UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Free Lunch For Few-Shot Learning: Distribution Calibration

Shuo Yang, Lu Liu, Min Xu

ICLR 2021

Presented by Chenpei Huang

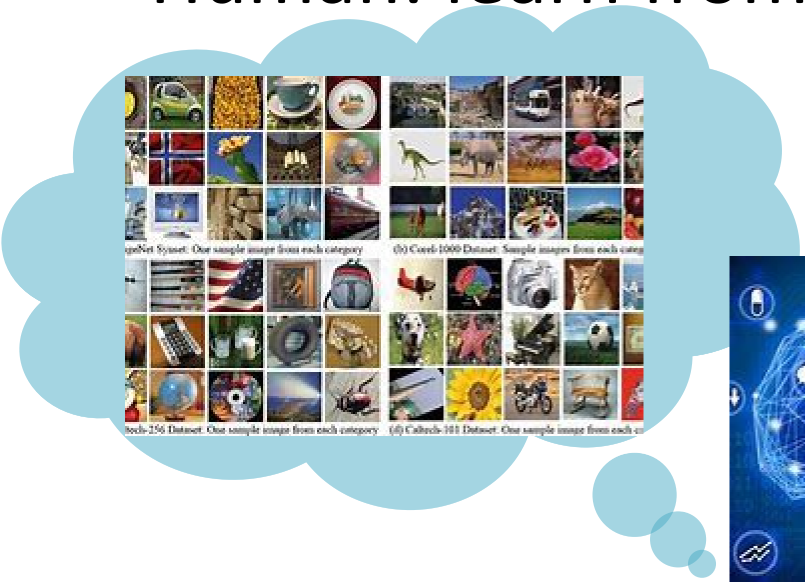
ANTS LAB

Outline

- Introduction
- Main Approach
- Experiments
- Conclusion

Background: Few-shot learning

- “Can machines think?”, by Alan Turing
- Machine: powerful device, advanced model, large dataset
- Human: learn from only one-shot



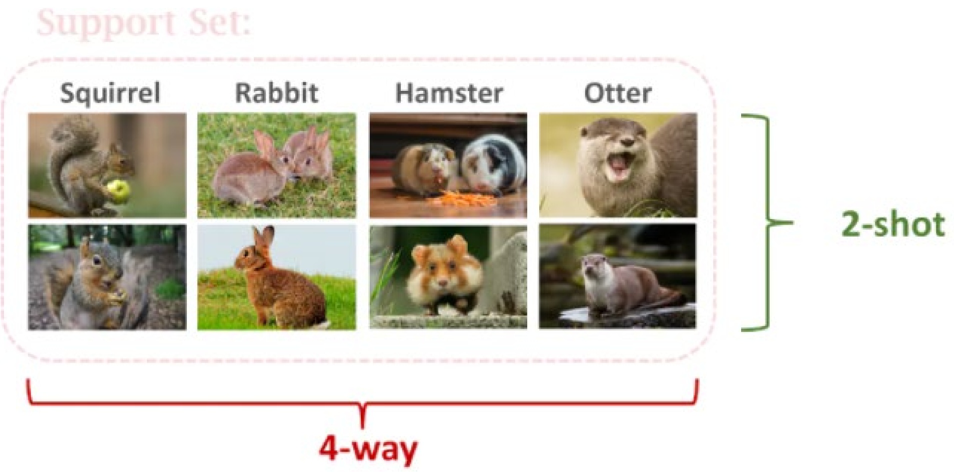
Prior knowledge

Background: Few-shot learning

- Dataset:
  1. (Big) base dataset – base classes
  2. (small) support dataset – novel classes

base classes C_b and *novel classes* C_n

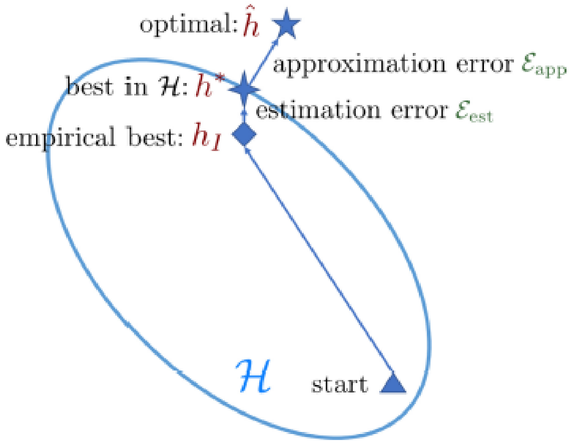
- N-way-K-shot: N classes, K samples in support dataset
- Query set: test samples in novel classes



$$\textit{support set } \mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N\times K}$$
$$\text{query set } \mathcal{Q} = \{(\boldsymbol{x}_i, y_i)\}_{i=N\times K+1}^{N\times K+N\times q}$$

Background: Few-shot learning

Few-shot problem



Few-shot solution

1. Model: constrain hypothesis space by prior knowledge
2. Algorithm: alter search strategy in hypothesis space by prior knowledge
3. Data: Augment training data by prior knowledge

Introduction

- Model trained with a wide coverage of data can generalize well.
- However, few training data makes the model overfit.
- Few-shot feature distribution needs calibration.

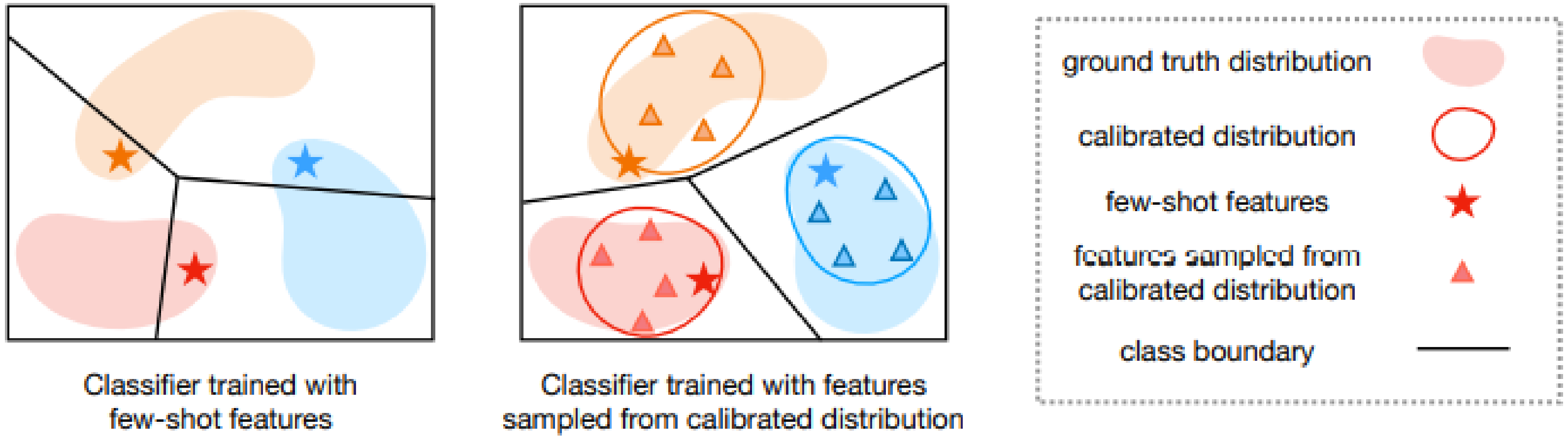


Figure 1: Training a classifier from few-shot features makes the classifier overfit to the few examples (Left). Classifier trained with features sampled from calibrated distribution has better generalization ability (Right).

Introduction

- What to generate: feature or data
    1. much lower dimensions => easier to calibrate
    2. Gaussian distribution

- How to generate augmented features

  From similar classes in the base dataset

  Estimate distribution => generate virtual features => supervise classification

	Arctic fox	
	mean sim	var sim
white wolf	97%	97%
malamute	85%	78%
lion	81%	70%
meerkat	78%	70%
jellyfish	46%	26%
orange	40%	19%
beer bottle	34%	11%

Outline

- Introduction
- Main Approach
- Experiments
- Conclusion

Main Approach

Main Algorithm

Algorithm 1 Training procedure for an N-way-K-shot task

Require: Support set features $\mathcal{S} = (\boldsymbol{x}_i, y)_{i=1}^{N \times K}$

Require: Base classes' statistics $\{\boldsymbol{\mu}_i\}_{i=1}^{|C_b|}, \{\boldsymbol{\Sigma}_i\}_{i=1}^{|C_b|}$

1: Transform $(\boldsymbol{x}_i)_{i=1}^{N \times K}$ with Tukey's Ladder of Powers as Equation 3

2: **for** $(\boldsymbol{x}_i, y_i) \in \mathcal{S}$ **do**

3: Calibrate the mean $\boldsymbol{\mu}'$ and the covariance $\boldsymbol{\Sigma}'$ for class y_i using $\boldsymbol{x}_i$ with Equation 6

4: Sample features for class y_i from the calibrated distribution as Equation 7

5: **end for**

6: Train a classifier using both support set features and all sampled features as Equation 8

① prepare base class mean/variance

② make them Gaussian like

③ For novel classes, calibrate distribution and generate samples

Main Approach

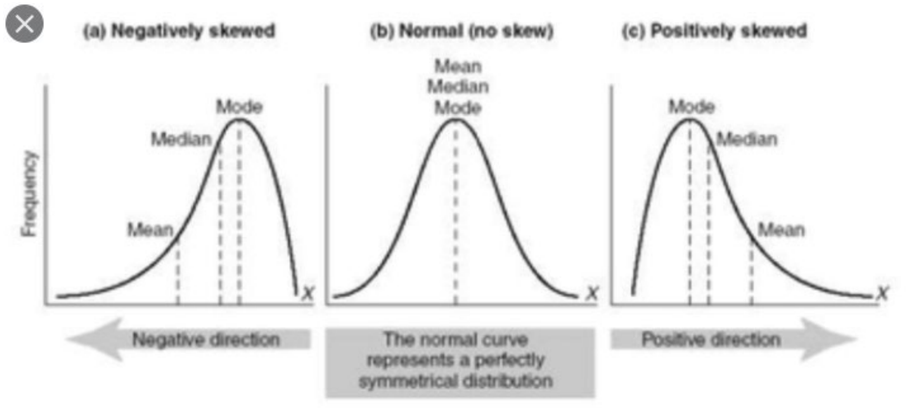
Base class statistics: calculate mean and variance of features

$$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^{n_i} \boldsymbol{x}_j}{n_i}, \qquad \boldsymbol{\Sigma}_i = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} \left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)^T.$$

Tukey's Ladder of Powers Transformation (if possibly skewed)

$$\tilde{\mathbf{x}} = \begin{cases} \boldsymbol{x}^{\lambda} & \text{if } \lambda \neq 0 \\ \log(\boldsymbol{x}) & \text{if } \lambda = 0 \end{cases}$$

Decreasing λ makes the distribution
less positively skewed and vice versa.



Examples of normal and skewed distributions

Main Approach

Calibration through statistics transfer

Find top-k similar base classes

$$\mathbb{S}_d = \{-\|\boldsymbol{\mu}_i - \tilde{\boldsymbol{x}}\|^2 \mid i \in C_b\},$$
$$\mathbb{S}_N = \{i \mid -\|\boldsymbol{\mu}_i - \tilde{\boldsymbol{x}}\|^2 \in topk(\mathbb{S}_d)\},$$

Calibrate to target Gaussian distribution

$$\boldsymbol{\mu}' = \frac{\sum_{i \in \mathbb{S}_N} \boldsymbol{\mu}_i + \tilde{\boldsymbol{x}}}{k+1}, \boldsymbol{\Sigma}' = \frac{\sum_{i \in \mathbb{S}_N} \boldsymbol{\Sigma}_i}{k} + \boxed{\alpha,}$$

Hyperparameter indicates dispersion

Main Approach

Leverage the calibrated distribution

Generate feature vectors with certain label

$$\mathbb{D}_y = \{(\boldsymbol{x}, y) | \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \forall (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in \mathbb{S}^y\}.$$

Train a task-specific classifier

$$\ell = \sum_{(\boldsymbol{x}, y) \sim \tilde{S} \cup \mathbb{D}_{y, y \in \mathcal{Y}^{\mathcal{T}}}} - \log \Pr(y | \boldsymbol{x}; \theta),$$

Outline

- Introduction
- Main Approach
- Experiments
- Conclusion

Experiment

Experiment Setup

Dataset: *mini*ImageNet, *tiered*ImageNet, CUB

Feature Extractor: WideResNet

Classifier: LR and SVM

Hyperparameter:

- Feature dimension 750
- Selected base class k=2
- Power transfer $\lambda = 0.5$
- $\alpha = 0.21, 0.21, 0.3$

Evaluation Metric

Top-1 accuracy on 5-way-1-shot and 5-way-5-shot settings

Experiment

Performance comparison to state-of-the-art

Methods	*mini*ImageNet 5way1shot	*mini*ImageNet 5way5shot	*CUB* 5way1shot	*CUB* 5way5shot
Optimization-based				
MAML (Finn et al. (2017))	48.70 ± 1.84	63.10 ± 0.92	50.45 ± 0.97	59.60 ± 0.84
Meta-SGD (Li et al. (2017))	50.47 ± 1.87	64.03 ± 0.94	53.34 ± 0.97	67.59 ± 0.82
LEO (Rusu et al. (2019))	61.76 ± 0.08	77.59 ± 0.12	-	-
E3BM (Liu et al. (2020c))	63.80 ± 0.40	80.29 ± 0.25	-	-
Metric-based				
Matching Net (Vinyals et al. (2016))	43.56 ± 0.84	55.31 ± 0.73	56.53 ± 0.99	63.54 ± 0.85
Prototypical Net (Snell et al. (2017))	54.16 ± 0.82	73.68 ± 0.65	72.99 ± 0.88	86.64 ± 0.51
Baseline++ (Chen et al. (2019a))	51.87 ± 0.77	75.68 ± 0.63	67.02 ± 0.90	83.58 ± 0.54
Variational Few-shot(Zhang et al. (2019))	61.23 ± 0.26	77.69 ± 0.17	-	-
Negative-Cosine(Liu et al. (2020a))	62.33 ± 0.82	80.94 ± 0.59	72.66 ± 0.85	89.40 ± 0.43
Generation-based				
MetaGAN (Zhang et al. (2018))	52.71 ± 0.64	68.63 ± 0.67	-	-
Delta-Encoder (Schwartz et al. (2018))	59.9	69.7	69.8	82.6
TriNet (Chen et al. (2019b))	58.12 ± 1.37	76.92 ± 0.69	69.61 ± 0.46	84.10 ± 0.35
Meta Variance Transfer (Park et al. (2020))	-	67.67 ± 0.70	-	80.33 ± 0.61
Maximum Likelihood with DC (Ours)	66.91 ± 0.17	80.74 ± 0.48	77.22 ± 0.14	89.58 ± 0.27
SVM with DC (Ours)	$\mathbf{67.31 \pm 0.83}$	$\mathbf{82.30 \pm 0.34}$	$\mathbf{79.49 \pm 0.33}$	$\mathbf{90.26 \pm 0.98}$
Logistic Regression with DC (Ours)	$\mathbf{68.57 \pm 0.55}$	$\mathbf{82.88 \pm 0.42}$	$\mathbf{79.56 \pm 0.87}$	$\mathbf{90.67 \pm 0.35}$

Keys:

1. Simple LR and SVM classifier, even MLE, outperform state-of-the-art
2. 10% surpass other generation-based method
3. Also compared to generation-based method, this method requires less training costs

Experiment

Visualization of generated samples

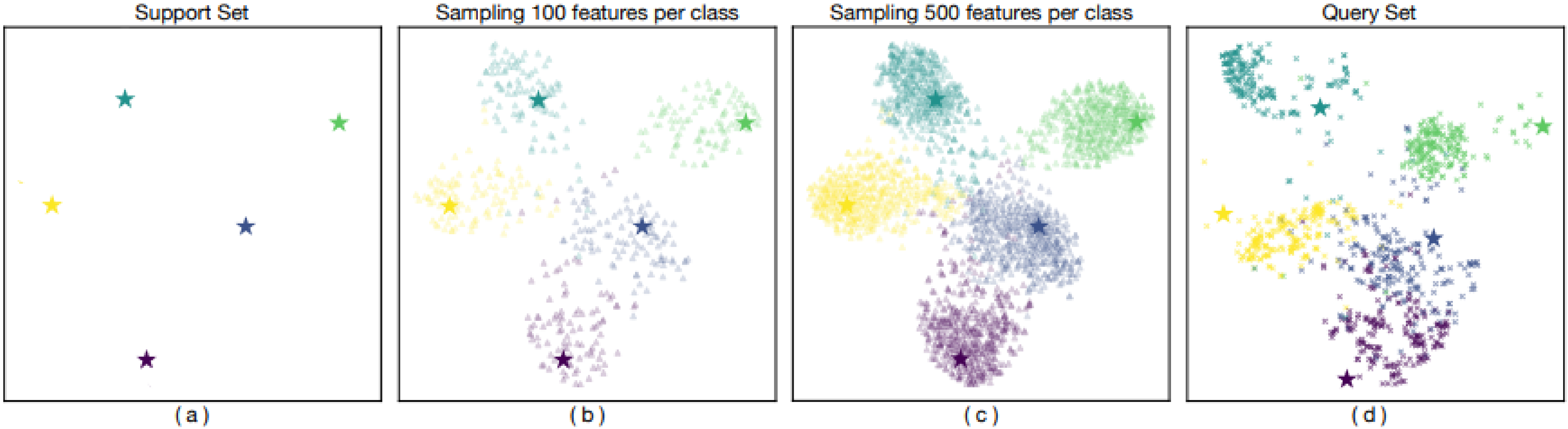


Figure 2: t-SNE visualization of our distribution estimation. Different colors represent different classes. '★' represents support set features, 'x' in figure (d) represents query set features, '▲' in figure (b)(c) represents generated features.

Experiment

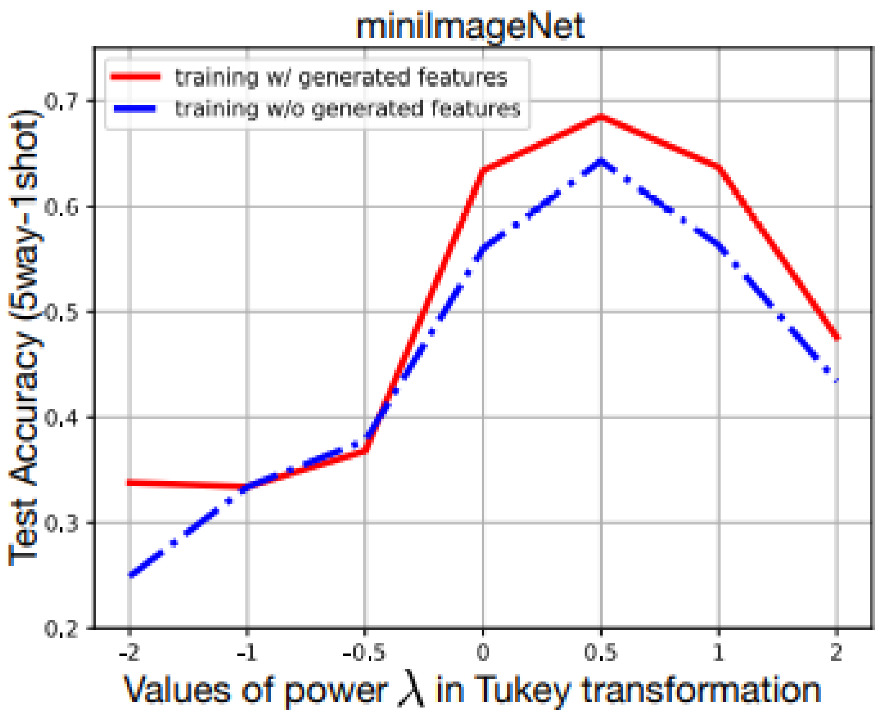
Apply DC on different backbones and baselines

Backbones	without DC	with DC
conv4 (Chen et al., 2019a)	42.11 ± 0.71	$\mathbf{54.62 \pm 0.64}$ **(↑ 12.51)**
conv6 (Chen et al., 2019a)	46.07 ± 0.26	$\mathbf{57.14 \pm 0.45}$ **(↑ 11.07)**
resnet18 (Chen et al., 2019a)	52.32 ± 0.82	$\mathbf{61.50 \pm 0.47}$ **(↑ 9.180)**
WRN28 (Mangla et al., 2020)	54.53 ± 0.56	$\mathbf{64.38 \pm 0.63}$ **(↑ 9.850)**
WRN28 + Rotation Loss (Mangla et al., 2020)	56.37 ± 0.68	$\mathbf{68.57 \pm 0.55}$ **(↑ 12.20)**

Method	without DC	with DC
Baseline (Chen et al., 2019a)	42.11 ± 0.71	$\mathbf{54.62 \pm 0.64}$ **(↑ 12.51)**
Baseline++ (Chen et al., 2019a)	48.24 ± 0.75	$\mathbf{61.24 \pm 0.37}$ **(↑ 13.00)**

Experiment

Role of Tukey's power transformation and feature generation



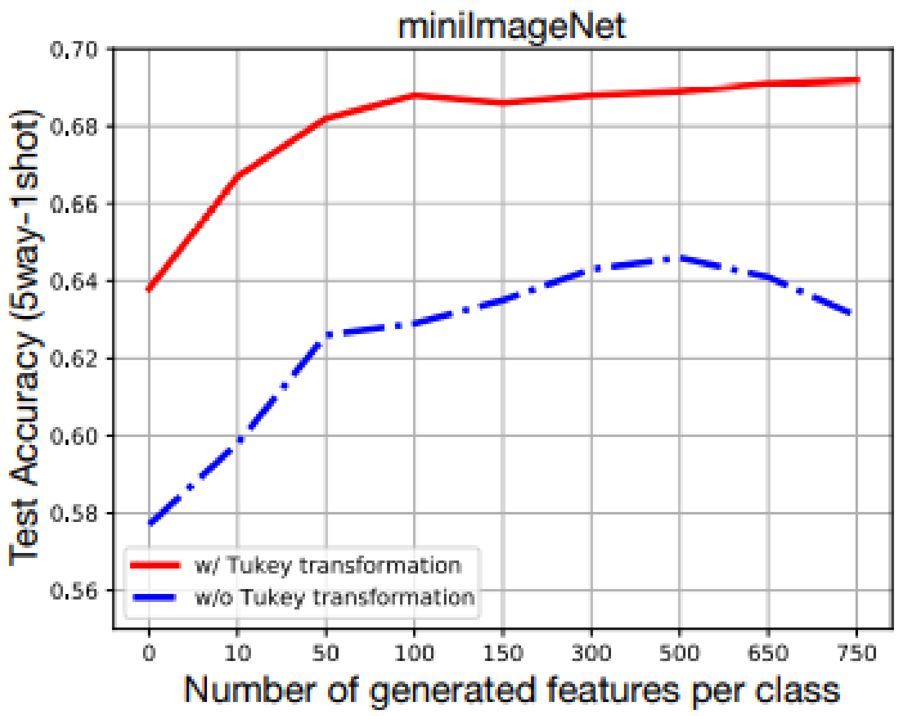


Figure 3: Left: Accuracy when increasing the power in Tukey's transformation when training with (red) or without (blue) the generated features. Right: Accuracy when increasing the number of generated features with the features are transformed by Tukey's transformation (red) and without Tukey's transformation (blue).

Experiment

Other Hyper-Parameters

Top-k base classes selection

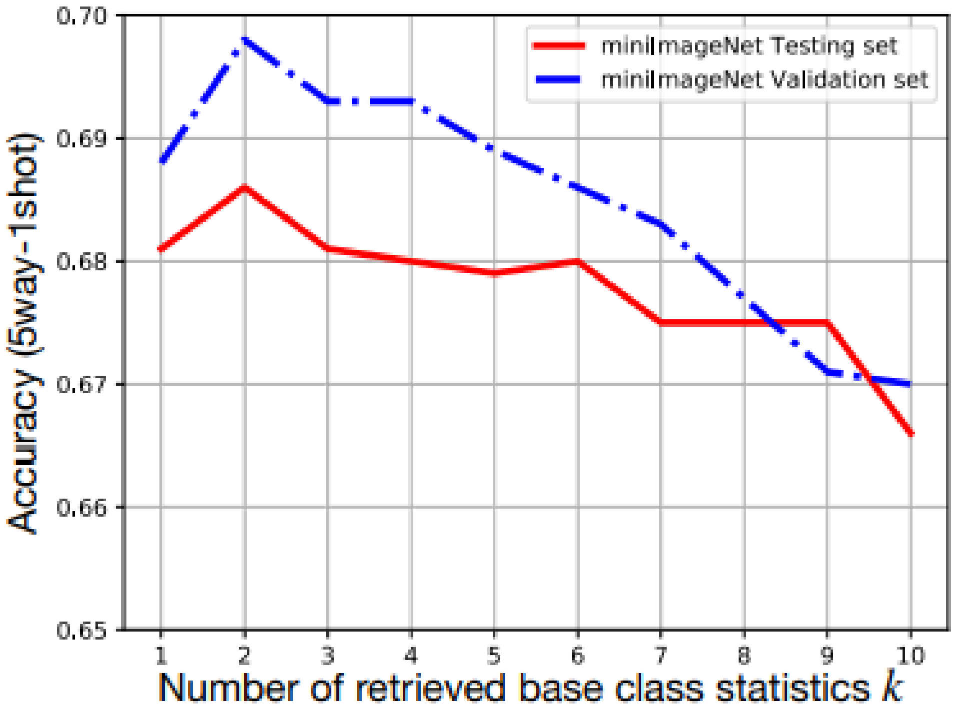


Figure 4: The effect of different values of k.

Dispersion of covariance matrix

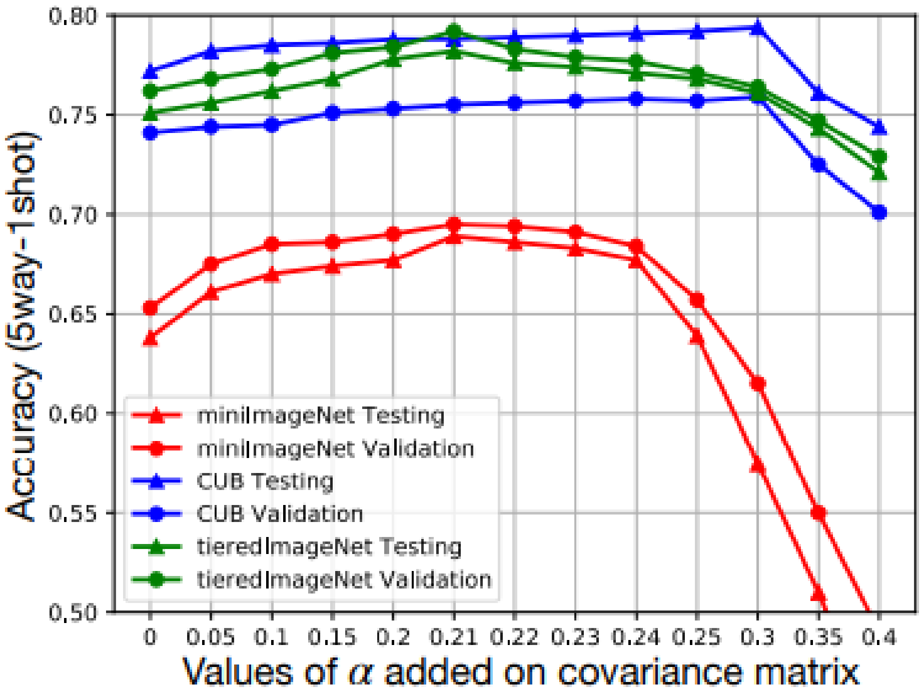


Figure 5: The effect of different values of α.

Outline

- Introduction
- Main Approach
- Experiments
- Conclusion

Conclusion

- They propose a simple but effective distribution calibration strategy for few-shot classification.

- Without much parameters to learn, a simple logistic regression trained with generated features outperform current state-of-the-art method .

- The calibrated distribution is visualized and demonstrates an accurate estimation of feature distribution.


THANK YOU
UNIVERSITY of HOUSTON | ENGINEERING