Supplemental Material for

Criteria Sliders: Learning Continuous Database Criteria via Interactive Ranking

James Tompkin* Brown University, USA
http://www.jamestompkin.com/
Kwang In Kim* University of Bath, UK
k.kim@bath.ac.uk
Hanspeter Pfister Harvard University, USA
https://vcg.seas.harvard.edu/
Christian Theobalt MPI for Informatics, Germany
http://gvv.mpi-inf.mpg.de/
* Equal contribution.

We describe our interactive ranking prototype (Sec. 1), present additional experimental results (Sec. 2) and discussion (Sec. 3), and provide both an introductory explanation to the graph Laplacian as a regularizer and more details of our active label suggestion algorithm (Sec. 4).

1 Interactive Prototype

Our prototype interactive interface allows users to describe criteria by ranking examples, receiving active label suggestions, and viewing and manipulating the recovered sliders (Fig. 1). The database is shown to the south, while the user rank is shown to the north. Initially, the database is shown in a random order. The user can drag suggested items to the rank, or drag any database item from the south up to the rank. The items are not 'removed' from the database on dragging; merely, this is the interaction metaphor for labeling an example. Every time an item is inserted into the rank or whenever the rank is reordered, we update the criteria and order all items in the database display to the south. Efficient learning is key to this interactivity.

Criteria may be defined by ranking in 1D. These can be displayed as a linear rank, or in western page order similar to a Web image search. As our underlying representation is continuous, the criteria can also be displayed as a continuous scale where items are separated by their estimated distances along the scale. For databases which use underlying generative models as their high-dimensional representation, this allows new examples to be generated at specific points along the scale, *e.g.*, at a specific point between two desirable examples (Fig. 2).

Criteria may also be defined in pseudo-2D by ranking with two independent 1D rankings (Fig. 3). The criteria output is then viewed as a 2D scatter plot, either as a matrix-style rank ordering or as a continuous 2D space from the underlying continuous representation. For experts, we can relax the rank interaction convenience and allow criteria to be defined directly by placing examples on the continuous scale, e.g., Figure 3 in the main paper.

Our databases can be large with many thousands of items, and so we allow the user to reduce the number of presented items to show the broader trend. When viewing the database organized by rank, we allow the user to control a subsampling of the database items. When viewing the database organized along continuous scales, we allow the user to zoom out. Users can also jump quickly to a relevant point in the output criteria by right-clicking on the rank, and vice versa.

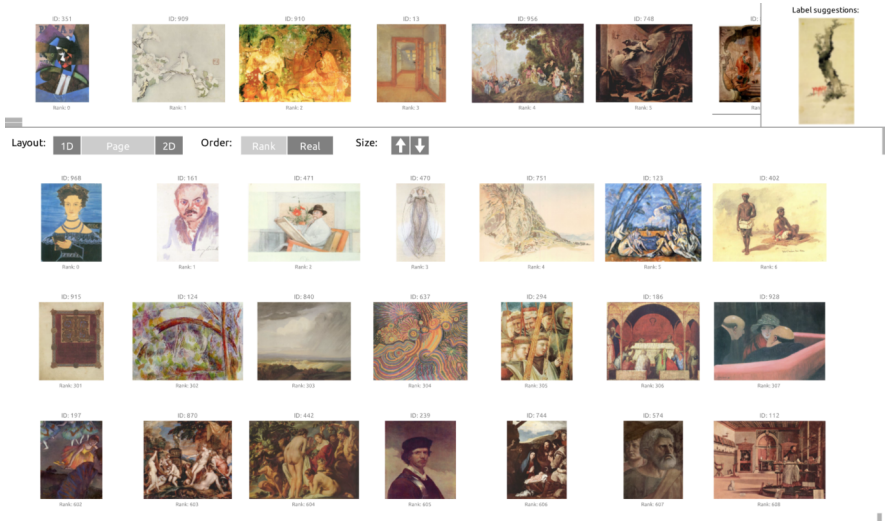

Figure 1: Our prototype interface to define criteria on visual databases, in this case of paintings. The user is attempting to describe a criteria which spans 'abstract to concrete'. *Top:* The user ranking of examples. To the far right is the suggested next item to label. *Bottom:* The database, ordered by the criteria, displayed in western page order. To see the broad trend across all items, the user has decided to show only a subset of the data sampled across all items. We can see that drawings and pictographic representations are at the top left, and more realistic or concrete depictions are at the bottom right.

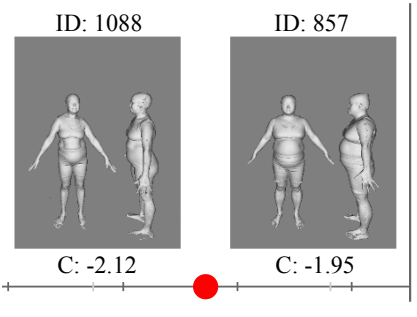

Figure 2: With an underlying high-dimensional generative model, we can create new examples by interpolating the discovered continuous criteria. For example, given the CAESAR database, we can generate a human form with a shape in between two examples on our criteria.

2 Additional Evaluation

1D criteria learning—CAESAR human body geometries. In our supplemental video, we demonstrate different ways of organizing the CAESAR human body database. Here, we show 2226 female body scans ordered by 'height with priority, then girth', which project two orthogonal but related physical attributes down to a single criteria (Fig. 4).

Rank usefulness—further details To try and discover the performance required for useful orderings, we asked 28 users to rate candidate orderings of 50 items against a hypothetical ideal ordering. Users assigned to each candidate rank a 7-point Likert scale score and a binary 'useful/not' classification. Candidate orderings were real results generated by our system, which spanned the performance space between 'random selection' and 'ground truth'. The ideal

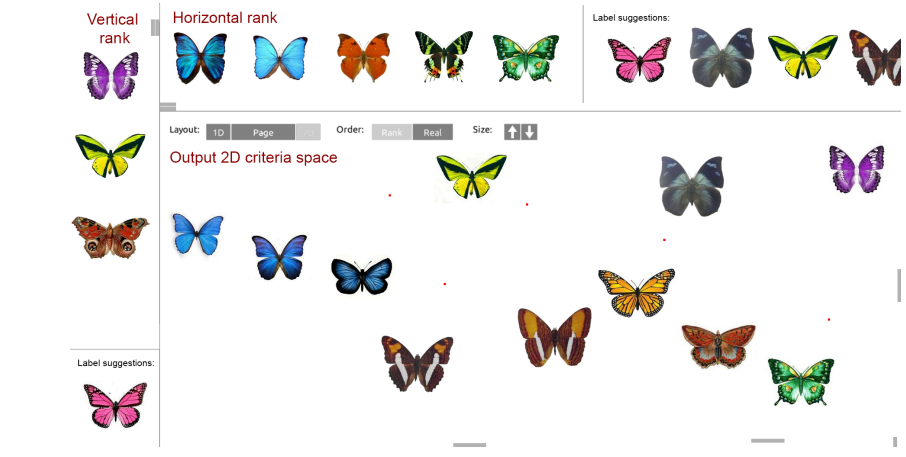

Figure 3: Our interface is used to define an output pseudo-2D criteria space from two independent 1D ranks. The horizontal criteria labels are ranked in the bar at the top left (*Label rank 1*), with label suggestions to the right. The vertical criteria is on the far left (*Label rank 2*). In this example, the user has zoomed in to a subregion of the learned criteria space, where representative data points are visible as images and red dots represent the remaining data points.

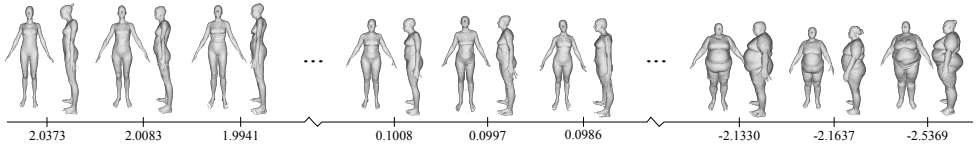

Figure 4: 2226 CAESAR female body scans are ordered by *height with priority, then girth*, expressed with 50 labels. While some local variation remains, we can identify the general trend.

orderings were visible for comparison. First, the experiment was thoroughly explained and users completed a training example. Then, users ranked 24 candidate rankings in a random order, taken in equal portion from the three 'Objects' geometry databases described in the main paper.

Figure 5 compares the Kendall's tau rank correlation coefficient of the different candidate rank orderings with the average Likert score attributed by the participants. We fit a line to compute τ thresholds for 70%, 80%, and 90% acceptance rates. For 70% of participants to be satisfied, $\tau = 0.61$; at this level, 4.3× more pairwise orderings must agree than disagree. For 80% of participants to be satisfied, $\tau = 0.70$, and 5.9× more pairs must agree than disagree. For 90% of participants to be satisfied, $\tau = 0.80$, and 9× more pairs must agree than disagree.

Human variation—further details In the main paper, we organize paintings along the criteria "abstract to concrete". This task has no well-defined answer: while there is a general trend that can be agreed by most people, the specific rank positions of the paintings are open to interpretation. As such, one measure for the level of performance that our system should aim to achieve to be useful is the level of agreement between humans.

To investigate the human variance of ranking paintings, we asked seven users to rank 201 images along the criteria "abstract to concrete". The users were aged 20–40, 2 female and 5 male, with varying artistic experience from regular painters to casual knowledge. During this 'ground truth' generation task, many visual comparisons must be made across the rank. Completing this

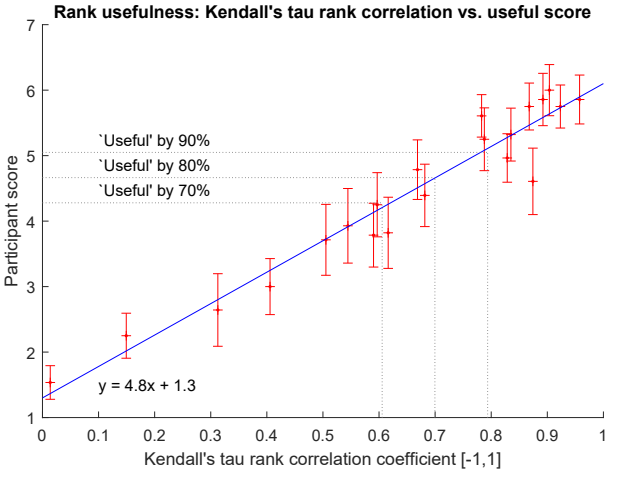

Figure 5: Ordering agreement plotted against participant score. At 90% acceptance, 9× more pair orderings must agree than disagree; at 80%, 5.9× more; at 70%, 4.3× more.

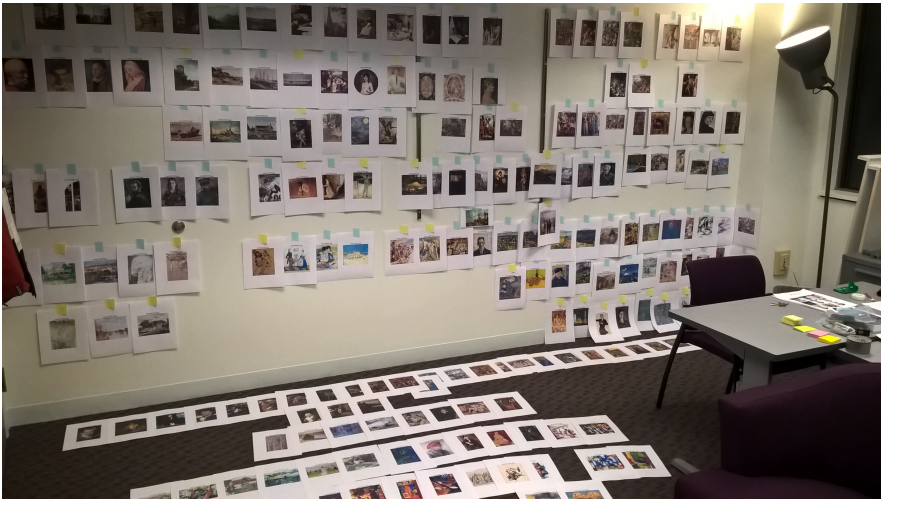

Figure 6: Ranking 201 paintings on paper over a large wall (and floor).

task on a computer is currently difficult because displays are too small to show all images at once with sufficient size for comparison. Thus, we printed each image on paper and asked participants to tape them to a very large wall. The wall acted as the 'rank space', and participants were free to use it as they wished. Most used some combination of bubble, insertion, and merge sorts (Fig. 6).

We compute Kendall's tau rank correlation coefficient between each pair of user rankings, producing 21 coefficients (Tab. 1). The mean coefficient is 0.619, with standard deviation of 0.043, and range 0.530–0.684. There is only moderate agreement between participants. This shows the difficulty of the task and the need for an interactive ranking system which can more easily create personal criteria.

On average, participants spent 158 minutes to produce their rank—ordering a complex database can take time! We initially developed a mouse-based list drag and drop computer interface for this task, with quick buttons to move items to different sections of the 201-item list as a quick approximate pre-sort (e.g., 'Move to start/10%/20%. . . /end'). However, this was slower than using paper and a large physical rank space (Fig. 6).

Par.	1	2	3	4	5	6	7
1	-	0.6296	0.6067	0.5269	0.6094	0.6028	0.6032
2	0.6296	-	0.6806	0.6443	0.6839	0.6667	0.6130
3	0.6067	0.6806	-	0.5760	0.6449	0.6237	0.5941
4	0.5269	0.6443	0.5760	-	0.6444	0.6401	0.5342
5	0.6094	0.6839	0.6449	0.6444	-	0.6586	0.5661
6	0.6028	0.6667	0.6237	0.6401	0.6586	-	0.6407
7	0.6032	0.6130	0.5941	0.5342	0.5661	0.6407	-

Table 1: Kendall's tau rank correlation coefficients computer pairwise between seven participants, each asked to rank 201 paintings by the subjective criteria 'abstract to concrete'. There is mild agreement across all participants, showing both the difficulty of the task and a baseline level of performance required to satisfy the group on average.

Data retrieval precision/recall We perform an experiment to provide an impression of the performance of our approach in the data retrieval setting, rather than in our interactive ranking setting. We use three databases from computer vision: the easier ETH-80 database [12], and the harder C-Pascal [12] and CIFAR-10 [9] databases. We compare both our approach and linear SVM in a one-vs.-all binary classification, then average performance across classes (Fig. 7). We compute precision and recall curves by varying the decision boundary threshold. Further, we compare each method across varying numbers of labels. To build the LG regularizer [8], we set the k nearest neighbors to 20 and the dimensionality of the underlying manifold m to 10.

For ETH-80 with 3,280 data points, we withhold 530 random points as test data ($\approx 1/6$). We follow Ebert et al. [5] and extract HOG descriptors for each image. For C-Pascal, we randomly withhold 700 of the 4,450 data point for testing. Similarly, we extract HOG descriptors. CIFAR-10 has 60,000 data points, 10,000 of which are for testing. For this database, we normalize the input pixel values to the range -1 to 1, then classify directly without a feature transform, and so we should not expect performance to be high. Current state of the art techniques jointly learn the feature representation and classifier, for instance, by employing a convolutional neural network.

Across all three datasets, our approach which makes use of the unlabeled data points is competitive with linear SVM.

3 Discussion

Ranking vs. metric distances. Interactive ranking makes the intuitive assumption that people find ranking easier than specifying explicit or relative distances between many objects on scales, where the difficulty of this task is proportional to the complexity of the criteria. This ranking choice assumes that scale metric distances depend linearly on the ordering. In principle, this does not incur any loss of generality: Once a criteria scale is generated, we can always non-linearly re-metrize it by performing an order-preserving regression on the final result.

Feature power. We focused on designing an interactive semi-supervised learning system suitable for high-dimensional spaces. However, ultimately, the performance of our algorithm also depends on the data features. We use existing feature representations from corresponding related problems (*e.g.*, distance fields for geometric objects); however, in general, the best features depend on the specifics of the database. Here, more sophisticated methods for feature extraction and selection are applicable, *e.g.*, using automatic relevance determination [16].

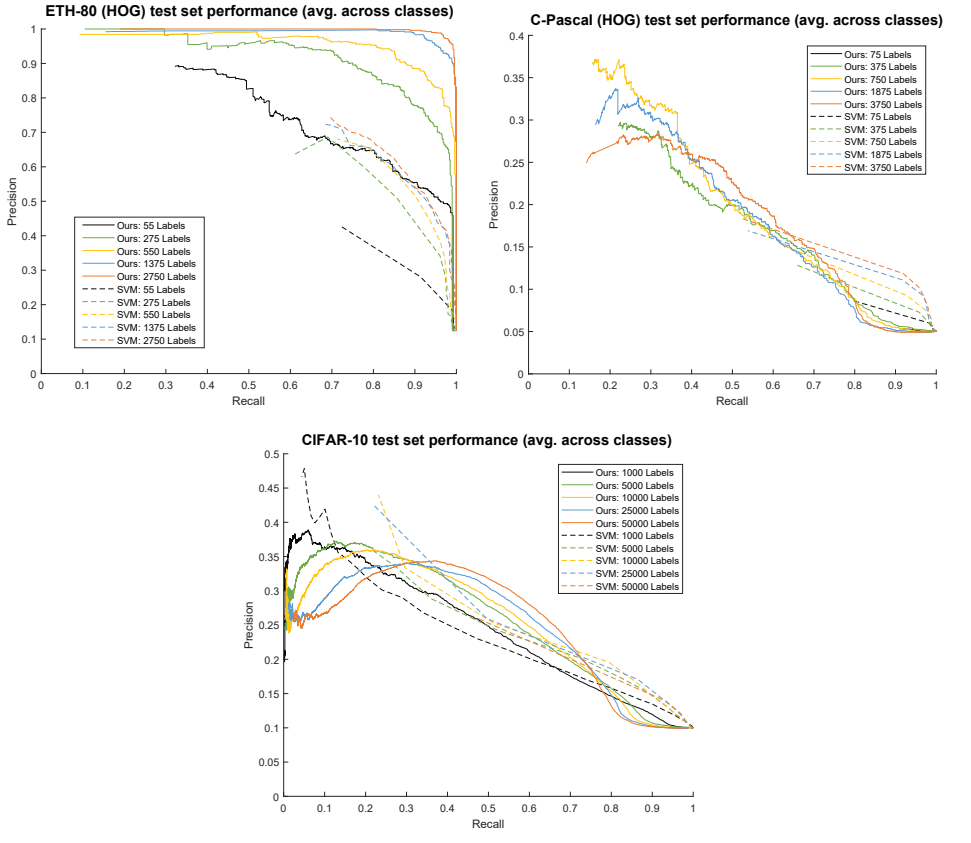

Figure 7: In the data retrieval setting, which compares classification performance instead of rank performance, our approach (full line) is competitive with linear SVM (dotted) across different numbers of labels (colors), so long as the threshold is set appropriately. The reader is recommended to compare like-colored lines to assess performance. All precision/recall curves are mean curve of retrieval over all classes. *Top left:* ETH-80 with HOG features. *Top right:* C-Pascal with HOG features. CIFAR-10 database.

Non-zero-knowledge vs. exploiting existing labels. We intentionally designed our system for the general 'zero-knowledge' case where there are no existing labels, *e.g.*, using criteria definition as queries to assess what is in a Web-scraped database. However, in the wild, some labels may exist. Taking inspiration from Chaudhuri *et al.* [4], who extend a supervised learning framework [15], it may be possible to create an interactive criteria exploration system where crowd-sourced labels 'guide' our regularizer. This may pull the solution towards the 'average' case, which removes absolute subjectivity, but it may also be possible for this guiding to be parameterized and under user control. Further, we only address single session labeling scenarios, but over multiple sessions it may be possible to apply structured labeling techniques to reinforce previous decisions or allow criteria evolution, as per Kulesza *et al.* [10].

4 Interactive ranking and active label suggestion—further details

This section contains 1) an introduction to the graph Laplacian regularizer, 2) motivation for our use of the local Gaussian (LG) regularizer [8] over the standard graph Laplacian regularizer [13, 18, 20] (par. 2, 'semi-supervised learning problem'), 3) a statement of the rigorousness of the Bayesian reformulation in Eq. 8 (par. 2, 'active learning formulation'), include explanations of our comparisons to Amershai et al. [1] and Zhu et al. [20], and 4) more detailed derivations of our large-scale database efficiency extension. Parts of this section are reproduced from the main paper to ease understanding.

Semi-supervised learning problem. For a set of data points $\mathcal{X} = \{X_1,...,X_u\} \subset \mathbb{R}^n$ plus the corresponding labels $\mathcal{Y} = \{Y_1,...,Y_l\} \subset \mathbb{R}$ for the first l points in $\mathcal{X}$ where $l \ll u$, our goal is to infer the labels of the remaining $u-l$ data points in $\mathcal{X}$. Our approach is based on regularized empirical risk minimization [8, 18, 20]:

$$\underset{f:\mathbb{R}^n \to \mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{l} (Y_i - f(X_i))^2 + \lambda \mathcal{R}_H(f), \tag{1}$$

where $\mathcal{R}_H(\cdot)$ is the regularization functional which quantifies how to *smooth* the propagated labels within their local context, and where λ balances between the smoothness of f and the deviation from the labeled examples. Here, we use the standard squared loss function for simplicity. This can be represented in matrix form as an energy functional to be minimized:

$$E(\mathbf{f}) = (\mathbf{f}-\mathbf{y})^\top L (\mathbf{f}-\mathbf{y}) + \lambda \mathbf{f}^\top H \mathbf{f}, \tag{2}$$

where L is a diagonal label indicator matrix and $\mathbf{y}$ is a vector of continuous label values: $L_{[i,i]} = 1$ and $\mathbf{y}_i = Y_i$ if i-th data point is labeled, and $L_{[i,i]} = 0$ and $\mathbf{y}_i = 0$, otherwise.

This problem can be solved in two ways: 1) by reconstructing the underlying function f which maps from $\mathcal{X}$ to $\mathcal{Y}$, which is often called *inductive learning* as it allows f to be applied to new data points not in $\mathcal{X}$; or 2) by identifying the values of $\mathcal{Y}$ for the unlabeled points in $\mathcal{X}$ only, which is often called *transductive learning*.

Datasets as graphs and the Laplacian matrix. One way to solve the transductive learning problem is to represented data points as a graph and use the Laplacian matrix as a regularizer. For the unfamiliar reader, the Laplacian matrix S is a representation of a simple graph $G = \{V,E\}$, where $V = \{v_0,...,v_n\}$ is the set of nodes and $E \subset V \times V$ is the set of edges. It can be built by subtracting the adjacency matrix A of a graph from the degree matrix D of a graph: $S = D - A$. For an unweighted graph, the elements of S are:

$$S_{[i,j]} := \begin{cases} deg(v_i), & \text{if } i = j \\ -1, & \text{if } i \neq j \text{ and if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

In our semi-supervised scenario, instead of an unweighted graph, we declare edge weights W based on the similarity of the input data points. Thus, the graph Laplacian matrix S is one of the best regularizers H for transductive learning because it allows label propagation to occur between data points based on the pair-wise deviations of $\{\mathbf{f}_i\}$ weighted by the similarity of the corresponding inputs [13, 18, 20]. The graph Laplacian matrix also finds applications in spectral clustering and embedding [13], ranking [19], and inducing diffusion processes on graphs [7].

Graph Laplacian as a regularizer. Given Equation 1, we can formulate the graph Laplacian S [13] as a regularizer $\mathcal{R}_S(f)$:

$$\mathcal{R}_H(f) := \mathbf{f}^\top S\,\mathbf{f} = \sum_{i,j=1}^{u} W_{[i,j]}(f_i - f_j)^2, \tag{4}$$

where $f_i = f(X_i)$, $\mathbf{f} := f|_{\mathcal{X}} = [f_1,...,f_u]^\top$, and W is a non-negative input similarity matrix which is typically defined based on a Gaussian:

$$W_{[i,j]} = \exp\left(-\frac{\|X_i - X_j\|^2}{b}\right). \tag{5}$$

with b being the width of the Gaussian kernel, and

$$S = D - W \tag{6}$$

where D is a diagonal matrix of column sums of W ($D_{[i,i]} = \sum_j W_{[i,j]}$).

One way of justifying the use of the graph Laplacian comes from its limit case behavior as $u \to \infty$ and $b \to 0$: When the data $\mathcal{X}$ is generated from an underlying manifold M with dimension $m \leq n$, i.e., the corresponding probability distribution P has support in M, the graph Laplacian converges to the Laplace-Beltrami operator Δ on M [2, 6]. The Laplace-Beltrami operator can be used to measure the first-order variations of a continuously differentiable function f on M:

$$\|f\|_\Delta^2 := \int_M f(X)[\Delta f|_X]dV(x) = \int_M \|\nabla f|_X\|_g^2 dV(X), \tag{7}$$

where g is the *Riemannian metric*, and dV is the corresponding *natural volume element* [11] of M. The second equality is the result of Stokes' theorem. Accordingly, a graph Laplacian-based regularizer $\mathcal{R}_L$ can be regarded as an empirical estimate of the first-order variation of f on M based on $\mathcal{X}$.

With predominant use of the graph Laplacian regularizer, semi-supervised learning has become established in classification and clustering tasks where the output is a *discrete* set of labels. However, estimating *continuous* outputs is still an area of active research due to degenerate behavior of the graph Laplacian regularizer: in high-dimensional spaces, minimizing E with $H = S$ tends to produce an output $\mathbf{f}^*$ where most elements are close to zero [14]. This is fine for classification as output magnitudes are irrelevant, *e.g.*, to classify data points into *positive* and *negative* classes, 1 and 1^{-10} are both positive; however, having close to zero outputs is meaningless for continuous outputs, as very little regularization occurs.

This problem has only recently been addressed with the *local Gaussian* (LG) regularizer [8], which explicitly addresses this degeneracy and reduces the computational complexity required to remove it. However, since the LG regularizer was developed for general batch semi-supervised learning, i.e., all labeled data points are provided before the learning process, it cannot be directly applied to interactive ranking. Hence, we adapt the LG regularizer to our setting.

Active learning formulation. Following the analogy between regularized empirical risk minimization and the maximum a posteriori (MAP) estimation [16] and adopting the Bayesian optimization [17], we reformulate $-1 \times E$ (Eq. 2) as (the logarithm of) a product of the prior $p(\mathbf{f})$ and a Gaussian noise model $p(\mathbf{y}|\mathbf{f})$. This leads us to assess the minimizer of E as the mean of the predictive distribution (the *posterior*):

$$-\log p(\mathbf{f}|\mathbf{y}) = (\mathbf{m} - \mathbf{f})^\top C^{-1}(\mathbf{m} - \mathbf{f}) + Z, \tag{8}$$

with mean $\mathbf{m} = CL\mathbf{y}$, covariance matrix $C = (L + \lambda H)^{-1}$, the normalization constant Z, and the local Gaussian regularization matrix H. This perspective informs a *predictive uncertainty* for each data point: The i-th diagonal component C_{ii} of the covariance matrix C contains informa-

tion on the uncertainty of the prediction on label f^i, which is typically low when X_i is labeled and is high otherwise. Active label suggestion presents results which minimize uncertainty.

Note that this Bayesian reformulation is mathematically rigorous: By the construction of H [8], C^{-1} is positive definite and therefore, $\mathbf{m}$ and C are valid mean vector and covariance matrix. This shapes the posterior $p(\mathbf{f}|\mathbf{y})$ as a sample from an underlying Gaussian process that factorizes to a prior on $\mathbf{f}$ and the corresponding likelihood $p(\mathbf{f}|\mathbf{y})$. This re-interpretation enables us to exploit well-established Bayesian uncertainty modeling. In particular, the variance of the predicted distribution $p(\mathbf{f}|\mathbf{y})$ indicates how *uncertain* we are about each prediction we make on data points $X \in \mathcal{X}$.

One simple and well-established strategy to exploit these modeled uncertainties for active label selection is to predict at each iteration t and choose the point X_i which has the largest uncertainty. However, Figure 8 shows that naively choosing data points with maximum uncertainty leads to poor performance with a higher error rate than random selection, as isolated outlier data points—which are not broadly informative—receive high variances and are chosen.

Instead, we construct the candidate data points that minimizes the predictive variance over the *entire* set of data points. At each time step t, we choose data points with the highest average *information gain* $\mathcal{I}$, defined as:

$$\mathcal{I}(X_i) = \sum_{j=1,\ldots,u} [C(t-1) - C(t)^i]_{[j,j]}. \tag{9}$$

The matrix $C(t)^i$ is constructed by adding 1 to the i-th diagonal element of $C(t-1)^{-1}$ and inverting it (i.e., i-th data point is regarded as labeled; see Eq. 2).

As suggested by the form of the matrix C, this score can be calculated without knowing the corresponding label value Y_i. This is due to the Gaussian noise model; in general, different noise models lead to the predictive variance (C) depending on the labels.

Naïvely estimating the information gain for all data points requires quadratic computational complexity: One has to estimate the minimizer of $E(\mathbf{f})$ (Eq. 2), which is $O(u^3)$ for each data point. However, in our iterative label suggestion scenario, $\mathcal{I}$ can be efficiently computed in linear time: Assuming that $C(t-1)$ is given from the previous step, calculating $\mathrm{diag}[C(t)^i]$ does not require inverting the matrix $L(t)+\lambda H$: Using *Sherman–Morrison–Woodbury* formula, $C(t)^i$ can be efficiently calculated from $C(t-1)$:

$$\mathrm{diag}[C(t)^i] = \mathrm{diag}[C(t-1)] - \frac{\mathrm{squ}[C(t-1)_{[:,i]}]}{1+\mathrm{diag}[C(t-1)]_i}, \tag{10}$$

where $A_{[:,j]}$ denotes a vector formed from the j-th column of the matrix A, $\mathrm{diag}[A]$ constructs a vector from the diagonal components of matrix A, and $\mathrm{squ}[B]$ is a vector obtained by taking element-wise squares of the vector B. Accordingly, after explicitly calculating $C(0)$,[1] subsequent updates in $C(t)$ and $C(t)^i$ can be performed efficiently for each iteration t.

Amershai *et al*. comparison. One of the first algorithms to propose active learning in end-user interaction application is Amershi *et al*.'s approach [1]. Their algorithm was originally developed for active labeling in metric learning contexts, and so it is not directly applicable to our problem. However, the re-interpretation of their label selection criterion enables transferring their algorithm to our current setting: Their strategy is based on a Gaussian process (GP) model with Gaussian noise: For a given set of labeled data points S and the unlabeled data points U, and for the i-th unlabeled point, the corresponding score $f(i)$ is defined as:

$$f = \frac{Var(i|S)}{Var(i|U)}, \tag{11}$$

[1] In practice, we limit $C(0)$ to 2,000 labels for faster interactivity.

where $Var(i|S)$ is the variance predicted for i-th data point based on the GP model trained based on S, and $Var(i|U)$ is the variance predicted based on the GP model trained on U, assuming that U are all labeled. Details of GP prediction can be found in [16]. This interpretation enables us to apply the same principle to our setting: for the i-th data point, the corresponding score $\mathcal{I}'(X_i)$ is:

$$\mathcal{I}'(X_i) = \frac{\mathrm{diag}[C(t-1)]_i}{\mathrm{diag}[U(t-1)]_i}, \tag{12}$$

where $U(t-1)$ is the variance predicted based on all unlabeled data points. Maximizing only the predictive uncertainty $\mathrm{diag}[C(t-1)]_i$ corresponds to active label selection based on the predictive uncertainty, which is likely to choose an outlier. The (reciprocal of the) denominator represents how well the point X_i is *connected* to the unlabeled data points [1]. Accordingly, normalizing the uncertainty by this values rules out the possibility of choosing the outlier. However, $\mathcal{I}'(X_i)$ is based on how much information the unlabeled data points contains about X_i, which does not directly represent our final goal: our goal is to choose a data point minimizes the *overall* uncertainty on every data points which is directly addressed by $\mathcal{I}$.

Zhu *et al*. comparison. Zhu *et al*. [20] proposed an active learning framework that selects the labels by minimizing an estimate of the expected risk. This framework relies on evaluating the prediction accuracy of the updated system for each hypothesized output values. Therefore, the application of this approach is limited to the case in which the output space is finite, *e.g*., $\{-1,1\}$ in the classification problem. Our approach does not have this limit.

Large-scale extension. The rank-one update model (Eqs. 9 and 10) relies on explicitly calculating the covariance matrix $C = (L+\lambda H)^{-1}$. Even though the original regularization matrix $L+\lambda H$ is sparse, its inverse C is in general dense. Therefore, this strategy is not directly applicable for large-scale problems where calculating and storing the inverse of $L+\lambda H$ are infeasible. For this case, we adopt sparse eigen-decomposition-based approximation of $L+\lambda H$.

Assume that $L+\lambda H$ is eigen-decomposed:

$$C^{-1} = L+\lambda H = E^F \Lambda^F {E^F}^\top$$
$$\Leftrightarrow C = E^F {\Lambda^F}^{-1} {E^F}^\top = \sum_{i=1}^{u} \frac{1}{\lambda_i} \mathbf{e}_i \mathbf{e}_i^\top,$$

where $u \times u$-sized matrix E^F stores eigenvectors ($\{\mathbf{e}_i\}_{i=1}^u$) column-wise, and Λ^F is a diagonal matrix of the corresponding eigenvalues ($\{\lambda_i\}_{i=1}^u$). We assume that the eigenvalues and the corresponding eigenvectors are arranged in increasing eigenvalues. Given the prescribed rank parameter r, our approximate covariance matrix $\overline{C}$ is given as the best possible L^2-approximation of rank r to C:

$$C \approx \overline{C} = \sum_{i=1}^{r} \frac{1}{\lambda_i} \mathbf{e}_i \mathbf{e}_i^\top = E \Lambda^{-1} E^\top, \tag{13}$$

where E is a $u \times r$ submatrix of E^F containing the first r eigenvectors and Λ is the corresponding $r \times r$ submatrix of Λ^F.

Now suppose that we add a label of the i-th data point at time 1. Applying the Sherman–

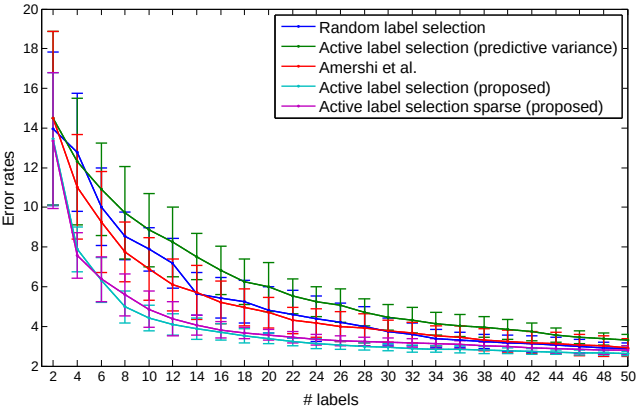

Figure 8: *Reproduced from the main paper for reading ease.* Mean absolute error for estimating the ground-truth *weight* parameter in the CAESAR database. Error bar lengths correspond to twice the standard deviations (std.). Note the smaller stds. of the proposed full and sparse methods. The error rates for two labeled data points cases are the same by construction.

Morrison–Woodbury formula to $\overline{C}$, we can represent the update equation:

$$C(1)^i = E\Lambda^{-1}E^\top - \frac{E\Lambda^{-1}E_{[i,:]}^\top E_{[i,:]}\Lambda^{-1}(t-1)E^\top}{1+H_{ii}^{-1}}$$
$$= E\left(\Lambda^{-1} - \frac{\Lambda^{-1}E_{[i,:]}^\top E_{[i,:]}\Lambda^{-1}}{1+H_{ii}^{-1}}\right)E^\top.$$

The corresponding information gain is given as

$$\mathrm{diag}[C-C^i]_k = E_{[k,:]}\left(\frac{\Lambda^{-1}E_{[i,:]}^\top E_{[i,:]}\Lambda^{-1}}{1+H_{ii}^{-1}}\right)E_{[k,:]}^\top$$
$$= \left(\frac{1}{1+H_{ii}^{-1}}\right)\left(E_{[k,:]}\Lambda^{-1}E_{[i,:]}^\top E_{[i,:]}\Lambda^{-1}E_{[k,:]}^\top\right). \quad (14)$$

At step t, adding a label to the $i(t)$-th data point leads the a new covariance estimate:

$$[EF_iE^\top]_{k,k} = E_{[k,:]}\Lambda^{-1}E_{[k,:]}^\top + \sum_{i=1,\ldots,t} E_{[k,:]}\mathbf{r}_i\mathbf{r}_i^\top E_{[k,:]}^\top,$$

where

$$\mathbf{r}_i = \left(\frac{1}{1+\overline{C}_{ii}^{-1}}\right)E_{[i,:]}\Lambda^{-1}. \quad (15)$$

Active label selection performance. For CAESAR, we randomly selected a set $\mathcal{A}$ of 2000 data points and a subset $\mathcal{B}\subset\mathcal{A}$ of 2 initial data points to be labeled (our user scenario). First, we train on $\mathcal{B}$. Then, we calculate the information gain $\mathcal{I}$ (Eq. 9) for each data point in $\mathcal{A}\backslash\mathcal{B}$. The best 2 data points were assigned labels and were added to $\mathcal{B}$. We iterated until $|\mathcal{B}|=46$. Figure 8 compares error rates of our sparse eigen-decomposition-based approximation (*Active label selection sparse (proposed)*) to the alternative algorithms (averaged over 10 trials): 1) random label selection, 2) predictive uncertainty as discussed previously, 3) Amershi *et al.* [1] adapted to our semi-supervised setting, 4) the full information-gain-based method (Eqs. 9 and 10; *Active label selection (proposed)*). We used only the first 100 ($r=100$) eigenvectors. Predictive variance only resulted in suggesting outliers which led to worse results than random selection. The adapted algorithm of Amershi *et al.* [1] improves upon random selection, while our two information gain-based approaches show further improvement, especially when the number of labels is low.

References

[1] S. Amershi, J. Fogarty, A. Kapoor, and D. S. Tan. Effective end-user interaction with machine learning. In *Proc. AAAI*, 2011.

[2] Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2005.

[3] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.

[4] S. Chaudhuri, E. Kalogerakis, S. Giguere, and T. Funkhouser. AttribIt: content creation with semantic attributes. In *Proc. UIST*, pages 193–202, 2013.

[5] Sandra Ebert, Diane Larlus, and Bernt Schiele. Extracting structures in image collections for object recognition. In *Proc. ECCV*, pages 720–733, 2010.

[6] M. Hein, J.-Y. Audibert, and U. von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph Laplacians. In *Proc. COLT*, pages 470–485, 2005.

[7] K. I. Kim, J. Tompkin, H. Pfister, and C. Theobalt. Context-guided diffusion for label propagation on graphs. In *Proc. ICCV*, pages 2776–2784, 2015.

[8] K. I. Kim, J. Tompkin, H. Pfister, and C. Theobalt. Local high-order regularization on data manifolds. In *Proc. IEEE CVPR*, pages 5473–5481, 2015.

[9] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[10] T. Kulesza, S. Amershi, R. Caruana, D. Fisher, and D. Charles. Structured labeling for facilitating concept evolution in machine learning. In *Proc. CHI*, pages 3075–3084, 2014.

[11] John M. Lee. *Riemannian Manifolds- An Introduction to Curvature*. Springer, New York, 1997.

[12] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–409–15 vol.2, June 2003. doi: 10.1109/CVPR.2003.1211497.

[13] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[14] B. Nadler, N. Srebro, and X. Zhou. Statistical analysis of semi-supervised learning: the limit of infinite unlabelled data. In *NIPS*, pages 1330–1338, 2009.

[15] D. Parikh and K. Grauman. Relative attributes. In *Proc. ICCV*, pages 503–510, 2011.

[16] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[17] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *NIPS*, pages 2951–2959, 2012.

[18] D. Zhou, O Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, pages 1330–328, 2004.

[19] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS*, pages 169–176, 2004.

[20] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In *Proc. ICML workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65, 2003.