Zero-Cost Benchmarks: Towards Lower Reliance on Spearman Rank Correlation

Anonymous[1]

[1]Anonymous Institution

Abstract Zero-cost proxies (ZCPs) have received increasing attention due to their potential for removing computational bottlenecks in Neural Architecture Search (NAS). Special attention has been given to the benchmarking of such proxies in the design process. So far, Spearman rank correlation has been used as a go-to similarity measure for these benchmarks. In this paper, we investigate the shortcomings of this abundantly used metric and find that, in opposition to the core goals of NAS, Spearman rank correlation wrongly estimates the performance of ZCPs when it comes to top-ranked architectures in the search space. We propose Rank-Biased Overlap (RBO) as an alternative measure to prevent overfitting of ZCP design in the future. Our RBO benchmarking reveals new insights on ZCPs that cannot be inferred from the Spearman benchmarking. The introduction of RBO as an additional criterion could help lower the reliance of the benchmarks on a single measure.

1 Introduction

While most neural network architectures are designed by human experts, Neural Architecture Search (NAS) proposes to instead find the most suitable architecture for a given task automatically. Various paradigms for NAS have emerged, each with their own limitations. In cases where the cost of training many architectures is not prohibitive, classic predictor-based methods inspired by early NAS works [24, 14] find better performing architectures [21, 7]. In practice, training candidate architectures from scratch is often too costly for common usage, hence the success of two-stage methods [18, 8, 3, 15] and more recently the rise of zero-cost approaches [16, 1], which aim to evaluate the performance of neural networks without training them.

Methods that fall into the zero-cost category are built around two key elements: a search algorithm to iteratively sample architectures from the search space, and a metric with which to evaluate the sampled architectures, also known as Zero-Cost Proxy (ZCP). Although both aspects are of similar importance, a stronger focus has been cast on the design of the ZCPs in recent literature. Specifically, to fully decouple ZCPs from the search algorithm, substantial efforts are made to evaluate them in isolation.

An example of a comprehensive benchmark for ZCPs in the zero-cost NAS context is NAS-Bench-Suite-Zero [10]. This benchmark is a collection of previous benchmarks from the NAS literature, namely NAS-Bench-101 [22], NAS-Bench-201 [5], NAS-Bench-301 [19] and TransNAS-Bench-101 [6]. Each benchmark corresponds to an architectural search space, where possible architectures in the search space have been exhaustively evaluated on several tasks, such as classic vision tasks [11, 4] or the Taskonomy task bank [23]. For each task, by comparing the architecture ranking given by the ZCP and the true architecture ranking, it is possible to assess all metrics against each other.

A common practice, in NB-Suite-Zero as well as other independent benchmarks conducted by ZCP works, is to evaluate the metrics with the Spearman rank correlation:

$$r_s = \frac{cov(rank(X), rank(Y))}{\sigma_{rank(X)}\sigma_{rank(Y)}} \tag{1}$$

Submitted to AutoML 2025 Non-Archival Track © 2025 the authors, released under CC BY 4.0

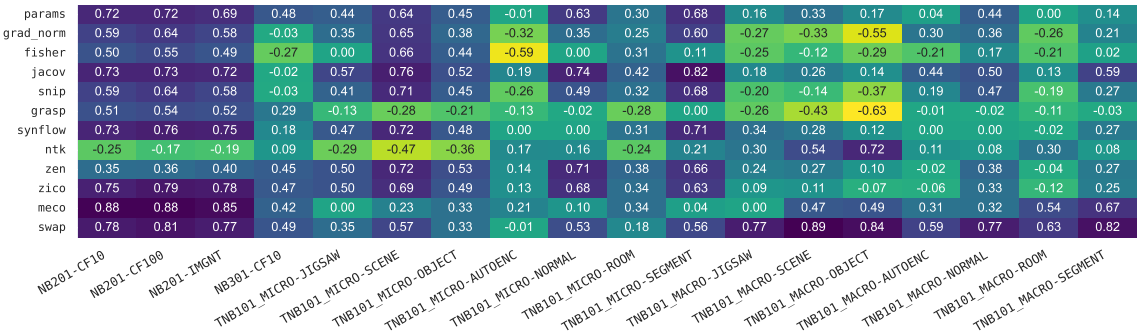

Figure 1: Spearman rank correlations of various ZCPs on the NB201, NB301 and TNB101 benchmarks. Spearman was computed over 3 seeds against the entire search space.

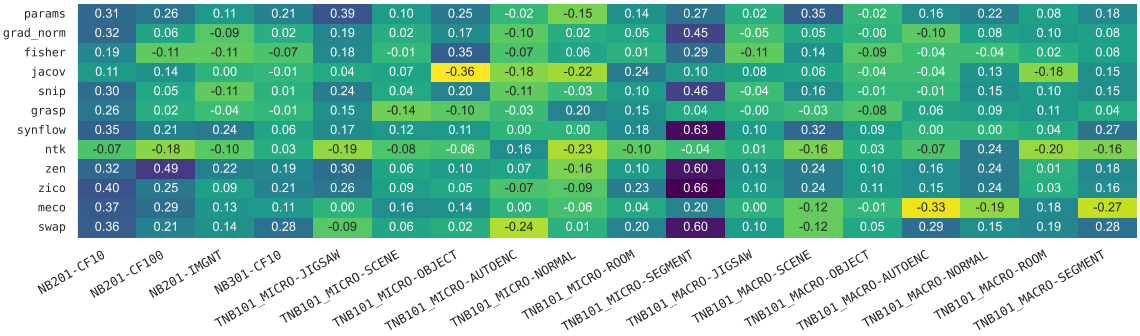

Figure 2: Spearman rank correlations of various ZCPs on the NB201, NB301 and TNB101 benchmarks. Spearman was computed on 3 seeds against the top 1% architectures in the space.

where X and Y are two rankable series of observations derived from the task at hand. Specifically, in the zero-cost context, X contains the ZCP scores for all architectures in the search space, while Y contains the "true" scores, given by direct evaluation on the task. The Spearman rank correlation is simple, intuitive and hyperparameter-free. This explains its success in the context of benchmarking various NAS methods, including but not limited to zero-cost NAS.

This paper exposes the limitations of using the Spearman rank correlation as the sole evaluation metric of ZCP benchmarks. Specifically, we highlight the discrepancy that exists when evaluating the entire search space or a restriction of the search space to the best architectures. In order to resolve this issue, we propose Rank-Biased Overlap (RBO) as an alternative and reproduce the NAS-Bench-Suite-Zero benchmark with this evaluation metric. Our findings suggest that both metrics are useful as they pinpoint different strengths of ZCPs.

2 Limitations of Spearman-based benchmarks

Since the Spearman rank correlation compares rankings at the scale of the entire search space, giving similar weight to all architectures, it is a metric that indicates global trends of the evaluated ZCP. Intuitively, the higher the Spearman rank correlation of a ZCP, the better it can separate bad architectures from good architectures. However, this does not hold at the local scale: the Spearman rank correlation over the whole space gives no indication of the ability of the ZCP to rank architectures with similar performance. No ZCP reaches a perfect Spearman rank correlation, which is acceptable as small differences in ranking for low-ranking architectures are of little interest. On the other hand, since the overarching goal of NAS is to find the best-suited architecture, small

fluctuations in ranking for top-ranking architectures are highly impactful. Therefore, the non-biased aspect of Spearman rank correlation causes it to be misaligned with the objectives of NAS.

In order to confirm this observation, we conduct benchmarking of current state-of-the-art ZCPs in the NAS-Bench-Suite-Zero benchmark[1]. First, we reenact the regular Spearman rank correlation benchmark using the following ZCPs: `params` (parameter count of the model), `grad_norm` [1], `fisher` [1], `jacov` [16, 1], `snip` [1], `grasp` [1], `synflow` [1], `ntk` [2], `zen` [13], `zico` [12], `meco` [9] and `swap` [17]. We report our results in Fig 1. For applicable methods, the observed Spearman rank correlation is similar to the correlation reported in their respective papers.

Secondly, for each task, we restrict the search space to only the top 1% of architectures on that task based on true performance. We report the results in Fig 2. For all ZCPs and on most tasks, we observe a significant drop in correlation, which indicates that the metrics are unable to rank top architectures in the correct order. Some SOTA metrics with excellent correlation over the entire space may exhibit no correlation or negative correlation when it comes to the top of the space.

This experiment displays the flaws of the Spearman-based paradigm in zero-cost benchmarking. While global Spearman correlation highlights the ZCPs' capabilities to trend towards good architectures, it does not indicate whether ZCPs are able to find the best architecture among the good ones. Furthermore, the over-reliance on Spearman rank correlation in the literature has led to the design of ZCPs that are especially suited for weeding out lower-end candidates, while little attention has been given to designing metrics for discerning between architectures of similar performance.

3 Rank-Biased Overlap: an alternative

Considering the drawbacks of Spearman rank correlation in the context of zero-cost benchmarking, we suggest that potential alternatives should be explored. Based on our earlier observations, an alternative evaluation metric to the Spearman rank correlation requires the following properties:

- the ability to quantify similarity between ranked lists;
- a direct relationship with the magnitude of disagreement for corresponding items;
- preferential treatment for better-ranked items.

The Rank-Biased Overlap (RBO) measure [20] satisfies all of the above. Consider the observation (ZCP) list X and the truth list Y, the RBO is defined as follows:

$$RBO(X, Y) = (1 - p) \sum_{d=1}^{N} p^{d-1} \mathcal{O}_d(X, Y) \tag{2}$$

where N is the size of the truth list Y, p is the exponential decay parameter, which controls the contribution of items at various depths of the lists. The overlap $\mathcal{O}_d(X, Y)$ is defined for depth d as:

$$\mathcal{O}_d(X, Y) = card(rank(X)_{:d} \cap rank(Y)_{:d}) \tag{3}$$

i.e the overlap is the number of items up to rank d that are present both in list X and list Y. In essence, RBO is the sum of the agreement of both lists with exponential decay starting from the top, meaning agreements in the top ranks are very impactful while disagreements for mid and low ranks have little impact.

Compared to Spearman rank correlation, RBO requires the tuning of hyperparameter p. However, remark that the contribution of the top d ranks is given by:

$$W_{:d}^{RBO} = 1 - p^{d-1} + \frac{1 - p}{p} * d * (ln(\frac{1}{1 - p}) - \sum_{i=1}^{d} \frac{p^i}{i}) \tag{4}$$

[1]NAS-Bench-101 was excluded due to the high cost of evaluation while being redundant with other benchmarks.

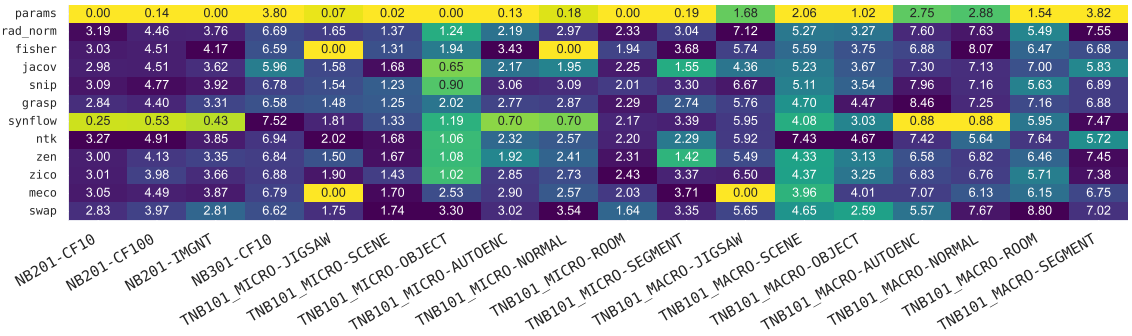

Figure 3: Normalized rank-biased overlap of various ZCPs on the NB201, NB301 and TNB101 benchmarks. RBO was computed on 3 seeds against the entire the space.

In order to tune p, we can simply select the desired contribution for the top $t\%$ architectures and tune accordingly, thereby decoupling the contribution from search space size N, the only constant that varies between search spaces.

Furthermore, in order to address the size difference between the search spaces which would cause the RBO to take vastly different ranges of values for each search space, we adopt normalized RBO:

$$RBO_{norm}(X,Y) = \frac{\sum_{d=1}^{N} p^{d-1}\mathcal{O}_d(X,Y)}{\sum_{d=1}^{N} p^{d-1}d} \tag{5}$$

We run our normalized RBO on the same benchmark tasks as Spearman rank correlation and report our results in Fig. 3. We observe that the results are very different from the Spearman-based benchmark. Similarly, the best metric varies based on the specific task. Some metrics which perform consistently well in the Spearman benchmark, such as `params` and `synflow`, experience collapse in the RBO-based benchmark. This could indicate that while they are able to distinguish low, mid and top-ranking architectures, they cannot properly order the rankings of the best architectures.

Conversely, a metric such as `ntk` performs poorly in the Spearman benchmark, but is among the top performers in the RBO benchmark. This could indicate that the metric orders the best architectures quite well, but may confuse low-ranking architectures for top-ranking ones, thereby hurting the overall correlation.

Remark that both the situations of performing poorly on the Spearman benchmark and the RBO benchmark are damaging. Indeed, low-scoring proxies in the RBO benchmark may be unable to identify the best architecture in the midst of the good ones, while low-scoring proxies in the Spearman benchmark may be unable to lead the search to a subspace containing good architectures in the first place. Therefore, we suggest that RBO may best be used as a companion to the traditional Spearman benchmark when designing new proxies.

4 Conclusion

We examine an alternative metric for benchmarking ZCPs based on the RBO similarity measure. This experiment depicts the metrics in a new light, revealing unknown properties. While ZCP design has largely overfitted to the Spearman benchmark, RBO benchmarking should not be considered a potential replacement as the insights are quite different. Rather, it is an additional measure for the ZCP design process, lowering its reliance on a single evaluation metric.

References

[1] M. S. Abdelfattah, A. Mehrotra, Ł. Dudziak, and N. D. Lane. Zero-cost proxies for lightweight nas. *arXiv preprint arXiv:2101.08134*, 2021.

[2] W. Chen, X. Gong, and Z. Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. *arXiv preprint arXiv:2102.11535*, 2021.

[3] X. Chu, B. Zhang, and R. Xu. Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. In *Proceedings of the IEEE/CVF International Conference on computer vision*, pages 12239–12248, 2021.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] X. Dong and Y. Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. *arXiv preprint arXiv:2001.00326*, 2020.

[6] Y. Duan, X. Chen, H. Xu, Z. Chen, X. Liang, T. Zhang, and Z. Li. Transnas-bench-101: Improving transferability and generalizability of cross-task neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5251–5260, 2021.

[7] L. Ericsson, M. Espinosa Minano, C. Yang, A. Antoniou, A. J. Storkey, S. Cohen, S. McDonagh, and E. J. Crowley. einspace: Searching for neural architectures from fundamental operations. *Advances in Neural Information Processing Systems*, 37:1919–1953, 2024.

[8] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun. Single path one-shot neural architecture search with uniform sampling. In *Computer vision–ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, part XVI 16*, pages 544–560. Springer, 2020.

[9] T. Jiang, H. Wang, and R. Bie. Meco: zero-shot nas with one data and single forward pass via minimum eigenvalue of correlation. *Advances in Neural Information Processing Systems*, 36:61020–61047, 2023.

[10] A. Krishnakumar, C. White, A. Zela, R. Tu, M. Safari, and F. Hutter. Nas-bench-suite-zero: Accelerating research on zero cost proxies. *Advances in Neural Information Processing Systems*, 35:28037–28051, 2022.

[11] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[12] G. Li, Y. Yang, K. Bhardwaj, and R. Marculescu. Zico: Zero-shot nas via inverse coefficient of variation on gradients. *arXiv preprint arXiv:2301.11300*, 2023.

[13] M. Lin, P. Wang, Z. Sun, H. Chen, X. Sun, Q. Qian, H. Li, and R. Jin. Zen-nas: A zero-shot nas for high-performance image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 347–356, 2021.

[14] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[15] H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[16] J. Mellor, J. Turner, A. Storkey, and E. J. Crowley. Neural architecture search without training. In *International conference on machine learning*, pages 7588–7598. PMLR, 2021.

[17] Y. Peng, A. Song, H. M. Fayek, V. Ciesielski, and X. Chang. Swap-nas: Sample-wise activation patterns for ultra-fast nas. *arXiv preprint arXiv:2403.04161*, 2024.

[18] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean. Efficient neural architecture search via parameters sharing. In *International conference on machine learning*, pages 4095–4104. PMLR, 2018.

[19] J. Siems, L. Zimmer, A. Zela, J. Lukasik, M. Keuper, and F. Hutter. Nas-bench-301 and the case for surrogate benchmarks for neural architecture search. *arXiv preprint arXiv:2008.09777*, 4:14, 2020.

[20] W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38, 2010.

[21] C. White, W. Neiswanger, and Y. Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10293–10301, 2021.

[22] C. Ying, A. Klein, E. Real, E. Christiansen, K. Murphy, and F. Hutter. NAS-Bench-101: Towards Reproducible Neural Architecture Search. 2 2019.

[23] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

[24] B. Zoph and Q. Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017.

Submission Checklist

1. For all authors...

   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

   (b) Did you describe the limitations of your work? [Yes]

   (c) Did you discuss any potential negative societal impacts of your work? [No]

   (d) Did you read the ethics review guidelines and ensure that your paper conforms to them? (see `https://2022.automl.cc/ethics-accessibility/`) [Yes]

2. If you ran experiments...

   (a) Did you use the same evaluation protocol for all methods being compared (e.g., same benchmarks, data (sub)sets, available resources, etc.)? [Yes]

   (b) Did you specify all the necessary details of your evaluation (e.g., data splits, pre-processing, search spaces, hyperparameter tuning details and results, etc.)? [Yes] Details that are not included in our paper can be found in the benchmarks' respective papers.

   (c) Did you repeat your experiments (e.g., across multiple random seeds or splits) to account for the impact of randomness in your methods or data? [Yes]

   (d) Did you report the uncertainty of your results (e.g., the standard error across random seeds or splits)? [No] We omitted standard errors due to readability concerns, but may provide it at an ulterior point.

   (e) Did you report the statistical significance of your results? [N/A]

   (f) Did you use enough repetitions, datasets, and/or benchmarks to support your claims? [Yes]

   (g) Did you compare performance over time and describe how you selected the maximum runtime? [N/A]

   (h) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]

   (i) Did you run ablation studies to assess the impact of different components of your approach? [No]

3. With respect to the code used to obtain your results...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results, including all dependencies (e.g., `requirements.txt` with explicit versions), random seeds, an instructive `README` with installation instructions, and execution commands (either in the supplemental material or as a URL)? [No] We will provide the code on a repository at an ulterior date to avoid breaching anonymity.

   (b) Did you include a minimal example to replicate results on a small subset of the experiments or on toy data? [N/A]

   (c) Did you ensure sufficient code quality and documentation so that someone else can execute and understand your code? [N/A]

   (d) Did you include the raw results of running your experiments with the given code, data, and instructions? [N/A]

(e) Did you include the code, additional data, and instructions needed to generate the figures and tables in your paper based on the raw results? [N/A]

4. If you used existing assets (e.g., code, data, models)…
   (a) Did you cite the creators of used assets? [N/A]
   (b) Did you discuss whether and how consent was obtained from people whose data you're using/curating if the license requires it? [N/A]
   (c) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you created/released new assets (e.g., code, data, models)…
   (a) Did you mention the license of the new assets (e.g., as part of your code submission)? [N/A]
   (b) Did you include the new assets either in the supplemental material or as a URL (to, e.g., GitHub or Hugging Face)? [N/A]
6. If you used crowdsourcing or conducted research with human subjects…
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to institutional review board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]
7. If you included theoretical results…
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]