PREDICTING SOIL PROPERTIES FROM HYPERSPECTRAL SATELLITE IMAGES

Frauke Albrecht[⋆†] *Caroline Arnold*[⋆†] *Roshni Kamath*[⋆‡]
Kai Konen[⋆§] *Rıdvan Salih Kuzu*[⋆¶]

[⋆] Helmholtz AI Artificial Intelligence Cooperation Unit, Germany
[†] German Climate Computing Centre DKRZ, Hamburg, Germany
[‡] Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Germany
[§] Institute for Software Technology, German Aerospace Center (DLR), Cologne, Germany
[¶] The Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Germany

ABSTRACT

The AI4EO HYPERVIEW challenge seeks machine learning methods that predict agriculturally relevant soil parameters (K, Mg, P_2O_5, pH) from airborne hyperspectral images. We present a hybrid model fusing Random Forest and K-nearest neighbor regressors that exploit the average spectral reflectance, as well as derived features such as gradients, wavelet coefficients, and Fourier transforms. The solution is computationally lightweight and improves upon the challenge baseline by 21.9%, with the first place on the public leaderboard. In addition, we discuss neural network architectures and potential future improvements.

***Index Terms*—** hyperspectral images, random forests, artificial neural networks, soil parameter estimation, regression.

1. INTRODUCTION

Machine learning methods are employed widely in remote sensing [1]. In particular, agricultural monitoring via remote sensing draws significant attention for various purposes ranging from early forecasting of crop yield amount [2] to the estimation of soil composite [3].

Predicting the fertility indicators of soil, such as percentage of organic matter, or amount of fertilizer, is one of the leading research topics in earth observation [4] due to the emerging needs for improving the agricultural efficiency without harming nature. Particularly, the European Union Green Deal gives special importance to supporting conventional farming practices with earth observation (EO) and artificial intelligence (AI) for resilient production as well as healthy soil and biodiversity [5].

The AI4EO platform seeks to bridge the gap between the AI and EO communities [6]. In the AI4EO HYPERVIEW challenge, the objective is to predict soil properties from hyperspectral satellite images, including potassium (K), magnesium (Mg), and phosphorus pentoxide (P_2O_5) content, and the pH value [7]. The winning solution of the challenge will be running on-board the Intuition-1 satellite.

In this manuscript, we present the solution to the AI4EO HYPERVIEW challenge developed by Team EAGLEEYES. Section 2 discusses the hyperspectral image dataset, Section 3 covers feature engineering and experimental protocols for different learning strategies, and Section 4 presents the preliminary performance results for predicting the given four soil properties. Eventually, we conclude in Section 5 and give an outlook on future work.

2. DATASET

The hyperspectral images are taken from airborne measurements from an unspecified location in Poland. In total, 1732 patches are available for training, and 1154 patches remain for testing. Each patch contains 150 hyperspectral bands, spanning $462 - 492\,\mathrm{nm}$ with a spectral resolution of $3.2\,\mathrm{nm}$ [7].

Samples in the dataset have been segmented into patches according to the boundaries of the agricultural fields. As shown in Figure 1, the patch size distribution is skewed: About one third of the samples is composed of $11 \times 11\,\mathrm{px}$ patches, and 60% of the patches are less than $50\,\mathrm{px}$ wide.

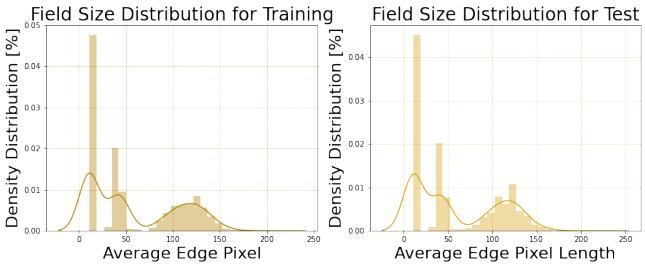

Fig. 1. Distribution of dataset in terms of different patch sizes.

The training data provides ground truth for all four soil parameters. The target values for P_2O_5 lie in the range of $[20.3 - 325]$, for K in $[21.1 - 625]$, for Mg in $[26.8 - 400]$, and for pH in $[5.6 - 7.8]$. As shown in Figure 2, for P_2O_5 and K, the target values follow a log-normal distribution with positive skewness, while the Mg and pH values are more

Gaussian distributed. Besides, pH measurements are mostly clustered in the intervals of 0.1.

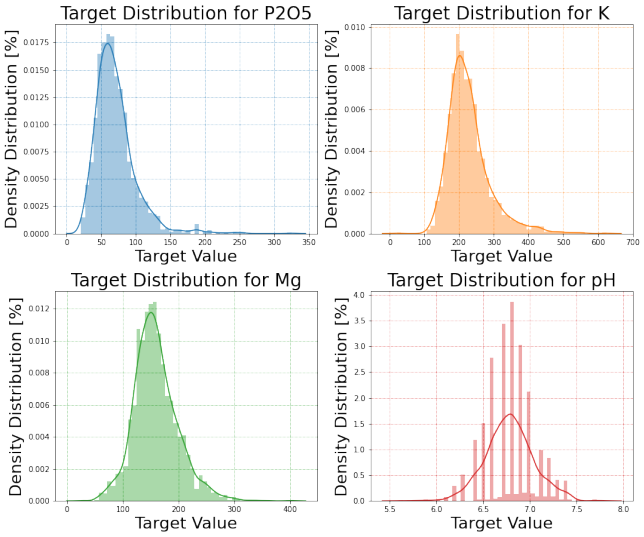

Fig. 2. Distribution of target values for each soil parameter.

3. EXPERIMENTAL FRAMEWORK

In this section, we present the feature engineering approaches, experimental protocols for different learning strategies and metrics utilized for validating our approach.

3.1. Data Processing and Augmentation

3.1.1. Feature engineering for traditional ML approaches

As mentioned in Section 2, the samples are 3-dimensional patches with dimension $(w \times h \times c)$ where width (w), and height (h) have varying sizes, but channel (c) is fixed to represent 150 spectral bands.

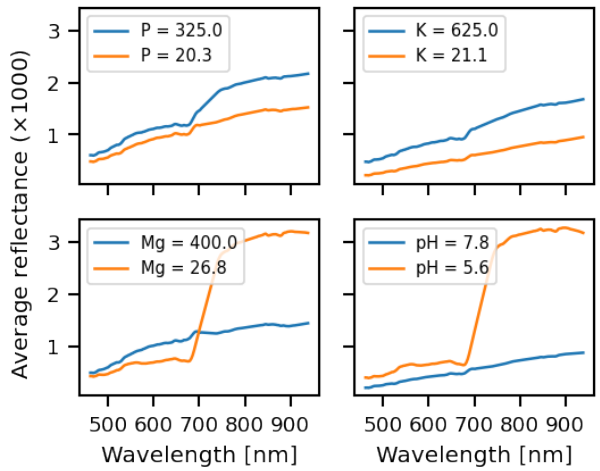

Fig. 3. Comparison of average reflectance for different agricultural fields for each soil parameter.

Figure 3 shows the average reflectance as a function of wavelength for the samples with minimum and maximum value of the respective soil parameter. These average reflectance curves are remarkably dissimilar for different values of the target variable. Thus, we use the average reflectance as a base feature in our experiments, and derive additional features from it. The list of the features is as follows:

- *average reflectance*, its 1^{st}, 2^{nd} and 3^{rd} order derivatives, ($[1 \times 150]$ dimension for each, $[1 \times 600]$ in total),
- discrete wavelet transforms of *average reflectance* with Meyer wavelet [8]: 1^{st}, 2^{nd}, 3^{rd}, 4^{th} level *approximation* and *detail* coefficients ($[1 \times 300]$ dims. in total),
- for each channel (c) of a field patch (P), singular value decomposition (SVD) has been conducted: $P_{(w \times h)} = U \Sigma V^T$, in which Σ is square diagonal of size $[r \times r]$ where $r \leq min\{w, h\}$. The first 5 diagonal values (σ_1, σ_2, σ_3, σ_4, $\sigma_5 \in \Sigma$) from each channel are selected as features ($[1 \times 750]$ dims. in total),
- the ratio of 1^{st}, 2^{nd} diagonals: σ_1/σ_2 ($[1 \times 150]$ dims.),
- Fast Fourier transform (FFT) of *average reflectance* and FFT of σ_1/σ_2: real and imaginary parts are included ($[1 \times 600]$ dims. in total).

To sum up, for each field patch, a $[1 \times 2400]$ dimensional feature array is extracted. Some of those features for different agricultural fields are illustrated in Figure 4. For data augmentation, 1% random Gaussian noise is added to both input features and target values.

3.1.2. Feature engineering for deep learning approaches

For experimenting on neural networks, either a raw patch, $P_{(w \times h \times c)}$, or random patch subsets or pixel subsets from the raw patch is treated as a feature from a field. For data augmentation, we randomly add Gaussian noise, scale, crop, and rotate the field patches.

For both feature engineering approaches, we experimented with different data normalization techniques, including min-max scaling, standard scaling, and robust scaling.

3.2. Exploited Models

During development, classical machine learning approaches, such as Random Forest (RF), K-Nearest Neighbour (KNN) and eXtreme Gradient Boosting (XGBoost) regressors were investigated. Additionally, different neural network architectures were explored. Since the final solution is supposed to run on the Intuition-1 satellite, solutions that require low computational resources are of special interest.

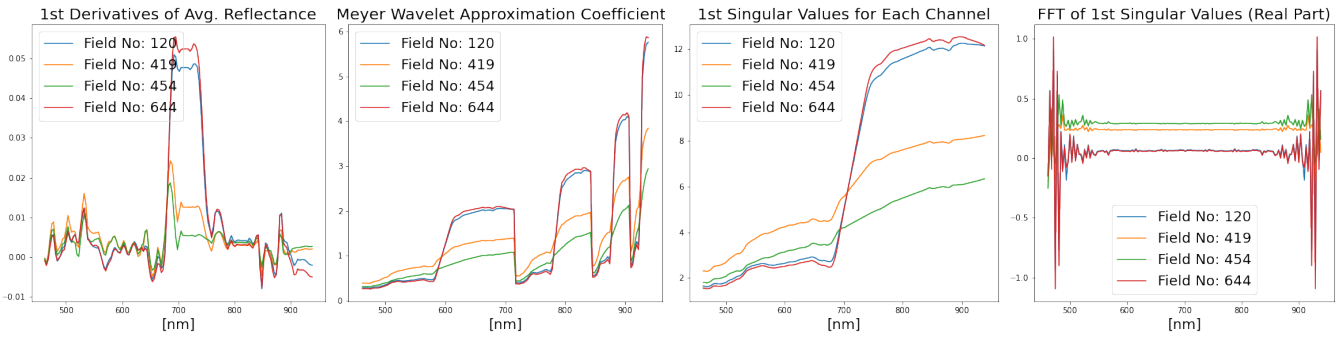

Fig. 4. Selected additional features derived from the agricultural field patches (explanations are given in Section 3.1).

3.2.1. Classical machine learning architectures

We used the RANDOMFORESTREGRESSOR (RF) and KNEIGHBORSREGRESSOR (KNN) implemented in the scikit-learn package [9], as well as XGBOOST [10] regressors. Since the latter does not support multiple-regression problems, the MULTIOUTPUTREGRESSOR, also from the scikit-learn package, was wrapped around the XGBOOST. For all model types, hyperparameter tuning was conducted using OPTUNA [11] with Bayesian optimization. However, we found the default parameters performed best and only changed the number of estimators to 1000. For all of our experiments, RFs perform better than the XGBoost and KNN algorithms.

3.2.2. Deep Neural networks

We experimented with various neural network architectures, including Transformers [12, 13], MobileNets [14], CapsuleNets [15], multilayer perceptrons, as well as autoencoder architectures [16] and attention networks, such as PSE+LTAE [17]. To exploit the pretrained weights of the networks, we experimented with several input modalities:

- channel-wise dimensional reduction from $(w \times h \times 150)$ to $(w \times h \times 3)$ via convolution operation and later feeding the samples into the pre-trained models;
- dropping the input layers of the pretrained networks mentioned above, and attaching our custom input layers that accept $(w \times h \times 150)$ dimensional samples;
- feeding each channel $(w \times h \times 1)$ of a sample into the 150 parallel and weight-sharing pretrained networks;
- expanding the input dimension as $(w \times h \times 150 \times 1)$ to exploit 3D neural networks such as CapsuleNets [15];
- flattening the input dimensions, or subsampling the input for exploiting 1D neural networks such as multilayer perceptrons or autoencoders.

Nonetheless, many of those trials performed worse than the RF, except for the Transformers with the pretrained weights (ImageNet-21k for Swin-T [12] and CLIP for ViT-L/14 [13]), when the channel-wise dimensional reduction operation was attached to it as an input layer.

For designing those experiments, we used the Keras framework with Tensorflow version 2.8.0 [18] and the Pytorch framework version 1.10.0 [19].

3.3. Evaluation Metrics

The evaluation metric takes into account the improvement upon the baseline of predicting the average of each soil parameter $(\mathrm{MSE_{bl}})$. For a given algorithm, it is calculated as:

$$\text{Score} = \frac{1}{4}\sum_{i=1}^{4}\frac{{\mathrm{MSE_{algo}}}^{(i)}}{{\mathrm{MSE_{bl}}}^{(i)}}, \quad \text{where:} \tag{1}$$

$${\mathrm{MSE_{algo}}}^{(i)} = \frac{1}{N}\sum_{j=1}^{N}(p_j^{(i)} - \hat{p}_j^{(i)})^2. \tag{2}$$

4. RESULTS AND DISCUSSION

Among our experiments listed in Section 3.2, the best performing ones on the public leaderboard of the challenge are:

- RF regression, by achieving 0.79476
- Swin Transformer (Swin-T), by achieving 0.80028
- Vision Transformer (ViT-L/14), by achieving 0.78799

The models show comparable performance, however, the RF is computationally more lightweight. Since this would be advantageous for running the model on the target Intuition-1 satelite, we selected the RF for further optimization.

As summarized in Table 1, the average of 5-fold cross validation on the training set with RF yields a validation score of 0.811. Note that while we improve on the baseline for all four soil parameters, the performance varies. Mg is predicted best (0.734), and $\mathrm{P_2O_5}$ is predicted worst (0.874).

For the RF regression, the feature importance can be determined. Figure 5 shows the derivatives of the average spectral reflectance contribute the most, followed by the features

derived from SVD and FFT. Figure 6 shows the importance of the spectral bands. The bands in the $650 - 670\,\mathrm{nm}$, and those exceeding $850\,\mathrm{nm}$ are considered to be most important.

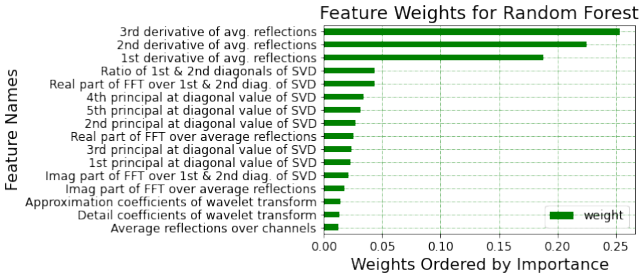

Fig. 5. Feature importance weights for RF regressor.

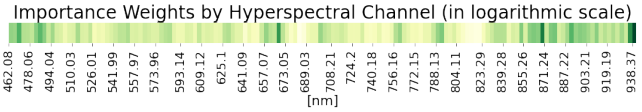

Fig. 6. Hyperspectral band importances for RF regressor.

In order to analyze if the data skewness affects performance, the prediction scores are reported for different patch sizes in Table 1. Thus, we observe that the smaller patches ($\leq 11 \times 11$ px) are the major source of the prediction error. This might stem from the higher variations in channel-aggregation due to a lower number of pixels. For mitigating this error source, alternative hyperparameter spaces and ML architectures were sought for smaller field patches. KNN regression with $k \geq 35$ improved performance on the smaller patches (from 0.985 to 0.915) but, on the other hand, it performs worse than RF on larger patches.

Therefore, a hybrid soil paramater estimator is proposed, combining KNN and RF regressors. Table 2 summarizes the performance of the hybrid model in which KNN predicts the soil parameters for smaller fields (mean edge length ≤ 11 px), while RF makes predictions for larger fields (mean edge length > 11 px). Thus, cross-validation performance on training set has been improved from 0.811 to 0.793. With this hybrid model, our team has preserved the top position in the public leaderboard by outperforming our former RF regressor (from 0.79476 to 0.78113).

Figure 7 shows density plots of the true vs the predicted soil parameters for the validation set. For all target variables, average values are predicted close to the $1:1$ line, while extreme values are hard to estimate correctly.

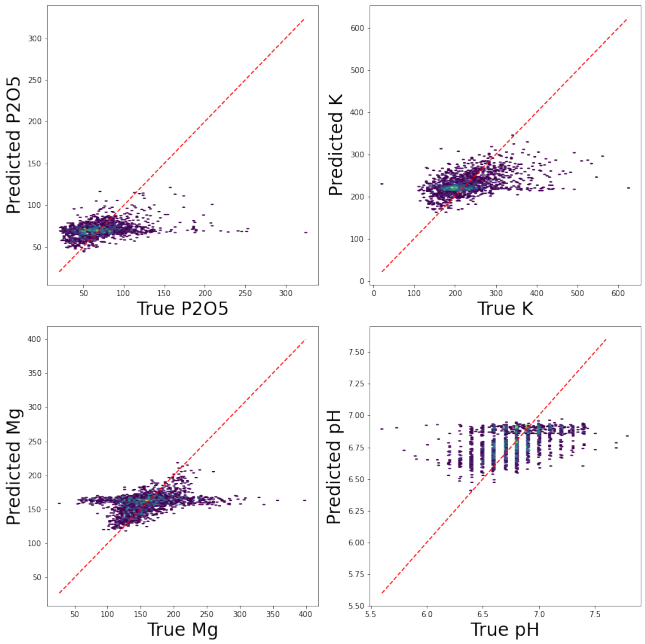

Fig. 7. Ground-truths vs predicted soil parameters.

Table 1. Cross validation with RF (the lower score is better).

Field Edge (pixel)	# of Fields	P2O5	K	Mg	pH	Average
0-11	650	1.050	1.008	1.019	0.866	0.985
11-40	94	0.491	0.581	0.539	0.981	0.648
40-50	326	0.724	0.754	0.416	0.777	0.668
50-100	138	0.683	0.660	0.618	0.749	0.677
100-110	113	0.911	0.591	0.398	0.764	0.665
110-120	118	0.883	0.812	0.614	0.731	0.760
120-130	132	0.895	0.776	0.644	0.656	0.742
130+	161	0.808	0.761	0.842	0.790	0.801
Entire Fields	**1732**	**0.874**	**0.828**	**0.734**	**0.807**	**0.811**
Public Leaderboard Score on the Test Set				**0.79476**		

Table 2. Cross validation with hybrid regressor (RF + KNN).

Field Edge (pixel)	Model	P2O5	K	Mg	pH	Average
0-11	KNN	1.002	0.953	0.993	0.710	0.915
11+	RF	0.766	0.720	0.564	0.772	0.706
Entire Fields	**Hybrid**	**0.855**	**0.807**	**0.725**	**0.749**	**0.793**
Public Leaderboard Score on the Test Set				**0.78113**		

5. CONCLUSION AND FUTURE WORK

In this paper, we demonstrated our solution to the AI4EO HYPERVIEW challenge which seeks for the most efficient approach to predict soil parameters (K, Mg, P_2O_5, pH). With comprehensive feature engineering, and by building a hybrid solution based on the fusion of KNN and RF regression models, we achieved 21.9% improvement compared to the baseline, and preserved the leadership so far. In the future, we will select features and train models individually for the four soil parameters to optimize performance. Besides, we will conduct further experiments with novel architectures tailored to the provided challenge dataset.

6. ACKNOWLEDGMENTS

We thank Lichao Mou for helpful discussions. This work was supported by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI [grant number: ZT-I-PF-5-01] and on the HAICORE@FZJ partition.

7. REFERENCES

[1] Aaron E. Maxwell, Timothy A. Warner, and Fang Fang, "Implementation of Machine-Learning Classification in Remote Sensing: An Applied Review," *International Journal of Remote Sensing*, vol. 39, no. 9, pp. 2784–2817, 2018.

[2] Yeshanbele Alebele, Wenhui Wang, Weiguo Yu, Xue Zhang, Xia Yao, Yongchao Tian, Yan Zhu, Weixing Cao, and Tao Cheng, "Estimation of Crop Yield From Combined Optical and SAR Imagery Using Gaussian Kernel Regression," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 10520–10534, 2021.

[3] Simon Schreiner, Dubravko Culibrk, Michele Bandecchi, Wolfgang Gross, and Wolfgang Middelmann, "Soil Monitoring for Precision Farming using Hyperspectral Remote Sensing and Soil Sensors," *at-Automatisierungstechnik*, vol. 69, no. 4, pp. 325–335, 2021.

[4] Liang Zhong, Xi Guo, Zhe Xu, and Meng Ding, "Soil Properties: Their Prediction and Feature Extraction from the LUCAS Spectral Library using Deep Convolutional Neural Networks," *Geoderma*, vol. 402, pp. 115366, 2021.

[5] European Comission, "Agriculture and the Green Deal," 2022, https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal/.

[6] Annekatrien Debien, Mauro Casaburi, Grega Milcinski, and Marcello Maranesi, "ESA's AI4EO Initiative: Bridging the Gap Between the AI and Earth Observation Communities," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 251–253.

[7] AI4EO Platform, "AI4EO Hyperview Challenge," https://platform.ai4eo.eu/seeing-beyond-the-visible.

[8] Daniel TL Lee and Akio Yamamoto, "Wavelet Analysis: Theory and Applications," *Hewlett Packard Journal*, vol. 45, pp. 44–44, 1994.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[10] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2016, KDD '16, pp. 785–794, ACM.

[11] Optuna, "Optuna Optimization," 2022, https://optuna.org/.

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012–10022.

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., "Learning Transferable Visual Models From Natural Language Supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[14] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.

[15] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton, "Dynamic Routing between Capsules," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] Ridvan Salih Kuzu, Emanuele Maiorana, and Patrizio Campisi, "Vein-based Biometric Verification using Densely-connected Convolutional Autoencoder," *IEEE Signal Processing Letters*, vol. 27, pp. 1869–1873, 2020.

[17] Vivien Sainte Fare Garnot and Loic Landrieu, "Lightweight Temporal Self-attention for Classifying Satellite Images Time Series," in *International Workshop on Advanced Analytics and Learning on Temporal Data*. Springer, 2020, pp. 171–181.

[18] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: A System for Large-scale Machine Learning," in *12th Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "Pytorch: An Imperative Style, High-performance Deep Learning Library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.