Simplex Clustering via sBeta With Applications to Online Adjustment of Black-Box Predictions

Florent Chiaroni, Malik Boudiaf, Amar Mitiche, and Ismail Ben Ayed

Abstract—We explore clustering the softmax predictions of deep neural networks and introduce a novel probabilistic clustering method, referred to as k-sBetas. In the general context of clustering discrete distributions, the existing methods focused on exploring distortion measures tailored to simplex data, such as the KL divergence, as alternatives to the standard euclidean distance. We provide a general maximum a posteriori (MAP) perspective of clustering distributions, emphasizing that the statistical models underlying the existing distortion-based methods may not be descriptive enough. Instead, we optimize a mixed-variable objective measuring data conformity within each cluster to the introduced sBeta density function, whose parameters are constrained and estimated jointly with binary assignment variables. Our versatile formulation approximates various parametric densities for modeling simplex data and enables the control of the cluster-balance bias. This yields highly competitive performances for the unsupervised adjustment of black-box model predictions in various scenarios.

Index Terms—Probability simplex clustering, softmax predictions, deep black-box models, pre-trained, unsupervised adaptation.

I. INTRODUCTION

Over the last decade, deep neural networks have continuously gained wide interest for the semantic analysis of real-world data. However, under real-world conditions, potential shifts in the feature or class distributions may affect the prediction performances of the pre-trained source models. To address this issue, several recent adaptation studies investigated fine-tuning all or a part of the model’s parameters using standard gradient-descent and back-propagation procedures [1], [2], [3]. However, in a breadth of practical applications, e.g., real-time predictions, it might be cumbersome to perform multiple forward and backward passes. This is particularly true when dealing with large (and continuously growing) pre-trained networks, such as the recently emerging foundational vision-language models [4]. In such scenarios, even fine-tuning only parts of the pre-trained model, such as the parameters of the normalization layers [1], might be computationally intensive. In addition, the existing adaptation techniques assume knowledge of the source model and its training procedures. Such assumptions may not always hold in practice due to data-privacy constraints [5] and the fact that many large-scale pre-trained models are only available through APIs [6], i.e., their pre-trained weights are not shared. Therefore, model-agnostic solutions, which enable computationally efficient adaptation of black-box pre-trained models, would be of significant interest in practice.

In this work, we propose to cluster the softmax predictions of deep neural networks. Solely based on the model’s probabilistic outputs, this strategy yields computationally efficient adaptation while preserving the privacy of both the model and data. Fig. 1 depicts an application example for road segmentation under domain shifts.

Clustering the softmax output predictions of deep networks can be formulated as a clustering problem on the probability simplex domain, i.e., clustering discrete distributions. Probability simplex clustering is not a new problem and has been the subject of several early studies in other application domains. However, in a breadth of practical applications, such as large-scale pre-trained models, would be of significant interest in practice.

With large (and continuously growing) pre-trained networks, such as the recently emerging foundational vision-language models [4]. In such scenarios, even fine-tuning only parts of the pre-trained model, such as the parameters of the normalization layers [1], might be computationally intensive. In addition, the existing adaptation techniques assume knowledge of the source model and its training procedures. Such assumptions may not always hold in practice due to data-privacy constraints [5] and the fact that many large-scale pre-trained models are only available through APIs [6], i.e., their pre-trained weights are not shared. Therefore, model-agnostic solutions, which enable computationally efficient adaptation of black-box pre-trained models, would be of significant interest in practice.

In this work, we propose to cluster the softmax predictions of deep neural networks. Solely based on the model’s probabilistic outputs, this strategy yields computationally efficient adaptation while preserving the privacy of both the model and data. Fig. 1 depicts an application example for road segmentation under domain shifts.

Clustering the softmax output predictions of deep networks can be formulated as a clustering problem on the probability simplex domain, i.e., clustering discrete distributions. Probability simplex clustering is not a new problem and has been the subject of several early studies in other application domains.

1This is the case, for instance, of large-scale foundation models in NLP such as the GPT family, Anthropic’s Claude or Google’s PaLM.

Manuscript received 12 March 2024; accepted 7 June 2024. Date of publication 25 June 2024; date of current version 5 November 2024. This work was supported by computational resources provided by Compute Canada. Recommended for acceptance by G. Hua. (Corresponding author: Florent Chiaroni.) Florent Chiaroni was with ÉTS Montreal and INRS, Montreal, QC H3C 1K3, Canada. He is now with Thales Research and Technology, Thales Digital Solutions (TDS), CortAIx, Montreal, QC H3C 1K3, Canada (e-mail: florent.chiaroni.ai@gmail.com).

Malik Boudiaf and Ismail Ben Ayed are with the ÉTS Montreal, Montreal, QC H3C 1K3, Canada (e-mail: malik.boudiaf.1@etsmtl.net; Ismail.BenAydı@etsmtl.ca).

Amar Mitiche is with the INRS, Montreal, QC H2X 1E3, Canada (e-mail: Amar.Mitiche@inrs.ca).

Our code and comparisons with the existing simplex-clustering approaches and our introduced softmax-prediction benchmarks are publicly available: https://github.com/fchiaroni/Clustering_Softmax_Predictions. This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3418776, provided by the authors. Digital Object Identifier 10.1109/TPAMI.2024.3418776.
such as text analysis [7], [8], [9], [10]. However, in the context of deep learning, clustering methods have been typically applied on deep feature maps, as in self-training frameworks [11]. This requires access to the internal feature maps of the source models, thereby violating the black-box assumption. In contrast, simplex-clustering-based adjustment of the probabilistic outputs of a deep pre-trained model complies with the black-box assumption but has remained largely under-explored, to the best of our knowledge.

The simplex clustering literature is often based on optimizing distortion-based objectives. The goal is to minimize, within each cluster, some distortion measure evaluating the discrepancy between a cluster representative and each sample within the cluster. Besides standard objectives like K-MEANS, which correspond to an $\| \cdot \|_2$ distortion, several simplex-clustering methods motivated and used distortion measures that are specific to simplex data. This includes, for instance, the Kullback-Leibler (KL) divergence [9], [10] and the Hilbert geometry distance [12].

In this work, we explore a general maximum a posteriori (MAP) perspective of clustering discrete distributions. Using the Gibbs model, we emphasize that the density functions underlying current distortion-based objectives (Table I) may not properly approximate the empirical marginal distributions of real-world softmax predictions.

Fig. 2 illustrates our empirical observations through histograms representing the marginal (per-coordinate) distributions of real softmax predictions. These predictions were obtained using a black-box source model applied to data from an unseen target domain. We juxtapose these empirical histograms to parametric density functions estimated on the same predictions via various clustering methods. For instance, curve $p(Eucl)$ represents the parametric Gibbs model corresponding to the euclidean distance used in K-MEANS. A close matching between the shape of the parametric density function and the one of the empirical histogram indicates a good fit of the simplex predictions. However, a significant disparity implies that the assumed parametric density function may not effectively model these predictions. For example, in Fig. 2(d), the modes of parametric density functions in order to preserve the assumption that the density functions in (a), (b), and (d) were extracted from the SVHN $\rightarrow$ MNIST benchmark and the one in (c) from the VISTDA-C one (details in Section V-A.2). Figs. 8 and 9 in the Appendix, available online, depict the whole set of histograms per class. Normalized probability density functions $N$, $p(Eucl)$, $p(KL)$, $p(HG)$, Beta and aBeta are obtained, respectively, from the normal density, euclidean distance, Kullback–Leibler divergence, Hilbert distance, Beta, and aBeta, all listed in Table I.

### TABLE I

| COMPARISON OF METRICS AND DENSITY FUNCTIONS, DEPENDING ON THE PROBABILITY SIMPLEX VECTOR $x_i = (x_{i,n})_{1 \leq n \leq D} \in \Delta^{D-1}$ AND THE PARAMETERS $\theta_k = (\theta_{k,n})_{1 \leq n \leq D}$ FOR EUCLIDEAN, KL, AND $HG_{D-1}$ DISTORTIONS |
| --- |
| **METRIC-BASED** |
| $\| x_i - \theta_k \|_2$ | $\exp(-\text{METRIC}(x_i; \theta_k))$ |
| $KL(x_i|\theta_k)$ | $\sum_{n=1}^{D} \exp(-\frac{1}{\lambda} x_{i,n} \log \frac{x_{i,n}}{\theta_{k,n}})$ |
| $HG_{D-1}(x_i; \theta_k)$ (EQ. (3)) | $\left(1 - \frac{1}{\lambda_1}\right)/(1 - \frac{1}{\lambda_0})$ |
| **PROBABILISTIC** |
| $N(x_i; \theta_k)$ | $\exp(-\frac{1}{2} (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k))$ |
| $f_{\theta_k}(x_i; \theta_k)$ | $\frac{\prod_{n=1}^{D} x_{i,n}^{\omega_{k,n} - 1}}{B(\alpha_{k,n}, \beta_{k,n})}$ |
| $g_{aBeta}(x_i; \theta_k)$ | $\frac{\prod_{n=1}^{D} (x_{i,n} + \alpha_{k,n} - 1) x_{i,n}^{\alpha_{k,n} - 1}}{B(\alpha_{k,n}, \beta_{k,n})}$ |
| $g_{aBeta}(x_i; \theta_k)$ | $\frac{\prod_{n=1}^{D} (x_{i,n} + \beta_{k,n} - 1) x_{i,n}^{\beta_{k,n} - 1}}{B(\alpha_{k,n}, \beta_{k,n})}$ |

$\theta_k$ respectively represents the density parameters $\theta_k = (\mu_k, \Sigma_k)$ with $\mu_k = (\mu_{k,n})_{1 \leq n \leq D}$ for $N$, $\theta_k = (\alpha_{k,n}, \beta_{k,n})_{1 \leq n \leq D}$ for $f_{\theta_k}$, and $\theta_k = (\omega_{k,n}, \alpha_{k,n})_{1 \leq n \leq D}$ for $g_{aBeta}$ and $g_{Beta}$. 

Contributions: Driven by the above observations, we introduce a novel probabilistic clustering objective integrating a generalization of the Beta density, which we coin aBeta. We derive several properties of aBeta, which enable us to...
impose different constraints on our clustering model, referred to as $K$-SBETAS, enforcing uni-modality of the densities within each cluster while avoiding degenerate solutions. We proceed with a block-coordinate descent approach, alternating optimization w.r.t assignment variables, and inner Newton iterations for solving the non-linear optimality conditions w.r.t the SBeta parameters. Furthermore, using the density moments, we derive a closed-form alternative to Newton iterations for parameter estimation. Our versatile formulation approximates various parametric densities for modeling simplex data, including highly peaked distributions at the vertices of the simplex, as observed empirically in the case of deep-network predictions. It also enables the control of the cluster balance. We report comprehensive experiments, comparisons, and ablation studies, which show highly competitive performances of our simplex clustering for unsupervised adjustment of black-box predictions in a variety of scenarios. Fig. 2 illustrates the ability of our method to consistently fit the empirical marginal distributions of real-world softmax predictions. For instance, Fig. 2(c) shows that the density peak of sBeta (orange curve) matches the highest point of the histogram, enabling a more accurate fitting than the other methods. In addition, unlike the standard Beta density, our approach deliberately avoids fitting bimodal distributions (Fig. 2(d)), thereby preventing degenerate solutions. This refinement not only ensures a more accurate representation of real data, but also maintains the assumption that the distribution of each class is unimodal.

II. PROBLEM FORMULATION

We start by introducing the basic notations, which will be used throughout the article:

- $\Delta^{D-1} = \{ x \in [0, 1]^D \mid x^T 1 = 1 \}$ stands for the $(D - 1)$-probability simplex.
- $\mathcal{X} = \{ x_i \}_{i=1}^N \overset{iid}{\sim} X$, with $X$ a given data set and $X$ denoting the random simplex vector in $\Delta^{D-1}$.
- $\mathcal{X}_k = \{ x_i \in \mathcal{X} \mid u_{i,k} = 1 \}$ denotes cluster $k$, where $u_i = (u_{i,k})_{1 \leq k \leq K} \in \Delta^{K-1} \cup \{ 0, 1 \}$ is a latent binary vector assigning point $x_i$ to cluster $k$: $u_{i,k} = 1$ if $x_i$ belongs to cluster $k$ and $u_{i,k} = 0$ otherwise. Let $U \in \{ 0, 1 \}^{NK}$ denote the latent assignment matrix whose column vectors are given by $u_i$. Note that $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_N$.

Clustering consists of partitioning a given set $\mathcal{X}$ into $K$ different subsets $\mathcal{X}_k$ referred to as clusters. In general, this is done by optimizing some objective functions with respect to the latent assignment variables $U$. Often, the basic assumption underlying objective functions for clustering is that data points belonging to the same cluster $k$ should be relatively close to each other, according to some pre-defined notion of closeness (e.g., via some distance or distortion measures).

In this study, we tackle the clustering of probability simplex data points (or distributions), with a particular focus on the output predictions from deep-learning models. Softmax$^3$ is commonly used as the last activation function of neural-network classifiers to produce such output probability distributions. Each resulting softmax data point $x$ is a $D$-dimensional probability vector of $D$ continuous random variables, which are in $[0, 1]$ and sum to one. Thus, the target clustering challenge is defined on probability simplex domain $\Delta^{D-1}$. Fig. 3 depicts mixtures of three class distributions on $\Delta^2$, simulating the impact of a domain shift on the softmax predictions, as well as the benefits of simplex clustering over the standard argmax prediction in such a scenario. Specifically, Fig. 3(a) shows the softmax predictions from a model trained on and applied to the same source domain. In this case, the standard argmax function, commonly used in deep networks, effectively separates the three classes. In contrast, Fig. 3(b) reveals that, in the case of a domain shift, the argmax function may lead to erroneous predictions. Fig. 3(c) illustrates how a simplex clustering successfully maintains class cohesion in this case. This synthetic example points to the potential of simplex clustering in dealing with domain-shift challenges while operating solely on the outputs (i.e., in a black-box setting).

III. RELATED WORK

This section presents related work in the general context of clustering, focusing on the probability simplex domain.

---

3Also referred to as the normalized exponential function, the softmax function is defined as $\sigma(z)_j = \frac{\exp(z_j)}{\sum_{i=1}^{D} \exp(z_i)}$, with $z = \{ z_1, \ldots, z_D \}$ denoting a vector of logits.

---

Fig. 3. Partitioning points within probability simplex $\Delta^2$. Specifically, these points are projected into a 2D equilateral triangle for a clear visualization, where each vertex represents one dimension of the probability space. The figures depict the probabilistic predictions from a hypothetical source model, with and without distribution shift. Fig. (b) depicts the standard argmax assignment, whereas Fig. (c) represents a Probability Simplex Clustering (PSC) using a cluster-to-class alignment. Symbols + (green), x (orange) and • (blue) indicate separate class assignments.
distances between two distributions $\Delta$ with respect to assignment variables and cluster representatives $\Theta = (\theta_k)_{1 \leq k \leq K}$ a mixed-variable function of the following general form:

$$L_{dist}(U; \Theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} u_{i,k} \|x_i - \theta_k\|_d = \sum_{k=1}^{K} \sum_{x_i \in X_k} \|x_i - \theta_k\|_d.$$  
(1)

The goal is to minimize, within each cluster $X_k$, some distortion measure $\|\cdot\|_d$ evaluating the discrepancy between cluster representative $\theta_k$ and each point belonging to the cluster. When $\|\cdot\|_d$ is the euclidean distance, general form (1) becomes the standard and widely used K-Means objective [13]. In general, optimizing distortion-based objective (1) is NP-hard.\(^4\) One standard iterative solution to tackle the problem in (1) is to follow a block-coordinate descent approach, which alternates two steps, one optimizes the objective w.r.t to the cluster prototypes and the other w.r.t. to assignment variables:

- U-update: $u_{i,k} = \begin{cases} 1 & \text{if } \arg \min_k \|x_i - \theta_k\|_d = k \\ 0 & \text{otherwise.} \end{cases}$
- $\Theta$-update: Find $\arg \min_{\Theta} L_{dist}(U; \Theta)$.

In the specific case of K-MEANS (i.e., using the $\|\cdot\|_2$ distance as distortion measure), optimization w.r.t. the parameters in the $\Theta$-update step yields closed-form solutions, which correspond to the means of features within the clusters:

$$\theta_k = \frac{\sum_{i=1}^{N} u_{i,k} x_i}{\sum_{i=1}^{N} u_{i,k}}, \forall k.$$  

One way to generalize K-MEANS is to replace the euclidean distance with other distortion measures. In this case, the optimal value of the parameters $\theta_k$ may no longer correspond to cluster means. For instance, when using the $\|\cdot\|_1$ distance as distortion measure in (1), the optimal $\theta_k$ corresponds to the cluster median, and the ensuing algorithm is the well-known K-MEDIAN clustering [15]. For exponential distortions, the optimal parameters may correspond to cluster modes [16], [17]. Such exponential distortions enable the model to be more robust to outliers, but this comes at the price of additional computations (inner iterations), as cluster modes could not be obtained in closed-form.

### B. Distortion Measures for Simplex Data

In our case, data points are probability vectors within simplex domain $\Delta^{D-1}$, e.g., the softmax predictions of deep networks. These points are $D$-dimensional vectors of $D$ continuous random variables bounded between 0 and 1, and summing to one. To our knowledge, the simplex clustering literature is often based on distortion objectives of the general form in (1). Besides standard objectives like K-MEANS, several simplex-clustering works motivated and used distortion measures that are specific to simplex data. This includes the Kullback-Leibler (KL) divergence $[9]$, $[10]$ and Hilbert geometry distance [12].

**Information-Theoretic k-Means:** The works in $[9]$, $[10]$ discussed KL K-MEANS, a distortion-based clustering tailored to simplex data and whose objective fits the general form in (1). KL K-MEANS uses the Kullback-Leibler (KL) divergence as a distortion measure instead of the euclidean distance in the standard K-MEANS:

$$\|x_i - \theta_k\|_d = KL(x_i | \theta_k) = \sum_{n=1}^{D} x_{i,n} \log \left( \frac{x_{i,n}}{\theta_{k,n}} \right),$$  
(2)

where $x_{i,n}$ and $\theta_{k,n}$ stand, respectively, for the $n$-th component of simplex vectors $x_i$ and $\theta_k$. KL$(x_i | \theta_k)$ is a Bregman divergence $[8]$ measuring the dissimilarity between two distributions $x_i$ and $\theta_k$. Despite being asymmetric [10], the machine learning community widely uses the KL divergence in a breadth of problems where one has to deal with probability simplex vectors $[18]$, $[19]$, $[20]$, $[21]$.

**Hilbert Simplex Clustering (HSC):** More recently, the study in [12] investigated the Hilbert Geometry (HG) distortion with minimum enclosing ball (MEB) centroids $[22]$, $[23]$ for clustering probability simplex vectors. For a given cluster set $k$, the MEB center represents the midpoint of the two farthest points within the set. Given two points $x_i$ and $\theta_k$ within simplex domain $\Delta^{D-1}$, let $x_i + t(\theta_k - x_i)$, $t \in \mathbb{R}$, denote the line passing through points $x_i$ and $\theta_k$, with $t_0 \leq 0$ and $t_1 \geq 1$ the two intersection points of this line with the simplex domain boundary $\partial \Delta^{D-1}$. Then, HG simplex distortion is given by:

$$HG_{\Delta^{D-1}}(x_i; \theta_k) = \left| \log \left( \frac{(1 - t_0) t_1}{(-t_0)(t_1 - 1)} \right) \right| = \log(1 - \frac{1}{t_0}) - \log(1 - \frac{1}{t_1}).$$  
(3)

Note that if one deals with centroids close or equal to the vertices of the simplex, then HG distortion would inconsistently output large or infinite distance values near such centroids.

### C. From Distortions to Probabilistic Clustering

Beyond distortions, a more versatile approach is to minimize the following well-known maximum a posteriori (MAP) generalization of K-MEANS $[24]$, $[25]$, $[26]$, coined by [26] as probabilistic K-MEANS:

$$L_{prob}(U; \pi; \Theta) = -\sum_{k=1}^{K} \sum_{i=1}^{N} u_{i,k} \log(\pi_k f(x_i | \theta_k)),$$  
(4)

where $f(\cdot | \cdot; \theta_k)$ is a parametric density function modeling the likelihood probabilities of the samples within cluster $k$, and $\pi_k$ the prior probability of cluster $k$, with $\pi = (\pi_k)_{1 \leq k \leq K} \in \Delta^{K-1}$. Hence, instead of minimizing a distortion $\|x_i - \theta_k\|_d$ in (1), we maximize a more general quantity measuring the posterior probability of cluster $k$ given sample $x_i$ and parameters $\theta_k$. It is easy to see that, for any choice of distortion measure, the objective in (1) corresponds, up to a constant, to a particular case of probabilistic K-MEANS in (4), with $\pi_k = 1/K$, $\forall k$, and
parametric density \( f \) chosen to be the \textit{Gibbs} distribution:

\[
f(x_i; \theta_k) \propto \exp(-\|x_i - \theta_k\|_2).
\]

For instance, the K-MEANS objective corresponds, up to additive and multiplicative constants, to choosing likelihood probability \( f(x_i; \theta_k) \) to be a Gaussian density with mean equal to \( \theta_k \), identity covariance matrix and prior probabilities verifying \( \pi_k = 1/K, \forall k \). This uncovers hidden assumptions in K-MEANS: The samples are assumed to follow Gaussian distribution within each cluster, and the clusters are assumed to be balanced.\(^5\) Therefore, the general probabilistic K-MEANS formulation in (4) is more versatile as it enables to make more appropriate assumptions about the parametric statistical models of the samples within the clusters (e.g., via prior knowledge or validation data), and to manage the cluster-balance bias [24].

\[\text{D. Do Current Simplex Clustering Methods Model Softmax Predictions Properly?}\]

Equation (5) emphasized that there are implicit density functions underlying distortion-based clustering objectives via Gibbs models; see Table I. Fig. 2 illustrates how the Gibbs models corresponding to the existing simplex clustering algorithms may not properly approximate the empirical marginal density functions of real-world softmax predictions:

\textit{Standard K-MEANS: Fast, But not Descriptive Enough and Biased.} The well-known K-MEANS clustering falls under the probabilistic clustering framework, as defined in Section III-C. In particular, it instantiates the data likelihoods as Gaussian densities with unit covariance matrices. An important advantage of K-MEANS is that the associated \( \Theta \)-update step can be analytically solved by simply updating each \( \theta_k \) as the mean of samples within cluster \( k \), according to the current latent assignments. Such closed-form solutions enable K-MEANS to be among the most efficient clustering algorithms. However, the Gaussian assumption yielding such closed-form solutions might be limiting when dealing with asymmetric data densities, such as exponential densities, with the particularity that the mode and the mean are distinct. As depicted by Fig. 2(a) and (b), the softmax outputs from actual deep-learning models commonly follow such asymmetric distributions. Note that K-MEANS and other distortion-based objectives correspond to setting \( \pi_k = 1/K, \forall k \) in the general probabilistic formulation in (4). Therefore, they encode the implicit strong bias towards balanced partitions [24], and might be sub-optimal when dealing with imbalanced clusters, as is often the case of realistic deep-network predictions.

\textit{Elliptic K-MEANS.} Elliptic K-MEANS is a generalization of the standard K-means: It uses the Mahalanobis distance instead of the euclidean distance, which assumes the data within each cluster follows a Gaussian distribution (i.e., Gibbs model for the Mahalanobis distance) whose covariance matrix is also a variable of the model (as opposed to unit covariance in K-MEANS); see Table I. Fig. 2(b) and (c) confirm that Gaussian density function \( \mathcal{N} \) is not appropriate for skewed distributions, as is the case of network predictions.

\[\text{HSC yields poor approximations of highly peaked distributions located at the vertices of the simplex: Fig. 2 shows that HSC, which combines MEB centroids and HG metric, is not relevant to asymmetric distributions whose modes are close to the vertices of the simplex. Specifically, in Fig. 2(a) and (b), the corresponding Gibbs probability is small near the vertices of the simplex and high in the middle of the simplex. This yields a poor approximation of the target empirical histograms (depicted in orange). In addition, the overall HSC algorithm is computationally demanding (see Table III).}\]

\textit{KL K-MEANS.} While KL could yield asymmetric density, it may poorly model highly peaked distributions at the vertices of the simplex (e.g., Fig. 2(a)), as is the case of softmax predictions.

\[\text{IV. Proposed Approach}\]

\[\text{A. Background}\]

\textit{Multivariate Dirichlet Density Function:} Commonly used for modeling categorical events, the Dirichlet density operates on \( D \)-dimensional discrete distributions within the simplex, \( \Delta^D \). Its expression, parameterized by vector \( \alpha = (\alpha_i)_{1 \leq i \leq D} \in \mathbb{R}^D \),

\[
\alpha \rightarrow \frac{\Gamma(\sum_{i=1}^{D} \alpha_i)}{\prod_{i=1}^{D} \Gamma(\alpha_i)} \prod_{i=1}^{D} x_i^{\alpha_i - 1},
\]

\[\text{TABLE II}\]

**COMPARATIVE PROBABILITY SIMPLEX CLUSTERING ON SIMU, SVHN \rightarrow MNIST AND VISDA-C DATASETS**

| METHOD         | LAYER USED | SIMU (NMI) | SVHN→MNIST (NMI) | VISDA-C (NMI) |
|----------------|------------|------------|------------------|---------------|
| K-MEANS FEATURE MAP | -         | 67.3       | 74.3             | 36.0          | 22.7          |
| K-MEANS LOGIT     | -         | 67.1       | 74.5             | 32.7          | 33.1          |
| ARGMAX            | -         | 60.1       | 68.9             | 36.5          | 33.1          |
| K-MEANS           | -         | 76.6       | 68.9             | 37.6          | 47.9          |
| GMM               |            | 75.8       | 61.9             | 36.3          | 49.4          |
| K-MEDIANS         |            | 76.8       | 58.6             | 35.9          | 40.0          |
| K-MUODIDS SIMPLEX |            | 60.8       | 58.9             | 36.5          | 46.8          |
| K-MODES (BLACK-BOX) |            | 76.2       | 59.5             | 71.3          | 34.2          | 31.8          |
| KL K-MEANS        | -         | 76.2       | 63.3             | 39.8          | 51.2          |
| HSC               |            | 9.3        | 59.2             | 28.7          | 16.1          |
| K-DIRS            |            | 81.3       | 57.7             | 68.8          | FAILS         |
| K-BETAS           |            | 81.1       | 53.3             | 51.5          | 36.7          | 19.8          |
| K-SBETAS          |            | 79.2       | 65.0             | 76.6          | 40.3          | 56.0          |

\[\text{The Feature Map layer is the Bottleneck layer on top of the convolutional layers. It outputs a feature vector of size 256 for each input image. The logit points are obtained with the fully connected classifier on top of the Bottleneck layer so that the logit point predicted for each input example is a vector of size } K. \text{ The probability Simplex points, which represent the predictions of a black-box model, are obtained with the standard softmax function on top of the logit layer.}\]

\[\text{TABLE III}\]

**COMPUTATIONAL TIME COMPARISON IN SECONDS**

| METHOD         | SVHN→MNIST \( (N = 10000, K = 10) \) | VISDA-C \( (N = 55388, K = 12) \) |
|----------------|-----------------------------|-----------------------------|
| K-MEANS - FEATURE MAP | 2.36                        | 18.01                       |
| K-MEANS - LOGIT/SIMPLEX | 0.06                        | 1.27                        |
| GMM            | 0.43                        | 10.39                       |
| K-MEDIANS      | 5.94                        | 48.39                       |
| K-MUODIDS      | 8.83                        | 195.00                      |
| K-MODES        | 0.08                        | 4.71                        |
| KL K-MEANS     | 0.27                        | 2.79                        |
| HSC            | 6494.54                     | >ONE DAY                    |
| K-SBETAS (MLE-1000) | 64.32                      | 107.43                      |
| K-SBETAS (MLE-500) | 34.22                      | 58.48                       |
| K-SBETAS (MOm) | 0.48                        | 3.61                        |
| K-SBETAS (MOm, GPU-BASED) | 0.13                      | 0.49                        |

\[\text{All methods are executed on the same hardware: CPU 11th Gen Intel(R) Core(TM) i7-11700K k 4.90GHz, and GPU NVIDIA GeForce RTX 2070 SUPER. MEB-1000 and MLE-500 refer to the MLE parameter estimation with a maximum of 1000 and 500 MLE iterations, respectively, while MoM refers to the method of moments.}\]

\[\text{IV. Proposed Approach}\]

\[\text{A. Background}\]

\[\text{Multivariate Dirichlet Density Function:} \text{ Commonly used for modeling categorical events, the Dirichlet density operates on } D \text{-dimensional discrete distributions within the simplex, } \Delta^D. \text{ Its expression, parameterized by vector } \alpha = (\alpha_i)_{1 \leq i \leq D} \in \mathbb{R}^D,\]

\[
\alpha \rightarrow \frac{\Gamma(\sum_{i=1}^{D} \alpha_i)}{\prod_{i=1}^{D} \Gamma(\alpha_i)} \prod_{i=1}^{D} x_i^{\alpha_i - 1},
\]

\[\text{Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.}\]
is given by:

\[ f_{\text{Dir}}(x; \alpha) = \frac{1}{B(\alpha)} \prod_{n=1}^{D} x_n^{\alpha_n - 1}, \quad (6) \]

with \( x \in \Delta^{D-1} \), and \( B(\alpha) \) the multivariate Beta function expressed using the Gamma function.\(^6\) While the Dirichlet density is designed for simplex vectors, its complex nature leads to computationally intensive optimization, often involving iterative and potentially sub-optimal parameter estimation [27]. To address this computational challenge, one could adopt the mean-field approximation principle [28]. This uses a relaxed product density, which is based on Dirichlet’s marginal density, i.e., \( \text{Beta} \) with parameters \( \alpha \) and \( \beta \):

\[ g_{\text{Beta}}(x; \alpha, \beta) = \prod_{n=1}^{D} f_{\text{Beta}}(x_n; \alpha_n, \beta_n) 
= \prod_{n=1}^{D} \frac{(x_n)^{\alpha_n - 1}(1-x_n)^{\beta_n - 1}}{B(\alpha_n, \beta_n)} 
\approx f_{\text{Dir}}(x; \alpha). \quad (7) \]

This approximation simplifies parameter estimation by treating each simplex coordinate independently. However, apart from the computational load, \( \text{Dir} \) and its marginal \( \text{Beta} \) may yield poor approximations of the empirical marginal distributions of real-world softmax predictions. For example, Fig. 2(a), (b), and (c) show how \( \text{Beta} \) has difficulty capturing empirical marginal distributions. In addition, \( \text{Dir} \) allows multimodal distributions, as shown with its \( \text{Beta} \) marginal in Fig. 2(d), opposing the common assumption that each semantic class distribution is unimodal. Indeed, discriminative deep learning models are optimized to output probability simplex vertices (i.e., one-hot vectors). For each class, the model is optimized to output softmax predictions following an unimodal distribution. In this context, \( \text{Dir} \) may erroneously represent two unimodal distributions as a single bimodal distribution.

In the next section, we propose \( \text{sBeta} \), an alternative capable of effectively addressing these limitations.

B. Proposed Density Function: A Generalization of Beta
Constrained to be Unimodal

Unlike \( \text{Beta} \), we seek a density function that can approximate highly peaked densities near the vertices of the simplex while satisfying a uni-modality constraint, thereby avoiding degenerate solutions and accounting for the statistical properties of real-world softmax predictions.

1) Generalization of Beta: We propose the following generalization of \( \text{Beta} \), which we refer to as \( \text{sBeta} \) (scaled \( \text{Beta} \)) in the sequel:

\[ f_{\text{sBeta}}(x; \alpha, \beta) = \frac{(x+\delta)^{\alpha-1}(1+\delta-x)^{\beta-1}}{B(\alpha, \beta)(1+2\delta)^{\alpha+\beta-2}}, \quad (8) \]

with \( \delta \in \mathbb{R}^+ \). Clearly, when parameter \( \delta \) is set equal to 0, the generalization in (8) reduces to \( \text{Beta} \) density \( f_{\text{Beta}} \) in (7).

Fig. 4(a) and (b) respectively illustrate \( \text{Beta} \) and \( \text{sBeta} \) density functions for different values of parameters \( \alpha \) and \( \beta \), showing the difference between the two densities. For instance, for \( \alpha = 3 \) and \( \beta = 9 \), one may observe that \( \text{sBeta} \) is characterized by a higher density than \( \text{Beta} \) in the proximity of the simplex vertex in this particular example. As illustrated in Fig. 2(a), (b) and (c), \( \text{sBeta} \) approximations of the empirical distributions (orange histograms) of the softmax predictions are better than those obtained with the \( \text{Beta} \) density. Overall, \( \text{sBeta} \) can be viewed as a scaled variant of \( \text{Beta} \). It is relatively more permissive, enabling it to fit a wider range of unimodal simplex distributions.

In the following, we provide expressions of the moments (first and second central) and mode of \( \text{sBeta} \) as functions of density parameters \( \alpha \) and \( \beta \). As we will see shortly, these properties will enable us to integrate different constraints with our \( \text{sBeta} \)-based probabilistic clustering objective, enforcing uni-modality of the distribution within each cluster while avoiding degenerate solutions. Furthermore, they will enable us to derive a computationally efficient, moment-based estimation of the density parameters.

2) Mean and Variance: Property 1. The mean \( E_{\text{sBeta}}[X] \) of \( \text{sBeta} \) could be expressed as a function of the density parameters as follows:

\[ E_{\text{sBeta}}[X] = \frac{\alpha}{\alpha + \beta} (1 + 2\delta) - \delta. \quad (9) \]

Proof: The mean \( E_{\text{sBeta}}[X] \) of \( \text{sBeta} \) could be found by integrating \( x f_{\text{sBeta}}(x) \). A detailed derivation is provided in Appendix A, available online.

\[ \Box \]
Property 2. The variance \( V_{SB}\{X\} \) of \( sBeta \) could be expressed as a function of the density parameters as follows:

\[
V_{SB}\{X\} = \frac{\alpha \beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}(1 + 2\delta)^2.
\]  
(10)

Proof: We use \( V_{SB}\{X\} = E_{SB}\{X^2\} - E_{SB}\{X\}^2 \), with \( E_{SB}\{X^2\} \) denoting the second moment of \( sBeta \), which could be found by integrating \( x^2 f_{SB}(x) \). The details of the computations of \( E_{SB}\{X^2\} \) and \( V_{SB}\{X\} \) are provided in the Appendix A, available online.

3) Mode and Concentration: The mode of a density function \( f(x) \) corresponds to the value of \( x \) at which \( f(x) \) achieves its maximum.

Property 3. The mode \( m_{SB} \) of \( sBeta \) could be expressed as a function of the density parameters as follows:

\[
m_{SB} = \frac{\alpha - 1 + \delta(\alpha - \beta)}{\alpha + \beta - 2}.
\]  
(11)

Proof: The mode \( m_{SB} \) of \( sBeta \) can be found by estimating at which value of \( x \) the derivative of \( f_{SB} \) is equal to 0. A detailed derivation can be found in Appendix A, available online.

Note that, when \( \alpha = \beta \), we have \( m_{SB} = E_{SB}\{X\} = \frac{1}{2} \). Otherwise, \( m_{SB} \neq E_{SB}\{X\} \). Thus, \( sBeta \) is asymmetric when \( \alpha \neq \beta \), and the variance is consequently inadequate to measure the dispersion around \( sBeta \) mode.

Concentration Parameter: Motivated by the previous observation, we present the concentration parameter \( \lambda \) to measure how much a sample set is condensed around the mode value. The concept of concentration parameter has been previously discussed in [29]. With respect to the mode (11), we express \( \lambda = \alpha + \beta - 2 \) such that we have

\[
m_{SB} = \frac{\alpha - 1 + \delta(\alpha - \beta)}{\lambda}.
\]  
(12)

Using (12), we can express \( \alpha \) and \( \beta \) as functions of the mode and concentration parameter:

\[
\begin{cases}
\alpha = 1 + \frac{m_{SB} + \delta}{1 + 2\delta} \\
\beta = 1 + \frac{1 + \delta - m_{SB}}{1 + 2\delta}
\end{cases}
\]  
(13)

Fig. 4(c) shows different concentrations around a given mode when changing \( \lambda \).

C. Proposed Clustering Model: K-SBETAS

Probabilistic Clustering: We cast clustering distributions as minimizing the following probabilistic objective (K-SBETAS), which measures the conformity of data within each cluster to a multi-variate \( sBeta \) density function, subject to constraints that enforce uni-modality and discourage degenerate solutions: where \( g_{SB} \) is a multivariate extension of \( sBeta \):

\[
g_{SB}(x_i; \alpha_k, \beta_k) = \prod_{n=1}^{D} f_{SB}(x_{i,n}; \alpha_k, \beta_k, \pi_k),
\]

and \( \lambda_{k,n} \) denotes the concentration parameter of univariate \( f_{SB}(x_{i,n}; \alpha_k, \beta_k, \pi_k) \). In our model in (14), \( \Theta = (\alpha_k, \beta_k)_{1 \leq k \leq K} \), with \( \alpha_k = (\alpha_{k,n})_{1 \leq n \leq D} \) and \( \beta_k = (\beta_{k,n})_{1 \leq n \leq D} \), \( \pi_k = (\pi_k)_{1 \leq k \leq K} \in \Delta^K \) and, following the general notation we introduced in Section II, \( U \) denotes binary point-to-cluster assignments.

1) Block-Coordinate Descent Optimization: Our objective in (14) depends on three different sets of variables: \( U; \pi; \) and \( sBeta \) parameters \( \Theta \). Therefore, we proceed with a block-coordinate descent approach, alternating three steps. Each step optimizes the objective w.r.t. a set of variables while keeping the rest of the variables fixed.

- \( \Theta \) updates, with \( U \) and \( \pi \) fixed: This section presents two different strategies for updating the \( sBeta \) parameters:
  (a) Solving the necessary conditions for the minimum of \( L_{SB} \) w.r.t. \( \alpha_{k,n} \) and \( \beta_{k,n} \). Our objective in (14) is convex in each \( \alpha_{k,n} \) and \( \beta_{k,n} \). The global optima could be obtained by setting the gradient to zero. This could be viewed as a maximum likelihood estimation approach (MLE). Unfortunately, this yields a non-linear system of equations that cannot be solved in closed-form. Therefore, we proceed to an inner iteration. In the appendix, available online, we derive the following iterative and alternating updates to solve the non-linear system of equations:

\[
\alpha_k^{(t+1)} = \psi^{-1}\left(\psi\left(\alpha_k^{(t)} + \beta_k^{(t)}\right) + \frac{1}{\sum_{i=1}^{N} u_{i,k}} \sum_{i=1}^{N} u_{i,k} \log \left(\frac{x_{i} + \delta}{1 + 2\delta}\right)\right)
\]  
(16)

\[
\beta_k^{(t+1)} = \psi^{-1}\left(\psi\left(\alpha_k^{(t)} + \beta_k^{(t)}\right) + \frac{1}{\sum_{i=1}^{N} u_{i,k}} \sum_{i=1}^{N} u_{i,k} \log \left(\frac{1 + \delta - x_{i}}{1 + 2\delta}\right)\right)
\]  
(17)

where \( t \) refers to iteration number and \( \psi \) corresponds to the digamma function. Note that neither \( \psi \) nor \( \psi^{-1} \) admit analytic expressions. Instead, we follow [27] to approximate those functions, at the cost of additional Newton iterations. The full derivation of the updates and all required details could be found in the Appendix, available online.

\[
L_{SB}(U; \pi; \Theta) = -\sum_{i=1}^{N} \sum_{k=1}^{K} u_{i,k} \log \left(\pi_k g_{SB}(x_i; \alpha_k, \beta_k)\right)
\]  
(14)

s.t. \[
\begin{cases}
\lambda_{k,n} \geq \tau^{-}\forall k, n : \text{Unimodal constraint, Section (IV-C-2)} \\
\lambda_{k,n} \leq \tau^{+}\forall k, n : \text{Avoids Dirac solutions, Section (IV-C-2)}
\end{cases}
\]  
(15)
(b) Method of moments (MoM):  
As a computationally efficient alternative to solving a non-linear system within each outer iteration, we introduce an approximate estimation of sBeta parameters $\alpha_k$ and $\beta_k$, which we denote $\alpha_k$ and $\beta_k$, as the solutions to the first and second central moment equations, following Property 1 and Property 2:

$$\begin{align*}
\mu_k &= \frac{\alpha_k}{\alpha_k + \beta_k} (1 + 2\delta) - \delta \\
u_k &= \frac{\beta_k}{\alpha_k + \beta_k} (1 + 2\delta)^2 \tag{18}
\end{align*}$$

where $\mu_k$ and $\nu_k$ denote, respectively, the empirical means and variances of cluster $k$:

$$\begin{align*}
\mu_k &= \frac{\sum_{i=1}^{N} u_{i,k} x_i}{\sum_{i=1}^{N} u_{i,k}} \quad \text{and} \quad \nu_k = \frac{\sum_{i=1}^{N} u_{i,k} (x_i - \mu_k)^2}{\sum_{i=1}^{N} u_{i,k}}. \tag{19}
\end{align*}$$

The system of two equations and two unknowns in $(18)$ could be solved efficiently in closed-form, which yields the following estimates of sBeta parameters:

$$\begin{align*}
\alpha_k &= \left( \frac{\mu_k^2}{\nu_k} - 1 \right) \mu_k \\
\beta_k &= \left( \frac{\nu_k}{\mu_k (1 + 2\delta)} - 1 \right) (1 - \mu_k^2) \tag{20}
\end{align*}$$

with $\mu_k^2 = \frac{\mu_k}{1 + 2\delta}$.

- **U updates, with $\Theta$ and $\pi$ fixed:** With variables $\Theta$ and $\pi$ fixed, the global optimum of our objective in $(14)$ with respect to assignment variables $U$, subject to constraints $u_i = (u_{i,k})_{1 \leq k \leq K} \in \Delta^{K-1} \cup \{0, 1\}$, corresponds to the following closed-form solution for each assignment variable $u_{i,k}$:

$$u_{i,k} = \begin{cases} 
1 & \text{if } \arg \max_k \pi_k \cdot g_{sBeta}(x_i; \alpha_k, \beta_k) = k \\
0 & \text{otherwise.}
\end{cases} \tag{21}$$

- **$\pi$ updates, with $U$ and $\Theta$ fixed:** Solving the Karush–Kuhn–Tucker (KKT) conditions for the minimum of $(14)$ with respect to $\pi$, s.t $\pi \in \Delta^{K-1}$, yields the following closed-form solution for each $\pi_k$:

$$\pi_k = \frac{\sum_{i=1}^{N} u_{i,k}}{N}. \tag{22}$$

2) Handling Parameter Constraints: Our model in $(14)$ integrates two constraints on concentration parameters $\lambda_{k,n}$; one discourages multi-modal solutions, and the other avoids degenerate, highly peaked (Dirac) densities.

Notice that the sBeta density is bimodal when $\alpha_k < 1$ and $\beta_k < 1$, which corresponds to $\lambda_{k,n} < 0$ (as $\lambda_{k,n} = \alpha_k n + \beta_k - 2$). Moreover, setting $\alpha_k = \beta_k = 1$, which yields $\lambda_{k,n} = 0$, corresponds to the uniform density. Thus, to constrain the sBeta densities to be strictly unimodal, we must restrict $\lambda_{k,n}$ to positive values. In practice, we constrain $\lambda_{k,n}$ to be greater or equal to a threshold $\tau^\ast > 0$, which we simply set to 1.

A high-density concentration around the sBeta mode corresponds to high values $\lambda_{k,n}$. Thus, to avoid degenerate, highly peaked (Dirac) densities, we constrain $\lambda_{k,n}$ to stay smaller than or equal to a fixed, strictly positive threshold $\tau^\ast$.

Overall, we enforce $\tau^\ast \leq \lambda_{k,n} \leq \tau^\ast$ on sBeta parameters using $(13)$, as detailed in Algorithm 1. This procedure allows for maintaining the same mode. Fig. 5(a) shows sBeta density estimation, following the constraint $\lambda_{k,n} \geq \tau^\ast$, on a bimodal distribution sample. Fig. 5(b) shows sBeta estimation, following the constraint $\lambda_{k,n} \leq \tau^\ast$, on a Dirac distribution sample.

3) Centroid Initialization: Clustering algorithms are known to be sensitive to their parameter initialization. They often converge to a local optimum. To reduce this limitation, the seeding initialization strategy k-means++, proposed in [30] and then thoroughly studied in [31], is broadly used for centroid initialization. However, in the context of the clustering of softmax predictions, we can assume that softmax predictions generated by deep learning models were optimized upstream to be sets of one-hot vectors.7 Thus, we propose initializing the centroids as vertices of the target probability simplex domain. More specifically, we initialize all $\alpha_{k,n}$ and $\beta_{k,n}$ parameters such that they model exponential densities on $[0,1]$ at the start. In other words, each initial mode is set as a vertex among all possible vertices on the probability simplex domain. Beyond improving k-sBetas, this simple initialization strategy unanimously improves the scores of every tested clustering method.8

4) From Cluster-Label to Class-Label Assignment: After clustering, we must align the obtained clusters with the target classes. One can align each cluster centroid with the closest one-hot vector by using the argmax function. However, in this case, a one-hot vector corresponding to a given class could be matched to several cluster centroids, thereby assigning them to

---

A one-hot vector both refers to a semantic class and to a vertex on the probability simplex.

Table XIV in Appendix, available online, empirically supports this statement.
the same class. To prevent this problem on closed-set challenges, i.e., with the prior knowledge $K = D$, we use the optimal transport Hungarian algorithm [32]. Specifically, we compute the euclidean distances between all the centroids and one-hot vectors. Then, we apply the Hungarian algorithm to this matrix of distances to find the lowest-cost way, assigning a separate class to each cluster.

V. Experiments

Throughout this section, we show different applications of the proposed method and compare it with the state-of-the-art clustering methods discussed in Section III. We first validate our implementations on synthetic datasets and on real softmax predictions from unsupervised domain adaptation (UDA) source models (Section V-A). Then, in Section V-B, we demonstrate the usefulness of our simplex-clustering method in the context of transductive zero- and one-shot predictions using contrastive language-image pre-training (CLIP) [4] for the source model. Finally, we provide an application example for a dense prediction task: real-time UDA for road segmentation (Section V-C). In each of these settings, our method is used as a plug-in on top of the output predictions from a black-box deep learning model. Along with the experiments and comparisons that follow, we also provide ablation studies, which highlight the effect of each component of the proposed framework.

Algorithm 2: $k$-sBetas Algorithm.

Input: Dataset of probability simplex points defined as $X = \{ x_i \}_{i=1}^N \in \Delta^{D-1}$, with $N$ the total number of points. Number of clusters $K$. The maximum number $t_{end}$ of iterations.

Output: $U$, the clustering hard label assignments

1: global variables
2: $\delta$, the sBeta hyper-parameter
3: $\tau^-$, the unimodal constraint
4: $\tau^+$, the constraint to avoid Dirac solutions
5: end global variables
6: $U \leftarrow \emptyset_{N,K}$ \hspace{5mm} $\triangleright$ Arbitrary label initialization
7: $\pi \leftarrow \{ \frac{1}{N} \}_{k=1}^K$ \hspace{5mm} $\triangleright$ Initialize uniform class proportions
8: Initialize $\Theta$ using vertex init. \hspace{5mm} $\triangleright$ Section IV-C.3.
9: $t \leftarrow 0$
10: repeat
11: if $t > 0$ then
12: for $k \leftarrow 1 \ldots K$ do
13: Estimate $\alpha_k, \beta_k$ with MoM/MLE \hspace{5mm} $\triangleright$ Section IV-C.1.
14: $\alpha_k, \beta_k \leftarrow \mathrm{constrain}(\alpha_k, \beta_k)$ \hspace{5mm} $\triangleright$ Algorithm 1
15: end for
16: end if
17: for $i \leftarrow 1 \ldots N$ do \hspace{5mm} $\triangleright$ hard label assignment
18: for $k \leftarrow 1 \ldots K$ do
19: if $\arg \max_{i,k} \pi_k : g_{sBeta}(x_i; \alpha_k, \beta_k) = k$ then
20: $u_{i,k} \leftarrow 1$
21: else
22: $u_{i,k} \leftarrow 0$
23: end if
24: end for
25: end for
26: $\pi \leftarrow \sum_{i=1}^N u_i$ \hspace{5mm} $\triangleright$ Estimate class proportions
27: $t \leftarrow t + 1$
28: until convergence or $t = t_{end}$
29: return $U$

Comparisons: We compare our method to several state-of-the-art clustering methods. Specifically, we include KL K-MEANS [9], [10] and HSC [12], which are specifically designed to deal with probability simplex vectors. We additionally compare our method to standard clustering methods: K-MEANS, GMM [33], K-MEDIANS [15], K-MEDOIDS [34] and K-MODES [16]. Both K-MEANS and KL K-MEANS use the mean as the cluster’s parameter vector, with the former being based on the euclidean distortion and the latter on the Kullback-Leibler one. Extensively studied in [15], K-MEDIANS uses the median as a cluster representative and the Manhattan distance. Clustering with K-MEDOIDS uses the euclidean distance while estimating medoid representations of the clusters with the standard PAM algorithm [34]. Finally, K-MODES [16] uses a kernel-induced distortion and computes cluster modes via the Meanshift algorithm [35].

K-DIRS. We also implemented a simplex-clustering strategy, which estimates a multivariate density per cluster using the
Dirichlet density function. We used the iterative parameter estimation proposed in [27]. This algorithm, which we refer to as K-DIRS, represents a baseline for the artificial datasets built from the Dirichlet distributions.

**Proposed k-sBETAS.** Our clustering algorithm uses \( f_{\text{sBetas}} \) as density function and the method of moments (MoM) detailed in Section IV-C-1 for parameter estimation. We empirically consolidate this choice in the ablation study in Section V-D. Note that \( \text{K-BETAS} \) corresponds to a non-scaled variant of k-sBETAS, i.e., when we set \( \delta = 0 \).

**Hyper-Parameters:** Across all the experiments, we set the scaling hyper-parameter \( \delta \) used in k-sBETAS to the same value 0.15. We set \( \tau^- = 1 \) and \( \tau^+ = 165 \) for parameter constraints. The maximum number of clustering iterations is set to 25 for all the clustering algorithms. For each method, we use the parameter initialization detailed in Section IV-C-3. This technique is also applied to the methods based on the feature maps, for consistency and fairness: For each probability simplex vertex (i.e., one-hot vector), we select the feature-map point whose softmax prediction most closely aligns with this one-hot vector. We provide comparisons and justifications for our design choices in the ablation studies in Section V-D.

**Reproducibility:** We used the implementation of the scikit-learn library for GMM,\(^9\) and the implementation provided by the authors for HSC.\(^10\) We have implemented all the other clustering algorithms.\(^11\)

**Evaluation Metrics:** We use the standard Normalized Mutual Information (NMI) to evaluate the clustering task for each dataset. To evaluate the class-prediction scores, we use the classification Accuracy (Acc) for the balanced datasets or the Intersection over Union (IoU) measure for the imbalanced ones. The process of aligning the clusters with the class labels, which is necessary for computing the ACC and mIoU, is detailed in Section IV-C-4.

### A. Unsupervised Domain Adaptation

1) **Synthetic Experiments:** The purpose of the following synthetic experiments is to benchmark the different clustering methods on a simple, artificially generated task.

**Simu Dataset:** We generate balanced mixtures composed of three Dirichlet distributions, defined on the 3-dimensional probability simplex, \( \Delta^2 \). The corresponding Dirichlet parameters are \( \alpha_1 = (1, 1, 5) \), \( \alpha_2 = (25, 5, 5) \), \( \alpha_3 = (5, 7, 5) \). Each component is made biased towards one different vertex of the simplex, thereby simulating the softmax predictions of a deep model for a certain class. Additionally, each component captures a different type of distribution\(^12\): \( \text{Dir}(\alpha_1) \) corresponds to an exponential distribution on \( \Delta^2 \), \( \text{Dir}(\alpha_2) \) to an off-centered distribution with small variance, and \( \text{Dir}(\alpha_2) \) to a more centered distribution, with a wider variance. All the components contribute equally to the final mixture. In total, \( 10^3 \) samples are sampled. We perform 5 random runs, each using new examples.

**Results:** The third column of Table II reports the NMI and Acc scores of different approaches on the Simu dataset. As expected, K-DIRS and K-BETAS yielded close scores as they both model the Beta marginal densities. K-sBETAS yields lower scores due to the scaling factor, \( \delta > 0 \). However, it behaves better than all the other clustering methods thanks to a proper statistical modeling of simplex data.

2) **Clustering the Softmax Predictions of Deep Models:** We now compare the different clustering approaches on real-world distributions, where data points correspond to the softmax predictions of deep-learning models.

**Setup:** First, we employ the \( \text{SVHN} \rightarrow \text{MNIST} \) challenge, where a source model is trained on \( \text{SVHN} \) [36] and applied to the MNIST [37] test set, which is composed of \( 10^4 \) images labelled with 10 different semantic classes. Additionally, we experiment with the more difficult \( \text{VISDA-C} \) challenge [38], which contains 55388 samples split into 12 different semantic classes. For both \( \text{SVHN} \rightarrow \text{MNIST} \) and \( \text{VISDA-C} \), we use the common network architectures and training procedures detailed in [39]. For these closed-set UDA experiments, we use the optimal-transport Hungarian algorithm to obtain cluster-to-class label assignment, as detailed in Section IV-C-4.

**Results:** Table II shows interesting comparative results for clustering real-world softmax predictions. In the \( \text{SVHN} \rightarrow \text{MNIST} \) setting, simplex-tailored methods KL-K-MEANS and k-sBETAS clearly stand out in terms of both the NMI and Accuracy. In particular, the proposed k-sBETAS achieves the best scores. On the VISDA-C challenge, however, KL-K-MEANS yielded a performance below the baseline, similarly to the other standard clustering methods. K-DIRS failed to converge. In contrast, k-sBETAS outperforms the baseline by a margin, achieving the best scores.

Table III reports the running times of all the methods. When using MoM for parameter estimation, k-sBETAS has a running time close to or even lower than the standard GMM approach. Furthermore, a GPU-based implementation of k-sBETAS could be made even faster, which makes it appealing for large-scale datasets such as \( \text{VISDA-C} \).

**Feature Maps versus Simplex Predictions:** Interestingly, Table II shows that k-means, when applied to the logit points, yields scores that are similar or even better than when applied to the feature maps. This suggests that the classifier head, which predicts the logit points, does not deteriorate the semantic information of the bottleneck layer. Table III shows that logit and simplex clustering are computationally less demanding than clustering the feature maps. This is due to the fact that the dimension of the simplex points is \( K \), which is considerably smaller than the dimension of feature maps. In addition, Table II shows that the proposed simplex-clustering method substantially outperforms the K-MEANS feature-map clustering on the most difficult/realistic dataset, \( \text{VISDA-C} \).

\(^9\)The GMM code is available at: https://github.com/scikit-learn/scikit-learn/blob/0eac6d09b/sklearn/mixture/_gaussian_mixture.py.

\(^10\)The HSC code is available at: https://franknielsen.github.io/HSG/.

\(^11\)The codes for KL-K-MEANS, K-MEANS, K-MEDIANS, K-MEOIDS, K-MODES, K-DIRS, K-BETAS and k-sBETAS are available at our repository: https://github.com/tchiaroni/Clustering_Softmax_Predictions

\(^12\)We provide the 2D visualizations of these simulated distributions in Appendix, Fig. 7, available online.
It is worth noting that the explored softmax-prediction datasets, SVHN → MNIST and VISDA-C, are challenging in this particular black-box setting. This enables interesting comparisons with model-agnostic clustering methods. For reproducibility and future improvements, we made these black-box softmax predictions publicly available: https://github.com/fchiaroni/Clustering_Softmax_Predictions.

B. Transductive Zero-Shot and One-Shot Learning

We now demonstrate the general applicability and effectiveness of our method in the practical and realistic few-shot problem, which involves only an unknown fraction of the original set of classes in a small, unlabeled query (test) set [40]. We tackle transductive zero- and one-shot inference using the foundational CLIP model [4] by clustering its zero-shot softmax predictions. CLIP is based on two deep encoders, one for image representation and the other for text inputs. Coupled with projections, this dual structure yields image and text embeddings within the same low-dimensional space. Trained on a large-scale dataset of text-image pairs, CLIP maximizes the cosine similarity between the embedding of an image and its corresponding text description, which makes it well-suited for zero-shot prediction.

At inference time, the classification of a given image into one of K classes is given by the zero-shot softmax prediction of CLIP: \( \sigma(x)_j = \frac{\exp(z_j)}{\sum_{k} \exp(z_k)} \), where \( \log z_j \) evaluates the cosine similarity between the image embedding and the representation of a text prompt describing the class, typically “a photo of a [name of class k]” [4]. In the one-shot setting, we cluster softmax predictions of the same form, except that, for each class, the text embedding is replaced by the visual embedding of the labeled shot of the class, i.e., the single labeled image of the class.

**Experimental Setup:** We assessed our method using four distinct architectures of the CLIP visual encoder: ResNet-50, ResNet-101 [41], ViT-B/32, and ViT-B/16 [42]. Our experiments were performed across eight datasets: CIFAR-100 [43], Stanford-Cars [44], FGVC Aircraft [45], Caltech101 [46], Food101 [47], Flowers102 [48], Sun397-100 [49] and ImageNetv2-100 [50]. For Sun397-100 and ImageNetv2-100, we restrict the evaluation to the first 100 classes. For each dataset, we generated 100 query sets, each containing 64 unlabeled examples sampled from a random selection of 2 to 10 classes.

We compare our method with the inductive baseline, i.e., CLIP zero-shot prediction, and with transductive state-of-the-art few-shot learning methods, which leverage the statistics of the unlabeled query set, and operate on the feature maps: BD-CSPN [51], LAP-SHOT [52] and TIM [53]. Additional comparisons include K-MEANS, KL K-MEANS, and our k-SBETAS method, which we apply directly to the softmax predictions of CLIP. In the one-shot setting, the initialization prototype vector for each class is the CLIP-encoded image from the labeled support set.

In the zero-shot setting, the initialization prototype vector for each class is the CLIP-encoded text prompt. Post-convergence, we directly employ the argmax function for cluster-to-class assignments. This function assigns each estimated cluster to the class whose one-hot vector is nearest to the cluster centroid.

**Results:** Tables IV and V show the results for these One-shot and Zero-shot settings. A first key observation is the enhanced effectiveness of the probability-simplex methods over those based on the feature maps. On another note, one may observe that feature-map methods TIM and k-means show similar results in these settings. This observation is consistent with a previous work [54], which showed that, under certain mild conditions, along with common posterior models and parameter regularization, such models become equivalent. Overall, the proposed k-SBETAS method demonstrates a clear and consistent improvement over the inductive CLIP baseline and the other transductive few-shot methods across a variety of datasets and network encoders. This indicates its potential as a robust and versatile approach in enhancing the performance of foundation models, like CLIP, in practical applications with limited labeled data and unknown class distributions.

C. Real-Time UDA for Road Segmentation

We now consider a question that was, to the best of our knowledge, hitherto unaddressed in UDA segmentation: Can we adapt predictions from a black-box source model on a target set in real-time?

**Setup:** We address this question on the road GTA5 → road Cityscapes challenge. A source model is trained on GTA5 [55] and applied on the Cityscapes validation set [56]. Following [57], we use Deeplab-V2 [58] as the semantic segmentation network.

In our case, the model is exclusively trained for the binary task of road segmentation. The validation set contains 500 images with a 1024*2048 resolution. In order to simulate a real-time application, we treat each image independently, i.e., as a new clustering task, in which each pixel represents a point to cluster.

Towards Real-Time Running Time: Furthermore, to maximize the speed of execution, we downsample the model’s output probability map prior to fitting the clustering methods and then use the obtained densities to perform inference at the original resolution. Downsampling by a factor of 8 allows for an increase in the frame rate by almost two orders of magnitude, without incurring any loss in mIoU. Specifically, k-SBETAS takes on average 0.0165 seconds for clustering on each 128*256 subset of pixels and 0.0056 seconds for predictions on original 1024*2048 images, which represents a processing frequency of 45 images per second. Hardware used: CPU 11th Gen Intel(R) Core(TM) i7-11700K 3.60GHz, and GPU NVIDIA GeForce RTX 2070 SUPER.

**Results:** Corresponding NMI and mean IoU scores are displayed on Table VI, and show k-SBETAS outperforming low computational demanding competitors K- MEANS and KL K- MEANS by a large margin, with an improvement of 14 mIoU points over the pixel-wise inductive baseline. Fig. 1 provides qualitative results, in which k-SBETAS yields a definite visual improvement over the baseline. In addition, and consistently with the previous results observed in Table II for the UDA classification challenge, Table VI also shows that clustering the

---

13See Table XIII in the Appendix, available online, for more details.
probability simplex points may be more effective than clustering the logits.

D. Ablation Studies

Joint versus Factorised Density: Our proposed method draws inspiration from a factorized Beta density, i.e., each component of the simplex vector is considered independent from each other.

While such a simplifying assumption has significant analytical and computational benefits, it may also fail to capture the complexity of the target distribution. We explore this trade-off by comparing K-BETAS with K-DIRS, in which the full joint Beta density – corresponding to a Dirichlet density – is fitted at each iteration. Because there exists no closed-form solution to this problem, we resort to the approximate estimation procedure described in [27].

The table shows that K-DIRS and K-BETAS further boost the results of the existing Zero-Shot CLIP model by clustering its soft predictions on the query set.

| METHOD | LAYER USED | NETWORK |
|--------|------------|---------|
| CLIP   | FEATURE MAP | RN50    |
| 55.4   | 17.2       | 61.6    |
| 15.8   | 77.7       | 64.5    |
| 38.8   | 29.4       | 74.0    |
| 58.5   | 65.3       | 48.3    |
| 48.3   | 63.2       | 31.5    |
| BD-CPNF | FEATURE MAP | RN50    |
| 55.4   | 13.8       | 60.3    |
| 15.1   | 77.4       | 64.5    |
| 38.8   | 29.4       | 74.0    |
| 58.5   | 65.3       | 48.3    |
| 48.3   | 63.2       | 31.5    |
| LAP-SHOT | FEATURE MAP | RN50    |
| 55.4   | 13.9       | 60.3    |
| 15.1   | 77.4       | 64.5    |
| 38.8   | 29.4       | 74.0    |
| 58.5   | 65.3       | 48.3    |
| 48.3   | 63.2       | 31.5    |
| TIM    | FEATURE MAP | RN50    |
| 55.4   | 13.9       | 60.3    |
| 15.1   | 77.4       | 64.5    |
| 38.8   | 29.4       | 74.0    |
| 58.5   | 65.3       | 48.3    |
| 48.3   | 63.2       | 31.5    |
| K-MEANS | SIMPLEX     | RN50    |
| 55.4   | 13.9       | 60.3    |
| 15.1   | 77.4       | 64.5    |
| 38.8   | 29.4       | 74.0    |
| 58.5   | 65.3       | 48.3    |
| 48.3   | 63.2       | 31.5    |
produce similar scores on mixtures of Dirichlet distributions. In the meantime, k-Dirs outperforms k-Betas on SVHN → MNIST but fails to converge on the more challenging VISDA-C challenge, making it ill-suited to real-world applications.

**Effect of Class Imbalance:** To investigate the problem of class imbalance, we generate heavily imbalanced datasets. First, we create an imbalanced version of our synthetic Simu dataset, which we refer to as iSimus. Specifically, we weighted the 3 components of the Simu mixture with six different combinations using the imbalanced class proportions \(\{0.75, 0.2, 0.05\}\). Second, we consider a variant of VISDA-C, where class-proportions are sampled from a Dirichlet distribution with \(\alpha = 1_k\). We refer to this variant as iVISDA-Cs. We perform 10 random re-runs.

Table VII compares the clustering models on these two datasets. Displayed NMI scores for iSimus correspond to the average NMI scores obtained across the six different mixture proportions. NMI and IoU scores displayed for iVISDA-Cs correspond to the average scores obtained across ten different highly imbalanced subset variants of VISDA-C. K-sBetas (biased) refers to the proposed approach without the marginal probability term \(\pi_k\) in (14). These comparative results clearly highlight the benefit of the proposed k-sBetas formulation. Note that our unbiased formulation could theoretically apply to metric-based approaches but was systematically found to produce degenerated solutions in which all examples are assigned to a unique cluster.

**Parameter Estimation:** Concerning the parameter estimation for k-sBetas, we can observe in Tables III and VIII that using the method of moments (MoM) is more beneficial than the iterative MLE in practice, both in terms of computational cost and prediction performances.

**Effect of the Unimodal Constraint:** Table VIII empirically confirms that constraining a density-based clustering model to only consider mixtures of unimodal distributions is appropriate with softmax predictions from pre-trained source models. Yet, we can observe that disabling this constraint has no impact on k-sBetas results. The following paragraph explains such observations by the presence of \(\delta\).

**Effect of \(\delta\):** As a matter of fact, Table VIII results suggest that \(\delta\) interest is finally twofold in our experiments: It enables a more flexible clustering. Meanwhile, it also prevents estimating bimodal density functions. The latter point can be explained as follows. Bimodal distributions have a higher variance than unimodal or unimodal distributions because they partition the major portion of the density at the vicinity of two opposite interval boundaries simultaneously. Thus, when \(\delta\) is sufficiently large (e.g., with \(\delta = 0.15\) in these experiments), the scaled variance of the projection of \(x_i\) coordinates into the interval \([1+\delta, 1-\delta]\) becomes sufficiently small to inhibit the modeling of bimodal distributions. Complementary, Fig. 6 shows that setting \(\delta = 0.15\) provides consistent outperforming results on different datasets, and it allows fast convergence in terms of clustering iterations. Thus, we set \(\delta = 0.15\) along all the other presented experiments.

**Pre-Clustering: Centroid Initialization:** Parameter initialization plays a crucial role in unsupervised clustering. In Table IX, we compare the widely-used k-means++ initialization, designed for generic clustering, with our simplex-tailored vertex initialization. Regardless of the assignment method used, our initialization demonstrates up to a 7% improvement in accuracy over k-means++.\(^{14}\)

\(^{14}\)Table XIV in the Appendix, available online, further extends this comparison to other clustering approaches, showing similar benefits across every tested approach.
TABLE IX
K-sBETAS PREDICTION PERFORMANCES DEPENDING ON THE PARAMETER INITIALIZATION AND CLUSTER TO CLASS ASSIGNMENT

| APPROACH  | INIT  | ASSIGNMENT | SVHN→MNIST | VISDA-C |
|-----------|-------|------------|-------------|---------|
| K-MEAN++  | ARGMAX| 69.0 ± 6.1 | 50.0 ± 5.0  |         |
| K-MEANS++ | HUNG  | 69.8 ± 8.1 | 47.2 ± 5.2  |         |
| VERTEX INIT | ARGMAX | 73.5 ± 0.0 | 53.8 ± 0.0  |         |
| VERTEX INIT | HUNG  | 76.6 ± 0.0 | 56.0 ± 0.0  |         |

We compare k-means++ with the proposed Vertex Init, and also the centroid assignment argmax with the proposed optimal transport strategy (Hung).

Post-Clustering: Cluster to Class Assignment: We also show on Table IX the interest of the Hungarian algorithm, referred to as Hung, for aligning cluster-labels with class-labels. Compared to the argmax assignment for centroids in the context of such closed-set challenges, we recall that Hung aims to ensure that each cluster is assigned to a separate class. We can observe that the proposed Hung strategy is particularly interesting when using the vertex init. This suggests that optimal transport assignment is naturally relevant in situations where each cluster is likely to represent a separate class.

In order to provide a fair comparison with the state-of-the-art, we jointly applied these pre- and post-clustering strategies on every compared approach.

VI. DISCUSSION

This section discusses the potential limitations and extensions of the proposed approach.

Maximum Performance May be Upper-Bounded: We have shown that clustering softmax predictions with the proposed approach can efficiently improve source model prediction performances at a reasonable computational footprint. However, it is worth noting that we could not reach the prediction performances of the recent state-of-the-art self-training [3], which exploits and updates the model parameters end-to-end through an iterative process. This is because, as for every clustering algorithm, the maximum performance of the proposed approach is upper-bounded by the quality of the output probability simplex domain representation, over which we have no control.

Spatial Analysis Could be Complementary: The clustering model does not consider global or local spatial information. That would be worthwhile in challenges such as semantic pixel-wise segmentation to complement it with spatial post-processing techniques [59].

Clustering Softmax Predictions for Self-Training: Throughout this article, we have motivated the use of simplex clustering for efficient prediction adjustment of black-box source models. Nevertheless, simplex clustering could also play the role of pseudo-labeling along self-training strategies. This could be a simple and light generic alternative to feature map clustering [11], which would not imply cumbersome manipulation of high-dimensional and hidden feature maps.

If the Number of Classes is Large: In some contexts, the number of classes $K$ may be potentially large. For example, $K$ may correspond to millions of words in neural language modeling. It would be then interesting to envision strategies that reorganize the softmax layer for more efficient calculation [60]. In addition, distortion-based and probabilistic clustering methods usually require a sufficiently large number of points per class to produce consistent partitioning. Thus, if $K$ is too large with respect to the target set size, then the clustering could produce degenerate solutions. It would be interesting to explore hierarchical clustering strategies [61, 62] on the probability simplex domain, as these models can deal with a small amount of data points per class.

Low-Powered Devices: Systems such as mobile robots and embedded systems may not have the resources to perform costly neural network parameter updates on the application time. The presented framework can be viewed as an efficient plug-in solution for prediction adjustment in the wild when using such low-powered devices.

VII. CONCLUSION

In this paper, we have tackled the simplex clustering of softmax predictions from black-box deep networks. We found that existing distortion-based objectives of the state-of-the-art do not adequately approximate real-world simplex distributions, which can, for example, be highly peaked near the simplex vertices. Thus, we have introduced a novel probabilistic clustering objective that integrates a generalization of the Beta density, which we coin sBeta. This scaled variant of Beta is relatively more permissive; meanwhile, we constrain it to fit only strictly unimodal distributions and to avoid degenerate solutions. In order to optimize our clustering objective, we proceed with a block-coordinate descent approach, which alternates optimization w.r.t assignment variables and parameter estimation of cluster distributions, s.t. the introduced parameter constraints. The resulting clustering model, which we refer to as K-sBETAS, is both highly competitive and efficient on the proposed simulated and real-world softmax-prediction datasets, including class-imbalance scenarios, along which we performed our comparisons. Complementary, we emphasized the interest of simplex clustering throughout practical applications: Zero-shot and One-shot image classification, and Real-Time adjustment for road segmentation on full-size images.

Our future perspectives include extending the insights of probability simplex clustering to other challenges, such as pseudo-labeling for self-training procedures and other applications that require rapid correction of source models. Moreover, it may be interesting to complement the clustering with spatial and temporal information depending on the target data properties.

Overall, we hope that the presented simplex clustering framework and softmax-prediction benchmarks will encourage the community to give more attention to such facilitating, low-computational demanding, model-agnostic, plug-in solutions.

REFERENCES

[1] D. Wang, E. Shellhammer, S. Liu, B. Olshausen, and T. Darrell, “Tent: Fully test-time adaptation by entropy minimization,” Int. Conf. on Learn. Representations, 2021.

[2] M. Wang and W. Deng, “Deep visual domain adaptation: A survey,” Neurocomputing, vol. 312, pp. 135–153, 2018.
[3] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng, “Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 11, pp. 3602–3617, Nov. 2022.

[4] A. Jabbur et al., “Learning transferable visual models from natural language supervision,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 8748–8763.

[5] H. Xia, H. Zhao, and Z. Ding, “Adaptive adversarial network for source-free domain adaptation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 9010–9019.

[6] P. Colombo et al., “Transductive learning for textual few-shot classification in API-based embedding models,” in Conf. Empirical Methods Natural Lang. Process., 2023, pp. 4214–4231.

[7] F. Pereira, N. Tishby, and L. Lee, “Distributional clustering of english words,” 1994, arXiv:cmp-lg/9408011.

[8] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty, “Clustering with Bregman divergences,” J. Mach. Learn. Res., vol. 6, pp. 1705–1749, 2005.

[9] J. Wu, H. Xiong, and J. Chen, “SAIL: Summation-based incremental learning for information-theoretic clustering,” in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2008, pp. 740–748.

[10] K. Chaudhuri and A. McGregor, “Finding metric structure in information theoretic clustering,” in Proc. 21st Ann. Conf. Learn. Theory, 2008, pp. 391–402.

[11] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 132–149.

[12] F. Nielsen and K. Sun, “Clustering in Hilbert’s projective geometry: The concave minimization,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2018, pp. 10 062–10 072.

[13] R. S. Bradley, O. L. Mangasarian, and W. N. Street, “Clustering via hard and soft assignment methods for clustering,” in Proc. Int. Conf. Mach. Learn., vol. 27, no. 2, pp. 2038–2047, 2000.

[14] T. Minka, “Estimating a Dirichlet distribution,” Tech. Rep., MIT, 2000.

[15] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty, “Clustering via hard and soft assignment methods for clustering,” in Proc. Int. Conf. Mach. Learn., vol. 38, pp. 293–306, 2020.

[16] M.-E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” in Proc. Int. Conf. Mach. Learn., 2008, pp. 740–748.

[17] J. Liang, D. Hu, and J. Feng, “Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 6028–6039.

[18] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” in Proc. 18th Annua. ACM-SIAM Symp. Discrete Algorithms, New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[19] I. Ziko, J. Dolz, E. Granger, and I. B. Ayed, “Laplacian regularized few-shot learning,” in Proc. 13th AAAI Conf. Artif. Intell., 2020, pp. 1459–1467.

[20] H. W. Kuhn, “The hungarian method for the assignment problem,” Nav. Res. Logistics Quart., vol. 2, no. 1/2, pp. 38–53, 1955.

[21] C. Biernacki, G. Celeux, and G. Govaert, “Assessing a mixture model for clustering with the integrated completed likelihood,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 7, pp. 719–725, Jul. 2000.

[22] I. Ziko, J. Dolz, E. Granger, and I. B. Ayed, “Laplacian regularized few-shot learning,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 11 660–11 670.

[23] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” in Proc. 18th Annua. ACM-SIAM Symp. Discrete Algorithms, New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[24] I. M. Ziko, E. Granger, and I. B. Ayed, “Scalable Laplacian K-modes,” in Proc. 13th AAAI Conf. Artif. Intell., 2020, pp. 1459–1467.

[25] I. Ziko, J. Dolz, E. Granger, and I. B. Ayed, “Laplacian regularized few-shot learning,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 11 660–11 670.

[26] I. Ziko, J. Dolz, E. Granger, and I. B. Ayed, “Laplacian regularized few-shot learning,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 11 660–11 670.

[27] I. Ziko, J. Dolz, E. Granger, and I. B. Ayed, “Laplacian regularized few-shot learning,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 11 660–11 670.

[28] I. Ziko, J. Dolz, E. Granger, and I. B. Ayed, “Laplacian regularized few-shot learning,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 11 660–11 670.

[29] I. Ziko, J. Dolz, E. Granger, and I. B. Ayed, “Laplacian regularized few-shot learning,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 11 660–11 670.

[30] I. Ziko, J. Dolz, E. Granger, and I. B. Ayed, “Laplacian regularized few-shot learning,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 11 660–11 670.
Malik Boudiaf obtained his MSc in Aeronautics & Astronautics from Stanford University in 2019, and his MEng in Aerospace Engineering from ISAE-Supaero, France in 2017, and the PhD degree from ÉTS Montréal, Canada, supervised by Prof. Ismail Ben Ayed and Prof. Pablo Piantanida. His research lied between computer vision, information theory, optimization, and their application to few-shot/unsupervised learning. He is currently a senior machine learning engineer in computer vision and natural language processing with Stealth Startup, Montreal, Canada.

Amar Mitiche received the licence Es sciences degree in mathematics from the University of Algiers, and the PhD degree in computer science from the University of Texas at Austin. He is currently a professor with the Department of Telecommunications (INRS-EMT), Institut National de la Recherche Scientifique (INRS), Montreal, QC, Canada. His research is in computer vision and pattern recognition. He has written several articles on the subjects, as well as three books: Computational Analysis of Visual Motion (Plenum Press, 1994), Variational and Level Set Methods in Image Segmentation (Springer, 2011), with Ismail Ben Ayed, and Computer Vision Analysis of Image Motion by Variational Methods (Springer, 2014), with J. K. Aggarwal. His current interests include image segmentation, image motion analysis, and pattern classification by neural networks.

Ismail Ben Ayed is currently a full professor with ÉTS Montréal. He is also affiliated with the CRCHUM. His research interests include computer vision, optimization, machine learning, and medical image analysis algorithms. He authored more than 100 fully peer-reviewed papers, mostly published in the top venues of those areas, along with two books and seven US patents. In recent years, he gave more than 30 invited talks, including four tutorials at flagship conferences (MICCAI’14, ISBI’16, MICCAI’19 and MICCAI’20). His research has been covered in several visible media outlets, such as Radio Canada (CBC), Quebec Science Magazine and Canal du Savoir. His research team received several recent distinctions, such as the MIDL’19 best paper runner-up award and several top-ranking positions in internationally visible contests. He served as Program Committee for MICCAI’15, MICCAI’17 and MICCAI’19, and as program chair for MIDL’20. Also, he regularly serves as a reviewer for the main scientific journals of his field and was selected several times among the top reviewers of prestigious conferences (such as CVPR’15 and NeurIPS’20).

Florent Chiaroni received the dipl- ing degree in computer science and electronics from ESTIA, France, in 2016, the MSc degree in robotics and embedded systems from the University of Salford Manchester, U.K., in 2016, and the PhD degree in signal and image processing from the University of Paris Saclay, in 2020, with VEDECOM Institute, Versailles, France, and Université Paris-Saclay, Centre national de la recherche scientifique (CNRS), CentraleSupélec, Gif-Sur-Yvette, France. He during the course of this research, was a postdoctoral fellow with ÉTS Montreal, Canada and Institut National de la Recherche Scientifique (INRS), Montréal, Canada. He is currently a research scientist with Thales Research and Technology (TRT) Canada, Thales Digital Solutions (TDS), CortAx, Montreal, Canada. His current research interests include civil unmanned aerial vehicles (UAVs) and efficient weakly-supervised learning for visual pattern analysis.