Accuracy is Not the Only Metric that Matters:

Estimating the Energy-Consumption of Deep Learning Models

Johannes Getzner, Bertrand Charpentier*, Stephan Günnemann
[getzner, charpent, guennemann]@in.tum.de

Published as a workshop paper at ”Tackling Climate Change with Machine Learning”, ICLR 2023

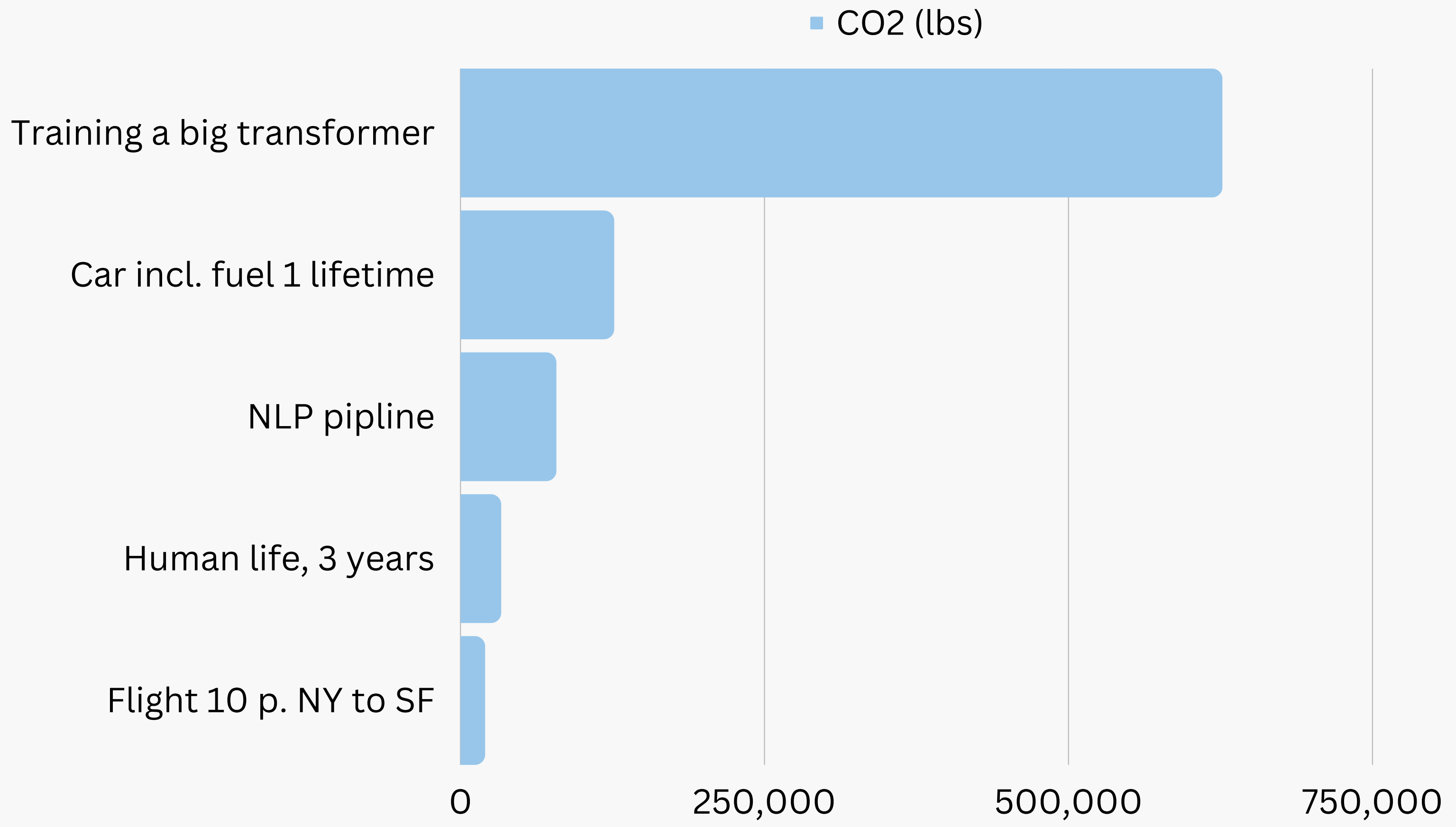
04.05.2023

DAML Group
Technical University of Munich

*Corresponding author

Deep Neural Networks consume astronomical amounts of power, incurring a large carbon footprint.

- Training & Inference require power-hungry hardware, often for multiple days [3][4].
- Estimating a model's energy consumption without running it is generally very difficult
- Models are usually not evaluated with respect to environmental impact



[1]

Our goal is to provide energy consumption estimates for deep neural nets based only on their configuration.

Benefits:

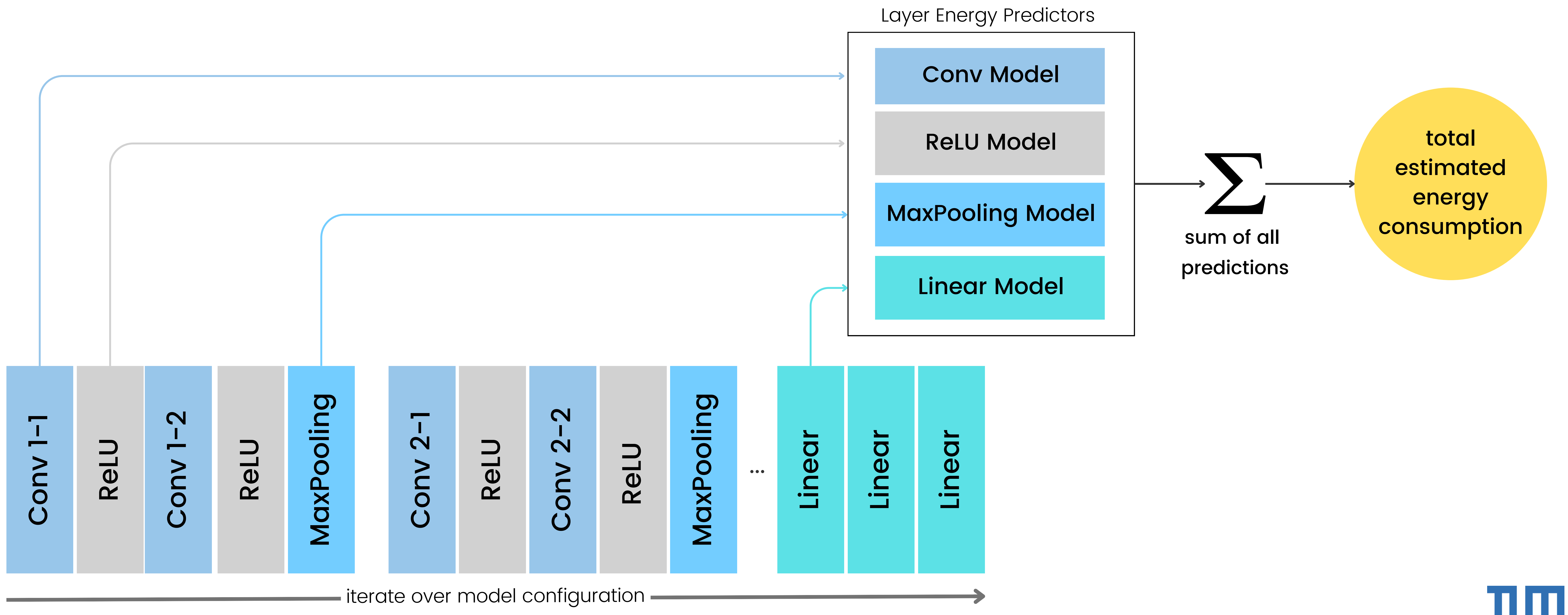
- Our method approximates energy consumption without running the model
- Promotes consideration of ecological footprint and running costs of models, raising environmental awareness

VGG16 configuration

```
(
 (features): Sequential(
  (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (1): ReLU(inplace=True)
  (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (3): ReLU(inplace=True)
  (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (6): ReLU(inplace=True)
  (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (8): ReLU(inplace=True)
  (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (11): ReLU(inplace=True)
  (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (13): ReLU(inplace=True)
  (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (15): ReLU(inplace=True)
  (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (18): ReLU(inplace=True)
  (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (20): ReLU(inplace=True)
  (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (22): ReLU(inplace=True)
  (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (25): ReLU(inplace=True)
  (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (27): ReLU(inplace=True)
  (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (29): ReLU(inplace=True)
  (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
 )
...
```

[2]

Compute the total energy consumption as the sum of layer-wise predicted energies.

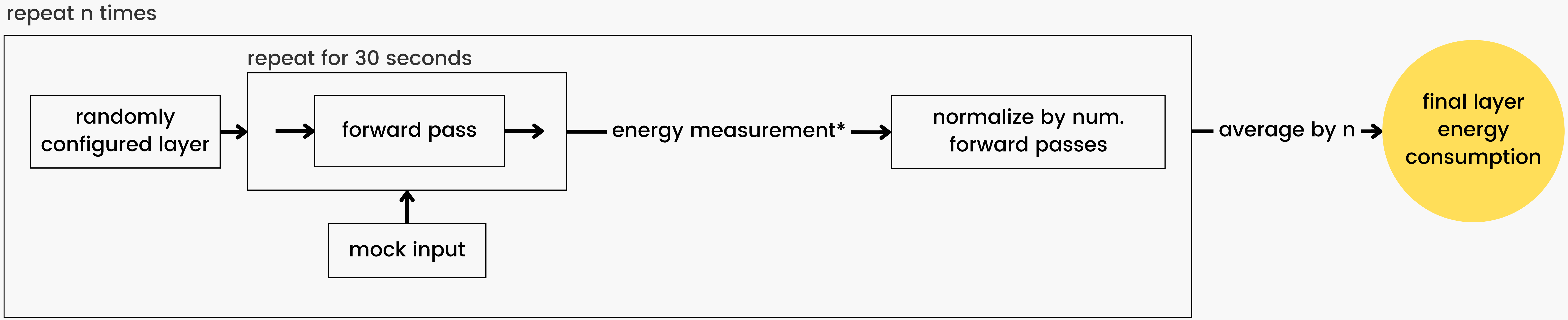


The Framework

1. Data collection process
2. Fitting the layer energy predictors
3. Estimating the total energy consumption

We release a robust and modular data collection process for CPU energy consumption.

- currently, 8 different layer types are implemented
- ~1000 data points collected for each one so far



*CPU energy collected with codecarbon [3] via Intel RAPL interface

Each layer type has a set of parameters that can be used to fit the energy estimation model.

layer	**model features** layer parameters	**energy contribution** in VGG13
Conv2D	kernel-size, image-size, in-channels, out-channels, padding, stride	88.42%
MaxPooling2D	kernel-size, image-size, in-channels, stride	9.14%
Linear	input-size, output-size	1.18%
Activations (ReLU, TanH, Sigmoid, Softmax)	input-size	1.19%

+ batch-size, log-transformed parameters, MAC count*

*the MAC count was calculated solely for the ReLU activation functions

For each layer, we selected the best set of features to predict its energy consumption.

- As no high-order dependencies were found, polynomial/linear regression models were chosen
- each model was evaluated concerning its avg. cross-validation MSE and R^2 score.
- features were standardized if they contained the MAC count

layer	model	model features
Conv2D	Linear	MAC count
MaxPooling2D	Polynomial²*	all**
Linear	Linear	MAC count
ReLU	Polynomial²*	MAC count
TanH	Polynomial²*	batch-size, input-size
Sigmoid	Polynomial²*	batch-size, input-size
Softmax	Polynomial²*	batch-size, input-size

*polynomial features, but restricted to interaction-only terms
** "all" corresponds to (log-transformed) layer parameters, (log-) batch-size, and the MAC count

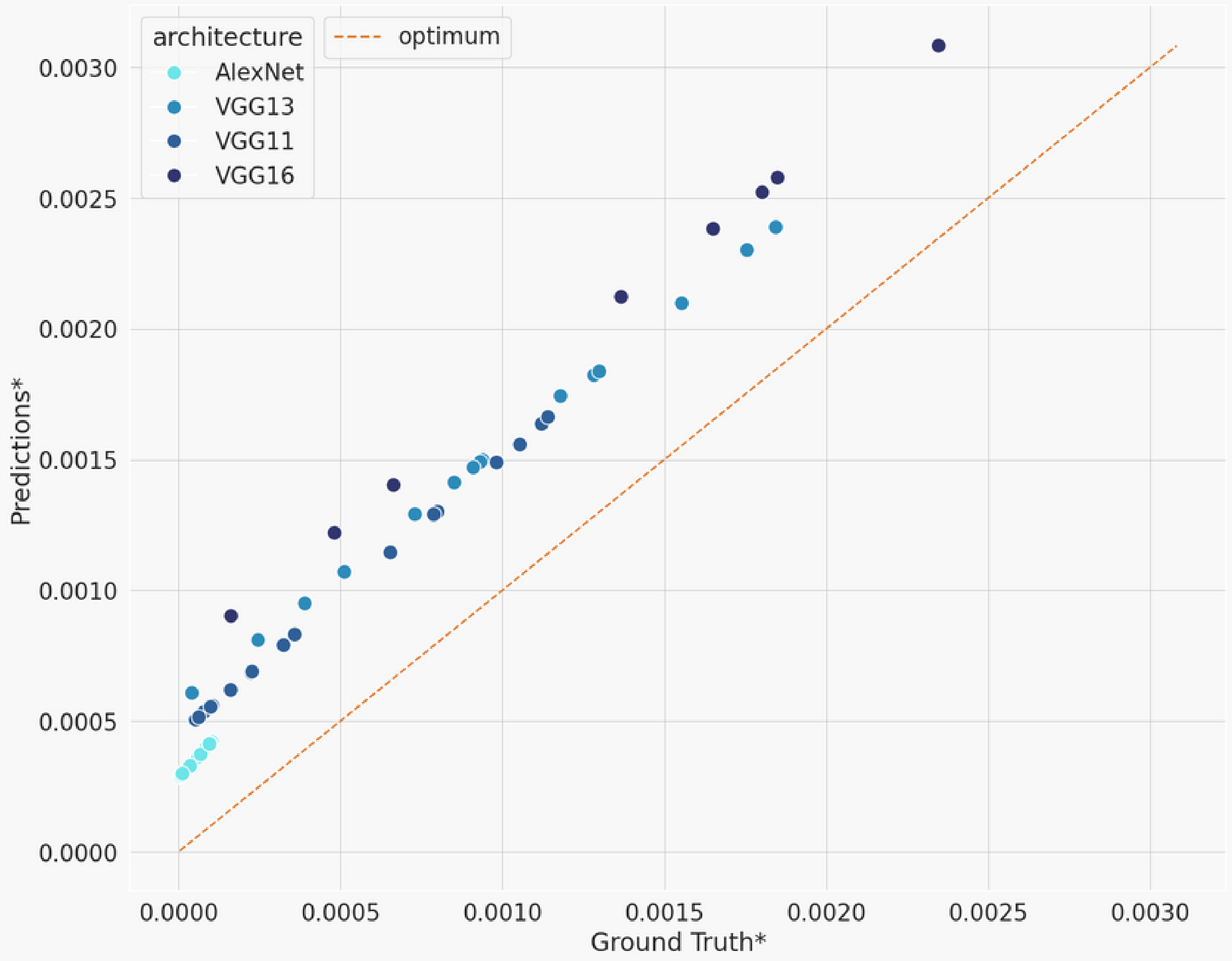
Although models demonstrated outstanding performance on random layer configurations, their generalization to layers from real architectures proved to be more challenging.

	R^2 score performance on test sets	
module	**random layer configurations**	**layer configurations from architectures**
Conv2D	0.9977	0.314
MaxPooling2D	0.9995	0.559
Linear	0.9992	0.977
ReLU*	0.9812	-21.51

*the other Activations are excluded as they are not present in any of the evaluated architectures

Together the models achieved an R^2 score of 0.352 for the total architecture energy consumption of AlexNet and VGG11/13/16.

- Together the models overestimate the total energy consumption slightly
- Largest contribution to the error comes from the Conv2D layers
- More energy-expensive and larger architectures suffer from greater overestimation



*axes are min-max scaled for visualization purposes

The main contributions of our work.

- We release a modular data-collection process along with an initial high-quality dataset on energy consumption of various architectures and layer types.
- We created predictors for different layer types as a simple energy estimation baseline for multiple DL architectures.
- We analyzed the predictive capabilities of various feature sets, providing insights into the energy behavior of different architectures and layer types.

Thank you for listening!

Questions?

Paper and Repository: https://www.cs.cit.tum.de/daml/energy-consumption-dl/#c35468

04.05.2023

DAML Group
Technical University of Munich

Johannes Getzner, Bertrand Charpentier*, Stephan Günnemann
[getzner, charpent, guennemann]@in.tum.de

V Appendix - References

[1] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. CoRR, abs/1906.02243, 2019. URL http://arxiv.org/abs/1906.02243.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. URL https://arxiv.org/abs/1409.1556v6, 2014

[3] Neil C. Thompson, Kristjan H. Greenewald, Keeheon Lee, and Gabriel F. Manso. The computational limits of deep learning. CoRR, abs/2007.05558, 2020. URL https://arxiv.org/abs/2007.05558.

[4] Peter H. Jin, Qiaochu Yuan, Forrest N. Iandola, and Kurt Keutzer. How to scale distributed deep learning? CoRR, abs/1611.04581, 2016. URL http://arxiv.org/abs/1611.04581.

[5] DataForGoodFR Mila. codecarbon. https://github.com/mlco2/codecarbon, 2022.