AutoML for Climate Change: A Call to Action

Renbo Tu[1*], **Nicholas Roberts**[2], **Vishak Prasad**[3], **Sibasis Nayak**[3], **Paarth Jain**[3], **Frederic Sala**[2], **Ganesh Ramakrishnan**[3], **Ameet Talwalkar**[4], **Willie Neiswanger**[5], **Colin White**[6]

[1]University of Toronto, [2]University of Wisconsin, [3]IIT Bombay,
[4]Carnegie Mellon University, [5]Stanford University [6]Abacus.AI

Abstract

The challenge that climate change poses to humanity has spurred a rapidly developing field of artificial intelligence research focused on climate change applications. The climate change ML (CCML) community works on a diverse, challenging set of problems which often involve physics-constrained ML or heterogeneous spatiotemporal data. It would be desirable to use automated machine learning (AutoML) techniques to automatically find high-performing architectures and hyperparameters for a given dataset. In this work, we benchmark popular AutoML libraries on three high-leverage CCML applications: climate modeling, wind power forecasting, and catalyst discovery. We find that out-of-the-box AutoML libraries currently fail to meaningfully surpass the performance of human-designed CCML models. However, we also identify a few key weaknesses, which stem from the fact that most AutoML techniques are tailored to computer vision and NLP applications. For example, while dozens of search spaces have been designed for image and language data, none have been designed for spatiotemporal data. Addressing these key weaknesses can lead to the discovery of novel architectures that yield substantial performance gains across numerous CCML applications. Therefore, we present a call to action to the AutoML community, since there are a number of concrete, promising directions for future work in the space of AutoML for CCML. We release our code and a list of resources at https://github.com/climate-change-automl/climate-change-automl.

1 Introduction

There is an increasing body of evidence which shows that climate change is one of the biggest threats facing humanity today [3, 7, 33, 39]. Taking action towards climate change must come in many forms, such as reducing greenhouse gases and facilitating the adaption of renewable energy. A rapidly developing area of artificial intelligence research, climate change ML (CCML), is focused on applications to address climate change [11, 26, 38].

On the other hand, the automated machine learning (AutoML) community has been focused on designing efficient algorithms for problems such as hyperparameter optimization (HPO) and neural architecture search (NAS) [22]. In general, the goal of AutoML is to develop algorithms that automate the process of designing architectures and tuning hyperparameters for a given dataset. Although AutoML would seemingly be most useful on understudied datasets where there is less human intuition [36, 47], most AutoML techniques, whether implicitly or explicitly, are tailored to CV and NLP tasks. Furthermore, a few recent works show that state-of-the-art AutoML techniques for common CV-based tasks do not transfer to other non-CV tasks [32, 47]. A natural question is therefore, *are AutoML techniques beneficial for high-impact CCML applications?*

*Work done while RT was part-time at Abacus.AI. Correspondence to: Colin White <colin@abacus.ai>.

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2022.

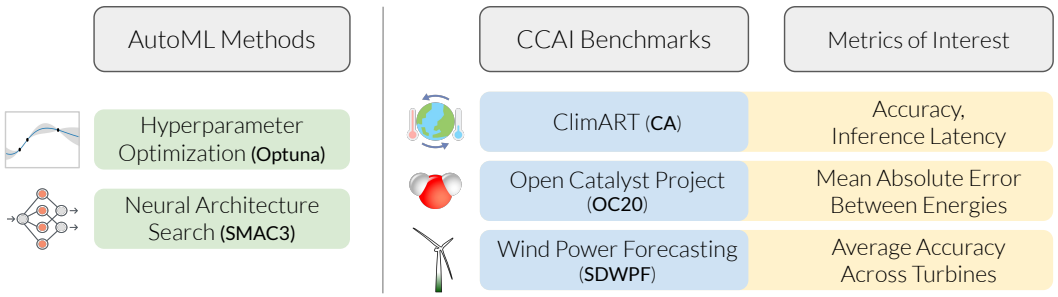

Figure 1: Overview of the main components of our study.

In this work, we benchmark popular AutoML libraries on three high-leverage CCML tasks: climate modeling, wind power forecasting, and catalyst discovery (see Fig. 1). We find that AutoML techniques currently do *not* work out of the box on these tasks, failing to meaningfully surpass the performance of human-designed models. At the same time, we identify concrete weaknesses stemming from the fact that AutoML techniques have not been designed for common CCML themes such as spatiotemporal data or physics-constrained ML. For example, designing a search space which interpolates among MLPs, CNNs, GNNs, and GCNs (all of which have been used for climate modeling [5, 6, 31, 34]) would allow NAS algorithms to discover novel combinations of existing architecture components, potentially leading to substantial performance gains across several spatiotemporal forecasting applications. Thus, we give a call to action to the AutoML community, with the aim of leveraging the full power of AutoML on challenging, high-impact CCML tasks.

Related work. In recent years, several techniques have been developed for atmospheric radiative transfer [4, 6, 35, 48], wind power forecasting [10, 49], catalyst prediction [9, 28, 46], and many more areas [24, 25, 50]. For a survey of machine learning tasks in the climate change space, see [38].

HPO [17] and NAS [16] are two popular areas of AutoML [22]. Recently, Tu *et al.* introduced NAS-Bench-360 [47], a benchmark suite to evaluate NAS methods on a diverse set of understudied tasks, in order to help move the field of NAS away from its emphasis on CV and NLP. Across ten tasks, Tu *et al.* showed that current state-of-the-art NAS methods do not perform well on diverse tasks. Another recent work similarly showed that the best techniques and hyperparameters on CV-based tasks do not transfer to more diverse tasks [32]. However, for both of these works, the analyses used a few fixed search spaces rather than identifying models hand-designed specifically for each task.

2 Methodology

In this section, we describe our methodology, driven by the following two research questions:

- **RQ 1:** Can current out-of-the-box AutoML techniques substantially improve performance compared to human-designed models in high-leverage climate-relevant applications?
- **RQ 2:** If not, then what are the key limitations and weaknesses of existing techniques?

In order to answer **RQ 1**, we select datasets which *(1)* correspond to impactful directions in climate change research, and *(2)* have existing strong human-designed baselines. For example, we choose datasets which were recently featured in large competitions, with top solutions now open-source. We describe the details of each dataset in Section 3.

For each of the datasets we choose, we first find open-source high-performing human-designed models. Then we run Optuna [1] or SMAC3 [30], two of the most widely used AutoML libraries today, using top human-designed models as the base. We compare the resulting searched models to top human-designed models.

In order to answer **RQ 2**, we check for general weaknesses in AutoML techniques applied to CCML tasks, which can be overcome with future work. For example, we look at whether the AutoML techniques are limited due to being implicitly tailored to CV tasks.

3 Experiments and Discussion

In this section, for three CCML tasks, we give a brief description of the task, dataset, and our AutoML experiments. Then, in Section 3.2, we use our experiments to answer **RQ 1** and **RQ 2**.

3.1 Experimental Setup

Atmospheric Radiative Transfer. Numerical weather prediction models, as well as global and regional climate models, give crucial information to policymakers and the public about the impact of changes in the Earth's climate. The bottleneck is atmospheric radiative transfer (ART) calculations, which are used to compute the heating rate of any given layer of the atmosphere. While ART has historically been calculated using computationally intensive physics simulations, researchers have recently used neural networks to substantially reduce the computational bottleneck, enabling ART to be run at finer resolutions and obtaining better overall predictions.

We use the ClimART dataset [6] from the NeurIPS Datasets and Benchmarks Track 2021. It consists of global snapshots of the atmosphere across a discretization of latitude, longitude, atmospheric height, and time from 1979 to 2014. Each datapoint contains measurements of temperature, water vapor, and aerosols. Prior work has tested MLPs, CNNs, GNNs, and GCNs as baselines [6].

We run HPO on the CNN baseline from Cachay *et al.* [6] using the Optuna library [1]. The CNN model is chosen because it had the lowest RMSE and second-lowest latency out of all five baselines from Cachay *et al.* We tune learning rate, weight decay, dropout, and batch size. We also run NAS using SMAC3 [30]. We set a categorical hyperparameter to choose among MLP, CNN, GNN, GCN, and L-GCN [5] while also tuning learning rate and batch size. See Appendix B.1 for more details of the dataset and experiments.

Wind Power Forecasting. Wind power is one of the leading renewable energy types, since it is cheap, efficient, and harmless to the environment [2, 19, 40]. The only major downside in wind power is its unreliablility: changes in wind speed and direction make the energy gained from wind power inconsistent. In order to keep the balance of energy generation and consumption on the power grid, other sources of energy must be added on short notice when wind power is down, which is not always possible (for example, coal plants take at least 6 hours to start up) [20]. Therefore, forecasting wind power is an important problem that must be solved to facilitate greater adoption of wind power.

We use the SDWPF (Spatial Dynamic Wind Power Forecasting) dataset, which was recently featured in a KDD Cup 2022 competition that included 2490 participants [49]. This is by far the largest wind power forecasting dataset, consisting of data from 134 wind turbines across 12 months. The features consist of external features such as wind speed, wind direction, and temperature, and turbine features such as pitch angle of the blades, operating status, relative location, and elevation.

We use a BERT-based model [44], and a GRU+LGBoost model [29], which placed 3rd and 7th in the competition out of 2490, respectively (and are 1st and 3rd among open-source models, respectively). We run HPO using Optuna for both the BERT-based model and the GRU+LGBoost model, and we also run NAS on the BERT-based model. For additional details, see Appendix B.2.

Open Catalyst Project. Discovering new catalysts is key to cost-effective chemical reactions to address the problem of energy storage, which is necessitated by the intermittency of power generation from growing renewable sources, such as wind and solar. Catalyst discovery is also important for more efficient production of ammonia fertilizer, which currently makes up 1% of the world's CO_2 emissions [21]. Modern methods for catalyst design use a simulation via density functional theory (DFT), which can be approximated with deep learning. Specifically, given a set of atomic positions for the reactants and catalyst, the energy of the structure can be predicted.

We use the Open Catalyst 2020 (OC20) dataset [8], which was featured in a NeurIPS 2021 competition [9]. Each datapoint is one reaction, where the features consist of the initial starting positions of the atoms, and the label consists of the energy needed to drive the reaction. In our experiments, we use a downsampled version of the OC20 IS2RE out-of-distribution adsorbates task, using 59 904 examples.

We use Graphormer [42], the winning solution from the NeurIPS 2021 Open Catalyst Challenge, developed by a team at Microsoft. We run Optuna on the learning rate, number of warmup steps, number of layers, attention heads, and blocks. See Appendix B.3 for additional details.

Dataset	Type	Base Model	Metric	Perf. human	Perf. AutoML	improv. %	search time
ClimART	NAS	Various	RMSE (W/m^2)	1.829	1.669	8.7%	12 GPU hrs
ClimART	HPO	CNN	RMSE (W/m^2)	1.829	1.538	15.9%	54 GPU hrs
SDWPF	NAS	BERT-based	RMSE+MAE (kW)	45.246	45.178	0.15%	26 GPU hrs
SDWPF	HPO	BERT-based	RMSE+MAE (kW)	45.246	45.329	-0.08%*	42 GPU hrs
SDWPF	HPO	GRU+GBDT	RMSE+MAE (kW)	45.074	45.074	0%	50 GPU hrs
OC20	HPO	Graphormer	MAE (eV)	0.399	0.396	0.65%	24 GPU hrs

Table 1: Empirical comparison between human-designed models and AutoML searched models. In the 'Perf. AutoML' column, we report the test set performance of the model with the best validation set performance during the AutoML search (* which may be worse than the original model, if the validation set performance is higher but the test set performance is lower).

3.2 Results and Discussion

See Table 1 for results and percentage improvement by running AutoML. Despite running for 10-50 hours on each task, the AutoML techniques did not meaningfully improve performance compared to the best human-designed model, with the exception of ClimART. However, for ClimART, we were unable to reproduce the originally reported RMSE of the CNN model [6] with the default parameters, and so the AutoML performance is compared to our own (worse) evaluation of the default model. Overall, although our experiments are not comprehensive, we find no indication that **RQ 1** is true; in other words, out-of-the-box AutoML techniques currently may not be able to substantially improve upon human-designed CCML models. We emphasize that our experiments were aimed specifically at evaluating AutoML methods *out-of-the-box*. For a discussion of limitations, see Appendix B.

We find that a key weakness of current AutoML methods is that the search spaces are designed for common tasks such as CV and NLP. For example, ClimART could benefit from search spaces that interpolate among MLPs, CNNs, GNNs, and GCNs, which do not currently exist. In general, many CCML applications would benefit from search spaces designed specifically to handle spatiotemporal forecasting tasks, both two-dimensional [23, 37, 49, 50] and three-dimensional [6, 31]. Furthermore, many CCML applications have physics constraints in some form [6, 9, 14, 15]. For example, ART and catalyst prediction follow the physics of thermodynamics. Architectures which incorporate physics constraints, and loss terms with several hyperparameters, are two common methods for handling physics constraints [27], and using AutoML to search for the best architecture and loss function is a promising area for future work. Therefore, our answer to **RQ 2** is that search spaces are currently focused on CV tasks, and designing search spaces for spatiotemporal forecasting and physics constraints would be particularly beneficial across CCML applications.

4 Conclusions and Future Work

In this work, we benchmarked popular AutoML libraries on datasets for climate modeling, wind power forecasting, and catalyst discovery, and we were unable to show that out-of-the-box AutoML libraries substantially improve over human-designed models.

There are many concrete, promising avenues for future work. First and foremost, designing search spaces for spatiotemporal forecasting and physics constraints, as mentioned in Section 3.2, would be particularly beneficial across many CCML applications. Next, while our work focused on HPO and NAS, there are still many other sub-areas of AutoML, such as data augmentation, data preprocessing, and continuous monitoring and maintenance of deployed models. Finally, while our work focused on three high-impact datasets, there are many other CCML applications for which AutoML could be tested, such as model predictive control for buildings [12, 13] and optimal power flow [18]. However, researchers must be careful to also consider the large carbon footprint caused by AutoML experiments [41, 45]. For a longer discussion on the broader impact of our work, see Appendix A.

Acknowledgments

The authors thank Priya Donti and Ján Drgoňa for their help with this project. This work was supported in part by the NSF (1651565), AFOSR (FA95501910024), ARO (W911NF-21-1-0125), CZ Biohub, Sloan Fellowship, National Science Foundation grants IIS1705121, IIS1838017, IIS2046613, IIS2112471, funding from Meta, Morgan Stanley, Amazon, Google, National Science Foundation

grants CCF2106707, and funding from Wisconsin Alumni Research Foundation (WARF). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of these funding agencies.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] Cristina L Archer and Mark Z Jacobson. Evaluation of global wind power. *Journal of Geophysical Research: Atmospheres*, 110(D12), 2005.

[3] David Archer and Stefan Rahmstorf. *The climate crisis: An introductory guide to climate change*. Cambridge University Press, 2010.

[4] Noah D Brenowitz and Christopher S Bretherton. Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, 45(12):6289–6298, 2018.

[5] Salva Rühling Cachay, Emma Erickson, Arthur Fender C Bucker, Ernest Pokropek, Willa Potosnak, Suyash Bire, Salomey Osei, and Björn Lütjens. The world as a graph: Improving el niño forecasts with graph neural networks. *arXiv preprint arXiv:2104.05089*, 2021.

[6] Salva Rühling Cachay, Venkatesh Ramesh, Jason NS Cole, Howard Barker, and David Rolnick. Climart: A benchmark dataset for emulating atmospheric radiative transfer in weather and climate models. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

[7] Projected Climate Change. Global warming of 1.5° c. *World Meteorological Organization: Geneva, Switzerland*, 2018.

[8] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, et al. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 11(10):6059–6072, 2021.

[9] Abhishek Das, Muhammed Shuaibi, Aini Palizhati, Siddharth Goyal, Aditya Grover, Adeesh Kolluru, Janice Lan, Ammar Rizvi, Anuroop Sriram, Brandon Wood, et al. The open catalyst challenge 2021: Competition report. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 29–40. PMLR, 2022.

[10] Xing Deng, Haijian Shao, Chunlong Hu, Dengbiao Jiang, and Yingtao Jiang. Wind power forecasting methods based on deep learning: A survey. *Computer Modeling in Engineering and Sciences*, 122(1):273, 2020.

[11] Priya L Donti and J Zico Kolter. Machine learning for sustainable energy systems. *Annual Review of Environment and Resources*, 46:719–747, 2021.

[12] Ján Drgoňa, Javier Arroyo, Iago Cupeiro Figueroa, David Blum, Krzysztof Arendt, Donghun Kim, Enric Perarnau Ollé, Juraj Oravec, Michael Wetter, Draguna L Vrabie, et al. All you need to know about model predictive control for buildings. *Annual Reviews in Control*, 50:190–232, 2020.

[13] Ján Drgoňa, Damien Picard, Michal Kvasnica, and Lieve Helsen. Approximate model predictive building control via machine learning. *Applied Energy*, 218:199–216, 2018.

[14] Ján Drgona, Aaron R Tuor, Vikas Chandan, and Draguna L Vrabie. Physics-constrained deep recurrent neural models of building thermal dynamics. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2020.

[15] Ján Drgoňa, Aaron R Tuor, Vikas Chandan, and Draguna L Vrabie. Physics-constrained deep learning of multi-zone building thermal dynamics. In *Energy and Buildings*, volume 243. Elsevier, 2021.

[16] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. In *JMLR*, 2019.

[17] Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated machine learning*, pages 3–33. Springer, Cham, 2019.

[18] Ferdinando Fioretto, Terrence WK Mak, and Pascal Van Hentenryck. Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 630–637, 2020.

[19] Aoife M Foley, Paul G Leahy, Antonino Marvuglia, and Eamon J McKeogh. Current methods and advances in forecasting of wind power generation. *Renewable energy*, 37(1):1–8, 2012.

[20] Shahram Hanifi, Xiaolei Liu, Zi Lin, and Saeid Lotfian. A critical review of wind power forecasting methods—past, present and future. *Energies*, 13(15):3764, 2020.

[21] Leif Hockstad and L Hanel. Inventory of us greenhouse gas emissions and sinks. Technical report, Environmental System Science Data Infrastructure for a Virtual Ecosystem, 2018.

[22] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2019.

[23] IARAI. Weather4cast neurips competition 2021. https://www.iarai.ac.at/weather4cast/, 2021.

[24] Jeremy Irvin, Hao Sheng, Neel Ramachandran, Sonja Johnson-Yu, Sharon Zhou, Kyle Story, Rose Rustowicz, Cooper Elsworth, Kemen Austin, and Andrew Y Ng. Forestnet: Classifying drivers of deforestation in indonesia using deep learning on satellite imagery. *arXiv preprint arXiv:2011.05479*, 2020.

[25] Piyush Jain, Sean CP Coogan, Sriram Ganapathi Subramanian, Mark Crowley, Steve Taylor, and Mike D Flannigan. A review of machine learning applications in wildfire science and management. *Environmental Reviews*, 28(4), 2020.

[26] Lynn H Kaack, Priya L Donti, Emma Strubell, George Kamiya, Felix Creutzig, and David Rolnick. Aligning artificial intelligence with climate change mitigation. *Nature Climate Change*, pages 1–10, 2022.

[27] K Kashinath, M Mustafa, A Albert, JL Wu, C Jiang, S Esmaeilzadeh, K Azizzadenesheli, R Wang, A Chattopadhyay, A Singh, et al. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379(2194):20200093, 2021.

[28] Adeesh Kolluru, Muhammed Shuaibi, Aini Palizhati, Nima Shoghi, Abhishek Das, Brandon Wood, C Lawrence Zitnick, John R Kitchin, and Zachary W Ulissi. Open challenges in developing generalizable large scale machine learning models for catalyst discovery. *arXiv preprint arXiv:2206.02005*, 2022.

[29] Fangquan Lin, Wei Jiang, Hanwei Zhang, and Cheng Yang. Kdd cup 2022 wind power forecasting team 88vip solution. *arXiv preprint arXiv:2208.08952*, 2022.

[30] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54):1–9, 2022.

[31] Ying Liu, Rodrigo Caballero, and Joy Merwin Monteiro. Radnet 1.0: Exploring deep learning architectures for longwave radiative transfer. *Geoscientific Model Development*, 13(9):4399–4412, 2020.

[32] Yash Mehta, Colin White, Arber Zela, Arjun Krishnakumar, Guri Zabergja, Shakiba Moradian, Mahmoud Safari, Kaicheng Yu, and Frank Hutter. Nas-bench-suite: Nas evaluation is (now) surprisingly easy. In *International Conference on Learning Representations*, 2022.

[33] Nebojsa Nakicenovic and Rob Swart. Emissions scenarios-special report of the intergovernmental panel on climate change, 2000.

[34] Anikesh Pal, Salil Mahajan, and Matthew R Norman. Using deep neural networks as cost-effective surrogate models for super-parameterized e3sm radiative transfer. *Geophysical Research Letters*, 46(11):6069–6079, 2019.

[35] Stephan Rasp, Michael S Pritchard, and Pierre Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018.

[36] Nicholas Roberts, Mikhail Khodak, Tri Dao, Liam Li, Christopher Ré, and Ameet Talwalkar. Rethinking neural operations for diverse tasks. *Advances in Neural Information Processing Systems*, 34:15855–15869, 2021.

[37] R Rohde, R Muller, R Jacobsen, S Perlmutter, A Rosenfeld, J Wurtele, J Curry, C Wickhams, and S Mosher. Berkeley earth temperature averaging process, geoinfor. geostat.-an overview, 1, 2. *Geoinformatics Geostatistics An Overview*, 1(2):20–100, 2013.

[38] David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2):1–96, 2022.

[39] Joseph Romm. *Climate change: What everyone needs to know*. Oxford University Press, 2022.

[40] Perry Sadorsky. Wind energy for sustainable development: Driving factors and future outlook. *Journal of Cleaner Production*, 289:125779, 2021.

[41] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020.

[42] Yu Shi, Shuxin Zheng, Guolin Ke, Yifei Shen, Jiacheng You, Jiyan He, Shengjie Luo, Chang Liu, Di He, and Tie-Yan Liu. Benchmarking graphormer on large-scale molecular modeling datasets. *arXiv preprint arXiv:2203.04810*, 2022.

[43] Neil C Swart, Jason NS Cole, Viatcheslav V Kharin, Mike Lazare, John F Scinocca, Nathan P Gillett, James Anstey, Vivek Arora, James R Christian, Sarah Hanna, et al. The canadian earth system model version 5 (canesm5. 0.3). *Geoscientific Model Development*, 12(11):4823–4873, 2019.

[44] Longxing Tan and Hongying Yue. Application of bert in wind power forecasting-teletraan's solution in baidu kdd cup 2022, 2022.

[45] Tanja Tornede, Alexander Tornede, Jonas Hanselle, Marcel Wever, Felix Mohr, and Eyke Hüllermeier. Towards green automated machine learning: Status quo and future directions. *arXiv preprint arXiv:2111.05850*, 2021.

[46] Richard Tran, Janice Lan, Muhammed Shuaibi, Siddharth Goyal, Brandon M Wood, Abhishek Das, Javier Heras-Domingo, Adeesh Kolluru, Ammar Rizvi, Nima Shoghi, et al. The open catalyst 2022 (oc22) dataset and challenges for oxide electrocatalysis. *arXiv preprint arXiv:2206.08917*, 2022.

[47] Renbo Tu, Nicholas Roberts, Mikhail Khodak, Junhong Shen, Frederic Sala, and Ameet Talwalkar. NAS-bench-360: Benchmarking neural architecture search on diverse tasks. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2022.

[48] Janni Yuval and Paul A O'Gorman. Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature communications*, 11(1):1–10, 2020.

[49] Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jiantao Su, Junfu Lyu, Yanjun Ma, and Dejing Dou. Sdwpf: A dataset for spatial dynamic wind power forecasting challenge at kdd cup 2022. *arXiv preprint arXiv:2208.04360*, 2022.

[50] Bryan Zhu, Nicholas Lui, Jeremy Irvin, Jimmy Le, Sahil Tadwalkar, Chenghao Wang, Zutao Ouyang, Frankie Y Liu, Andrew Y Ng, and Robert B Jackson. Meter-ml: A multi-sensor earth observation benchmark for automated methane source mapping. *arXiv preprint arXiv:2207.11166*, 2022.

A Broader Impact

Our goal in this work is to give evidence that current out-of-the-box AutoML techniques do not perform sufficiently on high-impact CCML applications, and then give a call to action to the AutoML community by identifying several concrete areas for future work. The successes would be for *(1)* AutoML researchers to design and test their methods on CCML tasks, and *(2)* CCML practitioners to use (future) AutoML tools to make progress in their respective domains.

Although automated machine learning is a powerful tool to make progress on climate change problems, the surprisingly large carbon footprint and financial cost of training machine learning models must be weighed [41, 45]. While we strongly believe that the AutoML community heeding our call to action will have a net positive impact on society, we urge AutoML researchers to conduct research in a responsible and climate-conscious manner, using the suggestions laid out by Tornede *et al.* [45].

B Details from Section 3

In this section, we give details from the experiments in Section 3. We also note that although we ran AutoML techniques across three different CCML tasks, our experiments should not be seen as a comprehensive evaluation of AutoML methods on CCML tasks. In particular, our experiments come with the limitations that only one trial was run per experiment (due to the a single run taking up to 50 GPU hours) and although we made reasonable choices for the AutoML methods (based on popularity) and hyperparameter ranges (based on default values), we did not run an exhaustive search across AutoML methods and hyperparameters.

Furthermore, we explicitly aimed to test AutoML performance *out-of-the-box*, and we therefore did not spend the time to carefully design tailored search spaces to the tasks at hand, which would be non-trivial (e.g., see [36]). However, we discuss this as a very promising area for future work in Sections 3.2 and 4.

B.1 ClimART

First, we give additional details about ClimART and the corresponding experiments.

ClimART. The ClimART dataset [6] consists of data that is simulated from CanESM5 [43]. This dataset takes global snapshots of the atmosphere split into a 128×64 latitude-longitude grid, every 205 hours from 1979 to 2014. Each datapoint is a "column" of the atmosphere at a specific time, with measurements of temperature, water vapor, and aerosols taken at 49 different heights. Each column also has global properties, such as optical and geographical information. Prior work has tested MLPs, CNNs, GNNs, and GCNs as baselines [6].

B.1.1 SMAC3 details

We use data from years 1990, 1999, and 2003 for training, and data from 2005 for validation, to match the setting of the benchmark experiments in the original ClimART paper [6]. Each model is trained for 5 epochs and validated. Hyperparameters for each model are set according to the original configurations provided by ClimART authors. We use a hyperparameter search space as follows:

- Unif{MLP, CNN, GNN, GCN, L-GCN}
- $\log_{10}$ (learning rate): Unif$[-5, -1]$
- $\log_{10}$ (weight decay): Unif$[-7, -4]$

B.1.2 Optuna Details

We use a hyperparameter search space as follows:

- $\log_{10}$ (learning rate): Unif$[-5, -1]$
- $\log_{10}$ (weight decay): Unif$[-7, -4]$
- dropout: Unif$[0.0, 0.8]$
- batch size: 2**int(Unif$[7.0, 9.0]$)

	Learning rate	Weight decay	Dropout	Batch size	Test RMSE
36 trials/ 20 epochs	1.43e-4	2.14e-5	0.0	256	1.538
24 trials/ 10 epochs	4.12e-4	1.96e-5	0.001	256	2.344
original	2e-4	1e-6	0.0	128	1.829

Table 2: Searched hyperparameters and performance comparison with original configuration.

We ran Optuna by training each architecture to 10 epochs during the search, and training each architecture to 20 epochs during the search. The best model according to validation accuracy is fully trained to 100 epochs and then the test accuracy is compared to the original (default) model (also fully trained to 100 epochs).

B.2 SDWPF

Next, we give details of the SDWPF dataset and experiments.

SDWPF. The SDWPF (Spatial Dynamic Wind Power Forecasting) dataset was recently featured in a KDD Cup 2022 competition that included 2490 participants [49].[2] This is by far the largest wind power forecasting dataset, consisting of data from 134 wind turbines across 12 months, with data sampled every 10 minutes. The features consist of external features such as wind speed, wind direction, temperature, and turbine features such as nacelle direction, pitch angle of the blades, operating status, relative location, and elevation. The problem is to predict the generated power for all 134 turbines every 10 minutes in a 48 hour time window.

We ran hyperparameter optimization over the BERT-based model with batch size and learning rate using Optuna. Due to computational constraints, we ran the search over 25% of the data and then trained the best model according to the validation set with the whole data.

We use a hyperparameter search space as follows:

- $\log_{10}$ (learning rate): Unif$[-7, -1]$
- batch size: 2**int(Unif$[5.0, 10.0]$)
- Feed Forward Network dropout: Unif$[0.0, 0.5]$
- Attention dropout: Unif$[0.0, 0.5]$

	Learning rate	Batch size	Test Score
70 trials/ 50% data	4.7e-3	512	-45.329
original	5e-3	1024	-45.246

Table 3: Searched hyperparameters via HPO and performance comparison with original configuration on SDWPF.

[2] https://aistudio.baidu.com/aistudio/competition/detail/152

Next, we ran neural architecture search on the same BERT-based architecture, using 50% of the data. The search space is as follows:

- No. of BERT Blocks {1,2,4}
- No. of heads in attention model {1,2,4}
- Attention dropout inside BERT block Unif[0.0,0.4]
- Feed Forward Network dropout inside BERT block Unif[0.0,0.4]
- Filter sizes inside BERT Block {8,16,32,64,128}

	num blocks	num heads	attention dropout	ffn dropout	Test Score
40 trials/ 50% data	1	1	0.224	0.097	-45.178
original	1	1	0.0	0.0	-45.246

Table 4: Searched model parameters via NAS and performance comparison with the original configuration on SDWPF.

Finally, we ran HPO on the GRU+LGBoost algorithm. We used the following hyperparameter search space:

- No. of numeric embedding dimension: int(Unif[32,64])
- No. of time embedding dimension: int(Unif[4,8])
- No. of ID embedding dimension: int(Unif[4,8])
- No of GRU hidden units: int(Unif[32,64])
- log_{10} (Learning rate): Unif([-6,-2])

And the hyperparameter search space for the LGBoost model is as follows:

- No. of leaves: int(Unif[2,128])
- Bagging frequency: int(Unif[1,7])
- Bagging fraction: Unif[0.4,1]
- Feature fraction: Unif[0.4,1]
- Learning rate : Unif[0.001,0.7]

	num_sz	time_sz	id_sz	hiden	GRU_lr	Test Score
20 GRU trials + 50 LGBoost trials	51	4	4	64	0.009538	-45.074
original	51	4	4	64	0.009538	-45.074

Table 5: Searched GRU model parameters and performance comparison with original configuration on SDWPF.

	num_lv	bag_freq	bag_frac	feat_frac	LGBoost_lr	Test Score
20 GRU trials + 50 LGBoost trials	128	5	0.998798	0.428377	0.00342	-45.074
original	128	5	0.998798	0.428377	0.00342	-45.074

Table 6: Searched LGBoost model parameters for a sample LightGBM model and performance comparison with original configuration on SDWPF.

B.3 OC20

Finally, we give the details of the OC20 dataset and experiments.

OC20. The Open Catalyst 2020 (OC20) dataset [8] was featured in a NeurIPS 2021 competition [9]. Each datapoint is one reaction, where the features consist of the initial starting positions of the atoms, and the label consists of the energy needed to drive the reaction. There are over 100 million examples in total in the original dataset. In our experiments, we use a down-sampled version of the OC20 IS2RE task with 10 000 examples where we report test accuracy on out-of-domain adsorbates.

B.3.1 Optuna Details

We use the following hyperparameter search space:

- $\log_{10}$ (learning rate): Unif$[-5, -3]$
- $\log_{10}$ (warm-up steps): Unif$[0, 4]$
- layers: Unif$[1, 12]$
- attention heads: Unif$[\{6, 12, 24, 32, 48\}]$
- blocks: Unif$[1, 4]$

	Learning rate	Warm up steps	Layers	Attention heads	Blocks	Test MAE
36 trials/ 4 epochs	2.9e-4	133	9	32	1	0.396
original	3e-4	100	12	48	4	0.399

Table 7: Searched hyperparameters and performance comparison with original configuration on OC20.