# Logarithm-transform aided Gaussian Sampling for Few-Shot Learning

Vaibhav Ganatra
BITS Pilani, Goa
Goa, India
f20190010@goa.bits-pilani.ac.in

## Abstract

*Few-shot image classification has recently witnessed the rise of representation learning being utilised for models to adapt to new classes using only a few training examples. Therefore, the properties of the representations, such as their underlying probability distributions, assume vital importance. Representations sampled from Gaussian distributions have been used in recent works, [19] to train classifiers for few-shot classification. These methods rely on transforming the distributions of experimental data to approximate Gaussian distributions for their functioning. In this paper, we propose a novel Gaussian transform, that outperforms existing methods on transforming experimental data into Gaussian-like distributions. We then utilise this novel transformation for few-shot image classification and show significant gains in performance, while simultaneously reducing computation.*

## 1. Introduction

Learning from limited data, and adapting neural networks to unforeseen tasks has attracted signification attention in recent years. This is essential since the development of large datasets for supervised training requires significant costs, in terms of finances and the amount of human effort required. Considerable progress has been made in machine learning in the limited data regime. After the development of sophisticated optimization-based meta-learning techniques such as Model-agnostic-Meta-Learning [5], and metric-based meta-learning techniques, such as ProtoNets and MatchingNets[13, 17], there has been a recent shift towards representation learning for few-shot learning [14, 3, 15, 9]. For example, Tian *et al.* [14] leverage self-supervision and regularization for learning meaningful representations, and state that "a good embedding is all you need" for few-shot image classification. They propose a meta-free model as a new State-of-the-Art for few-shot image classification. Therefore, studying the properties (such as the underlying probability distributions), of the

learned representations is a useful pursuit in decoding their usefulness in few-shot learning.

Gaussian representations play an essential role in few-shot learning. For example, instead of using point prototypes for few-shot image classification, Lin *et al.* [8] propose modelling prototypes as multi-dimensional Gaussian distributions, and rectify these prototypes using mutual information maximization. Yang *et al.* [19] propose a distribution calibration mechanism to calibrate the representations of few-shot data and train classifiers by sampling Gaussian data around the calibrated representations. They make use of Tukey's Ladder of Powers [1] to transform the data so that the underlying distribution of the transformed data is approximately Gaussian.

In this paper, we propose a novel "data-to-Gaussian" transform that "Gaussianizes" data. (Gaussianization of data or inducing "normality" into data refers to transforming the data so that its underlying distribution is approximately normal). We demonstrate the utility of our transformation method by transforming a variety of data distributions (including data from experimental datasets) and show (through qualitative and quantitative evaluation) that our transform produces data following a distribution that is closer to a Gaussian distribution. Finally, we replace the distribution calibration mechanism proposed by Yang *et al.* [19] with our transform and show small yet consistent improvements in the classification accuracies, while sampling a significantly lesser amount of data.

Therefore, in this paper, we make the following contributions -

- We propose a novel transformation (called the Log-Tukey transformation) to induce "Gaussianization" within experimental data.

- We replace the distribution calibration mechanism proposed by Yang *et al.* [19] with the novel transform and devise an alternative algorithm for few-shot learning.

- We perform experiments on commonly known benchmark datasets, and show significant performance gains, while sampling 5x lesser datapoints.

The rest of the paper is organized as follows - Sec.2 details the related work, both in terms of making data more Gaussian, and representation learning for few-shot learning. Sec.3 explains the Log-Tukey transform for Gaussianization of data. In Sec.4, we first briefly discuss the distribution calibration method proposed by Yang *et al.* [19], since our work is heavily based on it. Next, we detail how we incorporate the Log-Tukey transform into the algorithmic setup proposed by them. Sec.5 shows experimental results of how the method proposed in Sec.4 outperforms the baseline and Sec. 6 concludes the paper.

## 2. Related Work

**Making data Gaussian-like** - A significant amount of prior work exists in transforming experimental data such that it follows Gaussian-like distributions. This is because of the usefulness of Gaussian data. Tukey's Ladders of Powers [1] are one of the prominent methods available for this task. Other transforms such as Log transforms and Inverse transforms are also used for this purpose. In addition to these, the Box-Cox [2] transform is another useful method that is widely used for transforming data and making it more Gaussian-like, but can only be used for positive values. The Yeo-Johnson transform [20] is a modification of the Box-Cox transform and can also be used for negative values.

**Representation Learning for Few-Shot Learning** - Tian *et al.* [14] debunk the need for complicated meta-learning methods for few-shot learning, and emphasize the utility of representations learned through a proxy task such as image classification. Mangla *et al.* [10] make use of self-supervision and regularization to learn meaningful structures in the representations of data. Luo *et al.* [9] approach few-shot learning by learning global representations whereas Tokmakov *et al.* [15] learn compositional representations for few-shot learning.

## 3. Log-Tukey Transform

As stated earlier, "Gaussianization" of data plays an essential role in few-shot learning. Hence, there exist multiple techniques to transform existing data such that it follows an approximate Gaussian distribution. Tukey's Ladder of Powers transform [1], as shown in Eq. 1, is by far one of the most popular methods used to Gaussianize data.

$$\hat{x} = \begin{cases} x^{\lambda}, & if \lambda \neq 0 \\ log(x), & if \lambda = 0 \end{cases} \quad (1)$$

Commonly, the value of $\lambda$ used is 0.5. However, only using exponents of the data is not immune to data skew and does not ensure maximum "normality"/"Gaussianization" in the data.

Fig. 1 shows the probability distribution function of a data sample drawn from an Exponential(0.5) distribution after transforming it using Tukey's Square Root transform and that of the corresponding Gaussian distribution. The peaks of the Tukey-Transformed data distribution and the corresponding Gaussian distribution are misaligned (The peak of the Tukey-transformed data is to the left of the peak of the Gaussian distribution). This happens because the exponential distribution is positively skewed and the Tukey-transform is unable to sufficiently shrink the large values, consequently, the distribution of the transformed data is still positively skewed. (Gaussian distributions do not have any skew) This is a common problem with the method, therefore, maximum "normality" is not ensured in the transformed data. A transform that is used to Gaussianize data must significantly reduce/remove the skew in a skewed distribution. Skew (and long tails) in the data is often overcome by using the logarithm function [7]. Consequently, in this paper, we introduce the Log-Tukey transform, which makes use of logarithm along with Tukey's Square-root transform, as shown in Eq. 2.

$$\hat{x} = log(\sqrt{x} + \epsilon + 1) \quad (2)$$

where $\epsilon$ is a small value to prevent the transformation from zeroing out the input. In our experiments, we use a value of 1e-4 for $\epsilon$. The $+1$ is added to ensure that the resulting values after the transform are positive.

The Log-Tukey transform removes the data skew and shrinks the data in a way that the underlying distribution is much closer to a Gaussian-distribution, with the same mean and variance, as shown in Fig. 2 (The peaks are horizontally aligned, and the edges lie very close). A quantitative comparison of the various "Gaussianization" methods is carried out, where the "closeness" of the distributions is
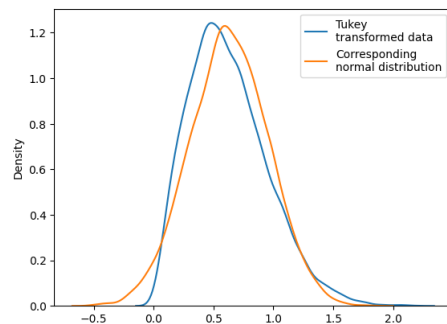


Figure 1. KDE plot (Probability distribution function) of Tukey-transformed data sampled from an Exponential(0.5) distribution. The "Corresponding normal distribution" shown in the figure is a Gaussian distribution with the same mean and variance as the Tukey-transformed data. Due to skew in the data, the transformed data is somewhat different from the corresponding normal distribution.
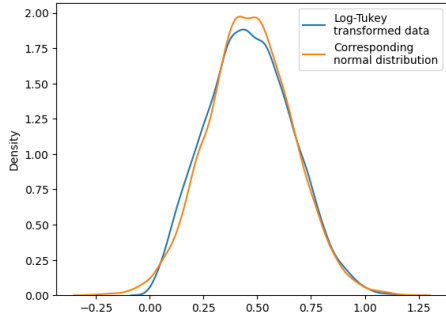
Figure 2. KDE plot (Probability distribution function) of Log-tukey-transformed data sampled from an Exponential(0.5) distribution. The "Corresponding normal distribution" shown in the figure is a Gaussian distribution with the same mean and variance as the Log-Tukey transformed data. The skew in the data is removed and the distribution is much closer to a Gaussian distribution with the same mean and variance.

| Distribution | Transform | Mean | Std Dev | Wasserstein Distance ↓ |
|---|---|---|---|---|
| Uni(0,1) | None | 0.5 | 0.289 | 0.0458 |
| Uni(0,1) | Tukey | 0.661 | 0.238 | 0.0402 |
| Uni(0,1) | Box-Cox | -0.598 | 0.384 | 0.0579 |
| Uni(0,1) | Yeo-Johnson | 0.456 | 0.258 | 0.0405 |
| Uni(0,1) | **Log-tukey** | 0.497 | 0.152 | **0.0308** |
| Exp(0.5) | None | 0.5 | 0.5 | 0.1482 |
| Exp(0.5) | Tukey | 0.625 | 0.326 | 0.0338 |
| Exp(0.5) | Box-Cox | -0.928 | 0.826 | 0.0227 |
| Exp(0.5) | Yeo-Johnson | 0.244 | 0.145 | 0.0169 |
| Exp(0.5) | **Log-Tukey** | 0.466 | 0.197 | **0.008** |
| Feature 0 | None | 5.843 | 0.825 | 0.0827 |
| from | Tukey | 2.411 | 0.17 | 0.0153 |
| Iris | Box-Cox | 1.549 | 0.109 | 0.01 |
| Dataset | Yeo-Johnson | 1.43 | 0.065 | 0.0057 |
| | **Log-Tukey** | 1.226 | 0.05 | **0.0042** |

Table 1. Quantitative comparison of "Gaussianization" methods. Uni stands for the continuous uniform distribution, and Exp stands for the exponential distribution.

estimated in terms of the Wasserstein distance [16], which can be interpreted as the minimum cost of transforming one of the distributions into the other. Here, we also consider the Box-Cox [2] and Yeo-Johnson [20] transforms, which are commonly used for Gaussianization of data. As is evident from Table. 1, the Log-Tukey transformation has the lowest Wasserstein distance in all cases, thereby showing the enhanced ability of the transformation in inducing "normality" within the data ( The table also includes data from the Iris dataset [6]). In deep learning, this ability is crucial in places where we do not know the ground truth distributions of data/weights, but assume that the underlying distribution is Gaussian. An example of this is shown in Sec.4 where we sample data for few-shot learning in the representation space, assuming it follows a multivariate Gaussian distribution.

# 4. Gaussian Sampling for Few-Shot Classification

In this section, we make use of the Log-Tukey transform for few-shot image classification.

## 4.1. Problem Setting

We adopt the typical few-shot classification problem formulation, where we have a labelled dataset $D = \{(x^i, y^i) | 1 \leq i \leq T\}$, where $x^i$ is a data sample, and $y^i$ is the corresponding label, $T$ denotes the size of the dataset. Each datapoint in the dataset is labelled as one of $|C|$ classes, where $C$ denotes the set of all classes. $C$ is partitioned into base classes $C_b$ and novel classes $C_n$, such that $C_b \cap C_n = \phi$ and $C_b \cup C_n = C$. The few-shot classification model is trained on the base classes, and the goal is to train the model in such a manner that it is able to adapt well to the novel classes, using only a few examples. Typ-

ically, abundant data is available for the base classes but only few samples are available for the novel classes. The generalization ability of the model is evaluated in terms of accuracy in N-way-K-shot tasks [17], where each task consists of $N$ novel classes sampled from $C_n$ and $K$ labeled samples from each of the $N$ classes. This few-shot labeled set available for model adaption is called the support set. The performance of the model is evaluated on the query set $Q$, which consists of $q$ test cases from each of the $N$ classes $Q = (x^i, y^i)_{i=N \cdot K+1}^{N \cdot K + Nq}$. Therefore, the average accuracy of the model on multiple N-way-K-shot tasks is used as an estimate of the model performance.

## 4.2. Distribution Calibration [19]

In accordance with the growing interest in using effective representations for few-shot learning , Yang *et al.* [19] propose a distribution calibration mechanism which uses statistics of the well-separated base-classes to calibrate the representations of novel classes, and make them separable in a few-shot setting, without over-fitting. They utilize the training method proposed by Mangla *et al.* [10] trained on the base classes, as a feature extractor $F$. After training, they record the classwise statistics - mean $\mu_{base} = \{\mu_i | 1 \leq i \leq |C_b|\}$ and covariances $\Sigma_{base} = \{\Sigma_i | 1 \leq i \leq |C_b|\}$ for the base-classes.

For each few-shot task, they calibrate the Tukey-transformed representation (obtained by using $F$ on the input image, followed by transformation using Eq. 1) of each image, using the base class statistics $\mu_{base}$ and $\Sigma_{base}$. Finally, they sample data around the calibrated representations, and train a linear classifier on data sampled around all points in the support set. They show that a logistic

regression/SVM classifier trained on the calibrated representations and sampled data outperform sophisticated optimization, metric and generation-based meta-learning methods. One point that must be noted is that they sample data around each point in the support set, therefore, if $p$ points are sampled for each image in an N-way-K-shot task, a total of $N$ x $K$ x $p$ points are sampled. Algorithm.1 outlines the distribution calibration mechanism proposed by Yang *et al.* [19].

---

**Algorithm 1** Algorithm for training a Few-Shot classifier using Distribution Calibration

---

**Require** : Support features $S = (x, y)_{i=1}^{NxK}$
**Require** : Base class statistics $\mu_{base}, \Sigma_{base}$
1: Transform $S$ with the Tukey transform (Eq. 1)
2: **for** $(x_i, y_i) \in S$ **do**
3:     Obtain calibrated mean $x_i'$ and covariance $\Sigma_i'$ for $x_i$ using the method proposed by Yang *et al.* [19]
4:     Sample multivariate Gaussian data using $x_i'$ and $\Sigma_i'$, and label them as $y_i$
5: **end for**
6: Train a linear classifier on the sampled + support set features

---

### 4.3. Gaussian Sampling

We build on the work of Yang *et al.* [19], since they utilize Tukey's Square Root transform. We make use of the 5-way-5-shot experimental setting. As we have seen in Sec. 3, the Tukey-transform alone is not optimal in inducing "normality" into the data. The sampled Gaussian data, therefore, is not as close to the ground truth representations of the novel classes, as possible and there is room for further correction.

Hence, we replace the Tukey-transform and distribution calibration steps with our Log-Tukey transformation on the representations of the novel classes in an attempt to make them more "Gaussian", following which we calculate classwise means and covariances using the images from the support set. Next, we sample data for each class around the mean and train a linear classifier on the sampled data. Therefore, if $p$ points are sampled around each mean, a total of $N$ x $p$ datapoints are sampled, $p$ datapoints for each of the $N$ classes. In an N-way-K-shot setting, this is $K$ times lesser than that of the distribution calibration proposed by Yang *et al.* [19]. Since we adopt a 5-shot setting, we sample 5x lesser data.

Thus, applying the Log-Tukey transform ensures that the data is more "Gaussian", therefore sampling Gaussian data around the class-mean generates more accurate representations of the novel class. This "Gaussian Sampling" of data from each class aids in few-shot learning and delivers a small yet consistent improvement in performance (as

shown in Sec. 5.3), while sampling 5x fewer data, thereby decreasing the computation. Algorithm. 2 shows the overall algorithm of training a few-shot classifier using Gaussian Sampling.

---

**Algorithm 2** Algorithm for training a Few-Shot classifier using Gaussian Sampling

---

**Require** : Support features $S = (x, y)_{i=1}^{NxK}$
1: Transform $S$ with the Log-Tukey transform (Eq. 2)
2: Calculate classwise-means $\mu$'s and covariance $\Sigma$'s from the transformed features
3: Sample multivariate Gaussian data using $\mu$'s and $\Sigma$'s for each class, label them with the corresponding class label
4: Train a linear classifier on the sampled + support set features

---

## 5. Experiments

### 5.1. Datasets

We validate our Gaussian Sampling method on the mini-ImageNet [11] and the CUB [18] datasets. A variety of classes including various animals and objects can be found in the miniImageNet dataset while CUB is a more fine-grained dataset that includes various species of birds. Datasets with different levels of granularity may have different distributions for their feature space. We show the validity of our sampling mechanism on both datasets

**miniImageNet** is derived from ILSVRC-12 dataset [12]. It has 100 diverse classes with 600 images per class, each of size 84 × 84 × 3. The data split used in the following experiments is as proposed by Ravi *et al.* [11], with 64 base classes, 16 validation classes, and 20 novel classes.

**CUB** is a fine-grained few-shot classification benchmark. It has a total of 11,788 images, each of size 84 × 84 × 3, of 200 different classes of birds. The dataset is split into 100 base classes, 50 validation classes, and 50 novel classes, following Chen *et al.*[4]

### 5.2. Implementation and Metrics

We follow the implementation provided by Yang *et al.* [19]. We use the method proposed by Mangla *et al.* [10] as the feature-extractor, trained on the base classes and evaluate its performance on the novel classes. We adopt the 5-way-5-shot setting for our experiments. The values of hyper-parameters are as per the implementation provided by Yang *et al.* [19]. We evaluate the performance in terms of classification accuracy on the query set. This is evaluated over a total of 500 tasks, in 5 runs of sets of 100 tasks each. Since the distribution calibration proposed by Yang *et al.* [19] already surpasses the performance of other existing methods, and since our method is heavily based on

| Trial | Without GS (%) | With GS (%) | Difference (%) |
|---|---|---|---|
| 1 | 82.733 | 83.053 | 0.320 |
| 2 | 83.053 | 84.067 | 1.014 |
| 3 | 84.067 | 84.177 | 0.11 |
| 4 | 83.987 | 84.24 | 0.253 |
| 5 | 84.187 | 84.973 | 0.786 |
| Avg | 83.605 | 84.102 | 0.497 |

Table 2. Accuracy on the query set of the MiniImageNet dataset. Each trial consists of 100 tasks, run with and without Gaussian Sampling (GS).

| Trial | Without GS (%) | With GS (%) | Difference (%) |
|---|---|---|---|
| 1 | 90.747 | 91.093 | 0.346 |
| 2 | 91.747 | 92.4 | 0.653 |
| 3 | 91.747 | 92.213 | 0.466 |
| 4 | 90.307 | 90.88 | 0.573 |
| 5 | 90.307 | 90.813 | 0.506 |
| Avg | 90.9707 | 91.48 | 0.509 |

Table 3. Accuracy on the query set of the CUB dataset. Each trial consists of 100 tasks, run with and without Gaussian Sampling (GS).

it, we only compare our method against theirs [19] [1]. We show the improvement in the performance across multiple runs for both the miniImageNet and CUB datasets.

### 5.3. Results

Table. 2 shows the accuracies of the distribution calibration mechanism [19] and the Gaussian sampling (ours) mechanism for the miniImageNet dataset. In all the trials, the average accuracy of all tasks is better with the Gaussian Sampling than without. Similar, results are observed for the CUB dataset, as seen in Table. 3. From the tables, it is clear that with the addition of a single log function, the accuracy can improve by ∼0.5%. This is a significant gain while using 5x lesser sampled data. All the results shown have been obtained after sampling 750 datapoints in total for the Gaussian sampling method, and 750 x 5 = 3750 datapoints for the distribution calibration method.

We also examine the effect of Gaussian Sampling with the variation of the number of datapoints sampled for each point in the support set. Fig. 3 shows the variation in average task accuracy for the miniImageNet dataset. Although the difference in accuracy is small, Gaussian sampling consistently outperforms distribution calibration [19]. Fig. 4 shows a similar trend in the variation in average accuracy for the CUB dataset with different amounts of sampled data.
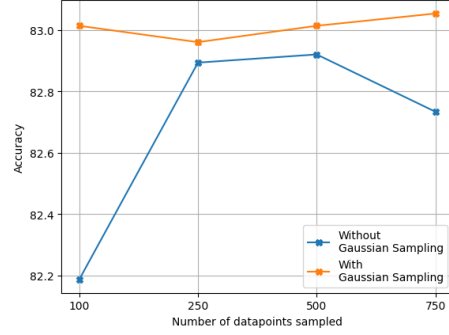
Figure 3. Variation in the accuracy with the number of sampled data per point in the support set, with/without Gaussian Sampling for the miniImageNet dataset. The number of datapoints sampled for the plot without Gaussian Sampling is 5x the value on the x-axis
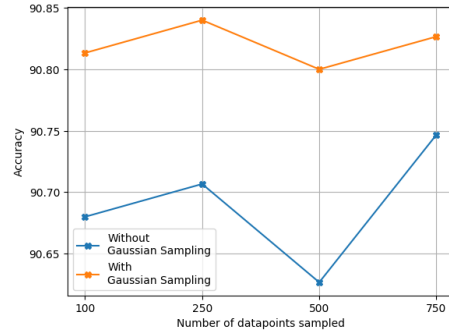


Figure 4. Variation in the accuracy with the number of sampled data per point in the support set, with/without Gaussian Sampling for the CUB dataset. The number of datapoints sampled for the plot without Gaussian Sampling is 5x the value on the x-axis

### 6. Conclusion

In this paper, we propose a new method to induce "normality" within experimental data, called the Log-Tukey transform. By transforming data sampled from various distributions, we show the effectiveness of this transform in making data more Gaussian-like as compared to the existing methods. Further, we employ this transform to sample Gaussian representations in a few-shot learning method, and show significant incremental gains while reducing the amount of computation. We also demonstrate the generality and utility of the method by conducting experiments on datasets of varied granularity, and with different amount of sampled data. A gain of ∼0.5% in accuracy by the addition of a single logarithm function seems like a good bargain. A possible direction for future work would be to examine the effectiveness of the Log-Tukey transform in other scenarios where Gaussian priors and sampling multi-variate Gaussians from experimental data are involved!

# References

[1] *Exploratory Data Analysis*, pages 192–194. Springer New York, New York, NY, 2008.

[2] G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964.

[3] Kuilin Chen and Chi-Guhn Lee. Meta-free few-shot learning via representation learning with weight averaging, 2022.

[4] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification, 2020.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.

[6] R. A. Fisher. Iris. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C56C76.

[7] Loet Leydesdorff and Stephen Bensman. Classification and powerlaws: The logarithmic transformation, 2009.

[8] Jinfu Lin, Junmin Shen, and Xiaojian He. Gaussian prototype rectification for few-shot image recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021.

[9] Tiange Luo, Aoxue Li, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations, 2019.

[10] Puneet Mangla, Mayank Singh, Abhishek Sinha, Nupur Kumari, Vineeth N Balasubramanian, and Balaji Krishnamurthy. Charting the right manifold: Manifold mixup for few-shot learning, 2020.

[11] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

[12] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.

[13] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017.

[14] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need?, 2020.

[15] Pavel Tokmakov, Yu-Xiong Wang, and Martial Hebert. Learning compositional representations for few-shot recognition, 2019.

[16] Cédric Villani. Optimal transport: Old and new. 2008.

[17] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[18] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[19] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration, 2021.

[20] In-Kwon Yeo and Richard A. Johnson. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, 12 2000.