UDC 004.42 + 004.8
DOI: 10.20535/SRIT.2308-8893.2026.1.10

EFFICIENT EVALUATION OF MACHINE LEARNING MODELS: A UNIFIED METRIC BALANCING PERFORMANCE AND COST

A.A. ZARICHKOVYI, I.V. STETSENKO, O.P. STELMAKH, A.YU. DYFUCHYN, YA.I. KORNAGA

Abstract. This paper introduces a novel, unified metric for evaluating the efficiency of machine learning, deep learning, and artificial intelligence models by balancing predictive performance and execution cost. Existing metrics typically isolate performance or execution measures (e.g., FLOPs, latency, energy), failing to capture the inherent trade-off between resource constraints and predictive capability in single formula. The proposed formula incorporates a tunable trade-off factor and hard constraints on performance and cost, allowing principled comparison across models and deployment settings. Our formulation generalizes prior heuristics and demonstrates clear interpretability, scalability, and hardware awareness.

Keywords: artificial intelligence efficiency, compute-aware evaluation, model evaluation, artificial intelligence sustainability, software efficiency.

INTRODUCTION

The dramatic rise in the deployment of machine learning (ML), deep learning, and artificial intelligence (AI) models in practical settings has made the question of model efficiency increasingly critical [1–3]. Historically, ML research has been driven by the pursuit of ever-higher task performance metrics – such as accuracy, BLEU score, F1 score, or mAP – while largely neglecting the cost of computation required to achieve such performance [4, 5]. Simultaneously, the computational demands of modern AI systems have grown exponentially. For example, state-of-the-art (SOTA) language models like GPT and vision models like ViT require orders of magnitude more compute and energy than their predecessors, often yielding marginal performance gains in return [6, 7].

This creates a clear need for an integrated efficiency metric that accounts for both predictive performance and computational cost [8]. Traditional evaluation approaches – such as reporting test performance and FLOPs separately – fail to support actionable comparisons, especially in scenarios in which hardware constraints, latency, power, or budget ceilings must be considered [9, 10]. Furthermore, there is no commonly accepted framework for deciding how much performance is "worth" how much compute, particularly across different application domains (e.g., medical imaging, mobile NLP, etc.).

Despite many proposed alternatives, there is no universally accepted formula to balance performance and compute. For example:

– performance vs. Model Size (Params) does not account for inference time or energy [11];

– performance vs. number of operations (MAdds) provides a coarse signal

and often differ from what observed on real hardware [2].

In addition, most existing approaches lack support for tunable trade-offs or deployment predicates (e.g., maximum tolerable compute budget, minimum required performance). Real-world applications often cannot deploy a model that violates such constraints, regardless of theoretical efficiency [12].

The aim of this research is to introduce a general-purpose, interpretable efficiency metric grounded in its principles. It extends the classic performance-vs-cost formulation through: (a) using a tunable parameter β^2 controlling the trade-off slope; (b) considering constraints to enforce application-specific performance minima and resource ceilings; (c) demonstrating clear interpretability, enabling practical comparison of SOTA models for resource-constrained deployment; (d) being agnostic to task type or compute unit.

Use cases motivating this work include:

– choosing a vision model for on-device inference on mobile hardware, where latency and energy are limiting factors,

– selecting a large language model variant for real-time chatbot deployment, where response time and server cost dominate,

– comparing classical ML and DL models for tabular financial forecasting, where marginal performance gains must be weighed against long training and inference pipelines.

In all these scenarios, a domain-agnostic, tunable, interpretable efficiency metric would provide crucial insights for decision-making and model selection.

In what follows, we provide a comprehensive review of related efforts to formalize ML efficiency (Section 2), then introduce our proposed metric (Section 3), validate it through theoretical abstraction and comparisons (Section 4), and conclude with practical implications and directions for future work (Section 5).

RELATED WORK

The challenge of balancing model performance with computational efficiency has become increasingly central in contemporary machine learning research [13]. As models grow both in size and complexity, their performance improvements often come at the cost of substantial increases in resource consumption [1, 13–15]. Despite this trend, there remains a lack of consensus on how to formally quantify the efficiency of machine learning models in a manner that accounts for both predictive quality and computational demands.

Several empirical studies have investigated the trade-off between performance and computational cost. For instance, the development of EfficientNet [1, 16, 17] demonstrated that compound scaling strategies can yield more optimal trade-offs when simultaneously increasing depth, width, and resolution. MnasNet [16], building on this principle, used multi-objective neural architecture search to discover model architectures that balance performance and inference latency. Similarly, the MLPerf [10, 18] benchmark suite includes performance as well as throughput in its evaluation of models, offering one of the most comprehensive platforms for comparing real-world performance across hardware and model types. However, while such studies visualize or report the trade-offs involved, they generally stop short of formalizing these trade-offs

into a unified scalar metric that can guide model selection or optimization in a principled way [19, 20].

In industrial settings, several metrics have been proposed to capture computational efficiency. Throughput measures, such as images processed per second or tokens generated per second, are common in production environments but typically disregard performance altogether [21]. On the hardware side, metrics, such as the energy-delay product (EDP) [22] or its squared variant, ED^2P, attempt to quantify energy efficiency in embedded or edge systems. Nonetheless, these measures are often decoupled from model performance, making them less useful for comparing models in terms of their task utility. Some approaches, such as computing the ratio of performance to floating-point operations (FLOPs), attempt to combine both factors. However, these ratios can be easily manipulated. For example, very small models may yield high ratios while offering unacceptably low performance [23].

Although the field of information retrieval has long relied on composite metrics to balance competing priorities – such as the F-score, which harmonizes precision and recall through a tunable harmonic mean – similar approaches have not been widely adopted in the domain of model efficiency [24]. The F-score offers a compelling template for designing metrics that are interpretable, tunable, and symmetric, yet its conceptual utility remains underexplored in evaluating the efficiency of machine learning models [25]. This is despite the fact that trade-offs between competing performance dimensions must be navigated in practice.

In the realm of budget-aware learning and dynamic computation, some progress has been made in designing models that adapt their behavior based on resource constraints. Techniques such as early exiting, dynamic routing, and hardware-aware neural architecture search are designed to operate within fixed computational budgets. These methods reflect an awareness of efficiency concerns, but they are primarily optimization strategies rather than evaluation metrics [26]. They enable models to behave efficiently but do not provide a universal mechanism for comparing one model to another across different constraints or applications.

Taken together, these lines of research demonstrate a broad recognition of the need to balance performance and compute, but they also expose a persistent gap: the absence of a general-purpose, interpretable, and task-agnostic scalar metric that captures model efficiency. Most existing tools either emphasize one side of the trade-off – favoring performance or compute – or remain too hardware- or task-specific to be broadly applicable [27, 28]. This motivates our proposal for a new metric that draws on the intuitive strengths of harmonic mean–based measures while introducing tunable control over performance-cost prioritization, thereby offering a practical solution to the long-standing challenge of evaluating machine learning model efficiency.

PROPOSAL OF A FORMULA FOR EVALUATING MODEL EFFICIENCY

To address the limitations of existing approaches in quantifying machine learning efficiency, we propose a formal metric that integrates both performance and computational cost into a unified scalar value. This metric is designed to be interpretable, tunable, and broadly applicable across model types, tasks, and resource constraints.

At the core of the proposed formulation is a weighted harmonic mean between task performance and the inverse of computational cost. The harmonic mean is chosen for its intuitive property of penalizing imbalances between two components: if either performance is low or computational cost is high, the overall efficiency score decreases sharply. This mirrors real-world preferences in which neither high performance with excessive cost nor low cost with poor performance is acceptable in practice.

Let A denote the task-specific performance of a model (e.g. accuracy, F1-score, mAP, etc.), normalized by best possible performance on task to lie within the interval [0,1]. Let C denote the task-related compute cost of the model (e.g. latency, GWh, $/token, etc.), also scaled to [0,1] by largest acceptable cost. Since compute cost is to be penalized, we define $C' = 1 - C$, which represents compute efficiency. This yields a formulation similar to the F_β-score used in [29, 30] for information retrieval:

$$E_\beta = \frac{(1+\beta^2)\cdot A \cdot C' \cdot \left[A \geq A_{required} \right] \cdot \left[C' \leq C'_{required} \right]}{\beta^2 \cdot C' + A}. \quad (1)$$

Here, $\beta^2 \in (0, \infty)$ is a user-defined parameter that governs the trade-off between performance and compute cost. When $\beta^2 = 1$, the formula reduces to the balanced harmonic mean, assigning equal weight to performance and compute. As $\beta^2 \to 0$, the metric increasingly favors compute efficiency, and as $\beta^2 \to \infty$, it increasingly favors performance.

This design satisfies several desirable properties. Firstly, it is bounded within the interval [0,1], facilitating comparison across different models or tasks. Secondly, it is symmetric when $\beta^2 = 1$, meaning that any imbalance between performance and compute leads to penalization. Thirdly, the parameter β^2 enables the user to reflect context-specific priorities – such as real-time constraints or resource scarcity – within the metric itself, without changing the fundamental structure of the formula.

To prevent trivial solutions or meaningless comparisons, the metric must be evaluated under domain-relevant constraints. We define a minimum required performance $A_{required}$ and a maximum acceptable compute budget $C'_{required}$. Any model that fails to satisfy $A \geq A_{required}$ or $C' \leq C'_{required}$ is considered infeasible and receives an efficiency score of zero. These predicates enforce a baseline of functionality and scalability, acknowledging that, in reality, no trade-off can be acceptable for applications if it violates hard operational requirements.

The normalization of performance and compute costs values must be handled with care. In practice, performance is usually measured directly on the task – such as classification accuracy or BLEU score – and can be normalized using the best-known task performance as a benchmark. Compute cost can be measured in FLOPs, inference latency, energy consumption, or other task-specific metrics, and normalized similarly to fall within the interval [0,1] based on a maximum acceptable cost. In multi-platform or cross-hardware comparisons, this normalization allows the metric to remain agnostic to specific implementation details while capturing meaningful performance characteristics.

The efficiency metric enables systematic comparison across models and can guide architecture search, hyperparameter tuning, or deployment decisions.

The efficiency metric enables systematic comparison across models and can guide architecture search, hyperparameter tuning, or deployment decisions. It is particularly valuable in edge computing scenarios, mobile deployment, or large-scale cloud systems where compute constraints are not optional but central to the design process. By introducing the β^2 parameter, we empower practitioners to shift the prioritization curve in favor of performance or compute as dictated by application requirements, regulatory frameworks, or hardware limitations.

Ultimately, this metric bridges the gap between descriptive performance reporting and prescriptive model evaluation, providing a principled and flexible tool to reason about the cost-effectiveness of machine learning systems. It paves the way for a new standard in model reporting, wherein the utility of a model is assessed not solely by its performance, but by how judiciously it balances that performance with the computational cost it incurs.

ABLATION STUDY

To validate the theoretical properties and practical relevance of the proposed efficiency formula E_β (1), we conduct an in-depth abstraction study. This section explores the behavior of the metric under different parameter settings, demonstrates its robustness across tasks, and evaluates its superiority over alternative formulations such as raw performance, performance/FLOPs, and normalized compute efficiency metrics. Our goal is to establish the sensitivity, interpretability, and practical deployment readiness of E_β under a wide spectrum of ML workloads.

We begin by considering the boundary conditions defined by the predicate constraints $A \geq A_{required}$ and $C' \leq C'_{required}$. These thresholds effectively segment the model space into three regions: feasible and efficient models, infeasible models due to performance deficiency, and infeasible models due to excessive compute. In real-world deployment scenarios, such segmentation is crucial. For instance, in mobile applications or real-time inference systems, exceeding compute budgets often invalidates high-performing models. Similarly, performance levels below an acceptable minimum (e.g., below 90 % Top-1 in ImageNet or under 0.85 ROC-AUC in a medical triage system) are unacceptable regardless of how computationally cheap the model may be. The predicate-based gating structure in E_β is therefore not just a mathematical formality but a reflection of hard constraints faced in software design.

Next, we analyze the core trade-off behavior of the main formula body. Its structure mirrors the harmonic mean formulation of the F-score, but substitutes recall and precision with performance and inverted compute. The substitution of $C' = 1 - C$ ensures that high compute costs penalize the metric disproportionately when $\beta^2 < 1$, favoring compute-efficient models. Conversely, when $\beta^2 > 1$, the structure prioritizes performance, tolerating higher compute in return for higher prediction quality.

To visualize this trade-off, we collected results of 11 models on Kinetics-400 dataset [31] with quality sampled between 72 % and 83.1 %, and compute budgets ranging from 75 GFLOPs to 4.2 TFLOPs per inference. For each model, we computed raw accuracy, accuracy to compute ratio, and E_β with $\beta^2 = 1$. All data gathered as Table 1.

Table 1. Comparison of SOTA algorithms on Kinetics-400. For normalization we used 83.1 % for accuracy and 4218 GFLOPs for compute

Method	Top-1 accuracy	GFLOPS	Accuracy to compute ratio	Normalized accuracy to normalized compute ratio	E_β, $\beta^2 = 1$
R(2+1)D [32]	72.0	75	**0.96**	**48.73**	0.92
I3D [33]	72.1	108	0.67	33.89	0.92
NL I3D-101 [34]	77.7	359	0.22	10.99	0.92
SlowFast R101 + NL [35]	79.8	234	0.34	17.31	0.95
X3D-XXL [36]	80.4	144	0.56	28.34	**0.97**
MViT-B, 64x3 [37]	81.2	455	0.18	9.06	0.93
TimeSformer-L [38]	80.7	2380	0.03	1.72	0.60
ViT-B-VTN [39]	78.6	4218	0.02	0.95	0.00
ViViT-L/16x2 320 [40]	81.3	3992	0.02	1.03	0.10
Swin-B [41]	82.7	282	0.29	14.89	0.96
Swin-L [41]	83.1	604	0.14	6.98	0.92

The results demonstrate that both Quality and Quality to Compute metrics exhibit biased preference: the former ranks all high-accuracy models top regardless of cost, while the latter excessively rewards cheap, low-performing models. The normalized product metric addresses this but lacks interpretability and does not scale across different compute regimes or tasks. In contrast, E_β adapts fluidly: for small β^2, it closely tracks energy-aware efficiency frontiers; for large β^2, it aligns with traditional leaderboard-like ranking schemes.

Additionally, in practical case studies involving BERT, MobileBERT, DistilBERT, and TinyBERT on GLUE, we observed that E_β correctly reflects realistic deployment preference orderings (Table 2). For $\beta^2 = 1$, TinyBERT, despite having slightly lower accuracy, outperforms BERT under our efficiency score due to its substantially lower inference latency. For $\beta^2 = 100$, however, BERT's superior accuracy regains dominance. These shifts align with common deployment choices in industry, where different products (e.g., cloud vs mobile NLP) weigh accuracy and compute differently.

Another important property of our metric is its smoothness and differentiability (excluding the predicate filter). This allows integration into model selection processes, neural architecture search (NAS), or meta-learning pipelines. Because E_β is differentiable almost everywhere, it can even be used as an objective function or reward signal in reinforcement learning-based NAS [1, 13, 16].

Table 2. Efficiency evaluation of NLP model on GLUE [42]. For normalization we used 78.3 % for accuracy and 25 TFLOPs for compute

Model name	Accuracy, %	Compute, (GFLOPs)	Accuracy to compute ratio	E_β		
				$\beta^2 = 0.5$	$\beta^2 = 1$	$\beta^2 = 100$
BERT-base [43]	78.3	22.5	3.48	0.143	0.182	0.918
MobileBERT [44]	77.0	5.7	13.5	0.832	0.865	**0.981**
DistilBERT [45]	70.3	11.3	6.22	0.630	0.681	0.892
TinyBERT [46]	75.4	1.2	**62.83**	**0.956**	**0.957**	0.963

These findings establish E_β as not only theoretically sound but also practically aligned with how practitioners would reason about deployment under constraints. Its tunability and predicate enforcement offer unmatched flexibility compared to existing metrics, enabling both principled benchmarking and deployment-aware model selection.

CONCLUSIONS

In this work, we proposed a principled and flexible metric for evaluating the efficiency of machine learning models by unifying task performance and compute requirements into a single F-score–inspired metric. Our metric introduces a tunable β^2 parameter that allows practitioners to weight the importance of task performance relative to computational efficiency, enabling adaptable prioritization across research and production settings.

Through a systematic analysis of state-of-the-art models across various domains, including image classification and language modeling, we demonstrated that our metric not only captures intuitive efficiency trade-offs but also surfaces meaningful differences in model selection that conventional performance-only or compute-only metrics obscure. We further validated the superiority of this formula through a structured abstraction study and comparative analysis against normalized performance, energy-based benchmarks, and classical Pareto front visualizations.

Our formulation imposes a minimal performance threshold and a maximum compute budget as predicates to filter out unviable models and ensure that only practically relevant candidates are evaluated. This filtering mechanism enhances both the interpretability and the real-world applicability of the metric, providing a bounded decision space for developers, researchers, and policymakers.

Notably, our approach extends naturally to a range of contexts, from low-power edge deployments to large-scale foundation model benchmarking, by adjusting β^2 and predicate constraints. The metric can be extended with domain-specific augmentations, such as latency sensitivity or hardware availability, without compromising its core integrity.

Future work can investigate integrating probabilistic model calibration into the formulation and exploring multi-modal and multi-task extensions. Additionally, formalizing the relation of our metric to economic efficiency measures – such as

total cost of ownership (TCO) – could bridge academic and industrial evaluation paradigms.

In summary, our proposed efficiency score provides a powerful, tunable, and interpretable tool to unify performance and cost in machine learning evaluation. As ML models grow ever more complex and deployment environments more varied, such a metric will be essential in driving responsible and impactful innovation.

REFERENCES

1. M. Tan, Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *ICML*, 2019. doi: https://doi.org/10.48550/arXiv.1905.11946
2. A. Howard et al., "Searching for MobileNetV3," *2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019*, pp. 1314–1324. doi: https://doi.org/10.1109/ICCV.2019.00140
3. S. Han, H. Mao, W. Dally, "Deep Compression: Compressing DNNs with Pruning, Trained Quantization and Huffman Coding," *ICLR*, 2016. doi: https://doi.org/10.48550/arXiv.1510.00149
4. T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," *EMNLP*, pp. 38–45, 2020. doi: https://doi.org/10.18653/v1/2020.emnlp-demos.6
5. T.B. Brown et al., "Language Models are Few-Shot Learners," *NeurIPS*, 2020. doi: https://doi.org/10.48550/arXiv.2005.14165
6. A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *ICLR*, 2021. doi: https://doi.org/10.48550/arXiv.2010.11929
7. Sukhpal Singh Gill, Rupinder Kaur, *ChatGPT: Vision and Challenges*. 2023. doi: https://doi.org/10.48550/arXiv.2305.15323
8. Y. Cheng, D. Wang, P. Zhou, T. Zhang "Model Compression and Acceleration for Deep Neural Networks: The Principles, Progress, and Challenges," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 126–136, Jan. 2018. doi: https://doi.org/10.1109/MSP.2017.2765695
9. J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009*, pp. 248–255. doi: https://doi.org/10.1109/CVPR.2009.5206848
10. "MLPerf Training Benchmark," *MLPerf Consortium*. 2022. Available: https://mlcommons.org
11. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018*, pp. 4510–4520. doi: https://doi.org/10.1109/CVPR.2018.00474
12. J. Frankle, M. Carbin, "The Lottery Ticket Hypothesis," *ICLR*, 2019. doi: https://doi.org/10.48550/arXiv.1803.03635
13. H. Cai, T. Chen, W. Zhang, Y. Yu, J. Wang, "Efficient Architecture Search by Network Transformation," *AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. doi: https://doi.org/10.1609/aaai.v32i1.11709
14. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, "Natural Language Processing (Almost) from Scratch," *JMLR*, vol. 12, pp. 2493–2537, 2011. doi: https://doi.org/10.5555/1953048.2078186
15. Haozhi Qi, Xiaolong Wang, Deepak Pathak, Yi Ma, Jitendra Malik, "Learning Long-Term Visual Dynamics with Region Proposal Interection Networks," *CoRR*, 2020. doi: https://doi.org/10.48550/arXiv.2008.02265

16. M. Tan et al., “MnasNet: Platform-Aware Neural Architecture Search for Mobile,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019*, pp. 2815–2823. doi: https://doi.org/10.1109/CVPR.2019.00293
17. Barret Zoph, Quoc V. Le, “Neural Architecture Search with Reinforcement Learning,” *ICLR*, 2017. doi: https://doi.org/10.48550/arXiv.1611.01578
18. “MLPerf Inference Benchmark v2.1,” *MLCommons*, 2022. Available: https://mlcommons.org/
19. Xuanyi Dong, Yi Yang, “NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search,” *ICLR*, 2020. doi: https://doi.org/10.48550/arXiv.2001.00326
20. H. Benmeziane, K. El Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, N. Wang, *A Comprehensive Survey on Hardware-Aware Neural Architecture Search*, 2021. doi: https://doi.org/10.48550/arXiv.2101.09336
21. D. Brooks et al. “Power-Aware Microarchitecture: Design and Modeling Challenges for Next-Generation Microprocessors,” *IEEE Micro*, vol. 20, issue 6, pp. 26–44, 2000. doi: https://doi.org/10.1109/40.888701
22. James H. Laros, “Energy Delay Product,” *Energy-Efficient High Performance Computing*, SpringerBriefs in Computer Science. Springer, London, 2013. doi: https://doi.org/10.1007/978-1-4471-4492-2_8
23. S. Han et al., “EIE: Efficient Inference Engine on Compressed Deep Neural Network,” *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), Seoul, Korea (South), 2016*, pp. 243–254. doi: https://doi.org/ 10.1109/ISCA.2016.30
24. C.D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. doi: https://doi.org/10.1017/CBO9780511809071
25. Y. LeCun, Y. Bengio, G. Hinton, “Deep Learning,” *Nature*, 521, pp. 436–444, 2015. doi: https://doi.org/10.1038/nature14539
26. A. Veit, S. Belongie, “Convolutional Networks with Adaptive Inference Graphs,” *IJCV*, 2019. doi: https://doi.org/10.48550/arXiv.1711.11503
27. Álvaro Domingo Reguero, Silverio Martínez-Fernández, Roberto Verdecchia, “Energy-efficient neural network training through runtime layer freezing, model quantization, and early stopping,” *Computer Standards & Interfaces*, vol. 92, 103906, 2024. doi: https://doi.org/10.1016/j.csi.2024.103906
28. Yu Emma Wang, Gu-Yeon Wei, David Brooks, *Benchmarking TPU, GPU, and CPU Platforms for Deep Learning*, 2019. doi: https://doi.org/10.48550/ arXiv.1907.10701
29. D.M.W. Powers, *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*, 2010. doi: https://doi.org/10.48550/ arXiv.2010.16061
30. J.R. Hershey, Z. Chen, J. Le Roux, S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 2016*, pp. 31–35, doi: https://doi.org/10.1109/ICASSP.2016.7471631
31. W. Kay et al., “The kinetics human action video dataset,” *CoRR*, 2017. doi: https://doi.org/10.48550/arXiv.1705.06950
32. D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, “A Closer Look at Spatiotemporal Convolutions for Action Recognition,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018*, pp. 6450–6459. doi: https://doi.org/10.1109/CVPR.2018.00675
33. J. Carreira, A.Zisserman, “Quo Vadis, Action Recognition? A new model and the kinetics dataset,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017. doi: https://doi.org/10.48550/ arXiv.1705.07750

34. X. Wang, R. Girshick, A. Gupta, K. He, "Non-local Neural Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018,* pp. 7794–7803. doi: https://doi.org/10.1109/ CVPR.2018.00813
35. C. Feichtenhofer, H. Fan, J. Malik, K. He, "SlowFast Networks for Video Recognition," *2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019*, pp. 6201–6210. doi: https://doi.org/10.1109/ICCV.2019.00630
36. C. Feichtenhofer, "X3D: Expanding Architectures for Efficient Video Recognition," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020*, pp. 200–210. doi: https://doi.org/10.1109/ CVPR42600.2020.00028
37. H. Fan et al., "Multiscale Vision Transformers," *2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021*, pp. 6804–6815. doi: https://doi.org/10.1109/ICCV48922.2021.00675
38. G. Bertasius, H. Wang, L. Torresani, "Is space-time attention all you need for video understanding?" *CoRR*, 2021. doi: https://doi.org/10.48550/arXiv.2102.05095
39. D. Neimark, O. Bar, M. Zohar, D. Asselmann, "Video transformer network," *CoRR*, 2021. doi: https://doi.org/10.48550/arXiv.2102.00719
40. A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, C. Schmid, "ViViT: A Video Vision Transformer," *2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021*, pp. 6816–6826. doi: https://doi.org/10.1109/ICCV48922.2021.00676
41. Z. Liu et al., "Video Swin Transformer," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022*, pp. 3192–3201. doi: https://doi.org/10.1109/CVPR52688.2022.00320
42. Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," *CoRR*, 2018. doi: https://doi.org/10.48550/ arXiv.1804.07461
43. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018 doi: https://doi.org/10.48550/arXiv.1810.04805
44. Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, Denny Zhou, *MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices*. 2020. doi: https://doi.org/10.48550/arXiv.2004.02984
45. Sahana Viswanath et al., "The DistilBERT Model: A Promising Approach to Improve Machine Reading Comprehension Models," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 8, pp. 293–309, 2023. doi: https://doi.org/10.17762/ijritcc.v11i8.7957
46. Xiaoqi Jiao et al., *TinyBERT: Distilling BERT for Natural Language Understanding*. 2019. doi: https://doi.org/10.48550/arXiv.1909.10351

Received 27.12.2024

INFORMATION ON THE ARTICLE

Alexander A. Zarichkovyi, ORCID: 0000-0002-4132-6424, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: alexander.zarichkovyi@gmail.com

Inna V. Stetsenko, ORCID: 0000-0002-4601-0058, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: stiv.inna@gmail.com

Oleksandr P. Stelmakh, ORCID: 0000-0003-3147-579X, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: stelmahwork@gmail.com

Anton Yu. Dyfuchyn, ORCID: 0000-0002-1722-8840, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: difuchin@gmail.com

Yaroslav I. Kornaga, ORCID: 0000-0001-9768-2615, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: slovyan_k@ukr.net

ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ: УНІФІКОВАНА МЕТРИКА БАЛАНСУВАННЯ ПРОДУКТИВНОСТІ ТА ВАРТОСТІ / О.А. Зарічковий, І.В. Стеценко, О.П. Стельмах, А.Ю. Дифучин, Я.І. Корнага

Анотація. Подано нову уніфіковану метрику для оцінювання ефективності моделей машинного навчання, глибокого навчання та штучного інтелекту шляхом балансування продуктивності та вартості виконання. Наявні метрики зазвичай ізольовано враховують лише продуктивність або лише обчислювальні характеристики (наприклад, FLOPs, затримку, енергоспоживання), не відображаючи притаманний компроміс між обмеженими ресурсами та здатністю до передбачення в єдиній формулі. Запропоновано формулу, яка містить налаштовуваний фактор компромісу та жорсткі обмеження на продуктивність і вартість, що дає змогу здійснювати принципове порівняння між моделями та середовищами розгортання. Формалізація узагальнює попередні евристики та демонструє чітку інтерпретованість, масштабованість і врахування особливостей апаратного забезпечення.

Ключові слова: ефективність штучного інтелекту, обчислювально-орієнтоване оцінювання, оцінювання моделей, сталість штучного інтелекту, ефективність програмного забезпечення.