Improving Out-of-Distribution Generalization in Cosmological Surveys

Bella Zhang
bz2428@nyu.edu

Cindy Luo
kl3108@nyu.edu

Yuwen Shen
ys3344@nyu.edu

Jingyue Huang
jh8522@nyu.edu

Abstract

This study explores the application of the Centered Kernel Alignment (CKA) metric in improving out-of-distribution (OOD) generalization in Convolutional Neural Networks (CNNs) for cosmological simulations. We propose a novel loss function incorporating CKA to encourage the learning of diverse layer representations, thereby enhancing model generalization. Our findings demonstrate that CKA-focused training improves model performance on both in-distribution (ID) and OOD datasets. Additionally, we employ CKA to inform the pruning of superfluous layers in CNNs, resulting in streamlined yet effective architectures. We further integrate domain adaptation techniques, observing that models with greater adaptability to data distribution shifts also exhibit more varied layer representations, aligning with the CKA metric. However, challenges remain in optimizing the CKA loss, particularly its computational efficiency and sensitivity to learning rates.

1 Introduction

In astronomy, it is a key challenge to extract the maximum amount of cosmological information from astronomical simulations, which describe the properties of our Universe. For the cases that traditional statistic methods can hardly address, researchers have made some progress in applying deep learning techniques in the extraction[1]. However, models trained on data from one simulation show a drop in performance when tested on another, due to their differences in the subgrid physics implementation and numerical approximations. In this project, we aim to address the out-of-distribution challenge by learning domain-invariant representations. We explore the potential of a performance-related similarity metrics on learned representations, named Centered Kernel Alignment (CKA), for both model training and model design. Specifically, we proposed a new loss function that incentivizes the learning of more distributed layer representations. The experimental results demonstrate the effectiveness of CKA loss on improving model generalization ability when evaluating on a toy dataset. We also refined the model architecture by removing unnecessary layers, inspired by the clusters in the CKA matrix. The CKA matrices calculated on models with domain adaptation methods further confirm the correlations between model performance and CKA metric, aligning with previous works.

2 Related Work

Performance-Related Similarity Metric CKA is a similarity measure of neural network representations, first introduced by [2]. Recent research has shown that when pre-trained CNNs are assessed using both in-distribution (ID) and out-of-distribution (OOD) samples from the CAMELS Multifield Dataset, there is a notable link between reduced test-time accuracy and greater similarity across

Preprint. Under review.

different model layers, as indicated by the non-diagonal entries in the CKA matrix [3]. Thus, this insight has led us to develop a similarity-focused approach as our model training strategy. Our goal is to minimize the non-diagonal values in the CKA matrix, thereby promoting the development of more diverse and distributed representations across the model's layers.

Domain Adaptation Techniques To utilize labeled data in a relevant source domain to execute tasks in a target domain, various domain adaptation (DA) methods have been proposed to solve the domain shift between two domains[4]. In astronomy, it has been shown that domain adaptation techniques can substantially improve model performance in cross-dataset applications when applying to different types of deep learning models[5–10]. Specifically, DA-GNNs[11] applied discrepancy-based method Maximum Mean Discrepancy (MMD)[12] to Graph Neural Networks and examined its generalization capabilities on cosmological simulations. In this project, We adopted MMD and domain-adversarial neural network (DaNN)[13], a kind of adversarial-based DA techniques, to Convolutional Neural Networks, addressing the out-of-distribution challenges.

3 Problem Definition and Methodology

3.1 Model Training

Our goal is to encourage different layers of the model to acquire distributed information from dataset. To achieve this, we hypothesize that as model performance improves, the CKA matrix should resemble the identity matrix. Consequently, we penalize the off-diagonal elements of the CKA matrix.

3.1.1 Model Architecture and Data

To implement the CKA loss, we applied it to a half-trained model for efficiency and to leverage self-supervised properties of CKA. This approach also circumvents potential accuracy drops or overfitting from continued training with the original loss function. Our goal is to assess CKA's impact on performance enhancement. We used a 10-layer CNN with seven convolutional and three fully-connected layers, featuring ReLU and max-pooling at specified intervals. CKA calculations utilize outputs from ReLU and linear transformations for their respective layer types.

We utilized the MNIST dataset for training, with in-distribution testing on original data and OOD testing on MNIST with superimposed white noise, to evaluate model performance under training.

3.1.2 Methodology

For the half-trained model, we utilized the negative log likelihood loss function (NLL) as the loss function, achieving an accuracy of 91.21% on the in-distribution test dataset and 87.43% on out-of-distribution samples. The kernel of the CKA loss lies in measuring the difference between the CKA matrix and the identity matrix. Our loss function is based on three aspects:

CKA Metric Let $X \in \mathbb{R}^{m\times p_1}, Y \in \mathbb{R}^{m\times p_2}$ contains representations from two layers, CKA matrix is calculated based on HSIC(Hilbert-Schmidt Independence Criterion):

$$K = XX^T,\ L = YY^T,\ H = I_m - \frac{1}{m}11^T,\ HSIC_0(K,L) = \frac{vec(HKH)vec(HLH)}{(m-1)^2}$$

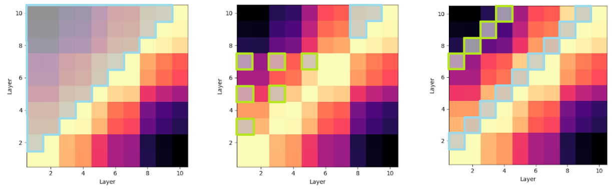
$$CKA(K,L) = \frac{HSIC_0(K,L)}{\sqrt{HSIC_0(K,K)HSIC_0(L,L)}}$$

For computational efficiency, we used the linear Kernel CKA, computed using small batch data rather than the entire dataset.

Regions for loss calculation As the CKA matrix is symmetric with 1 in the diagonal, we focused on the upper triangle part for loss calculation. We explored three different region selections: the entire upper triangle part, small upper triangles for selected layers, and k-th diagonals.

Weights of CKA loss term We combined the CKA loss term with the original NLL loss as the equation below shows. When $\alpha = 1$, the model is trained solely with the CKA loss term.

$$L_{new}(output, target) = \alpha L_{cka}(output) + (1-\alpha)NLL(output, target)$$



(a) Upper triangle (b) Selected layers (c) k-th diagonal

Figure 1: Examples of three types of regions for CKA loss calculation

3.2 Model Design

With the assumption that enhanced model performance correlates with highly dissimilar layers, our goal is to optimize the model architecture based on the computed CKA matrices. To further address the out-of-distribution challenge, we also aim to explore other approaches like introducing a pre-training model and applying domain adaptation techniques.

3.2.1 Data

The data used in this section is from the CAMELS Multifield Dataset (CMD)[14], featuring 2D maps generated from diverse cosmological simulations. We focus on IllustrisTNG and SIMBA simulation suites, particularly utilizing the LH set with 1000 simulations. Each simulation has unique values for two cosmological parameters (Ω_m, σ_8) and four astrophysical parameters ($A_{SN1}, A_{SN2}, A_{AGN1}, A_{AGN2}$). With 13 available physical fields, our project centers on the gas temperature field, comprising 15,000 2D maps in total. Figure 2 shows two examples of gas temperature maps when running two simulations with same random seed and initial parameters. We could also observe a significant distribution shift between maps of IllustrisTNG and SIMBA on the right part of the figure. Our goal is to estimate six parameters by training corresponding maps, considering IllustrisTNG simulations as in-distribution samples and SIMBA as out-of-distribution samples.

For data pre-processing, we normalized the values of all six parameters. We also re-scaled and normalized all maps, augmenting them through rotation and flipping. The data was then split into training, validation, and test sets in an 18:1:1 ratio, ensuring that all maps from the same simulation are grouped into one set.

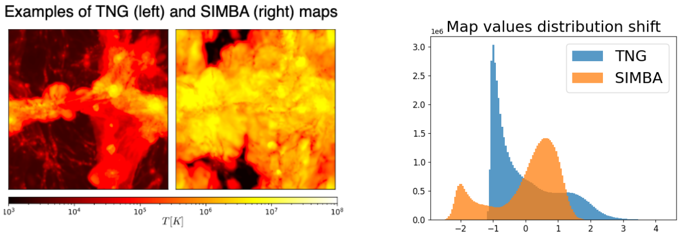


Figure 2: Examples of gas temperature fields (left) and distribution shift of map values (right)

3.2.2 Methodology

CKA-informed Model Refinement Our baseline model is a 21-layer CNNs[1] with 19 convolution layers and 2 fully-connected layers. Informed by the clusters in its CKA matrices 4, we removed several highly similar convolution layers, resulting in 18-layer CNNs and 15-layer CNNs.

Pre-training Model We further experimented with a pre-trained VGG13[15] model, which consists of 13 convolution and fully connected layers. Trained on large-scale image datasets, this pre-trained model has the capability to extract hierarchical features, providing us with a robust foundation.

Domain Adaptation We applied two famous domain adaptation methods, Domain-Adversarial Neural Networks (DaNN)[13] and Maximum Mean Discrepancy (MMD)[12]. DaNN introduces a domain confusion loss generated by a domain discriminator to learn domain-invariant features. For source domain $\mathcal{D}_s$, target domain $\mathcal{D}_t$ and a discriminator D, the loss is defined as

$$\mathcal{L}_{DaNN}(\mathcal{D}_s, \mathcal{D}_t) = \mathbb{E}_{x_i^s \sim \mathcal{D}_s} log[D(f_i^s)] + \mathbb{E}_{x_j^t \sim \mathcal{D}_t} log[1 - D(f_j^t)]$$

MMD applies a kernel function to map the data into a reproducing kernel Hilbert space (RKHS). It calculates the distance between the means of source and target domains in a RKHS to make two domains as similar as possible. With a kernel function ϕ, its loss function can be written as

$$\mathcal{L}_{MMD}(\mathcal{D}_s, \mathcal{D}_t) = \|\mathbb{E}_{x_i^s \sim \mathcal{D}_s}\phi(x_i^s) - \mathbb{E}_{x_j^t \sim \mathcal{D}_t}\phi(x_j^t)\|^2$$

3.2.3 Loss Function and Metrics

We trained on maps generated from IllustrisTNG and test on maps from both IllustrisTNG and SIMBA. For CNN variants and VGG13, We used maps to predict the posterior mean μ and standard deviation σ of the parameters and a modified MSE loss incorporating both was computed. Taking parameter Ω_m as an example, the loss is

$$\mathcal{L}(\mu, \sigma) = log(\sum_{i \in batch} (\Omega_{m,i} - \mu_i)^2) + log(\sum_{i \in batch} ((\Omega_{m,i} - \mu_i)^2) - \sigma_i^2)^2),$$

where $\Omega_{m,i}$ is the true parameter for i-th example. For domain adaptation methods, we used VGG13 as the backbone and minimized a weighted combination of the modified MSE loss and the domain adaptation loss, which was $\mathcal{L} = \mathcal{L}(\mu, \sigma) + \lambda \mathcal{L}_{DaNN/MMD}(\mathcal{D}_s, \mathcal{D}_t)$, where λ is a tradeoff that controls the weight of the domain adaptation loss. For assessment, we used the relative error ($RE = \delta\Omega_m/\Omega_m$) and the coefficient of determination (R^2). Additionally, we analyzed the generated CKA matrices to compare layer similarities and guide model design.

4 Experimental Evaluation and Results

4.1 Model Training

We conducted experiments with various loss configurations and evaluated the performance of the resulting models on both in-distribution (ID) and out-of-distribution (OOD) datasets. The table below presents the optimal training results for each configuration. Since the L1 and L2 norms yielded comparable outcomes, we have only included results from L1 norm configurations for brevity. Furthermore, no significant differences were detected whether using the entire upper triangle or just the first diagonal of the matrix. Consequently, given that the selection of layers can affect the number of diagonals considered, we opted to incorporate all diagonals in the computation of the CKA matrix.

	Model	ID Samples			OOD Samples		
		NLL Loss	**CKA Loss**	**Acc**	**NLL Loss**	**CKA Loss**	**Acc**
Loss Configs	Initial model	17.19	35.77	91.21%	24.33	35.84	87.43%
	CKA	17.20	35.71	91.22%	23.94	35.73	87.29%
	CKA_selected	17.24	35.75	**91.25%**	23.35	35.79	**88.19%**
	CKA + NLL	17.24	35.62	91.23%	23.72	35.63	87.42%

Table 1: Training result with different CKA-infused loss configurations.

Performance Enhancements on ID and OOD Data
As shown in Table 1, models trained with CKA loss from selected layers exhibited performance gains on both in-distribution (ID) and out-of-distribution (OOD) test sets. In contrast, employing CKA loss across all 10 layers or in conjunction with NLL loss did not result in comparable improvements. In particular, the presented training results for CKA_selected is based on using the CKA loss from all three fully-connected layers—layers 8, 9, and 10. Despite modest advances on ID data, the collective enhancements on both datasets indicate that our training approach is on the correct trajectory.

Focused Training with CKA_selected
By delving into the training trajectories of CKA_selected in Figure 3, we could observe that the CKA loss calculated from all 10 layers also effectively decreased during the course of training, signifying effective training with our custom-defined loss function. Key observations include:

- **Trade-Off Between ID & OOD Gains** Training that substantially improved ID performance tended to have a lesser impact on OOD enhancement.
- **Layer-Specific Effects** Utilizing CKA loss solely from convolutional layers improved ID performance, while CKA loss from fully-connected layers was more beneficial for OOD results.

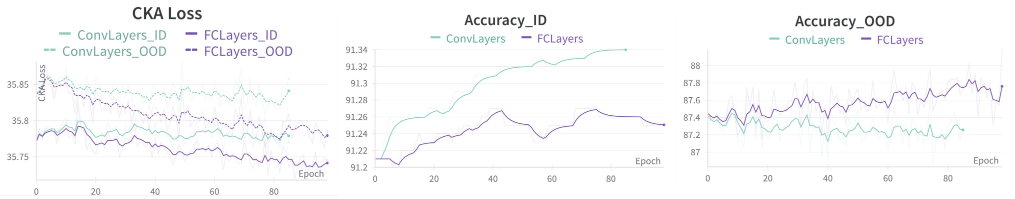


Figure 3: Loss and accuracy changes during the course of training using CKA_selected (from either the convolutional layers (1-6), or the fully-connected layers).

4.2 Model Design

CKA-informed Model Refinement

The CKA matrices of three CNN variants calculated on IllustrisTNG samples are shown in Figure 4. While the matrix of the baseline model, 21-layer CNNs, has two bright clusters within convolution layers, the matrices of two reduced-layer models show lower similarities within the last several layers, demonstrating the effectiveness of removing redundant information. Their parameter estimation performance on both in-distribution (IllustrisTNG) and out-of-distribution (SIMBA) samples is shown in Table 2 (left), where all three methods achieved comparable results on IllustrisTNG, but experienced significant performance drops on SIMBA after removing layers. These results confirm the potential of CKA matrices in informing the removal of unnecessary model layers while capturing enough information for cosmological parameters prediction. However, it is challenging to improve the generalization ability with fewer model parameters.

Methods	IllustrisTNG		SIMBA	
	$RE\downarrow$	$R^2\uparrow$	$RE\downarrow$	$R^2\uparrow$
CNN21	3.7%	0.985	**53.3%**	**-3.685**
CNN18	**3.4%**	**0.996**	476.5%	-74.175
CNN15	**3.4%**	0.990	361.6%	-41.735

Methods	IllustrisTNG		SIMBA	
	$RE\downarrow$	$R^2\uparrow$	$RE\downarrow$	$R^2\uparrow$
CNN21	**2.4%**	**0.995**	93.2%	-3.685
VGG13	5.9%	0.979	35.7%	0.384
VGG13+DaNN	6.3%	0.958	33.4%	0.470
VGG13+MMD	9.4%	0.939	**25.7%**	**0.598**

Table 2: Prediction and generalization performance of (1) three CNN variants designed according to CKA matrices, predicting on 6 parameters (left); (2) pre-training models and domain adaptation methods, predicting on 1 parameter (right).

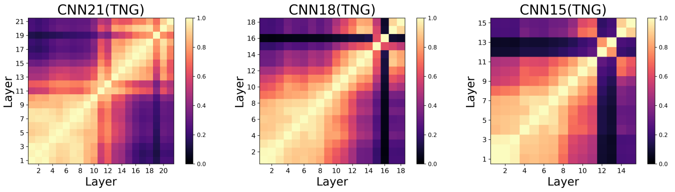


Figure 4: CKA matrices of three CNN variants calculated on IllustrisTNG samples

Domain Adaptation Methods

The performance of the pre-training model VGG13 and its combination with two domain adaptation methods, i.e., VGG13+DaNN and VGG13+MMD, are shown in Table 2 (right). With more visual information embedded during pre-training, VGG13 achieves significant improvements on both IllustrisTNG and SIMBA samples after fine-tuning on only IllustrisTNG samples. Moreover, domain adaptation methods are beneficial to the learning of domain-invariant representations, achieving lower relative errors and higher R^2 values. A trade-off between performance on in-distribution and out-of-distribution samples is observed, with more sacrifice of the performance on IllustrisTNG if having better generalization performance on SIMBA.

Figure 5 shows the CKA matrices calculated on both two simulations. Among all methods, CKA matrices for IllustrisTNG show less similarity along the non-diagonal than those for SIMBA, indicating strong redundancy between layers for SIMBA and thus poor performance. Although achieving best performance on IllustrisTNG, non-robust CNN21 has higher off-diagonal values, while robust methods such as those with domain adaptation have lower off-diagonal values. These observations align with the assumptions that model accuracy and robustness are related to representation similarities.

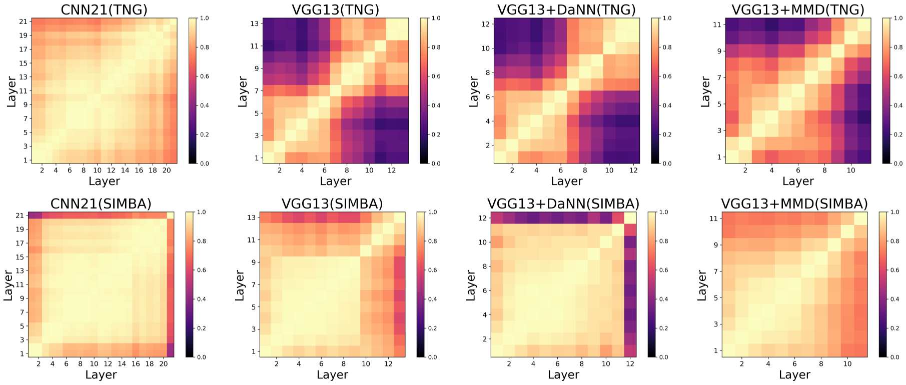


Figure 5: CKA matrices of VGG13 and domain adaptation methods

4.3 Discussion

Our study demonstrates a clear association between distributed layer representations, as quantified by the CKA metric, and overall model performance. Training focused on the CKA metric improved both ID and OOD outcomes, highlighting its potential to mitigate the OOD problem inherent in CNNs for cosmological studies. Additionally, the consistent performance of IllustrisTNG models following the pruning of non-essential layers, based on CKA insights, validates its utility in refining model architectures. The application of domain adaptation methods has further emphasized CKA's significance; models that were more adaptable to shifts in data distribution from IllustrisTNG to SIMBA showcased more varied layer representations, solidifying this metric's role in promoting model generalization. However, CKA loss lags behind NLL loss in training efficiency and is highly learning rate sensitive, with slight changes leading to varied outcomes. Future work will fine-tune this custom loss function and explore its application to CMD and complex models, areas not yet investigated due to time limitations.

5 Conclusions

Our study in CKA similarity matrix has provided valuable insights into enhancing model generalization ability. The proposed CKA loss achieved performance improvements for both in-distribution and out-of-distribution samples when incorporating with selected layers. The model design results highlight the distinctive role of CKA in evaluating model robustness and information redundancy.

Future Work The future endeavors on further investigating the OOD challenge with CKA metric including: (1) Incorporating CKA loss into more complex architectures & datasets to address real-world problems, such as the information extraction of cosmological simulations. (2) Exploring the potential of CKA metric in enhancing the interpretability of hyperparameter tuning, providing valuable insights into the model optimization process. This multifaceted approach ensures a holistic exploration of CKA's capabilities in advancing the field of domain adaptation.

Lessons Learned
One key insight from our study is the significant impact of training nuances, such as the choice between `float32` and `float64` data types or the scale of learning rates, on training outcomes. Furthermore, our research paves the way for innovative loss calculation methods, demonstrating that a model can learn from its own data representations, not just the data itself. The versatility and power of the CKA metric highlight the importance of applying it creatively. These practical lessons gleaned from our capstone project will undoubtedly enhance future endeavors.

Student Contributions
Model Training: Bella and Cindy. Model Design: Yuwen and Jingyue. Joint effort on final report.

Acknowledgments

This work was part of the Capstone Course for our Master's study. We extend our deep gratitude to our awesome mentors, Dr. Sultan Hassan and Dr. Naomi Saphra, for their invaluable mentorship and guidance. Their expertise was particularly instrumental in shaping the loss function definition and identifying an effective range of learning rates, which was pivotal in achieving the exciting training results concerning the CKA metric. We also express our sincere thanks to our course instructors for their constructive feedback and regular engagement with our project's progress. This research was made possible with the support of the Center for Data Science at New York University.

References

[1] Francisco Villaescusa-Navarro, Daniel Anglés-Alcázar, Shy Genel, David N. Spergel, Yin Li, Benjamin D. Wandelt, Andrina Nicola, Leander Thiele, Sultan Hassan, José Manuel Zorrilla Matilla, Desika Narayanan, Romeel Dave, and Mark Vogelsberger. Multifield cosmology with artificial intelligence. *CoRR*, abs/2109.09747, 2021.

[2] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited, 2019.

[3] Yash Gondhalekar, Sultan Hassan, Naomi Saphra, and Sambatra Andrianomena. Towards out-of-distribution generalization in large-scale astronomical surveys: robust networks learn similar representations, 2023.

[4] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[5] Stephon Alexander, Sergei Gleyzer, Hanna Parul, Pranath Reddy, Marcos Tidball, and Michael W. Toomey. Domain adaptation for simulation-based dark matter searches with strong gravitational lensing. *The Astrophysical Journal*, 954(1):28, aug 2023.

[6] Aleksandra Ćiprijanović, Diana Kafkes, Gregory Snyder, F Javier Sánchez, Gabriel Nathan Perdue, Kevin Pedro, Brian Nord, Sandeep Madireddy, and Stefan M Wild. Deepadversaries: examining the robustness of deep learning models for galaxy morphology classification. *Machine Learning: Science and Technology*, 3(3):035007, jul 2022.

[7] Ricardo Vilalta, Kinjal Dhar Gupta, Dainis Boumber, and Mikhail M. Meskhi. A general approach to domain adaptation with applications in astronomy. *CoRR*, abs/1812.08839, 2018.

[8] Aleksandra Ciprijanovic, Diana Kafkes, K. Downey, S. Jenkins, Gabriel N. Perdue, Sandeep Madireddy, T. Johnston, Gregory F. Snyder, and Brian Nord. Deepmerge II: building robust deep learning algorithms for merging galaxy identification across domains. *CoRR*, abs/2103.01373, 2021.

[9] Sankalp Gilda, Antoine de Mathelin, Sabine Bellstedt, and Guillaume Richard. Unsupervised domain adaptation for constraining star formation histories. *CoRR*, abs/2112.14072, 2021.

[10] Aleksandra Ciprijanovic, Ashia Lewis, Kevin Pedro, Sandeep Madireddy, Brian Nord, Gabriel N. Perdue, and Stefan M. Wild. Deepastrouda: semi-supervised universal domain adaptation for cross-survey galaxy morphology classification and anomaly detection. *Mach. Learn. Sci. Technol.*, 4(2):25013, 2023.

[11] Andrea Roncoli, Aleksandra Ciprijanovic, Maggie Voetberg, Francisco Villaescusa-Navarro, and Brian Nord. Domain adaptive graph neural networks for constraining cosmological parameters across multiple data sets. *CoRR*, abs/2311.01588, 2023.

[12] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alexander J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. In *Proceedings 14th International Conference on Intelligent Systems for Molecular Biology 2006, Fortaleza, Brazil, August 6-10, 2006*, pages 49–57, 2006.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. In Gabriela Csurka, editor, *Domain Adaptation in Computer Vision Applications*, Advances in Computer Vision and Pattern Recognition, pages 189–209. Springer, 2017.

[14] Francisco Villaescusa-Navarro, Shy Genel, Daniel Angles-Alcazar, Leander Thiele, Romeel Dave, Desika Narayanan, Andrina Nicola, Yin Li, Pablo Villanueva-Domingo, Benjamin D. Wandelt, David N. Spergel, Rachel S. Somerville, José Manuel Zorrilla Matilla, Faizan G. Mohammad, Sultan Hassan, Helen Shao, Digvijay Wadekar, Michael Eickenberg, Kaze W. K. Wong, Gabriella Contardo, Yongseok Jo, Emily Moser, Erwin T. Lau, Luis Fernando Machado Poletti Valle, Lucia A. Perez, Daisuke Nagai, Nicholas Battaglia, and Mark Vogelsberger. The CAMELS multifield dataset: Learning the universe's fundamental parameters with artificial intelligence. *CoRR*, abs/2109.10915, 2021.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.