Variational Few-Shot Learning

Jian Zhang[1] Chenglong Zhao[1] Bingbing Ni[1*] Minghao Xu[1] Xiaokang Yang[2]
[1]Shanghai Jiao Tong University, Shanghai 200240, China
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China
{stevenash0822,cl-zhao,nibingbing,xuminghao118,xkyang}@sjtu.edu.cn

Abstract

We propose a variational Bayesian framework for enhancing few-shot learning performance. This idea is motivated by the fact that single point based metric learning approaches are inherently noise-vulnerable and easy-to-be-biased. In a nutshell, stochastic variational inference is invoked to approximate bias-eliminated class specific sample distributions. In the meantime, a classifier-free prediction is attained by leveraging the distribution statistics on novel samples. Extensive experimental results on several benchmarks well demonstrate the effectiveness of our distribution-driven few-shot learning framework over previous point estimates based methods, in terms of superior classification accuracy and robustness.

1. Introduction

Relying on substantial labelled data, deep learning [27, 41, 20] based approaches have led a series of breakthroughs in computer vision community. However, collecting and annotating such a scale of data are both labor-intensive, seriously restricting their practicability in actual applications. Inspired by human visual system, which has the instinct to recognize novel objects by learning with only a few examples, few-shot learning [9] is proposed to mimic this ability.

More explicitly, we consider a common few-shot learning scenario where a learning agent, acquiring decision-making strategy with substantial data during training, is exerted on previously unseen classes with limited auxiliary supervision during test. Many prior works [25, 45, 7, 6] stress on learning a well-organized *matching* mechanism. This mechanism attempts to seek a best match between *support set* (scarce labelled data) and *target set* (unlabelled data), via parametric [35, 18] or non-parametric [45, 42] methods as measurement. While non-parametric methods utilize designated metric, its counterpart parametric methods leverage neural network to measure the similarity. The essence of both methods is to search for the nearest support class of target data within an embedding space.

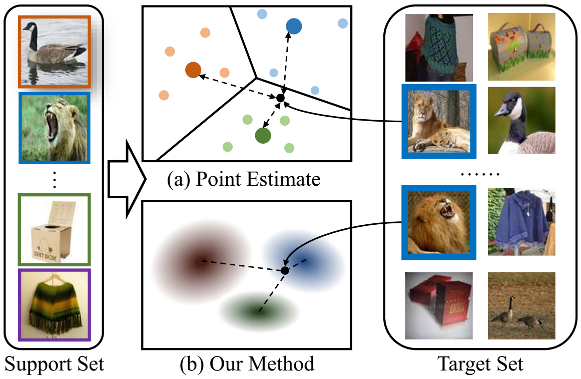


Figure 1. **Variational few-shot learning:** (a) Prior works choose a specific point to represent each class. (b) Our method estimates the distributions of each class. Best viewed in color.

As mentioned above, points of the same class scatter in a certain range of embedding space. A common manner is that choosing a specific point (*e.g.* central point) to represent the same class points, which can be represented as *point estimates*. Despite their favorable performance in public datasets, there exists two issues: 1) Estimating such a specific point is difficult when limited support points distribute unevenly in embedding space. For instance, prototype-based methods [42, 15, 1] compute the mean of embedded support examples for each class, which is vulnerable to noise since data is severely limited. 2) They are lack of interpretability since a single embedding is insufficient for indicating a class. We consider that each point of the same class is not isolated in embedding space, but is sampled from a high-dimensional distribution. As shown in Figure 1, it is obvious that the distributions of each class have superior descriptive power than several points.

To explicitly address the issues caused by point estimate methods, we propose a distribution estimate framework via *variational* inference. While exact inference is computationally intractable, variational inference is a theoretically attractive method and suitable for computation.

[*] Corresponding author: Bingbing Ni

Previous work [8] applying variational inference requires a full dataset through the model, inefficiently aggregating the embeddings of all elements first to generate the distribution. We instead share the distribution generation mechanism across elements of both support and target set and then merge them into large class-specific distributions. By estimating distributions of each class, we can straightforwardly calculate the confidence of each class the target point belongs to and perform classification by the confidence. This facilitates use of the entire distribution rather than sample from it, which makes our method immune from the sample bias. In a nutshell, the key of our method is to precisely infer these distributions and concisely predict outputs with distributions. We show that a combination between few-shot learning and variational inference can be extended to greatly eliminate the bias of original *point estimates* and improve performance in prediction. Furthermore, our method allows for full Bayesian analysis of the model, and it's significantly more interpretable over prior arts.

Notably, our variational learning method estimates distributions via tightening intra-class relationships. A classifier-free prediction is then obtained via calculating the sample probabilities of target samples from those estimated distributions. This metric is proven to be an extension of weighted Euclidean distance where class-specific features are strengthened and irrelevant features are suppressed.

Our method is evaluated on public benchmarks and results in state-of-the-art performance w.r.t. few-shot classification accuracy. Not only that, we achieve lowest variance compared with contemporary methods, which is meaningful for many real world applications where robustness of prediction is extremely desired, *e.g*., autonomous driving or medical diagnosis [49]. Additionally, we extend our variational-based learning strategy on a recently released one-shot segmentation benchmark [31], achieving incremental progress over the initial architecture. It further demonstrates the transportability of the proposed method.

2. Related work

Metric-learning based approaches. Metric learning approaches attempt to map the few-shot labelled and unlabelled points into a non-linear embedding space and perform classification by assessing which labelled points are closest to the unlabelled points. The key assumption of those approaches is that they can learn feature embeddings which preserve the intrinsic class relationships. Koch *et al*. [25] pioneer Siamese network to generate embeddings of same/different pairs and compute a weighted L_1 metric for measuring the similarity. Vinyals *et al*. [45] propose Matching Network to accumulate information on a given task with memory mechanism and utilize cosine distance in an attention kernel as measurement. In Prototypical Network [42], a metric space is learned in which nearest neighbor classification can be performed with prototype representations of each class. Sung *et al*. [43] introduce a learnable similarity metric by calculating the relation score between query images and the prototype of each class. On this basis, label-propagation-related approaches [29, 22] are developed to explore intra-class or inter-class relationships in the classification task. In addition, contrastive loss [17, 25] and triplet loss [40, 44] are used for strengthen learned metrics by fully exploring pair/triplet relationships within the dataset.

We emphasize that metric-learning methods are restricted in the quality of feature embeddings. This point estimate approach is sensitive to sample noise from random selection of support dataset and inductive bias from scarce data. In contrast, we estimate class-specific distributions instead, which possesses general stability and interpretability.

Meta-learning based approaches. The natural inconsistency between training and testing data is a bottleneck of contemporary few-shot learning. A generic term, "meta-learning", is first formulated to tackle this problem. The core of primary methods [10, 33, 11] is to initialize weight configuration that can be swiftly fine-tuned in test phase within a fewsteps. In MAML [10],parameters are optimized within a task pool so that they can be quickly adapted to a particular task. In "Optimization as a model" [37], a LSTM-based meta-learner is trained to coverage a learner classifier. Considering time-consuming drawback of fine-tuning in test phase, recent works [39, 4, 26] inference in a feed-forward pass. [15, 36, 48, 16, 3] further implicitly adopt meta-learning as an auxiliary phase to predict the parameters from the activations in the last stage.

Data augmentation based approaches. Data insufficiency and overfitting remain huge challenges. To alleviate it, GAN-based techs [46, 19, 47] exploit diverse training data patterns and apply it into test phase to expand the *support set* capacity. Ren *et al*. [38] and Garcia *et al*. [13] also benefit from leveraging unlabelled data, where the former is to refine class centroids and the latter is to construct graphical models. Nevertheless, these data-augmented methods fail to thoroughly solve the issue as they might introduce noise when improper patterns are deployed.

3. Methodology

3.1. Problem Formulation

Suppose we sample a small *support set* $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{K \times C} \subset X \times Y$ of pairs of inputs $x_i \in X$ and corresponding outputs $y_i \in Y$ (*e.g*., labels or masks in classification and segmentation tasks), where K labelled samples for each of C unique classes are obtained, a *target set* $\mathcal{T} = \{(\widetilde{x}_i, \widetilde{y}_i)\}_{i=1}^{N \times C} \subset X \times Y$ is similarly formed with another N samples for each of C classes. Notably, $\widetilde{y}_i$ is only available in training phase, remaining the target set unlabelled during testing. The large collection $X \times Y$ in training phase do not overlap with those in test phase, which means none of test classes will be seen during training.

A C-way K-shot learning problem is formulated to predict outputs of the target set $\mathcal{T}$ with the prior knowledge mined from support set $\mathcal{S}$, which is defined as:

$$\hat{y} = \arg\max_{\widetilde{y} \in C} p(\widetilde{y}|\widetilde{x}, \mathcal{S}), \tag{1}$$

where $p(\widetilde{y}|\widetilde{x}, \mathcal{S})$ is the probability of classifying a certain target sample $\widetilde{x}$ with the class $\widetilde{y}$ conditioned on $\mathcal{S}$. In the standard setting of few-shot learning [31], $\mathcal{S}$ and $\mathcal{T}$ are denoted as sample/query set in training phase and support/test set in test phase, respectively. Here we unify the notations into *support/target* set on both phases for clarity.

3.2. Point Estimate Revisit

Prior point estimate methods explicitly learn an embedding function $h(x)$, that maps support examples into a space where examples from the same class are close and those from different classes are distant. A designated measurement $s(\widetilde{x}, r)$ is then applied to a test sample $\widetilde{x}$ and former generated embeddings r to compute similarity scores. Generally, the whole process can be defined as:

$$\hat{y} = \arg\max_{\{y|(x,y)\in\mathcal{S}\}} s\big(\widetilde{x}, h(x)\big) \tag{2}$$

However, there are two important issues: 1) Scarce support data makes it hard to correctly estimate specific embeddings. 2) Data distribution inconsistency between training and test phase easily incurs overfitting of a point estimate system. To alleviate those problems, we are motivated to concentrate on estimating distributions of each class, which are robust to limited data than point estimate.

3.3. From Point to Distribution: A Variational Learning Strategy

3.3.1 Variational Inference for Few-shot Learning

Instead of finding such a biased point estimate in embedding space, distributions of each class can be predicted. From this perspective, we regard our goal as to infer the output of $\widetilde{x}$ by computing the confidence $p(\widetilde{y}|\widetilde{x}, z)$, where z denotes the distribution of entire dataset $\mathcal{T}$. However, to learn the distribution of z in Bayes rule $p(z|\mathcal{T}) = p(z)p(\mathcal{T}|z)/p(\mathcal{T})$ involves a computation intractable integral. One of approximating approaches is *variational* inference, which is a theoretically attractive method and easy to compute. In variational inference, we approximate the true posterior distribution with a parameterized distribution $q_\phi(z|\mathcal{S})$ conditional on $\mathcal{S}$ by minimizing the Kullback-Leibler divergence $D_{KL}(q_\phi(z|\mathcal{S})||p(z|\mathcal{T}))$. Concretely, minimizing the KL divergence is equivalent to maximize the evidence lower bound (ELBO) [24] in Eq. 3.

$$\begin{aligned} \log p(\mathcal{T}) &= \log \int \frac{p(\mathcal{T}, z)}{q_\phi(z|\mathcal{S})} q_\phi(z|\mathcal{S}) dz \\ &\geq \mathbb{E}_{q_\phi(z|\mathcal{S})} \log \frac{p(\mathcal{T}|z)p(z)}{q_\phi(z|\mathcal{S})}. \end{aligned} \tag{3}$$

To emphasize our maximization goal *ELBO*(ϕ), we extract the last term in Eq. 3, which has the same form as:

$$\textit{ELBO}(\phi) = \mathbb{E}_{q_\phi(z|\mathcal{S})}\big[\log p(\mathcal{T}|z)\big] - D_{KL}\big(q_\phi(z|\mathcal{S})||p(z)\big). \tag{4}$$

This objective function includes two terms. The first term tries to maximize the likelihood to improve the confidence of prediction, and the second term finds the approximate posterior distribution by minimizing the KL divergence. $p(z)$ represents the prior distribution in Bayes Learning, which is assigned to a certain distribution in most cases [24]. However, we emphasize that a manually fixed prior impedes our method's generalization capability due to the huge discrepancy between training and test phase. Thus, we make full use of target and support dataset to obtain prior distribution by $p_\theta(z|\mathcal{T}, \mathcal{S})$. In this paper, $q_\phi(z|\mathcal{S})$ and $p_\theta(z|\mathcal{T}, \mathcal{S})$ are both modelled as parameterized networks, and optimized in an end-to-end manner. As we leverage the neural network to obtain the prior, we rewrite the ELBO maximization goal as follows:

$$\begin{aligned} \textit{ELBO}(\phi, \theta) = &\ \mathbb{E}_{q_\phi(z|\mathcal{S})}\big[\log p(\mathcal{T}|z)\big] \\ &- D_{KL}\big(q_\phi(z|\mathcal{S})||p_\theta(z|\mathcal{T}, \mathcal{S})\big). \end{aligned} \tag{5}$$

Once the few-shot learning problem is casted as a variational inference problem, our task then lies in two-fold: how we precisely estimate the distribution of z and how we leverage the distribution to estimate the output $\widetilde{y}$ of target set, which are in detail discussed in Section 3.3.2 and 3.3.3.

3.3.2 Precise Estimation of Distribution

As shown in Eq. 5, a KL-divergence between the posterior and prior distributions need to be calculated. In few-shot learning problem, we estimate class-specific distributions by referring to the given output y. Specially in a C-way few-shot learning scheme, the support set is further split into C subset $\mathcal{S}_1, \ldots, \mathcal{S}_C$, each containing a certain class. Similar partition is also applied on target set. In this case, C posterior distribution conditional on the support set is to approach the same quantity of priors. As a consequence, we elaborate the second term in Eq. 5 with a class-specific KL divergence $\mathcal{L}_{intra}$ as:

$$\mathcal{L}_{intra} = \sum_{i=1}^{C} D_{KL}\big(q_\phi(z|\mathcal{S}_i)||p_\theta(z|\mathcal{T}_i, \mathcal{S}_i)\big). \tag{6}$$

Then the problem lies in how to define posterior distribution $q_\phi(z|\mathcal{S}_i)$ and prior distribution $p_\theta(z|\mathcal{T}_i, \mathcal{S}_i)$, namely,

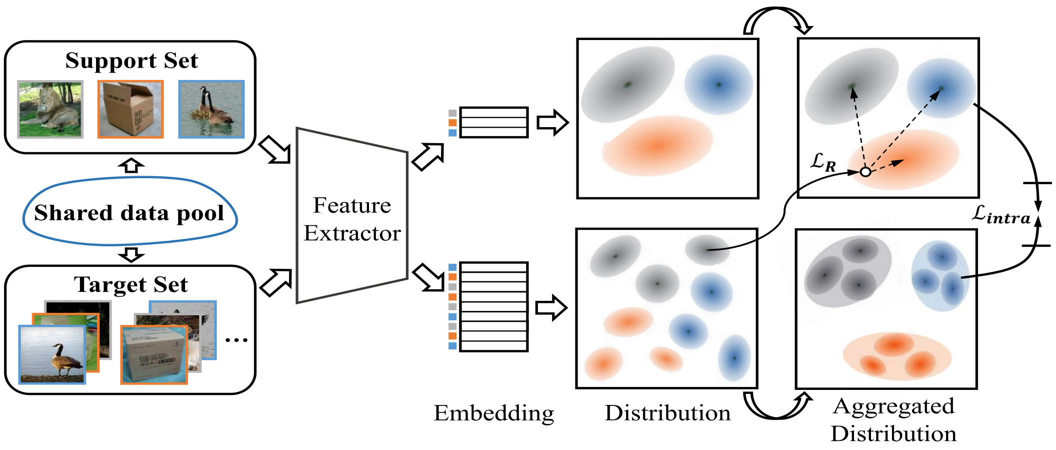


Figure 2. Diagram of the proposed method in a 1-shot, 3-way exemplar. Our variational method shares the identical mechanism to generate the distribution of each support and target data, and then aggregate them into class-specific distributions. By back-propagating errors of two loss terms (*i.e.*, $\mathcal{L}_R$ for penalizing classification mistake. $\mathcal{L}_{intra}$ for correcting the relationship within the same class), the framework acquire transferable distribution estimation knowledge that is amenable to unseen tasks in test phase. Best viewed in color.

how to obtain class-specific distributions. Details are discussed as follows:

Generate statistics for single data. While there is much freedom in the form q_ϕ and p_θ, we assume that latent variable z within the same class satisfies multivariate Gaussian with a mean and diagonal covariance structure. For the purpose of learning latent distribution(*i.e.*, learning the class mean and variance) quickly from a single instance, a statistics generating pipeline is tailored with two cascaded structures and an additional transform function: 1) A feature extractor $F(x;\varphi_F)$ extracts a representation r from single data x. 2) A raw generator $G(r;\varphi_G)$ takes the representation as input and directly split the output into two equivalent parts $\boldsymbol{\mu}_{raw}$ and $\boldsymbol{\sigma}^2_{raw}$ with equal dimensions. 3) A transform function is applied to $\boldsymbol{\sigma}^2_{raw}$ to yield the final variance vector $\boldsymbol{\sigma}^2$ while $\boldsymbol{\mu}_{raw}$ remains identical to yield the final mean $\boldsymbol{\mu}$. The entire process is formulated as follows:

$$\begin{aligned} \big[\boldsymbol{\mu}_{raw}, \boldsymbol{\sigma}^2_{raw}\big] &= G\big(F(x;\varphi_F);\varphi_G\big), \\ \boldsymbol{\mu} &= \boldsymbol{\mu}_{raw}, \\ \boldsymbol{\sigma}^2 &= |w| * S(\boldsymbol{\sigma}^2_{raw}) + |b|. \end{aligned} \tag{7}$$

In practical application, F and G are modeled as convolutional networks and fully-connected layers with learnable parameters φ_F and φ_G, respectively. The transform function is a Sigmoid mapping $S(\cdot)$ with learnable scale w and offset b, rescaling the raw variance vector $\boldsymbol{\sigma}^2_{raw}$ to a new range $(|b|, |w| + |b|)$. We clarify that predicting a well-constrained variance vector is of great significance for subsequent estimation in Section 4.2. The predicted mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$ for single data jointly comprise $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ to denote corresponding class distribution. Nevertheless, it is not reliable of the distribution generated from single data without extra supervision. Although each data contains sufficient class information, we still require estimating posterior distribution and approaching prior on a large set to eliminate class-irrelevant representation of a single data.

Estimate posterior distribution. It is of low confidence to generate posterior distribution with single data. We then impose support subsets of the same class on making a more precise distribution estimation. Unlike the strategy (*i.e.*, generate distributions with an averaged feature of a mini-batch) frequently adopted in [21, 14, 8], which implies every sample weighs the same, our method unifies distribution of every single data into the final one. Concretely, given n data with generated mean $\{\boldsymbol{\mu}_i | i = 1, \ldots, n\}$ and variance $\big\{\boldsymbol{\sigma}^2_i | i = 1, \ldots, n\big\}$, we estimate the overall distribution parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ on the following criterion:

$$\begin{aligned} \boldsymbol{\mu} &= \left(\sum_{i=1}^{n} \boldsymbol{\sigma}_i^{-2}\right)^{-1} \left(\sum_{i=1}^{n} \boldsymbol{\sigma}_i^{-2} \boldsymbol{\mu}_i\right), \\ \boldsymbol{\sigma}^2 &= \left(\frac{\sum_{i=1}^{n} \boldsymbol{\sigma}_i^{-2}}{n}\right)^{-1}. \end{aligned} \tag{8}$$

On one hand, the overall mean in Eq. 8 is a variance-weighted linear combination of individual components. And components with small variance are allocated big weights. On the other hand, the overall variance tends to approach the least variance component. To minimize disturbance, the network tends to predict small variance for those data which lies in the class center (more representative), but predict relatively large variance for those at the boundary (less representative). Otherwise either uniform variance or the converse situation results in misleading aggregation because of undersampling in support set. Thus, our aggregation choice aims to ease the burden of automatically selecting the most representative components.

Approach prior distribution. Similarly as the aggregating posterior distribution conditional on the support set, the prior distribution is accessed with a larger combination of support and target set. With smoothed class-irrelevant information along with the larger set, the prior focuses more on class-specific information and is thus well-fitted for a discriminative process. We then approach our estimated posterior with the prior, force the network capable of locating representative information from small support set.

3.3.3 Predict Output with Distribution

With class-specific distribution estimated from support set, we solve the classification task in a straightforward way to calculate the probability of target data. The final prediction is consistent with the maximum probability among classes.

Explicitly, we encode the target sample x into class distribution and utilize the mean vector $\boldsymbol{\mu}(x)$ to represent it, as the mean vector centralizes the class information extracted from this sample. The posterior distribution $q_\phi(z|\mathcal{S}_c)$ conditional on category c satisfies $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)$, where the variance vector indicates the significance of different dimensions. We thus denote $Pr\big(\boldsymbol{\mu}(x)|q_\phi(z|\mathcal{S}_c)\big)$ as the probability of $\boldsymbol{\mu}(x)$ if sampled from the distribution $q_\phi(z|\mathcal{S}_c)$, which is calculated as follows:

$$Pr\big(\boldsymbol{\mu}(x)|q_\phi(z|\mathcal{S}_c)\big) = \mathcal{N}\big(\boldsymbol{\mu}(x); \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2\big). \tag{9}$$

Those probabilities, are transformed into logarithmic form and then normalized with a softmax function to represent the confidence of which category the sample x belongs to. On this basis, we reformulate the first term in Eq. 5 to derive the recognition loss $\mathcal{L}_R$ for the whole target set $\mathcal{T}$ as:

$$\mathcal{L}_R = \sum_{(x_i,y_i)}^{\mathcal{T}} \log\left(1 + \sum_{c \neq y_i}^{C} \frac{Pr\big(\boldsymbol{\mu}(x_i)|q_\phi(z|\mathcal{S}_c)\big)}{Pr\big(\boldsymbol{\mu}(x_i)|q_\phi(z|\mathcal{S}_{y_i})\big)}\right). \tag{10}$$

Overall loss function. By tightening the intra-class relationship and fulfilling the recognition gap, the overall loss function is established as:

$$\mathcal{L} = \mathcal{L}_R + \mathcal{L}_{intra}. \tag{11}$$

3.3.4 Implementation Details

Figure 2 illustrates our overall framework. Similarly as many recent methods [43, 32], we follow their four-block convolutional architecture to form our feature extractor F. Each block contains a 3×3 convolution with 64 filters, a batch normalization, a ReLU non-linear layer and a 2×2 max-pooling layer. A little difference is that we clip the max-pooling layer of the last two blocks. Thus, the output size of the feature extractor is $64 \times 5 \times 5 = 1600$. Taking the representation as input, we then construct the raw generator G by a fully-connected layer of 128 dimensions. After applying the transformation function H and aggregation rules mentioned in Section 3.3.2, 64-dimensional distributions of support and target sets are achieved. We emphasize that our final distributions are consistent with embeddings of those comparable works w.r.t. the number of dimensions.

The training procedure is split into 2 stages. In first stage, we only train the feature extractor with the entire supervised training set. This is done in exactly the same way as any other standard recognition model. In second stage, we awaken the whole architecture and allocate distinct learning strategy for two learnable components. We use an initial learning rate of 10^{-4} and 10^{-3} for the feature extractor F and the raw generator G, respectively. Those two learning rates are cut by half every 5000 episodes. All of our trainable parameters are trained via Adam optimizer [23].

3.4. Discussion

Relation with previous variational arts. Recent works [8, 21] have already imposed variational inference on few-shot scenario. However, they are both generative models focusing more on reconstruction and utilizing fixed standard Gaussian prior to guide the latent space exploration. We argue that our framework is the first discriminative model to tackle with few-shot problem via variational inference, to the best of our knowledge. Also, our method improves generalization by exploiting the dataset to weaken irrelevant features and obtain class-specific prior. Our recognition performance degenerates to those two methods only if data distribution is identical between training and testing, which is not the case in few-shot learning.

Relation with euclidean distance metric. Even though widely used for evaluating the similarity between support and target features [42, 12], in the presence of noise Euclidean distance loses in performance, due to equal contribution over all features. In our framework, we predict the log-probability from Gaussian distribution as the metric, to overcome this drawback by weighting features on each dimension. Under the assumption of multivariate Gaussian densities, we can rewrite our metric D as:

$$\begin{aligned} D &= \log Pr\big(\boldsymbol{\mu}(x)|q_\phi(z|\mathcal{S}_c)\big), \\ &= -\frac{1}{2}\sum_{i=1}^{d} \frac{(\mu_i(x) - \mu_{c,i})^2}{\sigma_{c,i}^2} - \sum_{i=1}^{d} \log \sigma_{c,i} - \frac{d}{2}\log 2\pi, \end{aligned} \tag{12}$$

where d denotes the feature dimension and i denotes the i-th component of the vector. Thus, our proposed metric is equivalent to a combination of a weighted Euclidean distance, a variance regularization term and a constant. By giving low weight (*i.e.*, big variance) to noisy features and high weight (*i.e.*, small variance) to class-specific features, a reliable similarity measurement is achieved.

4. Experiments

We evaluate our approach on two few-shot related tasks: classification on Omniglot [28] and *mini*Imagenet [45] dataset and segmentation on a recent published benchmark called cluttered Omniglot [31].

4.1. Few-shot Classification

4.1.1 Omniglot

Omniglot [28] is a handwritten characters dataset of 1623 classes from 50 alphabets, where each class consists of 20 samples drawn by different people. Following the same setup first introduced in [45], we split the dataset into a training set of 1200 classes and a test set of the rest. We then resize the characters to 28 × 28, and augment the dataset through three rotated versions (90°, 180°, 270°).

Under the C-way K-shot setting, we form an episode by randomly picking K images from the support set and 15 from the target set for each of C sampled classes. We compute few-shot classification accuracy by averaging the results over 1000 generated episodes in test phase, as shown in Table 1. Our variational method achieves better recognition performance than state-of-the-arts in majority cases. Although we do not hit the best performance for 5-shot 20-way classification, our results are still in low variance compared with more complicated structures (like SNAIL [32]). Crucially, we reach those results with no extra classifiers, which saves memory overhead.

4.1.2 *mini*Imagenet

***mini*Imagenet** [45] is a more challenging dataset, consisting of 84×84 RGB images from 100 classes with 600 samples per class. Similarly as [37], the entire dataset is split into 64, 16 and 20 classes for training, validation and testing, respectively. For fair comparison, 16 validation classes are only used for model selection.

We mirrors the configuration of Omniglot experiment except for two minor changes: 1) As the input size is expanded to 84×84, we reserve the last two max-pooling layers in feature extractor F to maintain a 1600-dimensional representation r; 2) Considering *mini*Imagenet is more complex for generalizing well to unseen classes, we increase the initial learning rate of feature extractor to $2{*}10^{-4}$ in stage 2 for correcting the pre-trained features in stage 1.

We compute few-shot classification accuracy by averaging the results over 600 generated episodes in test phase, as summarized in Table 2. Recent works leverage ResNet [20]-like network to consolidate extracted features, which is of great assistance to strengthen the recognition capability. Thus, for fair comparison, we implement another backbone by substituting the relatively weak feature extractor (*i.e.*, four-blocks architecture) for the ResNet-12 as widely adopted in [32, 34, 2]. We argue that few-shot learning is more of a strategy-learning procedure than a representation-learning process, which is to some extent demonstrated by our superior performance for both simple ConvNet and ResNet implementations.

Setup	5-way Acc.		20-way Acc.	
	1-shot	5-shot	1-shot	5-shot
VHE [21]	-	-	95.2	98.8
NS [8]	98.1	99.5	93.2	98.1
MN [45]	98.1	98.9	93.8	98.5
PN [42]	98.8	99.7	96.0	98.9
MM-Net [5]	99.3	99.8	97.2	98.9
L2C [43]	99.6(0.2)	99.8(0.1)	97.6(0.2)	99.1(0.1)
SNAIL [32]	99.07(0.16)	99.78(0.09)	97.64(0.30)	**99.36**(0.18)
Ours	**99.67**(0.13)	**99.83**(0.06)	**97.95**(0.17)	99.24(0.10)

Table 1. Few-shot classification accuracy (%) on Omniglot. All results are averaged over 1000 episodes and reported with 95% confidence intervals (represented within the bracket). The best performance is indicated in **bold**.

Another observation is that the advantages of 5-shot classification results are less noticeable than of 1-shot scenario. Considering the 5-way setting with ConvNet, our method surpasses published state-of-the-art 0.95% for 1-shot but falls behind for 5-shot. We attribute this phenomenon to the stability in lower training shots, since our method can extract class information better from scarce support data via variational inference. When more support data is available, the bonus of variational inference will be decreased.

It is worth noticing that our method achieves the lowest variance, which means our prediction results are **not** fluctuant w.r.t. random selection of support and target set.

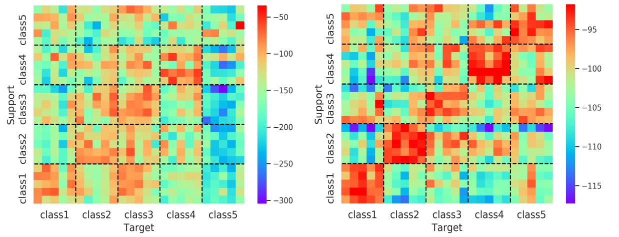
4.1.3 Experimetnal Analysis

Stability analysis. Figure 3 shows the similarity matrix learned by Prototypical Net and our method, with the same ConvNet backbone under 5-way 5-shot settings. The metrics are negative Euclidean distance and negative log-probability, respectively. We clarify that the matrix is asymmetric, as it is pair-wise constructed between randomly selected support subset (5 images per class, 5 classes) and corresponding target subset. Note that the 5 support images in the same class do **not** cluster into a class representation but perform independently as in 1-shot scenario, to examine the system stability if sampling different test batches. Thus, each cell consisting of 5×5 grids illustrates the divergence between two classes, as well as the intra-class similarities. The higher intra-class similarities demonstrates that our probability metric is more vulnerable to sampling noise under a well-designed Bayesian scheme, compared to standard Euclidean distance.

Training configuration analysis. Although it is a consensus to remain consistency between training and test

Pipeline Setup	Fine Tune	5-way 1-shot		5-way 5-shot	
Backbone		ConvNet	ResNet	ConvNet	ResNet
MAML [10]	Y	48.70 ± 1.84	-	63.11 ± 0.92	-
Prototypical Net [42]	N	49.42 ± 0.78	-	68.20 ± 0.66	-
Learning to Compare [43]	N	50.44 ± 0.82	-	65.32 ± 0.70	-
SNAIL [32]	N	-	55.71 ± 0.99	-	68.88 ± 0.92
TADAM [34]	N	-	58.50 ± 0.30	-	76.70 ± 0.30
Qiao *et al.* [36]	N	54.43 ± 0.40	59.60 ± 0.41	67.87 ± 0.20	73.74 ± 0.19
Without Forgetting [15]	N	56.20 ± 0.86	55.45 ± 0.89	**72.81** ± 0.62	70.13 ± 0.68
Our Variational Method	N	**57.15** ± 0.31	**61.23** ± 0.26	71.54 ± 0.23	**77.69** ± 0.17

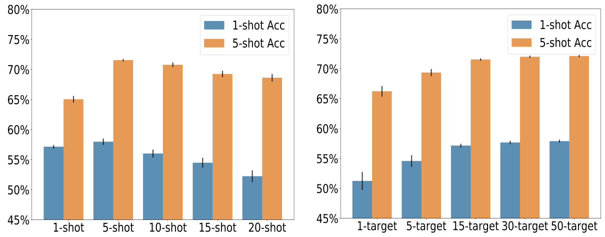
Table 2. Few-shot classification accuracy (%) on *mini*Imagenet. All results are averaged over 600 episodes and reported with 95% confidence intervals. The best performance is indicated in **bold** and the lowest variance is colored in red. Best viewed in color.



(a) Prototypcial Net [42]

(b) Ours

Figure 3. Similarity matrix of Prototypcial Net and our method on *mini*Imagenet. Horizontal axis: 5 target images per class. Vertical axis: 5 support images per class. Cell: 5*5 comparison between two classes. Warmer colors denotes higher similarities.



(a) Different support size

(b) Different target size

Figure 4. Model performance on 5-way *mini*Imagenet with various training configurations. x-axis: different size of support or target set during training. y-axis: 1-shot or 5-shot test accuracy. Error bars indicate 95% confidence intervals over 600 test episodes.

phase in standard few-shot learning, we argue not all frameworks benefit the most from this identical settings. We design two experiments to study the influence of various training configurations: 1) different support size; 2) different target size, which are denoted as "k-shot" and "k-target" in Figure 4. It is well observed that increasing target size during training will lead to the performance gain, while remaining the same support size during training is a better choice considering the stability.

4.2. Ablation Study

To assess the effects of different components, we conduct an ablation study on *mini*Imagenet, with detailed results in Table 3.

Aggregate embeddings / aggregate distributions. First we examine the performance of our system without the proposed aggregation rules on distributions. To maintain the structural integrity, we aggregate embeddings of the same class with an average-pooling layer and generate corresponding distribution on this basis instead. This alternative operation is abbreviated to "aggregate embeddings" while ours is denoted by "aggregate distributions" for clarity. We observe that the latter drastically exceeds the former w.r.t. 5-shot classification accuracy, which means our designated aggregation rules better expolit significance of each sample.

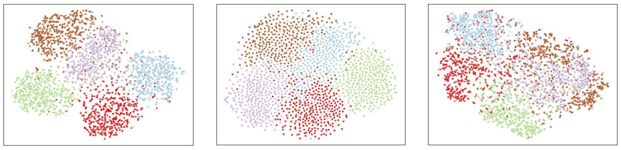
Without / with transform function. We validate the hypothesis that a well-constrained variance is of significance to the system performance. In our approach, a transform function H with two learnable scalars is applied to restrict the final variance to a certain range. If no constraint is exerted on the estimated variance (*i.e.*, the transform function is replaced with a simple ReLU nonlinearity), a sharp decrease in classification accuracy is observed in both 1-shot and 5-shot cases. Moreover, it incurs instability without transform function. We infer that it remains a complicated problem to simultaneously optimize the mean and the variance, where the network can be easily stuck in local optima under the KL constraint (*e.g.*, two distributions approach each other with small mean and big variance). With learnable scalars, the transform function is verified to generate non-trival variance while maintaining the model flexibility.

Multi-stage / end-to-end training. We further examine whether an end-to-end training strategy degrades the performance, in which case both feature extractor and distribution generator are mutually tuned from scratch. Our network is better fitted in a 2-stage training scheme as the pre-trained features assists in generalizing class distribution.

t-SNE visualization. We utilize t-SNE [30] to visualize the learned representation of target set in a 5-way classification task. In detail, we repeat sampling from our estimated distributions of each class to construct the final 64-dimensional features. As shown in Figure 5, representa-

			*mini*Imagenet	
AD	TF	MS	5-way 1-shot	5-way 5-shot
	✓	✓	54.89 ± 1.37	65.62 ± 0.94
✓		✓	53.18 ± 0.98	68.56 ± 0.71
✓	✓		55.17 ± 0.64	69.13 ± 0.50
✓	✓	✓	**57.15** ± 0.31	**71.54** ± 0.23

Table 3. Ablation study on our variational approach. All results (%) are averaged over 600 episodes and reported with 95% confidence intervals on 5-way *mini*Imagenet. The best performance is **highlighted**. AD: aggregate distributions. TF: transform function. MS: multi-stage training. Blank cell indicates operating in another way or operating without it, as discussed in analysis.



(a) Our full model (b) End-to-end training (c) Aggregate embeddings

Figure 5. The t-SNE visualization of target set in a 5-way problem under three different configurations.

tions of the same class are clustered closer in our full model than other two configurations. While the model trained in an end-to-end manner fails to split different categories, another model adopting the ”aggregate embeddings” strategy even slightly confuses the boundary of classes.

4.3. Few-shot Segmentation

We further evaluate our approach on a novel few-shot segmentation benchmark [31]. This benchmark, called *cluttered Omniglot*, is developed from the original Omniglot to form more complex cluttered scenes with multiple characters. Concretely, a cluttered scene of 96 × 96 pixels is composed of a target character and massive distractors (3-255 distinct characters), with each character of 32 × 32 pixels placed at a random location. Given distortion manually added to each character and occlusion in cluttered scenes, it is a challenging task to find the target character and produce a pixelwise segmentation map, even difficult for segmenting previously unseen targets under few-shot settings.

As huge discrepancy lies in classification and segmentation task, we make an incremental adjustment on the basis of proposed architecture MaskNet in this benchmark, rather than reconstruct our model to adapt to the new task. In training phase, MaskNet first generates some proposals with associated instance segmentations prediction, and then decides which of these proposals is the best match by a discriminator. We then deploy our variational learning strategy by two modifications: 1) Instead of predicting embeddings of generated proposals, we predict the distributions with auxiliary proposals. Those auxiliary proposals are generated in extra feed-forward passes with transformed versions of target characters as input. 2) In test phase, we makes the final decision of the best proposal by the distance between target embeddings and proposal distributions. Thus, by fairly comparing with the original well-performed architecture, this configuration ensures a precise evaluation of our variational learning method with minimal overhead.

Model	4	8	16	32	64	128	256
Siamese U-net	97.1	92.1	79.8	62.4	48.1	39.3	38.4
MaskNet	95.8	90.5	79.3	65.6	52.8	44.8	**43.7**
ours	**97.9**	**93.5**	**83.4**	**67.0**	**53.9**	**45.4**	43.2

Table 4. Few-shot segmentation accuracy (IOU in %) across different amounts of characters per cluttered scene. The best performance is highlighted.

We report segmentation results in Table 4, using Intersection over Union (IOU) as measurement. Siamese U-net also proposed in [31] serves as the baseline. We observe that our variational strategy strengthens the original architecture MaskNet by a considerable margin. This is because we improve the quality of segmentation proposals by making them insensitive to the distortion of the input. Without any auxiliary information, our assumption that we can well estimate the distribution of correct-located segmentation proposals fails in a seriously occluded situation, which results in a slight decrease on performance with 255 distractors per cluttered scene.

5. Conclusion

We propose a variational Bayesian framework for few-shot learning. Different from the deterministic point estimate methods, we approximate class-specific distributions instead and straightforwardly compute the probability of novel input. This probabilistic-based metric is an extension of weighted Euclidean distance, further consolidating the estimated distribution as irrelevant information is suppressed. Our method requires no fine-tuning before test and is easy to implement on other task, which is of great flexibility and transportability. It is further proven effective on few-shot recognition and segmentation benchmarks, in terms of superior classification accuracy and robustness.

6. Acknowledgement

The work was partly supported by State Key Research and Development Program (2016YFB1001003). This work was also supported by National Science Foundation of China U1611461 and China’s Thousand Youth Talents Plan, STCSM 17511105401, 18DZ2270700 and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China.This work was sponsored by CCF-Tencent Open Fund. This work was also jointly supported by SJTU-Minivision joint research grant.

References

[1] Kelsey R. Allen, Evan Shelhamer, Hanul Shin, and Joshua B. Tenenbaum. Infinite mixture prototypes for few-shot learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 232–241, 2019.

[2] Matthias Bauer, Mateo Rojas-Carulla, Jakub Bartlomiej Swiatkowski, Bernhard Schölkopf, and Richard E. Turner. Discriminative k-shot learning using probabilistic models. *arXiv preprint arXiv:1706.00326*, 2017.

[3] Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *7th International Conference on Learning Representations*, 2019.

[4] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Advances in Neural Information Processing Systems*, pages 523–531, 2016.

[5] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. Memory matching networks for one-shot image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 4080–4088, 2018.

[6] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *7th International Conference on Learning Representations*, 2019.

[7] Matthijs Douze, Arthur Szlam, Bharath Hariharan, and Herv Jgou. Low-shot learning with large-scale diffusion. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3349–3358, 2018.

[8] Harrison Edwards and Amos Storkey. Towards a neural statistician. In *5th International Conference on Learning Representations*, 2017.

[9] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on Pattern Analysis and Machine Intelligence*, pages 594–611, 2006.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135, 2017.

[11] Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. *arXiv preprint arXiv:1710.11622*, 2017.

[12] Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. In *Advances in Neural Information Processing Systems*, pages 983–993, 2018.

[13] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. In *6th International Conference on Learning Representations*, 2018.

[14] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

[15] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

[16] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[17] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1735–1742, 2006.

[18] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015.

[19] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[21] Luke B. Hewitt, Maxwell I. Nye, Andreea Gane, Tommi S. Jaakkola, and Joshua B. Tenenbaum. The variational homoencoder: Learning to learn high capacity generative models from few examples. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 988–997, 2018.

[22] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2019.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*, 2014.

[25] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, 2015.

[26] Jedrzej Kozerawski and Matthew Turk. Clear: Cumulative learning for one-shot one-class image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3446–3455, 2018.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[28] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33th Annual Meeting of the Cognitive Science Society*, 2011.

[29] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot

learning. In *7th International Conference on Learning Representations*, 2019.

[30] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, pages 2579–2605, 2008.

[31] Claudio Michaelis, Matthias Bethge, and Alexander S Ecker. One-shot segmentation in clutter. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[32] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *6th International Conference on Learning Representations*, 2018.

[33] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[34] Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 719–729, 2018.

[35] Hang Qi, Matthew Brown, and David G. Lowe. Low-shot learning with imprinted weights. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 5822–5830, 2018.

[36] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L. Yuille. Few-shot image recognition by predicting parameters from activations. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 7229–7238, 2018.

[37] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations*, 2017.

[38] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *6th International Conference on Learning Representations*, 2018.

[39] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, pages 1842–1850, 2016.

[40] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations*, 2015.

[42] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[43] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[44] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Web-scale training for face identification. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2746–2754, 2015.

[45] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[46] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[47] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems*, pages 2371–2380, 2018.

[48] Fang Zhao, Jian Zhao, Shuicheng Yan, and Jiashi Feng. Dynamic conditional networks for few-shot learning. In *The European Conference on Computer Vision*, pages 20–36, 2018.

[49] Wei Zhao, Jiancheng Yang, Yingli Sun, Cheng Li, Weilan Wu, Liang Jin, Zhiming Yang, Bingbing Ni, Pan Gao, Peijun Wang, Yanqing Hua, and Ming Li. 3d deep learning from ct scans predicts tumor invasiveness of subcentimeter pulmonary adenocarcinomas. *Cancer Research*, 78(24):6881–6889, 2018.