Zero Shot Learning with the Isoperimetric Loss

Shay Deutsch
Department of Mathematics
University of California Los Angeles
shaydeu@math.ucla.edu

Andrea Bertozzi
Department of Mathematics
University of California Los Angeles
bertozzi@math.ucla.edu

Stefano Soatto
Department of Computer Science
University of California Los Angeles
soatto@cs.ucla.edu

Abstract

We introduce the isoperimetric loss as a regularization criterion for learning the map from a visual representation to a semantic embedding, to be used to transfer knowledge to unknown classes in a zero-shot learning setting. We use a pre-trained deep neural network model as a visual representation of image data, a Word2Vec embedding of class labels, and linear maps between the visual and semantic embedding spaces. However, the spaces themselves are not linear, and we postulate the sample embedding to be populated by noisy samples near otherwise smooth manifolds. We exploit the graph structure defined by the sample points to regularize the estimates of the manifolds by inferring the graph connectivity using a generalization of the isoperimetric inequalities from Riemannian geometry to graphs. Surprisingly, this regularization alone, paired with the simplest baseline model, outperforms the state-of-the-art among fully automated methods in zero-shot learning bechmarks such as AwA and CUB. This improvement is achieved solely by learning the structure of the underlying spaces by imposing regularity.

1. Introduction

Motivating example. A pottopod is a pot with limbs.[1] Not even a single example image of a pottopod is needed to find one in Fig. 1. However, one has surely seen plenty of examples of animals with limbs, as well as pots. In zero-shot learning (ZSL) one aims to exploit models trained with supervision, together with maps to some kind of attribute or "semantic" space, to then recognize objects as belonging to classes for which no previous examples have ever been seen.

Figure 1. Find the Pottopod (Courtesy of [omitted for anonymous review]).

[1]Example courtesy of [omitted for anonymous review]

The ingredients of a ZSL method are illustrated in Fig. 2. Seen samples X_s and their corresponding labels Y_s are used to train a model ϕ, typically a deep neural network (DNN), in a supervised fashion, to yield a vector in a high-dimensional (visual embedding) space Z. At the same time, a function s maps semantic attributes such as "has legs," "is short," or simply a word embedding of the ground truth (seen and unseen) labels of interest $Y = \{Y_s, Y_u\}$, to some metric space. The name of the game in ZSL is to learn a map ψ, possibly along with other components of the diagram, from the visual embedding space Z to the semantic space S, that can serve to transfer knowledge from the seen labels (reflected in Z) to the unseen ones. Alternatively, one can try to cluster samples with unseen labels in the visual embedding space Z, and then associate clusters

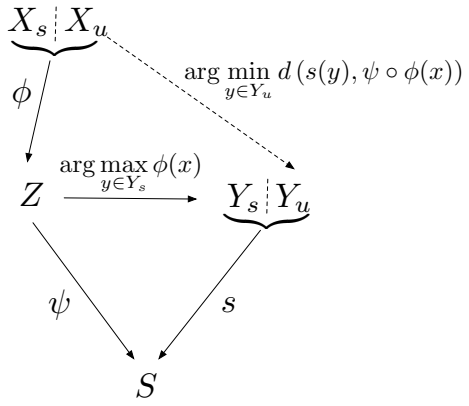

Figure 2. Illustration of the components of a ZSL algorithm: For images in the set with seen labels X_s, the labels can be estimated by maximum a-posterior over labels in the seen set Y_s on the visual representation $\phi(X_s)$. For the unseen labels, there is no direct connection to the data because they are not seen during training. Inference is then indirect: A visual representation is inferred, and from there a semantic representation, which is compared to the semantic representation of unseen labels, minimizing *some* distance in the semantic space, over all possible unseen labels Y_u.

to unseen labels.

Focus on regularization. Transfer of knowledge hinges on some kind of regularity of the maps involved in ZSL. In practice, the visual embedding space Z and the semantic embedding S are only known through discrete samples, and the maps are learned restricted to these finite samples. One crucial theme in ZSL is, interpreting sample embeddings as "noisy points" on the otherwise differentiable manifolds Z and S, to attempt to regularize the spaces Z, S, and/or the map between them.

Key contribution. Of all the various components of a ZSL method, we choose the simplest possible [27], except for the regularization of the semantic map. There, we introduce a sophisticated model, based on an extension of the isoperimetric inequalities from Riemannian geometry to discrete structures (graphs). We treat sample visual embeddings as vertices in a graph, with affinities as edges, and the visual-to-embedding map ψ interpreted as a linear function on the graph. We then introduce the isoperimetric loss (IPL) to enforce regularity on the domain Z based on flow of the function defined on it through the boundary of sets of a given volume. The resulting regularized graph is informed by both the visual and semantic maps. We use it to perform clustering and map clusters to labels. Therefore, we take a very simple visual-to-semantic embedding function, namely a linear transformation, and indirectly regularize it by regularizing its domain and range spaces.

We expected our regularization to significantly improve the baseline [27] on which our ZSL model is based. We did not expect it to improve it to surpass the (far more sophisticated) state-of-the-art in the two most common benchmark datasets used in ZSL, namely AwA [19] and CUB [30]. Yet, we show it did in Sect. 5. In some cases, it even outperformed methods that used human annotation for the unseen labels.

At heart, we solve a topology estimation problem. We determine the connectivity between nodes of the visual embedding graph, which defines a topology in that space informed by the semantic representation of seen attributes. Much of the literature in this area focuses on what kind of graph signal (embedding, or descriptor) to attribute to the nodes, whereas the connectivity of the graph is decided a-priori. We focus on the complementary problem, which is to determine the graph connectivity and learn the graph weights. Unlike other approaches, the connectivity in our method is informed both by the value of the visual descriptors at the vertices, and the values of the semantic descriptors in the range space.

Our framework allows us to use automated semantic representation to perform ZSL, resulting in a framework which is entirely free of human annotation.

Before we review related work in Sect. 6, we derive a general formalism for ZSL that enables relating the different contributions to one another in Sect. 2. With that baggage, we can immediately describe our approach in Sect. 3 and go into

the specific of our graph regularization and clustering in Sect. 4. we then return to discuss related work before drawing our conclusions in Sect. 6.

2. Preliminaries

We first describe a general formalization of ZSL that helps place our contribution in context. Every ZSL includes a supervised component, which results in a visual embedding, a collection of unseen labels or attributes, a map from these attributes to a vector (semantic) space, and a map from visual to semantic spaces. It is important to understand the assumptions underlying the transfer of information from seen to unseen attributes, which translates in regularity assumptions on the visual-to-semantic map.

Supervised component. In standard supervised classification, a dataset $\mathcal{D}_s$ is given where both the input data x and the output labels y_s are *seen*:

$$\mathcal{D}_s = \{x^i, y_s^i\}_{i=1}^N \tag{1}$$

where the set of seen labels, for instance 1 = "cat" and 2 = "dog," is indicated by Y_s, with cardinality $|Y_s| = n_s$. The data belong to X, for instance the set of natural images. The goal of supervised learning is to infer the parameters w of a function $\phi_w : X \to \mathbb{R}_+^K$ that approximates the (log)-posterior probability over Y_s,

$$\phi_w(x)_j \simeq \log P(y_s = j|x). \tag{2}$$

where the subscript j denotes the j-th component of the vector $\phi_w(x)$. At test time, given an unseen image x, one can infer the unknown label $\hat{y}$ associated with it as the maximum a-posteriori estimate

$$\hat{y}(x) = \arg\max_{y \in Y_s} \phi_w(x)_y. \tag{3}$$

We indicate with Z the (latent, or representation) space where the data X are mapped,

$$z^i = \phi_w(x^i) \in Z. \tag{4}$$

Visual embedding. Although z^i can be interpreted as log-probabilities, one can simply consider them as an element of a vector space of dimension at least K, $Z \subset \mathbb{R}^K$, called "visual embedding." It is also customary to use intermediate layers of a deep network, rather than the last one that is used for classification, as a visual embedding, so in general $K \neq n_s$. We want the formalism to be flexible, so we do not constrain the dimension of the embedding to be the same of the dimension of the seen classes.

Unseen labels. In ZSL there is a second set of "unseen" labels[2] Y_u, disjoint from the first $Y_u \cap Y_s = \emptyset$. We call $Y = Y_u \cup Y_s$. At training time we do not have any sample images with labels in Y_u. However, we do at test time.

Zero-shot learning. The goal of ZSL is to classify test images as belonging to the unseen classes. That is, to learn a map from X to Y_u. Absent any assumption on how the unseen labels are related to the seen labels, ZSL makes little sense.

Assumptions. In ZSL one assumes there is a "semantic" metric vector space $S \subset \mathbb{R}^M$, to which all labels – seen and unseen – can be mapped via a function $s : Y \to S$. If the metric is Euclidean, a distance between two labels, y^i, y^j, can be induced via $d_s(y^i, y^j) = \|s(y^i) - s(y^j)\|$. Otherwise, any other distance $d(s(y^i), s(y^j))$ on S can be used to find the label associated to an element $\sigma \in S$ of the semantic space (embedding), for instance using a nearest-neighbor rule

$$\hat{y}(\sigma) = \arg\min_{y \in Y} d(s(y) - \sigma). \tag{5}$$

Note that the minimum could be any label, seen or unseen. This is just a metric representation of the set of labels, independent of the ZSL problem.

The second assumption is that there exists a map $\psi : Z \to S$ from the visual to the semantic embedding, which can be learned to map embeddings of seen classes z to semantic vectors $\sigma = \psi(z)$ in such a way that they land close to the semantic embedding $s(y_s)$ of the seen labels:

$$\arg\min_{\psi} \sum_{i=1}^{N} d(s(y_s^i) - \psi \circ \phi_w(x_s^i)). \tag{6}$$

[2] 'A misnomer, since one knows ahead of time what these labels are, for instance 3 = "sailboat", 4 = "car." However, one is not given images with those labels during training. So, while the labels are seen, image samples with those labels are not seen during training.

One could learn both the visual embedding ϕ_w and the visual-to-semantic map ψ simultaneously, or fix the former and just learn the latter. In some cases, even the latter is fixed.

Validation. The merit of any ZSL approach is usually evaluated empirically, since the assumptions cannot be validated absent samples in the unseen label class or knowledge of the transfer between the seen and unseen tasks. Once training has terminated and we have embeddings $\phi_{\hat{w}}$ and ψ, given test data x^i, we can compare the imputed labels obtained via

$$\hat{y}(x_u) = \arg\min_{y \in Y_u} d(\psi \circ \phi_{\hat{w}}(x_u), y) \tag{7}$$

with labels y_u in the validation set. The construction of this loss function (6) is illustrated in Fig. 2.

Baseline. ZSL methods differ by whether the learn both the visual and semantic embedding, only one, or none, the method and criterion used for learning. Since the unseen labels are never seen during training, the transfer of information from seen to unseen labels hinges on the *regularity* of the learned maps. For this reason, much of the recent work in ZSL aims to explore different regularization methods for the learned maps. The simplest case, which requires no regularization, is to assume that all the maps are linear: $\phi(x) = Fx$ and $s(z) = Vz$ for suitable matrices F, V [27]. The results are not state-of-the-art (see Sect. 5), but we nevertheless adopt this baseline and focus on regularizing, rather than the map ψ directly, the spaces Z and S, which is our key contribution.

3. Description of our approach

We select a fixed visual embedding ϕ consisting of a ResNet101 architecture trained on ImageNet using all classes, to map images x onto a 2048-dimensional embedding $z = \phi(x)$. We assume $y_s^i \in Y_s$ is a subset of $n_s = 40$ to 150 classes depending on the dataset: In AwA [19] there are 50 classes, of which we consider 40 as seen and sequester $n_u = 10$ as unseen. In CUB [30] there are 200 classes, of which we consider 150 as seen and the rest unseen. We exploit a fixed semantic map s from text attributes, namely labels, onto a vector space $S = \mathbb{R}^M$ with dimension $M = 100$ (AwA) to 300 (CUB), using Word2Vec [22]: The map $\psi : Z \to S$ is assumed linear, $\psi(z) = Vz$, where V is an $M \times 2048$ matrix, learned as in [27] using the seen dataset $\mathcal{D}_s$. To facilitate comparison to some algorithms we also use a VGGverydeep-19 on CUB rather than ResNet101.

At test time, given data x_u^i with unseen labels, we compute the visual representation $z_u^i = \phi(x_u^i) \in \mathbb{R}^K$ and then semantic embeddings $s_u^i = Vz_u^i$ for all $i = 1, \dots, N_u$. We construct a graph $\mathcal{G} = (Z, W)$ with vertices[3] $z_u^i \in Z$ and edges $\{w_{ij}\} = W$ that measure the affinities between embeddings $w_{ij} = \langle z_u^i, z_u^j \rangle$. $\mathcal{G}$ is a discrete representation of the smooth manifold $\phi(X) \subset Z$. The function ψ, restricted to $\mathcal{G}$, yields s_u^i, with range $\psi(Z) \subset S$, which we also assume to be a smooth manifold. In practice, because of the finite sampling and the nuisance variability in the descriptors, both the domain and range of ψ are far from smooth.

Key idea. Rather than smoothing the map $\psi : \mathcal{G} \to S$, we assume it is linear in the embedding space, and instead smooth both its domain and range. We seek a non-parametric deformation represented by changes in the connectivity matrix W of the underlying graph, that minimizes the isoperimetric loss (IPL). This is a form of regularization which we introduce in the field of ZSL. The IPL measures the flow through a closed neighborhood relative to the area of its boundary. For two-dimensional (2-D) surfaces in 3-D, it is minimized when the neighborhood is a sphere. The IPL extends this intuition to higher dimensions.

Application to ZSL. The result of our regularization is a new graph $\mathcal{G}'$, informed by the domain, range and map of the function ψ. We perform spectral clustering on $\mathcal{G}'$ to obtain a set of $n_u = |Y_u|$ clusters $\{c_1, \dots, c_{n_u}\}$. Each of these clusters can then be associated with a label in the unseen set Y_u. We do not need to know the association explicitly to evaluate our method. However, one could align the clusters to the semantic representation of the unknown labels if so desired.[4] This does not have any impact on our method.

[3] We abuse the notation to indicate with Z the visual embedding space, the range of the function $\phi(X)$, which we assume to be a differentiable manifold, and the vertices of a discrete graph sampled from Z.

[4] For instance, by finding a transformation U that solves

$$\min_{U \geq 0, U = U^T} \sum_{y_j \in Y_u} d\left(Us(y_j), \frac{1}{|c_j|} \sum_{z_u^i \in c_j} s_u^i \right). \tag{8}$$

If so desired, one could also add to the regularization procedure a term to align the clusters to the semantic representations of the unseen labels:

$$\sum_{z_i \in c_j} d(Vz_i, Us(y_j)). \tag{9}$$

We, however, skip this as the alignment issue is beyond our focus in this paper.

Results. In general, there is no "right" regularizer, so we validate our approach empirically on the two most common datasets for ZSL, namely AwA and CUB. Compared to the current best methods that do not use any manual annotation, Zero-IPL reduces errors by 3.55 % on AwA2 (increased precision from 78.9% to 81.7%), and by 6.91% (increased precision from 36.9% to 39.45%) on CUB.

Next, we describe the specific contribution, which is the smoothing of the graph-representation of ψ, in detail.

4. Regularization

In this section we describe in more detail our graph smoothing based on the isoperimetric loss (IPL).

Our baseline gives us a graph $\mathcal{G}$ with weights w_{ij} that we want to modify. We can think of these weights as "noisy," and seek a way to regularize them, by exploiting also the function ψ defined on $\mathcal{G}$ that yields semantic embeddings. Our regularization criterion is to achieve some level of compactness of bounded subsets: For a collection of subsets of the vertices with fixed size (corresponding to the volume of a subset) we want to find the subsets with the smallest size boundary. Why this might be a good criterion rests on classical differential geometry of Riemannian manifolds, where in the most basic case, the most compact manifold that encloses a fixed area with minimum size boundary is a circle. Our criterion, quantified by the isoperimetric gap, generalizes this bias towards compactness to more general sets. In the continuous setting, for a Riemannian manifold the isoperimetric gap is zero when it is a sphere.

4.1. Isoperimetric loss

Let $B_r(\xi)$ be the ball around $\xi \in Z$ of radius r, that is the set of nodes within a distance $d_{\mathcal{G}}$ less than r. Let

$$\mu_i^{(\xi)} = \sum_{i\sim j,\ d_{\mathcal{G}}(j,\xi)<d_{\mathcal{G}}(i,\xi)} w_{ij} \tag{10}$$

be the flow from i towards ξ, that is, the sum of weights of edges connecting j with points closer to ξ. The geodesic flows $\mu_r^{(\xi)}$ are

$$\mu_r^{(\xi)} = \sum_{\substack{i \\ d_{\mathcal{G}}(i,\xi)=r}} \mu_i^{(\xi)}. \tag{11}$$

Note that $\mu_r^{(\xi)}$ equals $\mu(\partial B_r(\xi))$ - the sum of all the edges that connect vertices in $B_r(\xi)$ and its complement in Z, where μ is a measure on the edges in the boundary $\partial B_r(\xi)$. Next we define the isoprimetric inequality.

Definition 1 *[7] We say that a graph $\mathcal{G}$ has an isoperimetric dimension δ with a constant c_δ, if, for every bounded subset $B_r(\xi)$ of Z, the number of edges between $B_r(\xi)$ and the complement of $B_r(\xi)$, $Z \setminus B_r(\xi)$ satisfies*

$$\mu(\partial B_r(\xi)) \geq c_\delta(\mu(B_r(\xi)))^{1-\frac{1}{\delta}} \tag{12}$$

where $\partial B_r(\xi)$ denotes the boundary of $B_r(\xi)$.

In our notation, we have that $\partial B_r(\xi) = \mu_r^{(\xi)}$.

Next, we define the *isoperimetric gap* using the isoperimetric inequality above, which is the quantity to be minimized in the isoperiemtric loss:

Definition 2 *The* isoperimetric gap *is defined as*

$$\beta(\xi;\delta,W) \doteq c_\delta \left(\sum_{i,j\in B_r(\xi)} w_{i,j} \right)^{1-\frac{1}{\delta}} - \mu_r^{(\xi)} \tag{13}$$

To minimize the gap we propose solving the following optimization problem:

$$\begin{gathered} \min_{W\geq 0} \sum_{\xi\in Z} \sum_{\substack{f_{s_u^i,s_u^j} w_{i,j} + \lambda\beta(\xi;\delta,W) \\ z_u^i, z_u^j \in B_r(\xi)}} \\ \text{s.t. } 0 \leq w_{i,j} \leq 1 \,\forall\, i,j \end{gathered} \tag{14}$$

where $f_{s_u^i, s_u^j}$ is a function of the embedding distance between s_u^i and s_u^j, and λ is a positive scalar tuning parameter. Note that the gap β depends on δ, the isoperimetric dimension, which is unknown, and will have to be approximated. To minimize the gap we require to add constraints on the flows $\mu_r^{(\xi)}$, which we obtain by using the following inequality

$$\mu(B_r(\xi)) \leq \frac{\mu_r^{(\xi)}}{\delta}(r+1) \tag{15}$$

which holds for a bounded ball $B_r(\xi)$ for general graphs with isoperimetric dimension δ under some weak hypothesis [7], where it was suggested to approximate δ as follows

$$\delta = \nu_{r+1} w_0 \tag{16}$$

where the quantities ν_r and w_0 relates to properties of flow on the graph:

$$\nu_r = \inf_{\substack{\xi \in V \\ x \in B_r(\xi)}} \frac{d(x)}{\mu_x^{(\xi)}} \tag{17}$$

and

$$w_0 = \min_{i \in Z} \sum_{\substack{j \\ i \sim j}} w_{i,j} f_{i,j} \tag{18}$$

where f is some measure of the distance between nodes i and j on the graph.

Note that the upper bound in (15) is not tight, and moreover, the flows $\mu_r^{(\xi)}$ are affected most by noise, which typically increases the gap. Therefore, we choose a function g of the flows which provides a lower bound to the right hand side expression by using

$$g(\mu_r^{(\xi)}) = \min_{\xi \in Z} \frac{\mu_r^{(\xi)}}{\tilde{\delta}}(r+1) \tag{19}$$

where $\tilde{\delta} = w_0 \tilde{\nu}_{r+1}$, where the inf value in (17) is replaced with a median value in $\tilde{\nu}_r$ and thus our estimated $\tilde{\delta}$ satisfies $\delta < \tilde{\delta}$. Using the estimates for the geodesic flows above, we suggest solving the following relaxation to the isoperimetric loss

$$\begin{aligned} \min_{W \geq 0} & \sum_{\xi \in V} \sum_{\substack{z_u^i, z_u^j \in B_r(\xi)}} f_{s_u^i, s_u^j} w_{i,j} \\ \text{s.t.} & \sum_{z_u^i, z_u^j \in B_r(\xi)} w_{i,j} = g(\mu_r^{(\xi)}) \ \ \forall \xi \in Z \\ & 0 \leq w_{i,j} \leq 1 \, \forall i, j \end{aligned} \tag{20}$$

where $g(\mu_r^{(\xi)})$ is a function of the geodesic flows $\mu_r^{(\xi)}$ which further constrains them.

Thus, we obtain an approximation of the gap as

$$\tilde{\beta}(\xi) = \sum_{i,j \in B_r(\xi)} w_{i,j} - g(\mu_r^{(\xi)}). \tag{21}$$

which we minimize in (20). The algorithm is summarized in pseudo code in Algorithm 1.

4.1.1 Spectral Reduction

In addition to minimizing the IPL directly, we introduce a spectral reduction method for the isoperimetric inequalities, which reduces the isoperimteric gap directly in the spectral domain. The advantage of this method is that it does not require to estimate the isoperimetric gap directly and is also very fast. Specifically, we use the Spectral Graph Wavelet Transform [14] of the semantic embedding space s_u, and set eigenvalues $\{\lambda_i\}$ of the unnormalized Laplacian $\mathbf{L} = \mathbf{D} - W$ to zero beyond a nominal rank, and then recompute the adjacency matrix. More in detail, for each of the semantic dimensions e of s_u, let

Algorithm 1: Learning the Graph Connectivity Structure

1: **Input**: $z_u = \left\{z_u^i\right\}_{i=1}^{N_u}$, $s_u = \left\{s_u^i\right\}_{i=1}^{N_u}$ distance function f, k nearest neighbor parameter, r

2: **Step 1**: Construct k nearest neighbor graph $\mathcal{G} = (Z, W)$ from $\left\{z_u^i\right\}_{i=1}^{N_u}$ (using cosine similarity), extract $\left\{B_r(z_u^i)\right\}_{i=1}^{N_u}$ from $\mathcal{G}$ using the graph distance $d_{\mathcal{G}}$, r

3: **Output:** $\mathcal{G} = (Z, W)$, $\left\{B_r(z_u^i)\right\}_{i=1}^{N_u}$

4: **Step 2:** Using $\mathcal{G} = (Z, W)$, $\left\{B_r(z_u^i)\right\}_{i=1}^{N_u}$, s_u, r , estimate $\tilde{\delta}$, $\mu_r^{(\xi)}$ $g(\mu_r^{(\xi)})$ using (11) and (19)

5: **Step 3**: **Input:** graph $\mathcal{G} = (Z, W)$, $g(\mu_r^{(\xi)})$

6: **Step 4**:**Solve**

$$\begin{aligned} \min_{W_s \geq 0} \textstyle\sum_{\xi \in V} \sum f_{s_u^i, s_u^j} w_{i,j}^s \\ \textstyle\sum_{z_u^i, z_u^j \in B_r(\xi)} \\ \text{s.t.} \quad \sum_{z_u^i, z_u^j \in B_r(\xi)} w_{i,j}^s \quad = g(\mu_r^{(\xi)}) \;\; \forall \xi \in Z \\ 0 \leq w_{i,j}^s \leq 1 \, \forall i, j \end{aligned} \tag{22}$$

7: **Output:** A new Semantic embedding graph $\mathcal{G}' = (S, W_s)$

$s_u^i(e)$ be a component of s_u^i in a fixed dimension e

$$\begin{aligned} \tilde{s}_u^i(e) = \sum_{j=1}^{N} s_u^j(e) \sum_{k=0}^{r} a_k \sum_{l=1}^{N} \lambda_l^k \phi_l(j) \phi_l(i) = \\ \sum_{j=1}^{N} s_u^i(e) \sum_{k=0}^{r} a_k (\mathbf{L}^k)_{i,j} \end{aligned} \tag{23}$$

a filtered signal of a fixed dimension e of s_u^i. The coefficients a_k are constants of a polynomial function and for a specific choice correspond to spectral graph wavelet (SGW) coefficients. Note that the terms $\sum_{k=0}^{r} a_k (\mathbf{L}^k)_{i,j}$ can be interpreted as the localized spectral transform of the graph around the ball $B_r(z_u^i(i))$, which is non-zero for all z_u^j in $B_r(z_u^i)$ and vanishes for all $z_u^j \notin B_r(z_u^i)$. With the SGW transform, we employ a redundant representation with r polynomials $\kappa_{s(e)}(\lambda)$, $1 \leq e \leq r$ with corresponding scaling $s(e)$. Next choose the smallest $1 < r_0 < r$ such that the polynomial function $\kappa_{s(e)}(\lambda) = \sum_{k=1}^{r} a_k (\mathbf{L}^k)_{i,j}$ vanishes above $\lambda > d(i)$. Then, for all SGW coefficients $\tilde{s}_u^i(e)$ where the corresponding filter $\kappa_{s(e)}(\lambda) > 0$, $e \geq r_0$ for some $\lambda_l > d(i)$, we annihilate all terms $\tilde{s}_u^i(e)$, which has the effect of shrinking the boundaries of each ball around each vertex i and thus reducing the isoperimetric loss directly in the spectral domain. Take the inverse transform to obtain the denoised signal and construct the new graph.

4.1.2 Clustering and validation in ZSL

We employ a standard procedure for spectral clustering as follows: the input is the graph $\mathcal{G}' = (S, W_s)$ obtained from applying the IPL algorithm, and the number of clusters, n_u. Construct the Laplacian matrix $\mathbf{L}s$ using W_s and compute the first n_u eigenvectors of $\mathbf{L}s$. Letting $\Phi \in \mathbb{R}^{N_u \times n_u}$ correspond to the first n_u eigenvectors of $\mathbf{L}s$ stacked in a matrix form, we cluster the vectors $\Phi_i \in \mathbb{R}^{n_u}, i = 1, ..N_u$ corresponding to the rows of Φ into n_u classes, using the k-means algorithm. Once n_u clusters are found, we can associate each of them to a different unseen label. While it is not required that the semantic embedding of the unseen labels $s(y_u)$ correspond to the clusters in the same space, mapped from the visual embeddings, this alignment can be performed *post-hoc*. For the purpose of comparison, however, it is sufficient to perform the assignment by searching over permutation of the unknown labels. Since we have at most 50 unseen labels in our experiments, this is not a bottleneck. More in general, one may consider introducing the alignment as part of the regularization, but this is beyond our scope in this paper.

5. Experimental Results

Experimental Settings. In the first set of experiments, we restrict our comparisons to approaches that are fully automated beyond the definition of the visual embedding (best performance is marked in boldface). In addition, we also report the

Method/Data	AwA1	AwA2
EZSL [27]	58.2	58.6
SJE [4]	65.6	61.9
ALE [3]	59.9	62.5
LatEm [31]	50.8	-
DEVISE [12]	50.4	-
SynC [6]	58.6	-
UDA [16]	75.6	-
CAPD [26]	64.73	-
Kernel ZSL [34]	71.0	70.51
DEM [35]	68.4	67.1
RELATION NET [29]	68.2	64.2
QFSL [28]	-	79
MSMR[9]	80.21	78.9
Proposed	**80.6**	**81.7**

Table 1. Mean average precision accuracy (top-1 in %) results using our method compared to the state of the art in ZSL on the AwA1 and AwA2 datasets. Best performance using automated semantic representation is marked in boldface. The evaluation for the state of the art methods which are using human semantic annotation is also presented (all methods which are using human annotation are marked in red).

evaluation of the state of the art methods that have access to access to embeddings of ground-truth semantic attributes.

Using our approach we choose each component of our ZSL pipeline to be the simplest possible one, corresponding to the baseline [27]. A sanity check is whether our proposed regularization scheme improves over this baseline. Ideally, however, our method would take the baseline beyond the state-of-the-art.

To test this hypothesis, we use the two most common benchmarks for ZSL, AwA and CUB. AwA (Animals with Attributes) consists of 30,745 images of 50 classes of animals and has a source/target split of 40 and 10 classes, respectively. In addition we test on the new released dataset AWA2 which consists of 37,322 images of 50 classes which is an extension of AwA (which will be refereed from now and on AwA1). AwA2 also has source/target of 40 and 10 classes respectively with a number of 7913 unseen testing classes. We used the proposed new splits for AwA1 and AwA2 [33].

The CUB dataset contains 200 different bird classes, with 11,788 images in total. We use the standard split [6] with 150 classes for training and 50 disjoint classes for testing [33] which is employed in most automated based methods we compare to, while [33] also suggested a new split for the CUB dataset. Note that the CUB dataset is considered fine-grained, hence more challenging with both of the input features (visual and semantic) being very noisy. We present the evaluations in Tables 1, 2 and 3 using methods which are either representative or competitive for ZSL using automated attributes including [9, 16, 31, 12, 26, 6], as well as ones that used human annotation [34, 28, 35, 29] for a more general overview. **Implementation details:** For all the splits of AwA and CUB datasets, we fix $k = 15$, $r = 4$, and $k = 8$, $r = 2$ for the k nearest neighbor graph parameter and radius r of the ball around each point, respectively. The edges w_{ij} are chosen using the cosine similarity between the visual observations.

Experimental results on the AwA1 and AwA2 datasets using the new proposed splits [33] are shown in Table 1. Note that the new proposed AWA2 dataset is more challenging, as evident from the significant drop in performance compare to AwA for most of the state of the art methods. We also compare to state of the art methods which are employing human attributes (85 dimensional attribute vectors provided for each class in [19]). A "-" indicates that the performance of the method was not reported in the literature for the corresponding dataset. Mean average precision of the baseline is 58.6%. We improve it to 80.6% and 81.7% on the AwA1 and AwA2 datasets, respectively, by using our regularizer, taking the baseline past the state of the art, which is 80.21% and 78.9% using [9] on AwA1 and AwA2 respectively, reducing the error by 3.55 percentage points on AwA2. Note that among the state of the art methods using automated attributes which is the most competitive, [9] is using a much more computationally heavy method, which requires constructing several graphs and performing a diffusion process for each one of them [9]. Furthermore, for both AwA1 and AwA2 our method outperforms the state of the art methods which are using human attributes.

Experimental results on the CUB dataset is the next benchmark we consider. The baseline achieves a disappointing 23.8% precision on CUB. Surprisingly, our regularizer takes it past the state-of-the-art automatic method (36.9%), to 39.45%, corresponding to an error decrease of over 6.9%. The experimental results comparison on the CUB dataset is shown in Table

Method/Data	CUB
EZSL [27]	23.8
SJE [4]	28.4
LatEm [31]	33.1
Less Is more [25]	29.2
ALE [3]	54.9
Kernel ZSL [34]	57.1
DEM [35]	51.7
RELATION NET [29]	55.6
QFSL [28]	72.1
CAPD [26]	32.08
Multi-Cue Zero-Shot Learning [1]	32.1
DMaP [21]	30.34
MSMR [9]	36.9
Proposed (IPL reduction)	**39.45**

Table 2. Mean average precision accuracy (top-1 in %) results using our method compared to the state of the art methods in zero shot learning on the CUB dataset using Word2Vec or other automated semantic representation. All methods which are using human annotation are marked in red.

2.

Experimental results on generalized ZSL (GZSL) We also compare our performance in the generalized zero shot learning setting. We follow the standard protocol of the generalized ZSL (GZSL) [32] settings where the search space at evaluation time includes both the target and the source classes while the evaluation metric use the harmonic mean between while source and test data as the evaluation metrics. Thus, letting Acc_s, Acc_t the mean class accuracy achieved for the source and target classes, respectively, the harmonic mean H is given by:

$$H = \frac{2 * Acc_s * Acc_t}{Acc_s + Acc_t} \tag{24}$$

The settings of the GZSL is more challenging, as can be seen in the evaluation comparison (Table 3, for most methods the performance degrades significantly in comparison to the standard ZSL. We compare to the recent automated methods tested in the GZSL settings on the AwA1, AwA2 and CUB datasets (those which are available on GZSL and can scale to generalized ZSL settings). The experimental results summarized in Table 3 show not only a significant improvement over method using automatic attributes (error decrease of over 9.8% and 44% for the AWA1 and CUB datasets), but is also outperforming many of the recent state of the art methods which are using human annotation.

Computational complexity: The IPL with spectral reduction has a computational cost of $O(N_u K \log(N_u))$, which includes fast computation of the k nearest neighbor graph using $k-d$ tree, the SGW transform which is $O(N_u)$ for each dimension of the manifold for sparse graphs [14], thus total complexity of $O(N_u K \log(N_u))$ for N_u samples in K dimensional space.

The performance of our method degrades when the compactness values of the geodesic flows lie within a very wide range of different intervals corresponding to the bounded sets of the different input classes. One way to resolve this problem is to design a robust estimator such as M estimator to asses the error in the loss and modify the constraints for the geodesic flows in the optimization problem. In addition, the estimation of the isoperimetric dimension remains challenging for a very noisy data. Since our approximation of the isoperimetric dimension uses a bound which is far from tight, improving the approximation for isoperimetric dimension or suggesting tighter bounds for the isoperimetric inequalities would be beneficial to the IPL approach.

6. Related Work and Discussion

There are many variants for zero-shot learning (ZSL). The general formalism developed, along with the diagram in Fig.2, helps understanding the various approaches in relation to one another.

The problem of zero-shot learning dates back to the early days of visual recognition when the desire to transfer knowledge from painfully learned model to more general classes emerged [20, 5]. Early modern methods consistent with our approach include [19, 23], which can be described by a choice of fixed visual embedding ϕ, semantic embedding s, and an imposed

Method/Data	AwA1	AwA2	CUB
EZSL [27]	12.1	11.0	21.0
SJE [4]	19.6	14.4	33.6
LatEm [31]	13.3	20.0	24.0
ALE [3]	27.5	23.9	34.4
DEVISE [12]	22.4	27.8	32.8
SynC [6]	16.2	18.0	19.8
DMaP [21]	6.44	-	2.07
CAPD [26]	43.70	-	31.6
Kernel ZSL [34]	29.8	30.8	35.1
DEM [35]	47.3	45.1	13.6
RELATION NET [29]	46.7	45.3	47.0
QFSL [28]	-	77.4	73.2
Proposed	**48.0**	**49.2**	**45.6**

Table 3. Comparison results in generalized ZSL on the AwA1, AwA2, and CUB datasets. The harmonic mean is measured using the Mean average precision top-1% accuracy using the unseen and seen classes. Best performance using automated semantic representation is marked in boldface, and methods which are using human semantic annotation are marked in red.

structure of the visual-semantic map (eq. (2) of [23] in our notation)

$$\psi(z) = \sum_{i=1}^{T} z_i s(\hat{y}(z_i))$$

where the sum is truncated at the T largest elements of z, so nothing is learned.

A particularly simple approach, which we adopt as baseline, is [27], who assume that all the maps of interest are linear. In particular, they postulate

$$\psi(z) = Vz.$$

Although the map is linear, the domain and range where it is defined are not linear space (although their embedding space is). We adopt this choice and focus on smoothing the domain and range of the map.

Roughly speaking, zero shot learning methods can be classified into two main categories: inductive and transductive methods. In the inductive settings [36, 2, 18, 4], which has dominated zero shot learning, the unseen classes are introduced one by one and decision about each unseen instance is made instantly once it is introduced. In the transductive setting [21, 16, 13, 6, 9, 28] typically all unseen instances are processed simultaneously by constructing a graph where one exploits the underlying manifold structure, for example using the graph-based label propagation approach [13]. The problem of learning the graph Laplacian or the graph weights directly from given input data has been recently addressed in a number of works [15, 10, 11, 24, 17]. In the most general case, both the graph connectivity and the graph weights are unknown, in which case a common way to enforce smoothness is to use a regularizer which controls some level of sparsity. Perhaps the most widely used criterion is that the energy of the graph Laplacian computed from the graph signal at the vertices be small. Our approach is inspired from the isoperimetric problem, which is a classic problem in geometry. In Euclidean spaces the study of isoperimetric inequalities provides exact measures for general domains, while in Riemannian manifolds they provide some qualitative understanding of the geometry. Isoperimetric inequalities on manifolds were extended to graphs [8, 7], where the analysis shares some similarities and intuition from the continuous settings but is more complex in some sense.

We have introduced the use of isoperimetric inequalities, known for centuries, into clustering in general, and zero-shot learning in particular. We use the isoperimetric loss to indirectly regularize a learned map from visual representations of data to their semantic embedding. Regularization is done by representing the domain of the map as a graph, the map as a graph signal, and regularizing the graph, obtaining another "denoised" graph where clustering is performed to reveal the unseen labels, once cluster-to-label association is performed. This regularization appears to be so effective as to take the simplest possible ZSL approach, where all maps are assumed linear, and improve it to beyond the current state-of-the-art for fully automatic ZSL approaches.

Typical failure modes of our regularization and clustering algorithms are when the compactness values of the geodesic flows lie within a very wide range of different intervals corresponding to the bounded sets of the different input classes.

Since our model is general, it could be used in conjunction with more sophisticated ZSL components, including those where the various maps are not linear, and learned jointly with regularization. ZSL is an its infancy, so the benchmarks are not yet overfitted, and there is plenty of room to improve.

References

[1] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele. Multi-cue zero-shot learning with strong supervision. *CoRR*, 2016.

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[3] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1425–1438, 2016.

[4] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *IEEE Computer Vision and Pattern Recognition*, 2015.

[5] E. Bart and S. Ullman. In *CVPR*, pages 672–679. IEEE Computer Society, 2005.

[6] S. Changpinyo, W. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] F. Chung, A. Grigor'yan, and S. tung Yau. Higher eigenvalues and isoperimetric inequalities on riemannian manifolds and graphs, 1999.

[8] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[9] S. Deutsch, S. Kolouri, K. Kim, Y. Owechko, and S. Soatto. Zero shot learning via multi-scale manifold regularization. *CVPR*, 2017.

[10] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst. Learning laplacian matrix in smooth graph signal representations. *IEEE Trans. Signal Processing*, 64(23):6160–6173, 2016.

[11] H. E. Egilmez, E. Pavez, and A. Ortega. Graph learning from data under laplacian and structural constraints. *J. Sel. Topics Signal Processing*, 11(6):825–841, 2017.

[12] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[13] R. Gopalan, R. Li, and R. Chellappa. Unsupervised adaptation across domain shift by generating intermediate data representations. 36:2288–2302, 2014.

[14] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, pages 129–150, 2011.

[15] V. Kalofolias. How to learn a graph from smooth signals. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 920–929, 2016.

[16] E. Kodirov, T. Xiang, Z.-Y. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *International Conference on Computer Vision(ICCV)*, 2015.

[17] B. M. Lake and J. B. Tenenbaum. Discovering structure by learning sparse graph. In *Proceedings of the 33rd Annual Cognitive Science Conference*, 2010.

[18] C. H. Lampert and a. S. H. Hannes Nickisch. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[19] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between class attribute transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[20] F. Li, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):594–611, 2006.

[21] Y. Li, D. Wang, H. Hu, Y. Lin, and Y. Zhuang. Zero-shot recognition using dual visual-semantic mapping paths. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5207–5215, 2017.

[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, 2013.

[23] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *NIPS*, 2013.

[24] E. Pavez and A. Ortega. Generalized laplacian precision matrix estimation for graph signal processing. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 6350–6354, 2016.

[25] R. Qiao, L. Liu, C. Shen, and A. van den Hengel. Less is more: zero-shot learning from online textual documents with noise suppression. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[26] S. Rahman, S. H. Khan, and F. Porikli. A unified approach for conventional zero-shot, generalized zero-shot and few-shot learning. *IEEE Transactions on Image Processing*, 2018.

[27] B. Romera-Paredes and P. H. Torr. An embarrassingly simple approach to zero-shot learning. *Proceedings of The 32nd International Conference on Machine Learning (ICML)*, 2015.

[28] J. Song, C. Shen, Y. Yang, Y. Liu, and M. Song. Transductive unbiased embedding for zero-shot learning. *CoRR*, abs/1803.11320, 2018.

[29] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[30] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[31] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[32] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning - the good, the bad and the ugly. *CoRR*, abs/1703.04394, 2017.

[33] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning - the good, the bad and the ugly. *CVPR*, 2018.

[34] H. Zhang and P. Koniusz. Zero-shot kernel learning. *CoRR*, abs/1802.01279, 2018.

[35] L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[36] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6034–6042, 2016.