.org/10.1017/S0140525X1900061X
5. Liu, W., Zhou, P., Wang, Z., Zhao, Z., Deng, H., & Ju, Q. (2020). FastBERT: A self-distilling BERT with adaptive inference time. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 6035-6044). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.537
6. Peng, R. D. (2011). Reproducible research in computational science. Science, 334(6060), 1226-1227. https://doi.org/10.1126/science.1213847
7. Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. PLOS Computational Biology, 9(10), Article e1003285. https://doi.org/10.1371/journal.pcbi.1003285

8. Schwartz, R., Stanovsky, G., Swayamdipta, S., Dodge, J., & Smith, N. A. (2020). The right tool for the job: Matching model and instance complexities. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 6640-6651). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.593
9. Teerapittayanon, S., McDanel, B., & Kung, H. T. (2016). Branchy Net: Fast inference via early exiting from deep neural networks. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 2464-2469). IEEE. https://doi.org/10.1109/ICPR.2016.7900006
10. Xin, J., Tang, R., Lee, J., Yu, Y., & Lin, J. (2020). DeeBERT: Dynamic early exiting for accelerating BERT inference. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 2246-2251). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.204
11. Zhou, W., Xu, C., Ge, T., McAuley, J., Xu, K., & Wei, F. (2020). BERT loses patience: Fast and robust inference with early exit. Advances in Neural Information Processing Systems, 33, 18330-18341. https://doi.org/10.48550/arXiv.2006.04152
12. Zhu, W. (2021). LeeBERT: Learned early exit for BERT with cross-level optimization. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 2968-2980). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.231