

G2L: A High-Dimensional Geometric Approach for Automatic Generation of Highly Accurate Pseudo-labels

John R. Kender
Columbia University
jrk@cs.columbia.edu

Parijat Dube
IBM Research
pdube@us.ibm.com

Zhengyang Han
New York University
zh2033@nyu.edu

Bishwaranjan Bhattacharjee
IBM Research
bhatta@us.ibm.com

Abstract

Transfer learning is a deep-learning technique that ameliorates the problem of learning when human-annotated labels are expensive and limited. In place of such labels, it uses instead the previously trained weights from a well-chosen source model as the initial weights for the training of a base model for a new target dataset. We demonstrate a novel general technique for automatically creating such source models. We generate pseudo-labels according to an efficient and extensible algorithm that is based on a classical result from the geometry of high dimensions, the Cayley-Menger determinant. This G2L (“geometry to label”) method incrementally builds up pseudo-labels using a greedy computation of hypervolume content. We demonstrate that the method is tunable with respect to expected accuracy, which can be forecast by an information-theoretic measure of dataset similarity (divergence) between source and target. The results of 560 experiments show that this automatic technique generates base models that have similar or better transferability compared to a baseline of models trained on extensively human-annotated ImageNet1K labels, decreasing error in most divergent datasets tested.

1. Introduction

In the field of supervised deep learning, one often ends up with very little labeled training data. To alleviate this problem, a well-known technique used is Transfer Learning [39]. It uses trained weights from a source model as the initial weights for training a target dataset. A well-chosen source with a large amount of labeled data leads to significant improvement in accuracy for the target.

The task we address is the scenario of providing an efficient machine learning service for clients whose data,

particularly image data, varies greatly from the “natural scenes” that typically populate the image databases on which off-the-shelf classifiers are trained. Typically, these clients will present a rather small dataset taken from specialized environments, often industrial, which have been captured under unforeseen circumstances and parameters. Our method creates pseudo-labels for this data by determining their geometric relationships to the feature space of existing labeled data.

In this work, we therefore develop a content-aware labeling technique. First, we take data points such as images, and compute labels for them by calculating distances of these data points from a set of named anchor data points representing known and labeled categories, like *animal*, *plant*, *tool*, etc. Second, pseudo-labels are then constructed for the incoming data points based on these distances, or, more accurately, based on “contents”, which is the high-dimensional generalization of distances, calculated using a geometric approach. Each pseudo-label then consists of a sequence of semantically descriptive names: for example, $\langle tool, plant \rangle$, could be the pseudo-label for data from a previously unseen category like *rake*. Finally, we train a source model using these automatically labeled labels. We show that when an incoming dataset has a high divergence from the data on which a classifier has been trained, this method helps to focus transfer learning on the most relevant aspects of the training data, and the resultant model is more accurate than the original “vanilla” classifier.

Applying this G2L (“geometry to label”) method, we evaluate workloads from the Visual Decathlon [41] and other labeled datasets, comparing how well our pseudo-labeling scheme performs in generating sources for transfer learning against ImageNet1K data [32] using standard human annotated labels. We show that our purely automatic approach wins in many circumstances and that we can specify those circumstances.

Our contributions are: (1) a disciplined extensible algorithm to create semantically interpretable pseudo-labels, derived from methods in the geometry of high dimensions; (2) an analysis of the effects of algorithm hyperparameters on the amount and quality of these pseudo-labels; (3) a characterization of the relationship of source-to-target similarity (divergence) vs. pseudo-label variety, optimal transfer learning rate, and final accuracy; and (4) a discussion and visualizations of 560 experiments on a wide variety of transfer tasks to evaluate the efficacy of G2L.

2. Related Work

Several well-established approaches attempt to assign labels to unlabeled images automatically. Some utilize feature clusters to predict labels [16] or augment image data with linguistic constraints from sources such as WordNet [1, 17]. They address tasks by pretraining models using larger unlabeled data-sets.

Pretraining approaches have also improved results when attempting a target task with a limited amount of accurately labeled training data [24], by using *weakly* labeled data, such as social media hashtags, which are very plentiful. Again, however, effectiveness only appears to grow as the log of the image count. Some methods utilize a teacher model to generate labels to unlabeled data points [37, 18, 38], subsequently guiding the student model to discern underlying class relationships and more nuanced behaviors of the teacher. However, this approach may lead to the student model inheriting biases from the teacher.

Other approaches use generative models such as GANs [31] to explore and refine category boundaries between data clusters, which exploit the rich statistical structures of both real and generated examples, sometimes augmented with labels or linguistic constraints. These automatic approaches use the structures present in large unlabeled datasets to extend the expressivity of known labels and augment the raw size of training sets.

More broadly, various approaches attempt to learn a representation of a class of data and later use that representation in service of a target task. For example, [15] clustered images in an embedding space and developed a meta-learner to find classifications which distinguished various clusters within this embedding. Other approaches to mapping the feature space have used autoencoders [4]. Knowledge distillation techniques [26] teach a student an accurate but more compact feature space, similar in spirit to the work presented here.

Approaches like CLIP [30] trains an image encoder to align images with corresponding textual descriptions in a shared embedding space using a contrastive loss function. However, using natural language descriptions as pseudo labels is challenging as there is no control on the stability and distribution of labels during the training process.

The above existing literature attempts to find hybrid approaches that find productive ways to leverage machine-learned distributions of examples to find new ways of characterizing unlabeled data. The current work presents a novel approach in this domain.

3. Approach

In this section we give an overview of the conceptual flow of our method, discuss the feature space of one application of it to a computer vision task, and suggest a real-world analogy that illustrates some of the semantic considerations behind its central geometry-based algorithm.

3.1. Labeling Method

Generating rich pseudo-labels from models trained on distributionally similar data involves a trade off between an expressive, long label, and a generalizable, short label. Longer labels (obtained by concatenating multiple labels) carry more information about similarity between previous models and the target image, and differences between the previously trained models could be critical for adequately labeling new examples. For example, an incoming set of data including pictures of household objects might be well described by combining the labels of “tool, fabric, furniture.”

However, domains that possess substantial differences from previous data might be better defined by the magnitude and direction of such a difference. For example, a “flower” dataset would share some features with “plant,” but it is perhaps better defined by statements such as “flowers are very unlike furniture”. In other, ambiguous cases, negative features may be necessary to distinguish between overlapping cases: a suit of armor might have similarities with the body shapes of people but could be contrasted with these categories by its dissimilarity with “sport,” a category otherwise close to “person.”

3.2. Labeling Example

We illustrate our method using a specific case study involving images, and with source datasets created by vertically partitioning ImageNet22K [6] along these distinct subtrees: *animal, building, fabric, food, fruit, fungus, furniture, garment, music, nature, person, plant, sport, tool, tree, weapon*, illustrated later in Fig. 2. These subtrees vary in their number of images (from 103K images for *weapon* to 2,783K images for *animal*) and in their number of classes (from 138 for *weapon* to 4,040K for *plant*). These 16 subtrees were used since they were easy to partition from Imagenet22K, but our method could also be used with any other labeled data ontology.

We represent each such source dataset by a single average feature vector. This study generates this vector from the second to last layer of a reference VGG16 [33] model

trained on ImageNet1K, with the average taken over 25% of all the images in the dataset.

To label a new image, we first calculate its own feature vector, then compute its Euclidean distance from each of the representatives of the datasets. Together with other geometric computations in this high dimensional space, these distance measures are then used for a full pseudo-labeling process described in Sec. 4.

3.3. An Analogy

Our G2L approach for pseudo-labeling an image can be understood as being similar to the “Blind Men and the Elephant” parable, where blind men, who have never learned about an elephant, try to categorize an elephant just by touching it, then relating it to something that they already know. Their categorizations of Elephant then include Fan (ear), Rope (tail), Snake (trunk), Spear (tusk), Tree (leg), and Wall (flank). Basically, by touching and feeling an elephant, the blind men are measuring its closeness to things known by them. Our approach also measures the closeness of an unknown image, in feature space, to existing known categories and then generates a pseudo-label for it.

3.4. Analogy Extended

However, additionally, our work extends this analogy in three critical ways. First, we also compare unknown images to the existing categories that are *farthest* from them: the elephant is “not Feather”. Second, we observe a strong predictive relationship between (a) the measurement of the similarity of unknown imagery to existing categories, and (b) the computation of the *number* of pseudo-labels necessary to derive good transfer performance: the elephant could also use “Curtain, Leather, Bark, Mud”. Third, we also observe a strong predictive relationship between measured similarity and optimal *learning rate*: the elephant is most easily described starting from “Manatee”. The significance and computational advantages of these extensions are detailed in Sec. 4 and 6.

4. Geometric Pseudo-labels

In this section we present the main algorithm for generating semantically meaningful geometric pseudo-labels. We first start with five motivating principles and some necessary math background. Then we walk through the algorithm, and discuss some of its properties and results.

4.1. Motivation

Pseudo-labels for a target dataset can be generated by using a large *labeled dataset* organized within a semantic hierarchy such as ImageNet22K, and an off-the-shelf *robust classifier*, such as VGG16 trained on ImageNet1K. (The robust classifier need not be trained on the same labeled

dataset.) Our algorithm exploits these two tools in a way that promotes five desirable properties for pseudo-labels, to ensure that the pseudo-labels have an easily understandable meaning, and that their computation are reasonable efficient.

First, the pseudo-labels should be easily interpretable to humans. As an example, we can start by partitioning ImageNet22K into 16 non-intersecting sets, each of which carries the name of an object category, such as in Sec. 3.

Second, the pseudo-labels should therefore also follow a simple grammar. We refer to the 16 subsets that comprise the above partition as the source subsets. Then, a pseudo-label for an incoming target data item can be defined as the concatenation of some number of source subset names, such as the sequence $\langle person, music, tool \rangle$. This produces an informative natural language description of the incoming target data item.

Third, to make comparisons possible, these pseudo-labels should be geometrically interpretable within the space of feature vectors. Distances between pseudo-labels can be defined by various metrics: L^1 (city-block), L^2 (Euclidean), \sqrt{JS} (the square root of Jensen-Shannon divergence [11]), or others. This can be tricky; the feature vector spaces used in machine learning are difficult to visualize, and such high-dimensional spaces generate geometric paradoxes even at relatively low dimensions [13]. For example, each feature vector of a dataset is very likely to be on the convex hull of that dataset’s representation in that space [40]. Nevertheless, by methods like that of barycentric coordinates [19], particular “anchor” vectors can be used to represent regions within these spaces.

Fourth, the computation of pseudo-labels should leverage known efficient geometric algorithms that localize incoming data. Metrics defined over these spaces can be used to partition the space into cells that form equivalence classes of locations based on individual anchor points (“1st-order Voronoi diagram”). These locations are characterizable by geometric properties such as “the nearest point to this cell is P ”; see Fig. 1(a). These cells can be efficiently determined [9]. Further, these metrics can also partition the space into cells that form equivalence classes of locations based on *sets* of points (“ n^{th} -order Voronoi diagram”), characterizable by geometric properties such as “the n -nearest points to this cell are $\{P_1, P_2, \dots, P_n\}$ ”; see Fig. 1(b). The extreme case for N points is the $(N-1)^{th}$ -order partition (“farthest-point Voronoi diagram”); see Fig. 1(c). However, general farthest-point algorithms are provably hard, and the only efficient algorithms are approximate [29].

Fifth, the concepts of lengths and distances in these spaces should be further generalized to all measures of higher-dimensional “content”, following the progression of polytopes [3], as point, length, area, volume, hypervolume,

etc., to arbitrarily high dimension. Elegant algorithms exist for computing such content, in particular, the Cayley-Menger determinant [35].

Therefore, pulling these observations together, we seek to devise a method that composes a small number of semantically-named anchor vectors derived from the source datasets, into a sequence that defines location descriptions for target data items, based on the generalization of closest and farthest (Voronoi) distances into minimal and maximal (Cayley-Menger) contents. These location descriptions then become the pseudo-labels for the incoming data.

4.2. Mathematical Foundation

Some necessary mathematical preliminaries now follow.

Cayley-Menger determinant. Our method depends on the generalization of the concept of a single distance between a target and a single source, to that of the content of a d -dimensional simplex defined by the target and d sources.

The computation of content is a well-studied algorithm based on the Cayley-Menger determinant (“ CM ”). The determinant itself generalizes several earlier classic algorithms, including the familiar Heron formula for the area of a triangle, and the less familiar Piero formula for computing the volume of a tetrahedron.

Cayley-Menger computation. For an d -simplex, composed of $d+1$ anchors, the math to compute content C_d proceeds in three steps. The derivation of these steps is tedious, and is explained in [2].

First, it forms M_d , a particular symmetric $(d+2) \times (d+2)$ matrix. It incorporates a symmetric submatrix that expresses the *squares* of all pair-wise distances, that is, $D_{i,j} = \text{distance}(i,j)^2$.

$$M_d = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & 0 & D_{0,1} & D_{0,2} & D_{0,3} & \cdots & D_{0,n} \\ 1 & D_{1,0} & 0 & D_{1,2} & D_{1,3} & \cdots & D_{1,n} \\ 1 & D_{2,0} & D_{2,1} & 0 & D_{2,3} & \cdots & D_{2,n} \\ 1 & D_{3,0} & D_{3,1} & D_{3,2} & 0 & \cdots & D_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & D_{n,0} & D_{n,1} & D_{n,2} & D_{n,3} & \cdots & 0 \end{bmatrix}$$

Second, it computes the coefficient $a_d = (-1)^{d+1} 2^d (d!)^2$, which records the effects that various matrix operations have had on the determinant of M_d , during its simplification from more complex geometric volume computations into its above form. This coefficient also defines the integer sequence A055546 at [34], where it has an imaginative interpretation involving roller coasters.

Third, it solves for the value of C_d implicitly expressed by $a_d C_d^2 = \det M_d$.

The complexity of computing the determinant, by the usual and reasonably efficient method of LU decomposition, is $\mathcal{O}(d^3)$. No simpler approach involving the reuse

of previously computed subdeterminants appears feasible, as the determinant has been proven to be irreducible for dimensions greater than 3 [10]. Nevertheless, in the context of our overall machine learning problem, this cost has proven to be negligible with respect to training costs.

4.3. Pseudo-label Creation

Now, we give an overview of the algorithm. An incoming data point is compared at each step against a collection of named category anchor points. The name of the anchor point that minimizes (or maximizes, depending on a policy) its distance to the incoming point is chosen as the first component of an evolving sequence of names. Thereafter, the process repeats, and at each step the sequence is extended with the name of the anchor point that best extremizes the content—the area, volume, hypervolume, etc.—of the evolving polytope formed by these selected points. After a stopping criteria, this sequence gives the pseudo-label.

The full G2L algorithm is summarized in Algorithm 1. The algorithm requires a number of hyperparameters that are set by experiment. An example is shown for each of these choices, in the pseudocode of the precondition (“Require”) preamble. These examples use image classification as the domain, and record the exact configuration that is used in the experiments.

Most of the parameters (and hyperparameters) are straightforward. The indicator *Layer* is the choice of a particular layer within the data representation of f , usually but not necessarily the second-last. The function *Met* is the choice of a distance function that has been derived from an inner product, as required for the derivation of CM . The remaining three hyperparameters are more complex.

Parameters needing explanation. The method *Aggr* is the choice of an aggregation method that represents a set of *Layer* vectors in a sparser form. This can be as trivial as using a single mean vector, or as more elaborate as using a set of representatives derived from clustering methods. For example, as Fig. 2 suggests, the source *food* is probably adequately represented by a single aggregate vector, but the source *fruit* probably is better represented by a pair of aggregate vectors *fruit_{plant}* and *fruit_{food}*.

The integer d_{max} determines the number of dimensions to be explored using CM during the creation of the output pseudo-label name sequences. It also bounds the length of the pseudo-label name sequence pls_i , by $d_{max} \leq |pls_i| \leq 2d_{max}$. The exact length of pls_i , which is constant over a given execution of the complete algorithm, is determined by *Pol*.

Extremizing policies. The extrema decision sequence *Pol*, and its summarizing notation, are best explained by a walkthrough of the algorithm.

At $d=1$, the algorithm considers the length of the line (the 1-simplex) formed from the target data item t_i , and a

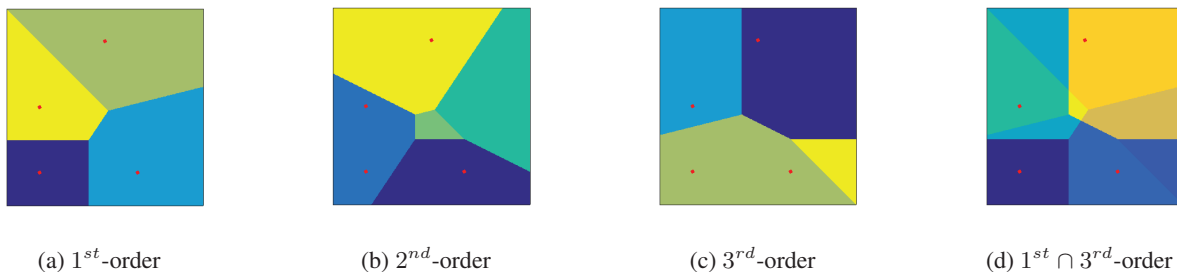


Figure 1: Voronoi tessellation regions in two-dimensions generated by the $N=4$ points shown in red. Colored regions depict equivalence classes of points that share: (a) “closest point” (policy c), (b) “top 2 closest points”, (c) “farthest point” (policy f), (d) “closest and farthest points” (policy C or F), the intersection of (a) and (c). Policies are explained in Sec. 4.3

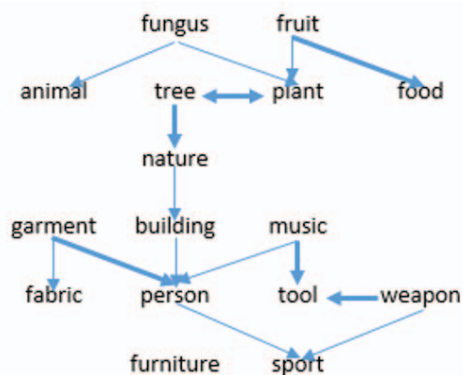


Figure 2: The 16 sources, using 1NN under Euclidean metric to define “closest”. Arrow $a \rightarrow b$ means “ a includes a b subcluster”; thickness is subcluster weight.

representative vector $sour_{j,k}$ from the source representation $Sour_j$. (If $Aggr$ was a simple mean, then each $Sour_j$ will be a singleton set.) Each $sour_{j,k}$ is examined, and the content (here, the length), computed by CM , is recorded in $cont_{i,j,k}$.

Now we can choose the first dimension’s extremizing pseudo-label sequence pls_1 for t_i , from one of four short sequences: (1) the source name of the closest vector, if Pol starts with $\langle c \rangle$, as shown in Fig. 1(a); or (2) the source name of farthest vector, if Pol starts with $\langle f \rangle$, as shown in Fig. 1(c); or (3) the source name of the closest vector followed by the source name of the farthest vector, if Pol starts with $\langle C \rangle$, as shown in Fig. 1(d); or (4) the source name of the farthest vector followed by the source name of the closest vector, if Pol starts with $\langle F \rangle$, as shown in Fig. 1(d) again (the repeat is expected in this case).

For example, if $Pol = \langle c \rangle$, one possible pseudo-label pls_1 for a particular t_i could be the sequence $\langle fruit_{food} \rangle$. Whereas, if $Pol = \langle F \rangle$, it could be $\langle fungus, fruit_{food} \rangle$ instead. The four choices of extremizing policy at any dimension are therefore captured by the quaternary al-

phabet $\{c, f, C, F\}$. And in particular, the policy $\langle C \rangle$ is the special case already explored in prior work [7], which forms pseudo-labels consisting of the names of $\langle closest, farthest \rangle$ pairs.

Proceeding to $d=2$, the algorithm considers the areas, computed by CM , of the triangle (2-simplex) formed by the target data item t_i , a representative vector $sour_{j,k}$, and a single prior extremizing vector, chosen according to the first dimension’s policy. This single vector would be the length-minimizing vector if the policy had been $\langle c \rangle$ or $\langle C \rangle$; or the length-maximizing vector if the policy had been $\langle f \rangle$ or $\langle F \rangle$. At this point, again we can efficiently choose one of four short sequences that capture the names of the area-extremizing sources for this dimension’s pseudo-label, which we then append to the evolving sequence pls_i .

At two dimensions, there are therefore 16 total policies, ranging from $\langle c, c \rangle$ to $\langle F, F \rangle$. These 16 policies create 4 different name sequences of length 2, 8 different name sequences of length 3, and 4 different name sequences of length 4. By establishing and solving straightforward recurrence relations that are similar to those describing Pascal’s triangle, we find that the number of possible name sequences at dimension d with length l is given by $P(d, l) = 2^l \cdot \binom{d}{l-d}$, and that the total possible sequences at dimension d is given by $\sum_l P(d, l) = 4^d$.

The algorithm proceeds likewise for each higher dimension, up to d_{max} , by first building simplices that extend the prior dimension’s simplex, and then selecting names according to this higher dimension’s policy.

4.4. Empirical Properties of Pseudo-labels

In what follows, we have used VGG16 trained on ImageNet1K as classifier f , and we have partitioned the *Full* ImageNet22K dataset into a *Part* collection of the 16 non-intersecting semantic subsets given in Sec. 4.1. We will now refer to policies without angle brackets or commas; $\langle C, f, f, f \rangle$ becomes simply $Cfff$.

Outliers and tractability. We note that a few of our 16 mean vectors, particularly *fungus*, *sport*, and *furniture*,

Algorithm 1 G2L pseudo-label algorithm, in pseudocode

Require: $Tar \leftarrow$ target dataset of data items, $Tar = \cup t_i$
Require: $Full \leftarrow$ semantically-partionable labeled dataset
{e.g. ImageNet22K}
Require: $Part \leftarrow$ partition of $Full$, $Part = \cup \{P_j\}$
Require: $f \leftarrow$ classifier {e.g. VGG16 on Imagenet1K}
Require: $Layer \leftarrow$ feature vector layer {e.g. second-last}
Require: $Aggr \leftarrow$ feature vector aggregator {e.g. mean}
Require: $Met \leftarrow$ feature vector metric {e.g. Euclidean}
Require: $d_{max} \leftarrow$ max simplex dimension {e.g. 4}
Require: $Pol \leftarrow$ extrema decision sequence {e.g. Cfff}
Ensure: pseudo-label sequence pls_i for each t_i

- 1: INITIALIZATION
- 2: **for** each data item $t_i \in Tar$ **do**
- 3: represent t_i by $vert_i \leftarrow Layer$ vector of t_i within f
- 4: **end for**
- 5: **for** each subset $P_j \in Part$ **do**
- 6: represent P_j by set $Sour_j = \cup \{sour_{j,k}\} \leftarrow$ aggregation of $Layer$ vectors of P_j within f , using $Aggr$
- 7: **end for**
- 8: PROCESS
- 9: **for** each $t_i \in Tar$ **do**
- 10: $pls_i \leftarrow \langle \rangle$
- 11: **for** $d=1$ to d_{max} **do**
- 12: **for** each $Sour_j$ **do**
- 13: **for** each $sour_{j,k} \in Sour_j$ **do**
- 14: $X_{i,j,k} \leftarrow$ simplex, using $vert_i, pls_i, sour_{j,k}$
- 15: **for** each vertex pair in $X_{i,j,k}$ **do**
- 16: compute edge distance, using Met
- 17: **end for**
- 18: $cont_{i,j,k} \leftarrow$ content of $X_{i,j,k}$, using CM
- 19: **end for**
- 20: **end for**
- 21: $e_d \leftarrow argextreme_{j,k}$ of $cont_{i,j,k}$, using Pol
- 22: $names_d \leftarrow$ names of $sour_{j,k}$, using $\langle e_d \rangle$
- 23: $pls_i \leftarrow pls_i$ concat $names_d$
- 24: **end for**
- 25: **end for**

repeatedly show up as outlier vertices in the polytopes under construction, as suggested by their relations shown in Fig. 2. They therefore tend to occur early in the output pseudo-label sequences.

We observe empirically, as suggested in [10], that the search for these maximal and minimal simplicies has to be done exhaustively. For example, through exhaustive search on our test dataset, we find that the 1-simplex with minimal content is $\langle plant, tree \rangle$, yet the minimal 2-simplex is $\langle fabric, garment, person \rangle$, and then the minimal 3-simplex is $\langle fabric, garment, plant, tree \rangle$.

Impact of the first policy. We illustrate an important statistical property of the resulting pseudo-labelings in the

heatmap of Fig. 3, which displays the entropy of pseudo-labels generated for $Part$ under the 256 possible policies of order $d=4$.

What is visually apparent is that the variability depends primarily on the initial policy in the sequence. The left half of the diagram shows policies that begin with c or C (first policy decision is “closest”); the right half begins with f or F (first policy decision is “farthest”). The top half shows policies that begin with c or f , which produce a single source name at $d=1$; the bottom half begins with C or F , which produce two source names at $d=1$.

The diagram shows that diversity of pseudo-labels increases roughly from upper left, $cccc$, to lower right, $FFFF$. The general progression is c, f, C, F , in the pattern of $\begin{bmatrix} c & f \\ C & F \end{bmatrix}$. Closer examination shows that the diagram is fractal, and that it follows this pattern at all scales.

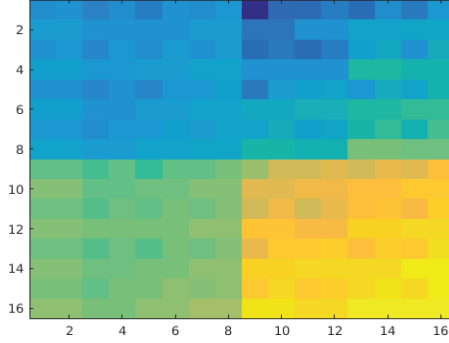
The major anomaly is $fccc$, at position (1, 9), colored strongly dark, which can be interpreted as “apply the 4-simplex consisting of the worst outlier source and its three nearest neighbors”—which for many different inputs is exactly the same, giving minimal entropy. In contrast, the rightmost column of the diagram, which traces policies from $ffff$ at (1, 16) to $FFFF$ at (16, 16), shows a monotonic and nearly linear increase in entropy to the global maximum.

Growth and predictability. The G2L algorithm creates pseudo-label sequences that are combinations of source names. The number of potential sequences of any given length can be very large, for example, $\binom{16}{4}=1,820$, $\binom{16}{6}=8,008$ and $\binom{16}{8}=12,870$. However, when the algorithm was applied to the ImageNet22K training dataset, the number of unique sequences were typically much less, as shown in Fig. 1. Even $FFFF$ of length 8 in the prolific extreme lower right corner produced only 2,644. This slower growth reflects the non-random correlated semantic clustering of the data.

5. Experimental Evaluation

Using our geometric technique, we created a number of pseudo-labeled datasets for the images in ImageNet1K, as shown in Table 1. We then trained both ResNet27 and ResNet50 models using six pseudo-labeled datasets, creating base models for further transfer learning. These six were: $cccc$, $Cfff$, $Ffff$, $FCCC$, $cccccc$, $cfffff$. These six were chosen because they represent a broad spectrum of unique label counts, they explore policies starting with different initial extremizing decisions, and they show the effect of increased dimensions. We observe that the CFA policy introduced in [7] can be obtained from our approach; by our notational convention it would be, simply, policy C .

We also created a baseline model using the vanilla ImageNet1K dataset of images and human-annotated labels.



(a) Entropy of all 256 policies for $d=4$.

```

1 cccc cccf ... cfff fccc fccf ... ffff
2 cccC cccF ... cfff fccC fccF ... ffff
3 ccCc ccCf ... cfFf fcCc fccF ... fffF
  :      :      :      :      :      :
8 cCCC cCCF ... cFFF fCCC fCCF ... fFFF

9 Cccc Cccf ... Cfff Fccc Fccf ... Ffff
10 CccC CccF ... Cfff FccC FccF ... Ffff
11 CcCc CcCf ... CfFf FcCc FcCf ... FffF
  :      :      :      :      :      :
16 CCCC CCCF ... CFFF FCCC FCCF ... FFFF
    1    2      8    9    10    16

```

(b) Policies corresponding to the heatmap at left.

Figure 3: Heatmap showing the entropy of the pseudo-labels generated for $d=4$ policies applied to ImageNet22K data.

Policy	#P-L	Policy	#P-L	Policy	#P-L
cccc*	41	ffff	226	vanilla*	1000
cccccc*	104	Cccc	346	Ffff*	1142
fccc	128	CCff	503	FCff	1391
cfffff*	160	Cfff*	527	FCCC*	1808
C*	202	Fccc	544	FFFF	2644

Table 1: Several base model datasets, sorted by number of pseudo-labels generated under ResNet27. The models with asterisks were used for transfer learning experiments.

This model attained a top-1 average accuracy of 65.6% and 80.2%, which is representative for a ResNet27 model and a ResNet50 model, respectively. The same hyperparameters and training setup was used for all the pseudo-labeling models. We chose these two architectures because residual networks [14] are considered state of the art, and having two networks enabled us to better evaluate the robustness and efficacy of G2L for improving transferability using unlabeled data during pretraining.

To evaluate the usefulness of these base models, we focused on eight target workloads taken from the Visual Domain Decathlon [41] and other fine-grained visual classification tasks. The choice of target datasets was made to have sufficient diversity in terms of number of labels, number of images, number of images per label, and divergence with respect to ImageNet1K. Divergence is here computed by first normalizing the representative vectors of each dataset so that their components (which are all non-negative) sum to 1, then computing Kullback–Leibler divergence [22].

Since we want to compare the performance of pseudo-labeling with respect to vanilla ImageNet1K, we selected only those datasets whose transfer learning accuracy under vanilla were not close to 1. This ensures that the comparison with vanilla is not trivial (otherwise, all policies also have

accuracies very close to 1). The target workloads evaluated included Aircraft [25], CIFAR100 [21], Describable Textures (DTD) [5], Omniglot [23], Street View House Number (SVHN) [27], UCF101 [20], Oxford VGG Flowers [28], and Caltech-UCSD Birds (CUBS) [36]. These span a range of divergence from ImageNet1K, and possess different labels and dataset sizes.

These target workloads were then learned from pseudo-labeled and human-annotated (vanilla ImageNet1K) source models over five different learning rates. The inner layers were set to learning rates ranging over 0.001, 0.005, 0.010, 0.015, and 0.020, and the last layer was set to a learning rate ten times that.

In order to evaluate the performance of G2L policy across frameworks, we implement and train ResNet27 using Caffe1 and ResNet50 using Pytorch. Each model was trained using SGD for 900K iterations, with a step size of 300K iterations, an initial learning rate of 0.01, a momentum of 0.9 and weight decay of 0.1. The target models were trained with identical network architecture but with a training method with one-tenth of iterations (90K) and step size (30K). A fixed random seed was used throughout all training. Thus a total of 560 transfer learning experiments (2 networks, 8 targets, 7 policies, 5 learning rates) with the same set of hyperparameters were conducted and compared.

6. Observations

Overall Accuracy. Table 2 compares the top-1 accuracy of vanilla transfer learning with our pseudo-labeling approach, for both ResNet27 and ResNet50 models. Table 2 shows that the divergence measure closely tracks the accuracy of both vanilla and G2L. For the five datasets whose divergence is above 0.6, G2L beats vanilla four times for ResNet27, and nearly four times for ResNet50. Where G2L underperforms, its performance is similar to vanilla.

Divergence vs. Number of Labels. Both the output

Dataset	Div.	Vanilla ₂₇	G2L ₂₇	Vanilla ₅₀	G2L ₅₀
DTD	0.32	0.4388	0.4282	0.6691	0.6176
CUBS	0.46	0.3485	0.2825	0.7106	0.6344
CIFAR100	0.51	0.7320	0.7065	0.7530	0.7437
UCF101	0.69	0.7500	0.7546	0.7637	0.7643
Oxford	0.70	0.7585	0.7628	0.9518	0.9513
Aircraft	0.71	0.4882	0.4744	0.7870	0.7694
SVHN	0.82	0.9351	0.9385	0.9465	0.9527
Omniglot	0.98	0.7961	0.7975	0.8334	0.8357

Table 2: The 8 target workloads tested, their divergence from ImageNet1K, and their accuracies. Vanilla_{NN} represents the accuracy obtained from ResNetNN under the vanilla policy, as does G2L_{NN} for the G2L policy. Accuracies are the maximum over five learning rates, and for G2L, the maximum over six policies, as described in Sec. 5.

of ResNet27 and ResNet50 demonstrate that, as divergence from Imagenet1K increased for the targets, pseudo-labeling schemes with a lesser number of unique labels performed better; see Fig. 4 which displays the output obtained from the ResNet50 model. In cases where pseudo-labeling schemes did better than the vanilla ImageNet1K labels, the label count was under 250, in contrast to 1,000 with vanilla.

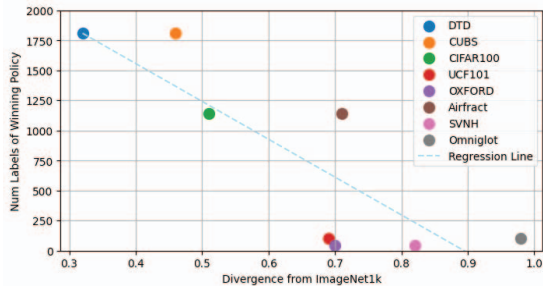


Figure 4: Divergence from ImageNet1K vs. number of labels in winning pseudo-label scheme under ResNet50, showing increasing G2L efficiency as divergence increases.

Divergence vs. Learning Rates. Both the output of ResNet27 and ResNet50 demonstrate that, as divergence from Imagenet1K increased for the targets, higher learning rates were better for the pseudo-labeling schemes; Fig. 5 illustrates the output from ResNet50. In the cases where pseudo-labeling schemes did better than vanilla ImageNet1K, learning rates between 0.015 and 0.02 did best. Even for the vanilla ImageNet1K labels, a higher learning rate was generally better for high divergent workloads.

Empirical observations reveal that the fine-tuning of models using pseudo-labeling schemes can yield comparable outcomes to the vanilla approach, and can exhibit better performance when conducting fine-tuning on datasets that relatively diverge from ImageNet1K. This improvement is

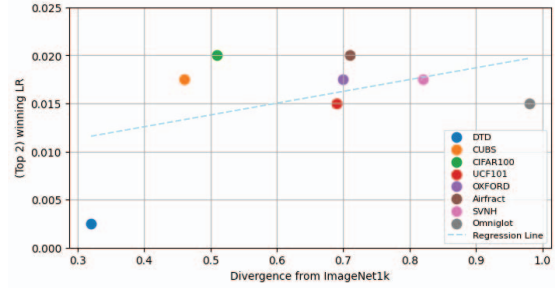


Figure 5: Divergence from ImageNet1K vs. average of the top-2 (sorted by accuracies) learning rates under ResNet50, across 30 fine-tuning configurations (5 learning rates for each of the 6 pseudo-labeling policies) for each dataset.

often accompanied by employing policies with a smaller number of unique labels and higher learning rates. Note that in our experiments the size of pseudo-labeled dataset was same as vanilla. Achieving similar performance as vanilla on transfer learning tasks with G2L substantiates the (human labeling) cost savings achievable with our approach.

7. Limitations and Future Work

Theory Limitations. The initial partitioning of the ImageNet22K dataset into 16 categories is currently heuristic. It known that the use of the second-last vector as a representative is not helpful if incoming data is significantly different on the signal level [12], and that vectors of other layers are. The aggregation method to aggregate the representative vectors of the sources relies on a simple mean. An efficient method for selecting the probable best policy is currently unexplored.

Practice Limitations. How well a given initial partition spans the available high-dimensional feature space has not been quantified. Learning rates for each inner layer in these experiments were identical. Policies were applied according to a fixed script. The value of d_{max} is a completely free meta-parameter.

Future Work. A method for optimizing the initial partition according to some figure of merit would be useful, particularly since human-annotated labels are quite sensitive to fine details such as color, texture, shape, etc. A more appropriate selection of learning rates for different layers of network can significantly improve accuracy during fine-tuning [8]; a thorough exploration, coordinated with knowledge of specific policy strengths and weaknesses, should be attempted. Search techniques other than greedy should be explored, that more intelligently select the best policy to execute next, and that stop without requiring a value for d_{max} . The G2L approach, particularly since it generates descriptive pseudo-labels, can be applied to the problem of data augmentation, and to human label error correction.

References

- [1] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching words and pictures. *J. Mach. Learn. Res.*, 3(null):1107–1135, Mar. 2003. [2](#)
- [2] Leonard Mascot Blumenthal. *Theory and applications of distance geometry*. Chelsea Publishing Company, 1970. [4](#)
- [3] Arne Brøndsted. *An introduction to convex polytopes*, volume 90. Springer Science & Business Media, 2012. [3](#)
- [4] Gabriel B. Cavallari, Leonardo Sampaio Ferraz Ribeiro, and Moacir Antonelli Ponti. Unsupervised representation learning using convolutional and stacked auto-encoders: a domain and cross-domain feature space analysis. *CoRR*, abs/1811.00473, 2018. [2](#)
- [5] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. [7](#)
- [6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on CVPR*, 2009. [2](#)
- [7] Parijat Dube, Bishwaranjan Bhattacharjee, Siyu Huo, Patrick Watson, Brian Belgodere, and John Kender. Automatic labeling of data for transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. [5](#), [6](#)
- [8] Parijat Dube, Bishwaranjan Bhattacharjee, Elisabeth Petit-Bois, and Matthew Hill. Improving transferability of deep neural networks. *CoRR*, abs/1807.11459, 2018. [8](#)
- [9] Rex A Dwyer. Higher-dimensional voronoi diagrams in linear expected time. *Discrete & Computational Geometry*, 6(3):343–367, 1991. [3](#)
- [10] Carlos D’Andrea and Martin Sombra. The cayley-menger determinant is irreducible for $n \geq 3$. *Siberian Mathematical Journal*, 46(1):71–76, 2005. [4](#), [6](#)
- [11] D.M. Endres and J.E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003. [3](#)
- [12] Dario Garcia-Gasulla, Ferran Parés, Armand Vilalta, Jonathan Moreno, Eduard Ayguadé, Jesús Labarta, Ulises Cortés, and Toyotaro Suzumura. On the behavior of convolutional nets for feature extraction. *CoRR*, abs/1703.01127, 2017. [8](#)
- [13] Brian Hayes. An adventure in the n th dimension. In *The Best Writing on Mathematics 2012*, pages 30–42. Princeton University Press, 2012. [3](#)
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on CVPR*, 2016. [7](#)
- [15] Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning. *CoRR*, abs/1810.02334, 2018. [2](#)
- [16] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’03, pages 119–126, New York, NY, USA, 2003. ACM. [2](#)
- [17] Rong Jin, Joyce Y Chai, and Luo Si. Effective automatic image annotation via a coherent language model and active learning. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 892–899, 2004. [2](#)
- [18] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 67–84. Springer International Publishing, 2016. [2](#)
- [19] Usman A Khan, Soumya Kar, and Jose MF Moura. Linear theory for self-localization: Convexity, barycentric coordinates, and cayley–menger determinants. *IEEE Access*, 3:1326–1339, 2015. [3](#)
- [20] S Khurram, R Z Amir, and S Mubarak. Ucf101: A dataset of 101 human action classes from videos in the wild. Technical Report CNS-TR-2010-001, University of Central Florida, 2012. [7](#)
- [21] A Krizhevsky. Learning multiple layers of features from tiny images. In *Technical Report*, 2009. [7](#)
- [22] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951. [7](#)
- [23] B. M Lake, R Salakhutdinov, and J. B Tenenbaum. Human-level concept learning through probabilistic program induction. In *Science*, 2015. [7](#)
- [24] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, and Kaiming He. *Exploring the Limits of Weakly Supervised Pretraining: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, pages 185–201. Springer Science+Business Media, 09 2018. [2](#)
- [25] s Maji, J Kannala, E Rahtu, M Blaschko, and A Vedaldi. Fine-grained visual classification of aircraft. In *Technical Report*, 2013. [7](#)
- [26] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020. [2](#)
- [27] Y Netzer, T Wang, A , Coates, A Bissacco, B , Wu, and A Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. [7](#)
- [28] M-E. Nilsback and A Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006. [7](#)
- [29] Rasmus Pagh, Francesco Silvestri, Johan Sivertsen, and Matthew Skala. Approximate furthest neighbor in high dimensions. In *International Conference on Similarity Search and Applications*, pages 3–14. Springer, 2015. [3](#)
- [30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. [2](#)

- [31] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. [2](#)
- [32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. [1](#)
- [33] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. [2](#)
- [34] Neil JA Sloane et al. The on-line encyclopedia of integer sequences. <http://oeis.org/A055546>, 2003. [4](#)
- [35] Duncan M'Laren Young Sommerville. *An Introduction to the Geometry of n Dimensions*, volume 512. Dover New York, 1958. [4](#)
- [36] 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. [7](#)
- [37] Ismet Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Kumar Mahajan. Billion-scale semi-supervised learning for image classification. *ArXiv*, abs/1905.00546, 2019. [2](#)
- [38] Weiyi Yang, Richong Zhang, Junfan Chen, Lihong Wang, and Jaemin Kim. Prototype-guided pseudo labeling for semi-supervised text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16369–16382, Toronto, Canada, Jul 2023. Association for Computational Linguistics. [2](#)
- [39] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014. [1](#)
- [40] Roozbeh Yousefzadeh. Deep learning generalization and the convex hull of training sets. *arXiv preprint arXiv:2101.09849*, 2021. [3](#)
- [41] L Zitnick, K He, A Torralba, and K Murphy. Visual domain decathlon. In *Detail Workshop Challenge, CVPR*, 2017. [1](#), [7](#)