Research on BERT Model Compression and Acceleration Based on Knowledge Distillation and Dynamic Routing

Li Zhang[1], Yanqin Cao[1]
[1]Sichuan Vocational College of Finance and Economics, Chengdu 610101, Sichuan, China
Corresponding author: Li Zhang (e-mail: lizhanguestc@163.com)

ABSTRACT The BERT model has achieved strong performance in natural language processing, but its large parameter scale and high computational cost limit deployment in resource-constrained and real-time scenarios. This study proposes a lightweight BERT compression and acceleration method that integrates knowledge distillation and dynamic routing. In the knowledge distillation stage, a multi-layer feature alignment and attention transfer mechanism is developed to transfer deep semantic knowledge from the teacher model to a compact student model. The strategy uses output-layer distillation, intermediate-layer feature alignment, and attention-map transfer to preserve semantic representation ability. In the dynamic routing stage, an adaptive inference structure based on early exit is constructed, enabling the model to adjust computational depth according to the complexity of different input samples and reduce redundant computation. A joint optimization strategy is used to balance final-output accuracy, early-exit classifier performance, and distillation effectiveness. Experimental results on multiple natural language processing tasks show that the proposed method substantially reduces parameter count, storage space, and inference latency while maintaining competitive performance relative to the original BERT model.

INDEX TERMS BERT, model compression, knowledge distillation, dynamic routing, inference acceleration

I. INTRODUCTION

BERT, as a pre-trained language model, is based on the Transformer architecture with leading performances at numerous tasks of natural language processing because of its capabilities to encode bidirectionally. However, BERT models are usually implemented with hundreds of millions or even billions of parameters, and this requires a considerable amount of storage and computing resources in practical deployments. Especially for mobile, embedded or real-time applications the inference latency and the power consumption are serious bottlenecks. Therefore, solving the problem of efficient model compression and acceleration while ensuring the same model performance has become an academic and industrial issue of common concern.

Existing model compression techniques mainly include pruning, quantization, low-rank decomposition and knowledge distillation. Knowledge distillation in particular is able to miniaturise models without substantial performance cost by transferring the output distribution of models or even their intermediate representations of complex teacher models to compact student models. However, traditional knowledge distillation methods usually only care about the final output of the teacher model, not the rich structural information contained in the intermediate layers, and therefore the student model may result in a limited ability in learning semantic representations. On the other hand, inference acceleration methods usually optimize the model structure at a higher level, e.g. designing more efficient attention mechanisms or reducing the number of network layers. However, there is no way to exploit the difference between samples in these methods since they use the same computation path for all samples for on-the-fly computation.

To overcome the above problems, this paper proposes a BERT model compression and acceleration method that combines knowledge distillation and dynamic routing. At the compression level, a multi-layered knowledge distillation strategy is adopted, which does not only transmit the output probabilities of the teacher model, but also adds the transmission of intermediate layer feature alignment and attention map, so that the student model can learn the comprehensive semantics of the teacher model. Based on confidence level, an early exit mechanism is designed at the acceleration level. During inference, it dynamically decides if it wants the current layer computation to be done or not based on the confidence level of the previous layer's output, thus preventing redundant computation to be performed on simple samples.

II. RELATED WORK

With its powerful contextual semantic modeling capabilities, BERT and its variants have made significant progress in many natural language processing tasks such as entity matching, sentiment analysis, question answering, and code defect prediction. Paganelli et al. [1] pointed out that BERT is significantly better than traditional methods and non-pre-trained deep models in entity matching tasks, especially in handling text-intensive attributes and complex semantic matching. Kondra et al. [2] reviewed and analyzed the representative models of multimodal BERT. Danyal et al. [3] used an ensemble learning strategy to fuse the prediction results of BERT and XLNet, and took advantage of the complementary advantages of the two models in semantic representation. Karabila et al. [4] constructed a BERT-based sentiment analysis module to perform fine-grained sentiment classification on user comment texts, and combined sentiment features with user history behavior, product attributes, etc., and input them into the recommendation model. Özkurt [5] compared and analyzed the performance of the current mainstream Transformer-based question answering models (BERT, RoBERTa, DistilBERT, ALBERT) on SQuAD v2.0 to evaluate their applicability in real-world scenarios. Mazari et al. [6] constructed multiple BERT variants as base learners and adopted voting or weighted fusion strategies to form an ensemble model. Deshmukh and Raut [7] used BERT to jointly encode resume text and job description, and constructed a candidate ranking model by calculating semantic similarity and key skill matching degree. Mujahid et al. [8] used RoBERTa and BERT as dual backbone networks and constructed an ensemble classification model by feature fusion or decision fusion. Zaki [9] compared the architectural differences, training strategies and application effects of these three representative Transformer models in various NLP tasks, including text processing and potential translation applications, and discussed their respective advantages and limitations. Uddin et al. [10] used BERT to pre-train the representation of source code text, extracted deep semantic features, and then input them into a BiLSTM network to capture the sequence dependency relationship in the code. This paper focuses on the research of BERT model compression and acceleration based on knowledge distillation and dynamic routing. By transferring the capabilities of large models to a compact structure through knowledge distillation, and introducing a dynamic routing mechanism to realize on-demand computation during the inference process, the aim is to build a lightweight BERT model that balances performance and efficiency, and provide a feasible solution for efficient deployment in resource-constrained environments.

III. METHODS

A. DESIGN OF MULTI-LEVEL KNOWLEDGE DISTILLATION MODULE

In order to effectively reduce the number of parameters of the BERT model and ensure the semantic representation ability, a multi-level knowledge distillation mechanism was built. The mechanism: The original BERT-base model is used as the teacher model and the Transformer model with fewer layers and smaller hidden layer dimensions is used as the student model. The distillation process not only soft label supervision of final output of teacher model, but also adds the intermediate layer feature alignment, attention matrix transfer, to ensure that the student model learns the semantic structure of the teacher model at multiple abstraction levels [11].

In output layer distillation, the usual knowledge distillation loss function is used to fit the class probability distribution of the student model to the probability distribution of the teacher model. Specifically, for each input sample, the Softmax outputs of the teacher and the student models are computed separately and the probability distributions will be softened using a temperature parameter. The difference between the two is measured as Kullback-Leibler divergence. This loss term helps the student model in learning the hidden knowledge given by the teacher model between the categories and thereby enhancing the generalization ability.

For the purpose of intermediate layer feature alignment layer by layer mapping mechanism is designed. Since the number of layers in teacher model and student model are different, a layer to layer correspondence is first established, matching each layer of student model with a semantically similar layer in teacher model. For both matched layers, mean square error of their hidden layer representations is determined as the feature alignment loss. This loss term requires the student model to learn the corresponding feature representation of the teacher model at each layer, which ensures the preservation of the intermediate information of the teacher model in semantic encoding.

In as far as attention matrix transfer is concerned, attention graph distillation is further introduced. The Transformer model attention mechanism controls the interaction between words and is the key to semantic modeling. The multi-head attention matrix of each layer of the teacher model is adopted as the knowledge source to guide the attention matrix of the student model to be consistent with the former. By finding the mean squared error between the attention matrices of the teacher and student, the student model learns the fine-grained pattern of the teacher model in word relationship modeling. This transfer strategy can be effective for enhancing the ability of the student model to capture the long-distance dependency and complicated semantic structure [12-13].

Summing up the three losses above in the distillation functions yields the overall distillation loss function. In the preparation of the teacher model, the model is pre-taught and the parameters are fixed. Then, classification loss and distillation loss of student model are jointly optimized in labelled data of target task. Through this multi-level distillation mechanism the student model can preserve the semantic expressive power of the teacher model to the maximum possible degree while greatly reducing the number of

parameters.

B. CONSTRUCTION OF DYNAMIC ROUTING MECHANISM

Based on the lightweight student model obtained through distillation, a dynamic routing mechanism is further introduced to accelerate inference. The core idea of this mechanism is to dynamically determine the computational depth of the model according to the complexity of the input samples. For simple samples, shallow computation is sufficient to output results, while for complex samples, a full network is used for fine-grained inference, thereby avoiding redundant computation of the same depth for all samples. A dynamic routing structure based on an early-retreat strategy was constructed. A classifier is inserted after each layer of the Transformer encoder. Each classifier consists of a pooling layer and a fully connected layer, and can generate a prediction result based on the output representation of the current layer. During inference, the input sample passes through the Transformer encoder layer by layer. After each layer, the classifier corresponding to that layer calculates the confidence score of the current prediction. If the confidence score exceeds a preset threshold, subsequent calculations are immediately stopped, and the result of that classifier is directly output; if none of the layers reach the threshold, the final result is output by the last layer classifier.

The confidence score is calculated using the maximum predicted probability as a metric. While maximum predicted probability is used here for its computational efficiency, it is acknowledged that this metric can occasionally overestimate confidence in out-of-distribution scenarios. Future iterations may incorporate calibration techniques such as temperature scaling to further refine the reliability of the early-exit trigger. For classification tasks, the maximum predicted probability reflects the degree of certainty of the model in the current prediction. The higher the value, the clearer the model's judgment of the sample. By setting a reasonable confidence score threshold, a flexible adjustment can be made between accuracy and efficiency. When the threshold is high, more samples need to undergo deep computation, resulting in higher accuracy but limited acceleration; when the threshold is low, more samples drop out early, resulting in significant acceleration but potentially introducing accuracy loss [14-15].

The advantage of this dynamic routing mechanism lies in its ability to adaptively adjust the computation depth based on the characteristics of the samples themselves. In practical applications, many input samples often have relatively obvious class features and can be accurately classified without going through all Transformer layers. Through the early termination mechanism, these samples only require a small amount of computation to complete inference, thus significantly reducing the average inference latency. At the same time, for boundary samples or complex samples, the prediction accuracy can still be guaranteed through the complete network.

C. JOINT OPTIMIZATION AND REASONING STRATEGY

A complete model compression and acceleration framework is built, which involves the joint optimization of multi-level knowledge distillation and dynamic routing mechanisms. During the training phase, multi-level distillation is applied first to transfer knowledge from the teacher model to the student model in order to provide the student model with a good initial semantic representation capability. Subsequently, when dynamic routing training is performed, the early-leave classifier is included in the optimization objective at the same time, and multi-task learning is adopted to achieve the purpose of making each layer of the classifier have independent prediction ability.

At the time of joint training, the loss function is made up of several parts. The basic classification loss is used to supervise the last layer classifier to ensure the accuracy of the final output, the early regression classification loss is used to supervise the intermediate classifiers in each layer, to enable each layer classifier to output the corresponding accurate prediction results, and the distillation loss is used to continuously guide the student model to learn the output distribution and intermediate features of the teacher model. The overall joint optimization loss function is formulated as: $L_{total} = \alpha L_{cls} + \beta \sum L_{earlycls} + \gamma L_{distill}$, where α, β, γ are hyper-parameters that balance the accuracy of the final output, the efficiency of early exits, and the effectiveness of knowledge transfer. In our experiments, we empirically set $\alpha = 1.0$, $\beta = 0.3$, and $\gamma = 0.7$ based on grid search to achieve the optimal trade-off between model performance and reasoning speed. By balancing the weights of each loss term, the model accuracy and the inference efficiency is synergistically optimized.

During the inference phase, a dynamic early termination strategy is used for acceleration purposes. With each input sample the computation is done layer by layer, i.e. it starts from the first layer. After each layer the classifier after the layer makes its prediction result and also computes its confidence score. If the confidence score hits a certain score threshold, then computation immediately stops and the current prediction result is returned, otherwise the process continues its path to the next layer. This process fully considers the difference between samples, so that the model can be adaptively adjusted according to the input complexity during the actual deployment, and finally the on-demand inference is achieved.

Through the joint optimization and inference strategies as above, the average inference latency of the proposed framework is substantially reduced while high prediction accuracy is maintained. At the same time, because the number of parameters in the student model itself has been greatly reduced, storage overhead and the consumption of calculation resources are also well controlled.

IV. RESULTS AND DISCUSSION

A. ANALYSIS OF MODEL COMPRESSION EFFECT

In order to validate the proposed method in model compression, experiments were performed on several tasks in GLUE benchmark. One of the chosen models was BERT-base with 110M parameters, the teacher. The model used by the students was a 6-layer Transformer model with 384 hidden layers, 6 multi-head attention heads and around 32M parameters. In multi-level knowledge distillation, the compression performance was compared with three settings: output layer distillation, output layer distillation and feature alignment, and complete multi-level distillation. Experimental results show that when only output layer distillation is used, the student model has an average accuracy of about 91.2% of the teacher model on the validation set. Adding feature alignment loss makes the average accuracy better to 93.5%. Further introducing attention matrix transfer boosts further the average accuracy of 94.8 percent of the teacher model. This shows that the multi-level distillation strategy is effective to boost the student model to absorb the semantic knowledge from the teacher model, and the improvement is more significant in complex tasks, where where the performance gains are achieved through the attention transfer from the teacher model to the student model. In terms of parameter size, the student model is down 70.9% from the teacher model's 110MB. Under the standard 32-bit floating-point (FP32) precision used in this study, the model storage space reduced from about 440MB to about 128MB, greatly saving the storage capacity demand of the device. In actual deployment, model storage space reduced from about 440MB to about 128MB, greatly saving the storage capacity demand of the device. Table 1 presents a comparison between the size of the parameter produced and their relative performances of the student model using different distillation strategies.

As shown in Table 1, with a large reduction in the number of parameters and storage space, the complete multi-level distillation strategy can retain the performance of the student model with approximately 94.8% of that of the teacher model, thereby verifying the effectiveness of the proposed compression method to reserve the semantic capabilities. The resulting increase in performance from the transfer of attention is especially important, suggesting that modeling the relationship between words plays an important role in the semantic representation of BERT. Using it as the target of knowledge transfer allows the student model to learn the semantic structure of the teacher model in a more accurate way. To evaluate the model's consistency, we further report the detailed results on individual GLUE tasks. For example, on the SST-2 sentiment analysis task, the student model with complete multi-level distillation achieved 91.5% accuracy, while on the MNLI (Matched/Mismatched) natural language inference task, it maintained 83.2% and 83.5% respectively. These results demonstrate that the proposed method performs consistently across different linguistic tasks, including sentiment classification and semantic inference, without significant degradation in specific domains.

B. ACCELERATED ANALYSIS OF MODEL INFERENCE

Based on the compressed model, the influence of the dynamic routing mechanism on inference speed was further tested. To ensure reproducibility, experiments were performed on a server equipped with an Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz, 32GB of DDR4 RAM, running Ubuntu 20.04 LTS. The average inference latency was measured on this CPU environment for the original BERT-base model, the compressed student model without the early termination mechanism, and the student model with the dynamic early termination mechanism. The test dataset contained three different types of samples including simple samples (short texts with clear class features), medium samples (medium length, relatively clear class boundaries), and complex samples (long texts with blurred class boundaries). The confidence threshold was set at 0.85, that is, the model would stop early if the maximum predicted probability of each classifier layer reached 85%. Experimental results show that when the original model, BERT-base, is used, the average inference latency is 12.4 milliseconds per sample. The compressed student model, due to the lower number of parameters, has an average inference latency of 8.7 milliseconds per sample which is about 70% of the original model. By adding a dynamic early termination mechanism further, the average inference latency is reduced to 7.1 milliseconds per sample. Analysis of the categories of samples reveals the following percentage of samples exiting at the second or third layer early in simple samples the percentage is about 86% and the average number of milliseconds about 3.2 milliseconds, in medium samples the percentage is about 58% and the average number of milliseconds is about 6.5 milliseconds, in complex samples the percentage of samples that exit early is about 12% and the percentage of samples that must go through all the six layers is about 88% for an average latency time of 9.8 milliseconds. This means that the dynamic routing mechanism is successful to distinguish sample complexity and on-demand computation. Table 2 shows the comparison of average inference latency for different models on various types of samples.

As shown in Table 2, the dynamic early termination mechanism has the most significant acceleration effect on simple samples, reducing latency to 25.8% of the original BERT-base. Medium and complex samples also show varying degrees of acceleration. The overall reduction in average latency indicates that the dynamic routing mechanism can fully utilize the differences between samples and effectively improve inference efficiency in practical deployments.

C. COMPREHENSIVE PERFORMANCE COMPARISON AND ABLATION ANALYSIS

To further verify the overall effectiveness of the proposed framework, a comprehensive performance comparison was conducted on multiple natural language processing tasks, including sentiment classification, natural language infer-

TABLE 1. Comparison of parameter count and performance of the student model under different distillation strategies

Distillation strategy	Number of parameters (M)	Relative teacher model performance (%)	Storage space (MB)
Output layer distillation only	32	91.2	128
Output layer + feature alignment	32	93.5	128
Output layer + feature alignment + attention transfer	32	94.8	128
Teacher model (BERT-based)	110	100	440

TABLE 2. Average inference latency (milliseconds/sample) for different models on various types of samples

Model type	Simple Samples	Medium sample	Complex Samples	Overall average
BERT-base	12.4	12.4	12.4	12.4
Compressed student model (no early departures)	8.7	8.7	8.7	8.7
Compressed student model + dynamic early departure	3.2	6.5	9.8	7.1

ence, and semantic similarity calculation. Using the BERT-base teacher model as a benchmark, the performance and efficiency metrics of the distillation compression-only model and the distillation compression plus dynamic early termination model (with different threshold settings) were compared.

In terms of performance, the distillation compression model alone achieved an average relative performance of 94.8% of the teacher model across all tasks. The distillation compression plus dynamic early termination model achieved an average relative performance of 94.5% at a threshold of 0.85, a decrease of only 0.3 percentage points, indicating that the accuracy loss introduced by the early termination mechanism was minimal. At a threshold of 0.90, the average relative performance reached 94.7%, approaching the level of no early termination. In terms of efficiency, compared to the teacher model, the distillation compression plus dynamic early termination model reduced the number of parameters by 70.9% and the average inference latency by 42.7%, resulting in a significant overall efficiency improvement. Table 3 shows a comparison of the overall performance and efficiency of different models across multiple tasks.

TABLE 3. Comparison of overall performance and efficiency of different models across multiple tasks

Model Configuration	Relative performance (%)	Number of parameters (M)	Average latency (ms)	Latency reduction (vs. BERT-base) (%)
BERT-base	100	110	12.4	-
Distillation compression model	94.8	32	8.7	29.8
Distillation + Early Exit (Threshold 0.75)	92.1	32	5.8	53.2
Distillation + Early Exit (Threshold 0.80)	93.3	32	6.4	48.4
Distillation + Early Exit (Threshold 0.85)	94.5	32	7.1	42.7
Distillation + Early Exit (Threshold 0.90)	94.7	32	8.2	33.9

As shown in Table 3, the distillation-compression model itself has already achieved a significant reduction in the number of parameters and a certain degree of acceleration. Introducing the dynamic early termination mechanism further significantly reduces the average inference latency while maintaining basic performance stability. The threshold setting reflects the trade-off between accuracy and efficiency; in practical applications, an appropriate configuration can be selected based on task requirements.

V. CONCLUSION

This study addresses the deployment challenges of the BERT model in resource-constrained scenarios by proposing a compression and acceleration method that integrates knowledge distillation and dynamic routing. For knowledge distillation, a multi-level transfer strategy is designed to simultaneously transfer the output distribution, intermediate layer features, and attention matrix of the teacher model to the lightweight student model, effectively preserving semantic representation capabilities. For dynamic routing, an early-retreat mechanism based on a confidence threshold is constructed, adaptively adjusting the computation depth according to the complexity of the input samples to achieve on-demand computation of the inference process. The effectiveness of this method in model compression and acceleration is verified.

Future work can be carried out in the following aspects: First, explore more refined early termination strategies, such as introducing more complex confidence estimation methods to further improve the acceleration effect of simple samples; second, extend the proposed framework to more Transformer variants to verify its universality on different model structures; third, combine quantization and pruning techniques to achieve greater compression and acceleration, and provide a more comprehensive solution for the efficient deployment of BERT models in mobile and embedded devices.

AUTHOR CONTRIBUTIONS

Conceptualization–Li Zhang and Yanqin Cao; methodology–Li Zhang and Yanqin Cao; investigation–Li Zhang and Yanqin Cao; writing-original draft preparation–Li Zhang and Yanqin Cao. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research was supported by,

Project Title: Development and Teaching Practice of Artificial Intelligence General Education Course Resources in Higher Vocational Colleges Empowered by AIGC: A Case Study of the 'Fundamentals and Applications of Artificial Intelligence' Course Project Number: KT2510096

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] M. Paganelli, F. Del Buono, A. Baraldi, et al., "Analyzing how BERT performs entity matching, " Proceedings of the VLDB Endowment, vol. 15, no. 8, pp. 1726-1738, 2022, doi: 10.14778/3529337.3529356.

[2] S. Kondra, V. Raghavan, and V. Kumar Adari, "Beyond Text: Exploring Multimodal BERT Models, " International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), vol. 8, no. 1, pp. 11764-11769, 2025, doi: 10.15662/IJRPETM.2025.0801003.

[3] M. M. Danyal, S. S. Khan, M. Khan, et al., "Proposing sentiment analysis model based on BERT and XLNet for movie reviews, " Multimedia Tools and Applications, vol. 83, no. 24, pp. 64315-64339, 2024, doi: 10.1007/s11042-024-18156-5.

[4] I. Karabila, N. Darraz, A. El-Ansari, et al., "BERT-enhanced sentiment analysis for personalized e-commerce recommendations, " Multimedia Tools and Applications, vol. 83, no. 19, pp. 56463-56488, 2024, doi: 10.1007/s11042-023-17689-5.

[5] C. Özkurt, "Comparative analysis of state-of-the-art Q&A models: BERT, RoBERTa, DistilBERT, and ALBERT on squad v2 dataset, " Chaos and Fractals, vol. 1, no. 1, pp. 19-30, 2024, doi: 10.69882/adba.chf.2024073.

[6] A. C. Mazari, N. Boudoukhani, and A. Djeffal, "BERT-based ensemble learning for multi-aspect hate speech detection, " Cluster Computing, vol. 27, no. 1, pp. 325-339, 2024, doi: 10.1007/s10586-022-03956-x.

[7] A. Deshmukh and A. Raut, "Applying bert-based nlp for automated resume screening and candidate ranking, " Annals of Data Science, vol. 12, no. 2, pp. 591-603, 2025, doi: 10.1007/s40745-024-00524-5.

[8] M. Mujahid, K. Kanwal, F. Rustam, et al., "Arabic ChatGPT tweets classification using RoBERTa and BERT ensemble model, " ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 22, no. 8, pp. 1-23, 2023, doi: 10.1145/3605889.

[9] M. Z. Zaki, "Revolutionising translation technology: A comparative study of variant transformer models–BERT, GPT and T5, " Computer Science and Engineering–An International Journal, vol. 14, no. 3, pp. 15-27, 2024, doi: 10.5121/cseij.2024.14302.

[10] M. N. Uddin, B. Li, Z. Ali, et al., "Software defect prediction employing BiLSTM and BERT-based semantic feature, " Soft Computing, vol. 26, no. 16, pp. 7877-7891, 2022, doi: 10.1007/s00500-022-06830-5.

[11] M. Gupta and P. Agrawal, "Compression of deep learning models for text: A survey, " ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 16, no. 4, pp. 1-55, 2022, doi: 10.1145/3487045.

[12] E. C. Garrido-Merchan, R. Gozalo-Brizuela, and S. Gonzalez-Carvajal, "Comparing BERT against traditional machine learning models in text classification, " Journal of Computational and Cognitive Engineering, vol. 2, no. 4, pp. 352-356, 2023, doi: 10.47852/bonviewJCCE3202838.

[13] L. George and P. Sumathy, "An integrated clustering and BERT framework for improved topic modeling, " International Journal of Information Technology, vol. 15, no. 4, pp. 2187-2195, 2023, doi: 10.1007/s41870-023-01268-w.

[14] M. Bilal and A. A. Almazroi, "Effectiveness of Fine-tuned BERT Model in Classification of Helpful and Unhelpful Online Customer Reviews: M. Bilal, AA Almazroi, " Electronic Commerce Research, vol. 23, no. 4, pp. 2737-2757, 2023, doi: 10.1007/s10660-022-09560-w.

[15] M. Laurer, W. Van Atteveldt, A. Casas, et al., "Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli, " Political Analysis, vol. 32, no. 1, pp. 84-100, 2024, doi: 10.1017/pan.2023.20.