Transductive Zero-Shot and Few-Shot CLIP

Supplementary Material

A. Illustration of a Dirichlet distribution

Figure 2 presents examples of Dirichlet distributions on the unit simplex of $\mathbb{R}^3$.

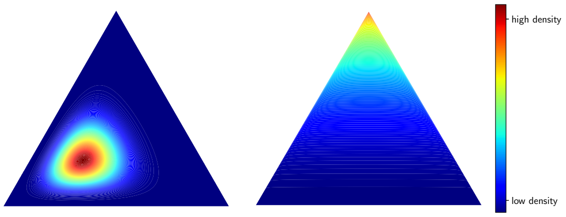

Figure 2. Examples of Dirichlet distributions on the simplex of $\mathbb{R}^3$, for $\boldsymbol{\alpha} = (10, 5.0, 5.0)$ (left) and $\boldsymbol{\alpha} = (0.975, 0.975, 3.0)$ (right)

B. Majorization-Minimization algorithm

We provide the details for our new MM Algorithm 1 for minimizing (10). Our approach is based on constructing a quadratic bound of the function $\ln\Gamma(\cdot + 1)$, which is a consequence of the following lemma.

Lemma 2 ([15])**.** *Let ψ be a twice-continuously differentiable function on $[0, +\infty[$. Assume that ψ'' is decreasing on $[0, +\infty[$. Let $z \in [0, +\infty[$ and let*

$$c_\psi(z) = \begin{cases} \psi''(0) & \text{if } z = 0 \\ 2\dfrac{\psi(0) - \psi(z) + \psi'(z)z}{z^2} & \text{otherwise.} \end{cases} \tag{17}$$

Then, for every $x \in [0, +\infty[$,

$$\psi(x) \le \psi(z) + \psi'(z)(x - z) + \frac{1}{2}c_\psi(z)(x - z)^2. \tag{18}$$

We are now ready to prove Lemma 1.

Proof. We first observe that $\boldsymbol{\alpha}_k \mapsto -\ln\Gamma\left(\sum_{i=1}^K \alpha_{k,i}\right)$ is concave. Consequently, we can upper-bound this term at $\boldsymbol{\beta}_k$ using its first-order Taylor expansion around $\boldsymbol{\beta}_k$. Furthermore, considering the relation

$$\forall t \in (0, +\infty), \quad \ln\Gamma(t) = \varphi(t) - \ln t, \tag{19}$$

and given that the prerequisites of Lemma 2 are fulfilled by φ, the result in (13) follows immediately. □

For a fixed value of $\boldsymbol{\beta}_k \in (0, +\infty)^K$, the minimizer $\widehat{\boldsymbol{\alpha}}_k$ of the majorant given by Lemma 1 is such that, for every $i \in \{1, \ldots, K\}$, $\widehat{\alpha}_{k,i}$ is the unique positive root of the second order polynomial equation

$$c(\beta_{k,i})\alpha_{k,i}^2 + b_{k,i}(\boldsymbol{\beta}_k)\alpha_{k,i} = 1, \tag{20}$$

with

$$\begin{aligned} b_{k,i}(\boldsymbol{\beta}_k) = \varphi'(\beta_{k,i}) - (\ln\Gamma)'\Big(\sum_{j=1}^K \beta_{k,j}\Big) - c(\beta_{k,i})\beta_{k,i} \\ - \left(\sum_{n=1}^N u_{n,k}\right)^{-1} \sum_{n=1}^N u_{n,k} \ln z_{n,i}. \end{aligned} \tag{21}$$

Hence,

$$\widehat{\alpha}_{k,i} = \frac{-b_{k,i}(\boldsymbol{\beta}_k) + \sqrt{\big(b_{k,i}(\boldsymbol{\beta}_k)\big)^2 + 4c(\beta_{k,i})}}{2c(\beta_{k,i})}, \tag{22}$$

which yields the MM updates described in Algorithm 1.

In Table 3, we compare the convergence speed of the MM Algorithm 1 and the Block MM Algorithm 2, using our majorant (13) versus the one proposed by Minka in [35]. For Algorithm 1, the convergence criterion is defined as $\frac{\|\boldsymbol{\alpha}^{(m+1)} - \boldsymbol{\alpha}^{(m)}\|^2}{\|\boldsymbol{\alpha}^{(m)}\|^2} \le \varepsilon$, and for Algorithm 2 as $\frac{\|\boldsymbol{\alpha}^{(\ell+1)} - \boldsymbol{\alpha}^{(\ell)}\|^2}{\|\boldsymbol{\alpha}^{(\ell)}\|^2} \le \varepsilon$, where $\varepsilon = 10^{-13}$. Our MM algorithm is approximately twice as fast as Minka's.

	Algo. 1	Algo. 2
Minka's [35]	2.04×10^{-1}	2.09
Ours	7.62×10^{-2}	1.04

Table 3. Time before reaching the convergence criterion in seconds, for Algorithm 1 and 2. The displayed time is the average execution time per task, computed over 1,000 tasks, on the ImageNet dataset with 4 shots.

C. Estimation step on assignments in our algorithm

We provide more details on the derivation of the closed-form update of variable $\boldsymbol{u}_n$ at each iteration $\ell \in \mathbb{N}$. Consider the function F given by

$$\begin{aligned} F(\boldsymbol{u}_n) = -\sum_{k=1}^K u_{n,k} \ln\left(\mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_k\right)\right) + \iota_{\Delta_K}(\boldsymbol{u}_n) \\ - \frac{\lambda}{|\mathbb{Q}|}(\ln(\boldsymbol{\pi}^{(\ell+1)}) + \mathbf{1})^\top(\boldsymbol{u}_n - \boldsymbol{u}_n^{(\ell)}) + \sum_{k=1}^K u_{n,k} \ln u_{n,k}, \end{aligned} \tag{23}$$

where ι_{Δ_K} is the indicator function of the simplex Δ_K, assigning zero to points within the simplex and $+\infty$ elsewhere.

Let us see how to compute the minimizer of (23) via the proximal operator (see [2, Eq. 24.2] for a definition). We define the function ψ on $\mathbb{R}^K$ as

$$\psi(\boldsymbol{x}) = \begin{cases} \sum_{k=1}^K x_k \ln(x_k) - \frac{x_k^2}{2}, & \text{if } \boldsymbol{x} \in \Delta_K, \\ +\infty, & \text{otherwise.} \end{cases} \tag{24}$$

The proximal operator of ψ, which is well-established as the softmax function, allows for the practical computation of the minimizer [13, Example 2.23]. Since F is proper, lower semi continuous and convex, finding the minimizer of F is equivalent to finding $\boldsymbol{u}_n$ such that $0 \in \partial F(\boldsymbol{u}_n)$. This reads

$$\begin{aligned} & 0 \in \partial F(\boldsymbol{u}_n) \\ \iff\quad & 0 \in -\ln(\mathrm{p}\,(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_k)) - \frac{\lambda}{|\mathbb{Q}|}(\ln(\boldsymbol{\pi}^{(\ell+1)}) + 1) \\ & \qquad + \partial\psi(\boldsymbol{u}_n) + \boldsymbol{u}_n, \\ \iff\quad & \boldsymbol{u}_n = \mathrm{softmax}\left(\left(\ln \mathrm{p}\,(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_k) + \frac{\lambda}{|\mathbb{Q}|}\ln \pi_k^{(\ell+1)}\right)_k\right), \end{aligned}$$

where we used the characterization of the proximity operator [2, Prop. 16.44].

D. Class-assignment in the zero-shot setting

Figure 3 gives an illustration of our graph matching procedure for assigning each cluster to a unique class.

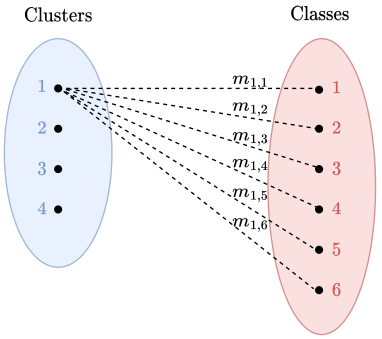

Figure 3. Illustration of the bipartite matching for class assignment.

Note that it is possible to not perform the graph matching procedure and simply assign to each cluster $k \in \mathcal{K}$ the class $\ell^* \in \{1,\ldots,K\}$ such that $\ell^* = \underset{\ell\in\{1,\ldots,K\}}{\operatorname{argmax}}\ m_{k,\ell}$, where $\mathbf{m}_k = (m_{k,\ell})_{1\le\ell\le K}$ is the average of simplex features assigned to cluster k. However, this leads in practice to multiple clusters being assigned to the same class. We nevertheless provide the zero-shot accuracy results in Table 6.

E. Links with the EM algorithm

We give a proof of Proposition 1.

Proof. Given the mixture model (16), the EM algorithm aims at maximizing the log-likelihood function

$$L(\boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{n\in\mathbb{Q}} \ln\left(\sum_{k=1}^K \pi_k \mathrm{p}\,(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_k)\right) \tag{25}$$

with respect to $\boldsymbol{\pi}$ and $\boldsymbol{\alpha}$. The process involves two steps: **expectation** and **maximization**, and the algorithm iteratively generates sequences $\{\boldsymbol{\pi}^{(\ell)}\}_{\ell\in\mathbb{N}} \subset \Delta_K$ and, for every $k \in \{1,\ldots,K\}$, $\{\boldsymbol{\alpha}_k^{(\ell)}\}_{\ell\in\mathbb{N}} \subset (0,+\infty)^K$.

During the **expectation step**, for a given iteration number $\ell \in \mathbb{N}$, we compute the expected responsibilities. For each query sample $n \in \mathbb{Q}$, we define $\boldsymbol{u}_n^{(\ell)} = (u_{n,k}^{(\ell)})_{1\le k\le K}$ by

$$u_{n,k}^{(\ell)} = \frac{\pi_k^{(\ell)} \mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_k^{(\ell)}\right)}{\sum_{i=1}^K \pi_i^{(\ell)} \mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_i^{(\ell)}\right)}. \tag{26}$$

This quantity corresponds to the probability of the data point n belonging to class k based on the current estimates of $\boldsymbol{\pi}^{(\ell)}$ and $\boldsymbol{\alpha}_k^{(\ell)}$.

In the **maximization step**, we derive an upper bound for the log-likelihood at the current iterate using the responsibilities calculated in the expectation step, along with Jensen's inequality. This majorization reads

$$L(\boldsymbol{\pi}, \boldsymbol{\alpha}) \le q((\boldsymbol{\pi}, \boldsymbol{\alpha}); (\boldsymbol{\pi}^{(\ell)}, \boldsymbol{\alpha}^{(\ell)})), \tag{27}$$

where $q(\,\cdot\,; (\boldsymbol{\pi}^{(\ell)}, \boldsymbol{\alpha}^{(\ell)}))$ is defined, for all $\boldsymbol{\pi} \in \Delta_K$ and $\boldsymbol{\alpha} \in ((0,+\infty)^K)^K$, by

$$q((\boldsymbol{\pi}, \boldsymbol{\alpha}); (\boldsymbol{\pi}^{(\ell)}, \boldsymbol{\alpha}^{(\ell)})) = \sum_{n\in\mathbb{Q}}\sum_{k=1}^K u_{n,k}^{(\ell)} \ln\left(\frac{\pi_k \mathrm{p}\,(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_k)}{u_{n,k}^{(\ell)}}\right).$$

This upper bound is separable and defines a tight majorant, i.e., $q((\boldsymbol{\pi}^{(\ell)}, \boldsymbol{\alpha}^{(\ell)}); (\boldsymbol{\pi}^{(\ell)}, \boldsymbol{\alpha}^{(\ell)})) = L(\boldsymbol{\pi}^{(\ell)}, \boldsymbol{\alpha}^{(\ell)})$. Next, one maximizes the majorant with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$ under the simplex constraints. This yields the expression

$$(\forall k \in \{1,\ldots,K\}) \quad \pi_k^{(\ell+1)} = \frac{1}{|\mathbb{Q}|}\sum_{n\in\mathbb{Q}} u_{n,k}^{(\ell)}, \tag{28}$$

i.e., the mixing coefficients are the average of the responsibilities for each class over all data points in the query set. On the other hand, for each class $k \in \{1,\ldots,K\}$, the parameters $\boldsymbol{\alpha}_k^{(\ell+1)}$ are set by solving the optimization problem

$$\underset{\boldsymbol{\alpha}_k\in(0,+\infty)^K}{\text{maximize}} \sum_{n\in\mathbb{Q}} u_{n,k}^{(\ell)} \ln \mathrm{p}\,(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_k)\,. \tag{29}$$

We can then show that the updates are identical to those performed in Algorithm 2 when $\lambda = |\mathbb{Q}|$ and $\mathbb{S} = \varnothing$. The identity of the updates on $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$ are obvious. For $\boldsymbol{u}$, note that Equation (26) can be rewritten

$$\begin{aligned} u_{n,k}^{(\ell+1)} &= \frac{\pi_k^{(\ell+1)} \mathrm{p}(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_k^{(\ell+1)})}{\sum_{i=1}^K \pi_i^{(\ell+1)} \mathrm{p}(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_i^{(\ell+1)})}, \\ &= \frac{\exp\left(\ln \pi_k^{(\ell+1)} + \ln \mathrm{p}(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_k^{(\ell+1)})\right)}{\sum_{i=1}^K \exp\left(\ln \pi_i^{(\ell+1)} + \ln \mathrm{p}(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_i^{(\ell+1)})\right)}, \end{aligned}$$

or equivalently,

$$\boldsymbol{u}_n = \mathrm{softmax}\left((\ln \pi_k^{(\ell+1)} + \ln \mathrm{p}(\boldsymbol{z}_n \mid \boldsymbol{\alpha}_k^{(\ell+1)}))_k\right), \quad (30)$$

thus aligning with the update in Algorithm 2.

□

F. Zero-shot performance as a function of the size the query set

We point to Figure 4 which displays the accuracy of our methods EM-Dirichlet and Hard EM-Dirichlet in the zero-shot setting versus the number of samples in the query set.

G. Additional results in the few-shot setting

In addition to the results in the 4-shot case presented in Table 2, we provide the results for other number of shots. Figure 5 displays the accuracy as a function of the number of shots. This analysis includes our methods EM-Dirichlet and Hard EM-Dirichlet, other transductive methods (BDC-SPN, Laplacian Shot, α-TIM, PADDLE), and the inductive Tip-Adapter method. We did not evaluate CoOp because of the prohibitive time required to run the method, as underlined in Table 2. We observe that our method significantly outperforms its closest competitor, TIP, on the challenging SUN397 and ImageNet datasets, as well as on the average of the 11 datasets. This gap gets even wider when the number of shots increases. Complete results for all datasets are given in Figure 6.

H. Ablation study on each term of the objective

We provide an ablation study on our objective function, which minimizes $-\mathcal{L} + \Phi + \Psi$ under simplex constraints, where $\mathcal{L}$ is the log-likelihood, Φ a barrier term, and Ψ a partition complexity term promoting fewer clusters. Note that, when removing barrier term Φ, our update step for the assignment variables (Eq. (15) without the barrier term) amounts to solving a linear programming problem, resulting in integer solutions (i.e., hard assignments), akin to what we coined "Hard EM-Dirichlet".

Table 4 demonstrates the effect of each term. The partition complexity term Ψ significantly enhances performance. In contrast, the barrier term Φ, in isolation, does not improve performance. However, when combined with Ψ, it shows utility in the 4-shot scenario. The inclusion of Φ was primarily to maintain a soft assignment approach and to make the link with the EM algorithm (Proposition 1).

	Criterion	Acc.
0-shot	$-\mathcal{L}$	50.8
	$-\mathcal{L} + \Phi$	42.7
	$-\mathcal{L} + \Psi$ (= Hard EM-Dirichlet)	67.6
	$-\mathcal{L} + \Phi + \Psi$ (= EM-Dirichlet)	65.8
4-shot	$-\mathcal{L}$	59.5
	$-\mathcal{L} + \Phi$	58.8
	$-\mathcal{L} + \Psi$ (= Hard EM-Dirichlet)	72.9
	$-\mathcal{L} + \Phi + \Psi$ (= EM-Dirichlet)	73.6

Table 4. Average accuracy on the 11 datasets, over 1,000 classification tasks. Inference is performed on the text-vision probability features.

I. Using the similarity scores as feature vectors

One might consider directly using the visual-textual embeddings as input features (specifically, the cosine similarities) without applying a softmax function. It could be hypothesized that methods targeting a Gaussian distribution might perform more effectively with these raw features than with probability features. However, as indicated in Table 5, this is not the case. Employing a Gaussian distribution within the joint visual-textual embedding space actually leads to decreased accuracy when compared to our method that utilizes probability features.

Method	Acc.	Loss in acc.
Soft K-means	28.2	2.1
EM-Gaussian (diag. cov.)	34.9	14.8

Table 5. Average accuracy on the 11 datasets, over 1,000 zero-shot tasks using text-vision features (without softmax). The accuracy loss is measured against the results with probability features.

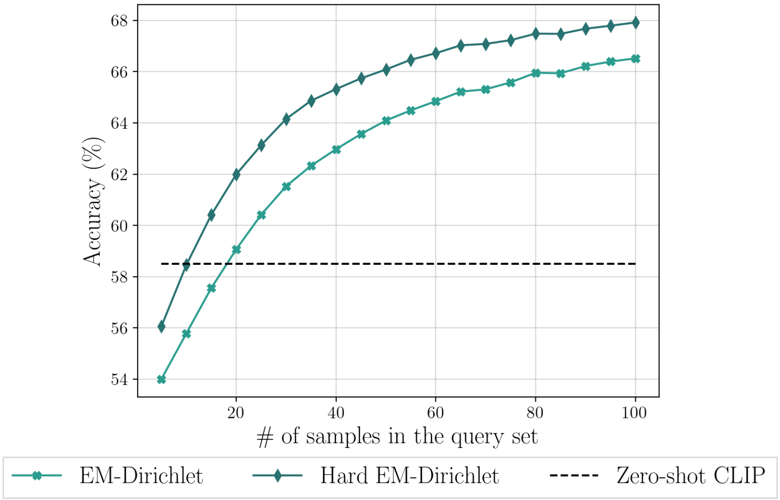

Figure 4. Average accuracy on the 11 datasets as a function of the number of samples in the query set, over 1,000 tasks generated following the protocol described in Section 6.1. As anticipated, the efficiency of transduction increases with the number of samples in the query set.

		Food101	EuroSAT	DTD	OxfordPets	Flowers102	Caltech101	UCF101	FGVC Aircraft	Stanford Cars	SUN397	ImageNet	**Average**
	Zero-shot CLIP	77.1	36.5	42.9	85.1	66.1	84.4	61.7	17.1	55.8	58.6	58.3	58.5
Vis. embs.	Hard K-means	78.4	34.5	46.2	86.3	70.2	87.3	66.1	19.2	58.7	62.9	60.9	61.0
	Soft K-means	79.3	28.4	42.8	67.5	64.7	86.0	62.7	17.7	57.5	59.0	59.3	56.8
	EM-Gaussian (Id cov.)	14.0	14.5	9.4	6.9	5.3	30.3	7.4	1.9	2.5	5.3	3.9	8.3
	EM-Gaussian (diag cov.)	77.1	**37.1**	44.1	86.9	68.9	85.8	63.8	18.4	57.3	60.1	59.3	59.9
Probabilities	Hard K-means	80.2	34.7	45.9	88.7	69.0	86.9	66.6	20.1	59.7	63.7	61.0	61.5
	Soft K-means	43.4	22.1	18.7	67.7	36.2	54.7	31.7	7.6	36.3	18.9	19.1	32.4
	EM-Gaussian (Id cov.)	21.4	14.5	16.5	21.1	23.1	33.6	19.3	6.8	18.5	18.7	19.1	19.3
	EM-Gaussian (diag cov.)	78.9	33.4	44.8	87.9	69.3	86.6	65.7	20.2	63.5	66.1	63.0	61.8
	Hard KL K-means	84.3	34.4	46.2	90.3	72.3	88.3	69.5	21.4	68.6	62.4	61.0	63.5
	EM-Dirichlet	89.0	32.9	48.7	91.2	73.1	90.4	70.5	21.4	69.5	78.1	78.0	67.5
	Hard EM-Dirichlet	**90.7**	33.5	**49.8**	**92.6**	**73.9**	**91.1**	**71.3**	**22.0**	**70.8**	**79.1**	**78.5**	**68.5**

Table 6. Evaluation of the methods computing the accuracy without the graph matching. Average accuracy of clustering methods over 1,000 zero-shot classification tasks. Inference is performed both on the visual embeddings and on the text-vision probability features.

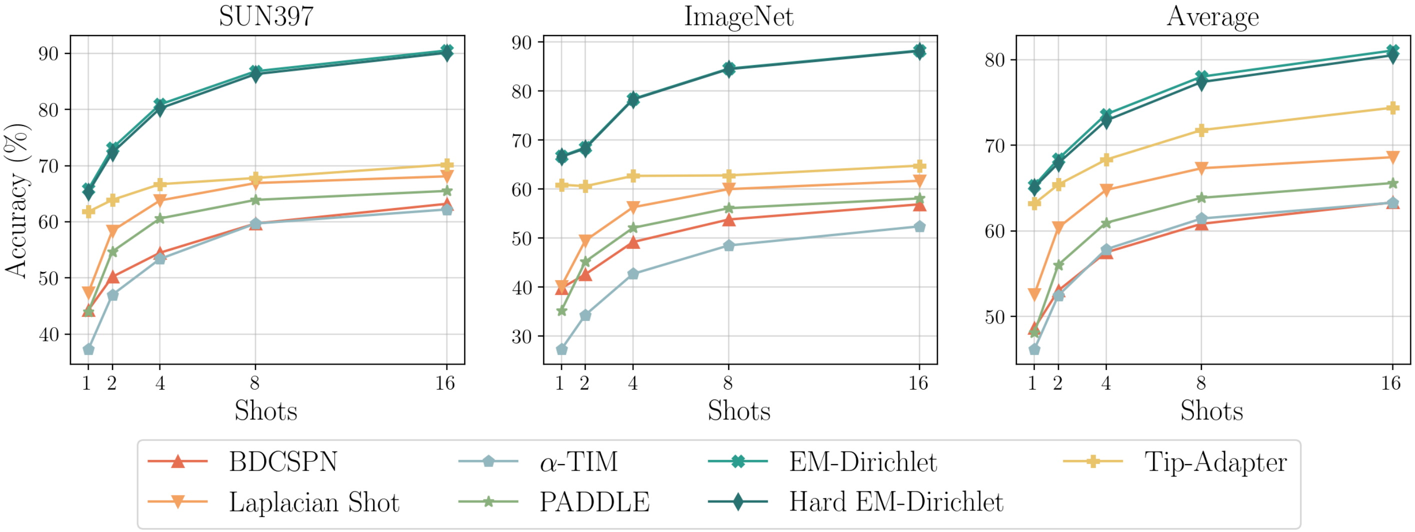

Figure 5. Accuracy versus shots for seven methods from Table 2 on SUN397, ImageNet, and the average across the 11 datasets.

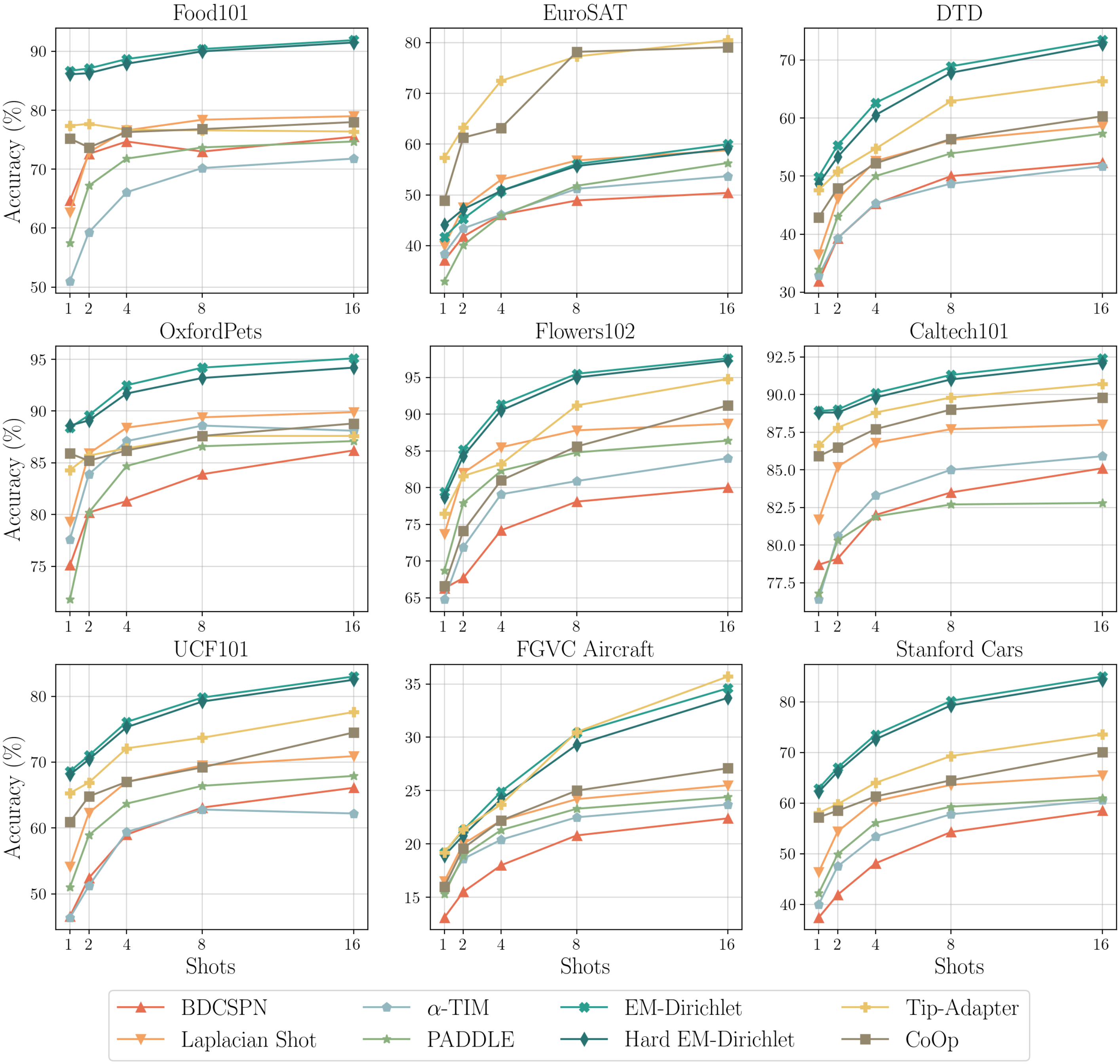

Figure 6. Accuracy versus shots for eight methods from Table 2 on 9 datasets.