Proceedings of Machine Learning Research 140:29–37, 2021 AAAI Workshop on Meta-Learning and MetaDL Challenge

Transfer learning based few-shot classification using optimal transport mapping from preprocessed latent space of backbone neural network

Tomáš Chobola CHOBOTO1@FIT.CVUT.CZ
Daniel Vašata DANIEL.VASATA@FIT.CVUT.CZ
Pavel Kordík PAVEL.KORDIK@FIT.CVUT.CZ

Editors: Isabelle Guyon, Jan N. van Rijn, Sébastien Treguer, Joaquin Vanschoren

Abstract

The MetaDL Challenge 2020 focused on image classification tasks in few-shot settings. This paper describes second best submission in the competition. Our meta learning approach modifies the distribution of classes in a latent space produced by a backbone network for each class in order to better follow the Gaussian distribution. After this operation which we call Latent Space Transform algorithm, centers of classes are further aligned in an iterative fashion of the Expectation Maximisation algorithm to utilize information in unlabeled data that are often provided on top of few labelled instances. For this task, we utilize optimal transport mapping using the Sinkhorn algorithm. Our experiments show that this approach outperforms previous works as well as other variants of the algorithm, using K-Nearest Neighbour algorithm, Gaussian Mixture Models, etc.

1. Introduction

Few-shot learning is increasingly popular because it can handle machine learning tasks with just a few learning examples. It is also more biologically plausible and closer to what we observe in nature. While learning a new task, one normally does not start from a randomly initialised neural network presenting hundreds of thousands of examples in several thousand epochs.

When you are told to remember a person from a picture, you are able to distinguish this person from others even when you see her in different positions or environments. In machine learning, this is called one shot learning. The task of one shot learning is to learn new classes given only one instance available for each class. Three-way five-shot learning means learning three classes given five training instances each. You do not learn classifiers from scratch, but you typically use neural networks trained on similar tasks using much more data. This also reflects the natural situation when the visual perception is already well trained on similar tasks when trying to remember a new person from the picture. This process can be also called meta learning or transfer learning as one uses a pretrained neural network called a backbone network. Also, in a few-shot learning scenario, you can often utilise unlabelled instances apart of those few labelled samples that are available for the task.

The MetaDL challenge 2020[1] focused on few shot learning of image classification tasks. Participants trained a meta-learner on a meta-train set and produced a learner which was subsequently used to train on classification

1. https://competitions.codalab.org/competitions/26638

tasks generated from the meta-test set and evaluated. The goal was to discover learners with the ability to quickly adapt to new unseen image classification tasks.

Our submissions scored second on the final leaderboard. This paper describes methods we have experimented with and the architecture of the meta-learning pipeline responsible for second best result in the competition. The architecture of our solution mainly follows Hu et al. (2020) with important improvements in the preprocessing of latent space output of the backbone model B. The main improvement is in the different normalization of the transformed feature vectors which resembles the Gaussian distribution assumption better. Since this is the key assumption for the proper functionality of the Sinkhorn mapping algorithm, it leads to more accurate results.

2. Related Work

There are several different approaches to few shot learning. The survey (Wang et al., 2020) is a good resource to learn about the general overview and taxonomy of few shot learning methods. Prototypical networks (Snell et al., 2017) and the Siamese networks (Koch et al., 2015) focus on learning embeddings transforming the data in a way that it can be recognised with a simple classifier. This approach is further enhanced by relation networks (Sung et al., 2018) which is able to classify images of new classes by predicting distances between query images and the few examples of each new class.

Another interesting direction aims at the learning process itself. In Ravi and Larochelle (2017) a recurrent network based meta-learner model learns the exact optimization algorithm used to train another learner neural network classifier in the few-shot setup. Meta-transfer learning (Sun et al., 2019) adapts a deep neural network for few shot learning tasks. The transfer is achieved by learning the scaling and shifting functions of DNN weights for each task. We further extend the direction of few-shot learning research that is leveraging classification capabilities in robust backbone models (neural networks) pretrained on similar tasks. These transfer learning based methods need to find a mapping of few-shot classes to similar classes used to train the backbone model.

In Rohrbach et al. (2013) the Propagated Semantic Transfer has been applied to employ semantic knowledge transfer to original classes, combine the transferred predictions with labels for the novel classes, exploit the manifold structure of novel classes by graph based learning and improve the local neighborhood in such graph structures by replacing the raw feature-based representation with an attribute-based representation.

When transferring the knowledge, deep embeddings are far superior, compared to weight transfer, as a starting point for novel tasks as investigated by Scott et al. (2018). Another similar approach is TransMatch (Yu et al., 2020), where a feature extractor is pretrained on original classes and subsequently used to initialize few-shot classifier weights for the novel classes, the classifier is also updated with a semisupervised learning method.

Our research proceeds from Hu et al. (2020), where the latent space produced by a backbone deep network is preprocessed by a power transform and optimal-transport algorithm maps original classes to novel classes while centres on new classes are iteratively adjusted. This approach has shown significant improvement in accuracy in our experiments. The importance of feature transformation for few-shot learning is confirmed by Wang et al. (2019).

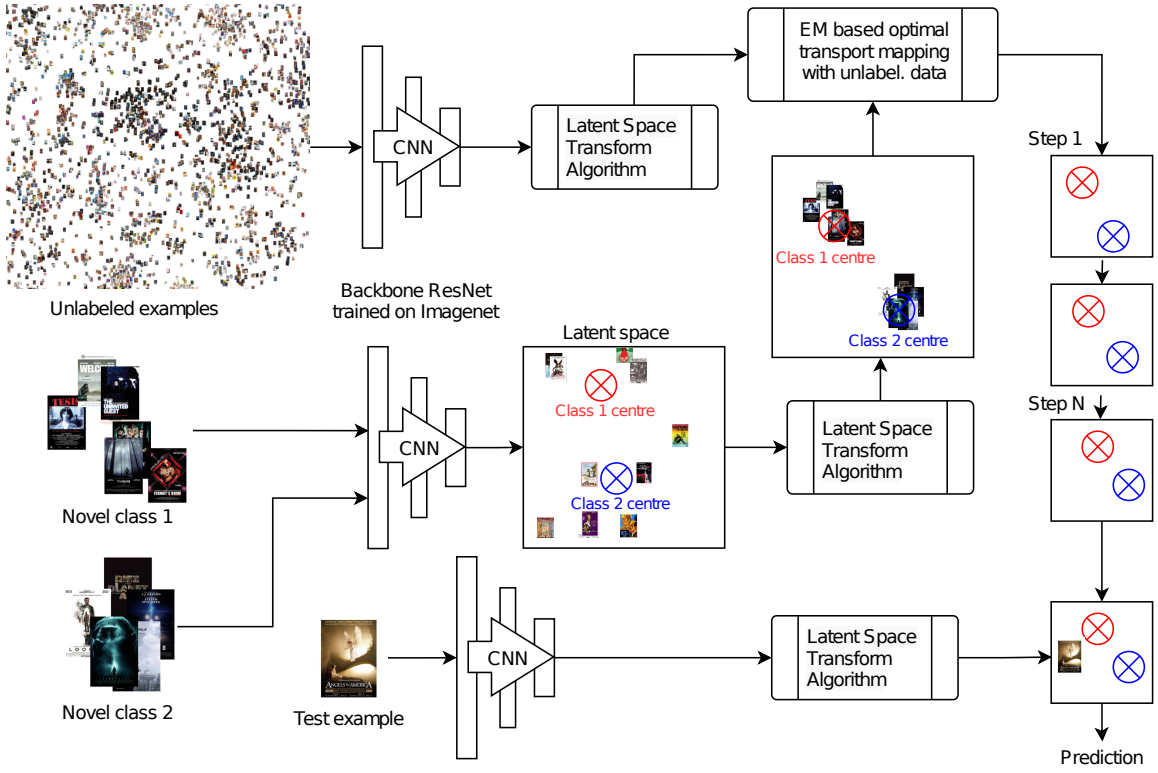

Figure 1: In order to predict the class label of a test example, we transform the image using a backbone CNN to the latent space and preprocess vectors by the Latent Space Transform algorithm that helps to transform the distribution of individual classes to Gaussian like. Then a test example is processed and compared to the class centres that have been iteratively adjusted using a Sinkhorn mapping with unlabeled data projected to the latent space in the same way. The closest class is assigned to the test example as the prediction.

3. Model description

Formally, in a few-shot learning task one has a dataset D containing a part D_S with a few labelled samples from w classes and a part D_Q with some unlabelled samples. The goal is to predict the classes for samples in D_Q. We will assume that D_S contains exactly s labelled samples for each class and D_Q contains exactly q unlabelled samples for each class. Hence, there are ws samples in D_S and wq samples in D_Q. The i-th sample from D will be denoted by x_i and if it is from D_S we will denote its label by y_i.

Moreover, let us assume that there is another dataset D_B corresponding to some related task, such as image classification to some novel classes. This dataset can be used to train the backbone model b which maps the initial space into some latent feature space $\mathcal{L} = \mathbb{R}^d$. In order to train such a model one might train the neural network for classification and then remove the last classification layers as we did in the experiments. Or an encoder part of an autoencoder might be used.

The next step is to preprocess the points in the latent space to be prepared for the final prediction algorithm that estimates the labels. As was recently researched this step is crucial and may lead to significant improvements of the result, see Wang et al. (2019). To proceed we will further assume that the features obtained from the backbone model B are non-negative, i.e. $\mathcal{L} = \mathbb{R}^d_+$. This is often the case when one extracts b as a part of some neural network with the ReLU activation function on inner layers. Let us denote by B the dataset D transformed by b and by B_S and B_Q its parts corresponding to D_S and D_Q, respectively.

In the preprocessing, we transform the dataset D of points in the latent space $\mathcal{L}$ to a final dataset F of points in the final feature space $\mathcal{F} = \mathbb{R}^r$, where the dimension $r = \min\{d, w(s+q)\}$ is the minimum of the dimension d of $\mathcal{L}$ and the number of points in the dataset D. The preprocessing is a composition of three steps and we will call it the Latent Space Transform algorithm (LST). The first is the power transform combined with the semi-normalization of each point given by

$$f_1(u) = \frac{(u+\varepsilon)^\beta}{\|(u+\varepsilon)^\beta\|_2^\delta} \quad \text{for all } u \in \mathcal{L},$$

where the power is taken component-wise, $\varepsilon = 10^{-6}$ is the normalization parameter, and $\|\cdot\|$ is the Euclidean norm. The hyperparameter β controls the strength of the power transform and the hyperparameter δ controls the strength of the normalization, where $\delta = 1$ means the full normalization and $\delta = 0$ yields no normalization at all. The power transform is known to help stabilising the variance and making the data more Gaussian distribution-like by reducing its skewness, see Box and Cox (1964). The normalization on the other hand leads to the projection on the unit sphere which is not compatible with the assumption used later in the optimal-transport that the components of points in the same class are independent with Gaussian distribution of the same variance. Hence, the semi-normalization controlled by the hyperparameter δ enables for having some variance in the perpendicular direction to the unit sphere surface and thus does not a priori break the compatibility of the resulting distribution with the Gaussian assumption. Let us denote the dataset with all points in B transformed using f_1 by F_1 and $F_{1,S}, F_{1,Q}$ analogously.

The second step is the removal of unnecessary dimensions using the QR decomposition of the transposition of the already preprocessed data matrix $\mathbf{F}_1 \in \mathbb{R}^{w(s+q),d}$ corresponding to dataset F_1,

$$\mathbf{F}_1^T = \mathbf{QR}$$

and thus we define

$$\mathbf{F}_2 = \mathbf{F}_1\mathbf{Q}$$

so that $\mathbf{F}_2 \in \mathbb{R}^{w(s+q),r}$, where $r = \min\{d, w(s+q)\}$, and the corresponding dataset is denoted by F_2. We again denote by $F_{2,S}$ and $F_{2,Q}$ the parts of F_2 that corresponds to samples originally in D_S and D_Q, respectively. It corresponds to the change of the orthonormal basis in the $\mathbb{R}^d$ and throwing away the dimensions that are zero for the data points.

The last preprocessing step is the centering and further semi-normalization given by

$$f_3(u) = \frac{u - \bar{u}}{\|u\|_2^\gamma},$$

where

$$\bar{u} = \frac{1}{w(s+q)} \sum_{i=1}^{w(s+q)} u_i$$

is the centroid (component-wise average) of the dataset F_2. Again, the hyperparameter γ allows to control the strength of the normalization. For $\gamma < 1$ the resulting points are

only partially normalized and one may expect to better resemble the Gaussian distribution assumed in the next step. The typical result for the final Euclidean norms of transformed points is shown in Figure 2.

Let us denote the final preprocessed dataset by F and its respective parts corresponding to original parts D_S and D_Q by F_S and F_Q, respectively.

Once the preprocessing of the dataset is finished, the actual optimal-transport can begin. In this part we directly follow Hu et al. (2020). The preliminary assumption of the method is the independent Gaussian distributions of all components of points in individual classes with class centres $c_1, \ldots, c_w$ as parameters. Moreover, it is assumed that all the Gaussian distributions have the same variance $\lambda/2$, where λ is the hyperparameter. Under this assumption the maximum a posteriori estimate (MAP) $\hat{y}_1, \ldots, \hat{y}_{wq}$ of the labels of unlabelled samples $f_1, \ldots, f_{wq}$ from F_Q corresponds to

$$\begin{aligned}
&\{\hat{y}_j\}_{j=1}^{wq}, \{\hat{c}_k\}_{k=1}^{w} \\
&= \arg \max_{\{y_j\},\{c_k\}} \prod_i P(y_i|f_i) \\
&= \arg \max_{\{y_j\},\{c_k\}} \prod_i P(f_i|y_i)P(y_i) \\
&= \arg \max_{\{y_j\},\{c_k\}} \prod_i e^{-\lambda^{-1}\|f_i - c_{y_i}\|_2^2} P(y_i).
\end{aligned}$$

This is directly related to the Optimal Transport theory, see Hu et al. (2020); Cuturi (2013); Berman (2020); Villani (2003), and one may use the iterative expectation-maximization like approach incorporating the Sinkhorn algorithm to get the MAP estimate. It consists of repeating two steps, where the first is the construction of the mapping matrix $\mathbf{M}^*$ with elements $\mathbf{M}^*_{ij} = P(y_i = j)$ which is maximizing the previous term for a given centres $c_1, \ldots, c_w$ and the second step is the estimation of class centres that is for the fixed mapping matrix again optimizing the previous term. For the Sinkhorn algorithm, see

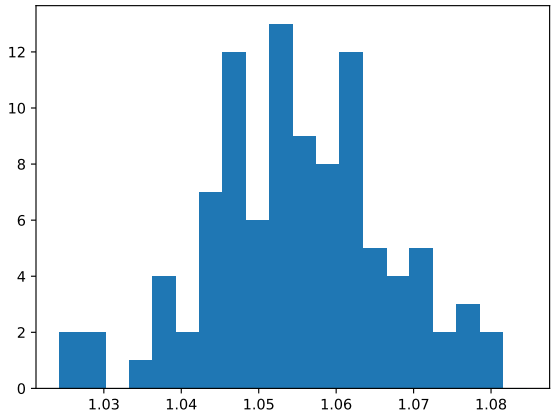

Figure 2: Latent Space Transform algorithm produces Gaussian like distribution also for the norms of the transformed samples. The figure was produced for one batch from the CUB dataset with $s = 5, q = 15, \beta = 0.5, \delta = 0.3$, and $\gamma = 0.9$.

Cuturi (2013) the mapping matrix is defined as

$$\begin{aligned}
\mathbf{M}^* &= \text{Sinkhorn}(\mathbf{L}, a, b, \lambda) \\
&= \arg \min_{\mathbf{M} \in U(a,b)} \sum_{i,j} \mathbf{M}_{ij}\mathbf{L}_{ij} + \lambda H(\mathbf{M}),
\end{aligned}$$

where $U(a, b)$ is a set of positive matrices in $\mathbb{R}^{wq \times w}$ for which the rows sums to a vector a and columns sums to a vector b, $\mathbf{L} \in R^{wq \times w}$ is the cost function consisting of Euclidean distances between unlabelled instances and class centres, that is $\mathbf{L}_{ij} = \|f_i - c_j\|_2^2$, the hyperparameter λ is a regularisation coefficient forcing the entropy $H(\mathbf{M}) = -\sum_{ij} \mathbf{M}_{ij} \log \mathbf{M}_{ij}$ to become smaller, a denotes the distribution of the amount that each unlabelled example uses for class allocation, i.e. a is the vector of ones with wq elements, and b denotes the distribution of the amount of unlabelled examples allocated to each class, i.e. b is the vector with w elements that equals to q.

The iterative approach starts with initialising the class centres from the labelled samples in F_S. Then the mapping matrix $\mathbf{M}^*$ is calculated using the Sinkhorn algorithm. It is then used to re-estimate the class centres via the update using

$$\mu_j = \frac{\sum_{f_i \in F_Q} \mathbf{M}^*_{ij} f_i + \sum_{f_k \in F_S, y_k = j} f_k}{s + \sum_{i=1}^{wq} \mathbf{M}^*_{ij}}.$$

To avoid unnecessarily big steps in centre estimations, the new centre is set to be $c_j = c_j + \alpha(\mu_j - c_j)$, where the α is the learning rate. The number of iterations is fixed to n_{steps}. Once the iteration process finishes, the labels of the samples from F_Q might be estimated from the last mapping matrix as

$$\hat{y}_i = \arg\max_j \mathbf{M}^*_{ij}.$$

The overview of the algorithm is given in Algorithm 1. The overall process of our approach is depicted in Figure 1. The code is available at https://github.com/ctom2/latent-space-transform.

Algorithm 1: Optimal map algorithm

Parameters: $w, s, q, \lambda, \alpha, n_{\text{steps}}$
Initialisation: $c_j = \frac{1}{s} \sum_{f_k \in F_S, y_k = j} f_k$
repeat n_{steps} **times:**
  $\mathbf{L}_{ij} = ||f_i - c_j||^2, \forall i, j$
  $\mathbf{M}^* = \text{Sinkhorn}(\mathbf{L}, p = 1_{wq}, q = q1_w, \lambda)$
  Calculate μ_j $c_j = c_j + \alpha(\mu_j - c_j)$
end
return $\hat{y}_i = \arg\max_j \mathbf{M}^*_{ij}$

4. Experiments

The performance of the stated methods was measured based on standardised few-shot classification datasets CIFAR-FS (Bertinetto et al., 2019) and CUB (Wah et al., 2011). CIFAR-FS dataset consists of images with a size of 32×32 distributed into 100 classes, each containing 600 images. The dataset is split into 64 base classes, 16 validation classes and 20 novel classes. CUB dataset contains 11,788 images of birds, each with size 84×84, distributed over 200 classes. The dataset is split into 100 base classes, 50 validation classes and 50 novel classes.

Table 1: Hyperparameters used in the final evaluation of the LST+MAP model.

	1-shot		**5-shot**	
Parameter	CIFAR-FS	CUB	CIFAR-FS	CUB
β	0.5	0.5	0.5	0.5
λ	10	10	10	10
α	0.3	0.4	0.2	0.2
n_{steps}	20	30	20	20
δ	0.3	0.7	0.4	0.3
γ	0.98	0.95	0.95	0.9

In each testing run, w classes are randomly and uniformly drawn from novel classes, where each class consists of s instances with a label and q instances without a label.

We chose the WideResNet (Zagoruyko and Komodakis, 2017) augmented with the S2M2 method (Mangla et al., 2020) as the backbone architecture for our model because of its high performance in the few-shot setting. The latent representation of images produced by the backbone is a vector with a dimension of 640. The QR decomposition reduces the said dimension to 80 in 1-shot setting, and to 100 in 5-shot setting.

All experiments are based on $w = 5, q = 15$ and $s = 1$ or 5. To evaluate the performance of the models we run 10,000 random draws to obtain mean accuracy with 95% confidence scores. By tuning the hyperparameters of the model we observed evolution in accuracy in both 1-shot and 5-shot setting with dependency on tested dataset. The overview with the hyperparameters can be found in Table 1. The final accuracy can be seen in Table 2 and Table 3 for 1-shot and 5-shot setting, respectively. Moreover,

Table 2: 1-shot accuracy of models based on Power Transform (PT), our proposed Latent Space Transform (LST) and WideResNet backbone.

Method	Backbone	1-shot CIFAR	1-shot CUB
PT+MAP	WRN	$87.69 \pm 0.23\%$	$91.55 \pm 0.19\%$
PT+GMM	WRN	$86.96 \pm 0.22\%$	$90.06 \pm 0.18\%$
PT+KNN	WRN	$86.17 \pm 0.19\%$	$89.07 \pm 0.17\%$
LST+MAP	WRN	$\mathbf{87.79 \pm 0.23\%}$	$\mathbf{91.68 \pm 0.19\%}$
LST+GMM	WRN	$87.01 \pm 0.21\%$	$89.9 \pm 0.18\%$
LST+KNN	WRN	$85.76 \pm 0.19\%$	$89.26 \pm 0.17\%$

the tables include results obtained by substituting MAP with different clustering algorithms, Gaussian Mixture model and k-means model, that take the transformed features as their input. The k-means model is initiated with centres corresponding to the labeled instances in a testing run. The centres are then iteratively refined to produce better representations of the class centres. Similarly, Gaussian Mixture model is provided with initial means corresponding to the labeled examples at the beginning of each run. To compare our proposed transform method with the Power Transform (PT) (Hu et al., 2020), we performed the same substitutions for the PT+MAP model.

The scores show that even by omitting the MAP part from the architecture and replacing it with simpler classification approaches while keeping the transformation intact produces competitive results. Moreover, to compare the statistical significance of the superiority of the LST+MAP model against the PT+MAP model we performed the paired t-test with p-values presented in Table 4. We can see that except for the CUB dataset in 5-shot scenario the LST+MAP model is significantly better than the PT+MAP model.

In terms of execution time, we measured an average of $0.0026s$ per run in 1-shot setting and $0.003s$ per run in 5-shot setting with the GPU backend.

Table 3: 5-shot accuracy of models based on Power Transform (PT), our proposed Latent Space Transform (LST) and WideResNet backbone. The authors of the PT+MAP model presented accuracy $93.99 \pm 0.10\%$ in 5-shot setting for CUB dataset, however we were able to obtain higher accuracy with their described model configuration.

Method	Backbone	5-shot CIFAR	5-shot CUB
PT+MAP	WRN	$90.68 \pm 0.15\%$	$94.09 \pm 0.09\%$
PT+GMM	WRN	$87.16 \pm 0.21\%$	$90.04 \pm 0.20\%$
PT+KNN	WRN	$86.70 \pm 0.19\%$	$89.72 \pm 0.18\%$
LST+MAP	WRN	$\mathbf{90.73 \pm 0.15\%}$	$\mathbf{94.09 \pm 0.09\%}$
LST+GMM	WRN	$87.33 \pm 0.20\%$	$90.06 \pm 0.18\%$
LST+KNN	WRN	$86.56 \pm 0.18\%$	$89.64 \pm 0.18\%$

Table 4: p-values of the paired t-test with the null hypothesis that the accuracy of the PT+MAP model is greater or equal than the accuracy of the LST+MAP model against the alternative that the accuracy of the PT+MAP model is smaller than the accuracy of the LST+MAP model.

	1-shot CIFAR-FS	1-shot CUB	5-shot CIFAR-FS	5-shot CUB
p-value	9.09e−5	1.99e−9	1.68e−7	0.78

5. Challenge submission

In this section, we describe the modification to our method we have elaborated for the MetaDL challenge 2020. The main limitation of the challenge was the submission runtime which had to include backbone training time and was limited to two hours. Therefore we were not able to utilise the WRN backbone as we suggest above.

Our best performing solution was relying on a lighter backbone network based on the

ResNet architecture. During the backbone training, the fed images could either be left as they were, or their saturation or brightness could be changed with the probability set to 1/3 for each alteration. Moreover, the training batches also included the same images rotated by 90, 180 and 270 degrees to further improve the backbone capabilities and augment the training overall.

6. Conclusion

Extracted features from backbones often do not resemble Gaussian-like distributions, even though multiple algorithms are built on that assumption. In this paper we show how to transform feature vectors into better Gaussian-like distributions. By applying an iterative optimal-transport algorithm to estimate class centres empirically, the subsequent clustering method gains significant improvement over other few-shot classification methods.

Our experiments confirmed that the Latent Space Transform algorithm introduced above outperforms other forms of feature preprocessing including the Power Transform. We have also compared our approach based on optimal transport mapping to other classification methods based on Gaussian mixtures and nearest neighbours. For both CIFAR and CUB datasets, our approach proved to be superior in both 1-shot and 5-shot learning scenarios. We have adjusted our method for the MetaDL challenge 2020 competition and scored second on the final leaderboard.

Acknowledgments

Our research has been supported by the Grant Agency of the Czech Technical University in Prague (SGS20/213/OHK3/3T/18) and the Czech Science Foundation (GAČR 18-18080S).

References

Robert J. Berman. The Sinkhorn algorithm, parabolic optimal transport and geometric Monge–Ampère equations. *Numerische Mathematik*, 145:771–836, 2020. doi: https://doi.org/10.1007/s00211-020-01127-x.

Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers, 2019.

G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964. ISSN 00359246. URL http://www.jstor.org/stable/2984418.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 2292–2300. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf.

Yuqing Hu, Vincent Gripon, and S. Pateux. Leveraging the feature distribution in transfer-based few-shot learning. *ArXiv*, abs/2006.03806, 2020.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

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.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=rJY0-Kcll.

Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In *Advances in neural information processing systems*, pages 46–54, 2013.

Tyler Scott, Karl Ridgeway, and Michael C Mozer. Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning. In *Advances in Neural Information Processing Systems*, pages 76–85, 2018.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

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.

Cédric Villani. *Topics in optimal transportation*. American mathematical society, Providence, Rhode Island, 2003. ISBN 0-8218-3312-X.

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.

Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *ArXiv*, abs/1911.04623, 2019.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12856–12864, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017.