UCLA
UCLA Previously Published Works

Title

PDEs on graphs for semi-supervised learning applied to first-person activity recognition in body-worn video

Permalink

https://escholarship.org/uc/item/8vt5f3vg

Journal

Discrete and Continuous Dynamical Systems, 41(9)

ISSN

1078-0947

Authors

Li, Hao
Chen, Honglin
Haberland, Matt
et al.

Publication Date

2021

DOI

10.3934/dcds.2021039

Peer reviewed

PDES ON GRAPHS FOR SEMI-SUPERVISED LEARNING
APPLIED TO FIRST-PERSON ACTIVITY RECOGNITION
IN BODY-WORN VIDEO

Hao Li and Honglin Chen

University of California, Los Angeles, Department of Mathematics
520 Portola Plaza, Box 951555, Los Angeles, CA 90095-1555, USA

Matt Haberland

California Polytechnic State University, BioResource and Agricultural Engineering Department
BRAE 8-101, 1 Grand Ave, San Luis Obispo, CA 93407,USA

Andrea L. Bertozzi*

University of California, Los Angeles, Department of Mathematics
520 Portola Plaza, Box 951555, Los Angeles, CA 90095-1555, USA
University of California, Los Angeles
Department of Mathematics and Mechanical and Aerospace Engineering
420 Westwood Plaza, Box 951555, Los Angeles, CA 90095-1555, USA

P. Jeffrey Brantingham

University of California, Los Angeles, Department of Anthropology
375 Portola Plaza, 341 Haines Hall, Box 951553, Los Angeles, CA 90095-1555, USA

(Communicated by José A. Carrillo)

Abstract. This paper showcases the use of PDE-based graph methods for
modern machine learning applications. We consider a case study of body-worn
video classification because of the large volume of data and the lack of training
data due to sensitivity of the information. Many modern artificial intelligence
methods are turning to deep learning which typically requires a lot of training
data to be effective. They can also suffer from issues of trust because the heavy
use of training data can inadvertently provide information about details of the
training images and could compromise privacy. Our alternate approach is a
physics-based machine learning that uses classical ideas like optical flow for
video analysis paired with linear mixture models such as non-negative matrix
factorization along with PDE-based graph classification methods that parallel
geometric equations from PDE such as motion by mean curvature. The up-
shot is a methodology that can work well on video with modest amounts of
training data and that can also be used to compress the information about
the video scene so that no personal information is contained in the compressed
data, making it possible to provide a larger group of people access to these
compressed data without compromising privacy. The compressed data retains
information about the wearer of the camera while discarding information about
people, objects, and places in the scene.

1. **Introduction.** There is a long tradition of using nonlinear PDEs in image processing starting with the Total Variation minimization for image denoising [38] and energy minimization methods for image segmentation, e.g. [8]. More recently, ideas from these energy minimization methods have been used for graphical models for classifying high dimensional data. Here we discuss a particular application of those ideas to the problem of ego-activity recognition in body-worn video. This particular application takes advantage of both PDE-based image processing of the video using classical optical flow techniques and discrete graph clustering of the video frames according to their ego-activity.

With the development of body-worn camera technology, it is now possible and convenient to record continuously for a long period of time, enabling video capture of entire days. Classifying ego-activities in body-worn video footage is well-studied in the context of sports videos [21] and life-log videos [41, 35, 37, 13]. Most ego-activity recognition methods require a substantial amount of training footage; supervision is either used to determine the importance of extracted low-level features in a bottom-up system (e.g. [36]) or used to learn features in a deep-learning approach (e.g. [37]). We consider a semi-supervised approach, in which we utilize a much smaller amount of labeled training data than a typical supervised method.

The proposed semi-supervised approach is based on similarity graphs. It first quantifies the similarities between pairs of data points, i.e. short pieces of video, according to handcrafted, motion-based features adapted from [39]. Then, it spreads the label information from a small set of manually labeled fidelity points to unlabeled data. This process is inspired by three interrelated dynamical processes on graphs: the Allen-Cahn equation, the Merriman-Bence-Osher scheme, and mean curvature flow [4]. These processes have been shown to be descending flows for the Total Variation semi-norm, which is the objective function of graph cuts [45]. We propose the use of handcrafted features instead of deep-learning features so that we can ensure that the features do not compromise personally identifiable information in the scene. We employ features based exclusively on motion cues without object detection and tracking. With the aid of the Nyström extension, the graph-based semi-supervised classification method is scalable to handle the enormous size of body-worn video data sets. We benchmark the proposed method on two publicly available data sets involving routine activities and demonstrate its comparable performance to supervised methods that use significantly more training data. We also illustrate the performance of the method on a set of body worn camera videos from the Los Angeles Police Department, demonstrating the method's effectiveness over a prior method applied to some of that data. This is an academic study and the method is not intended for field use by law enforcement. However it does provide an alternative approach to deep learning methods that rely heavily training data. The method also focuses exclusively on the actions of the wearer of the camera, and would be more useful for for sorting information related police activity and accountability rather than on crime. The paper is organized as follows. In section 2 we survey related work on analyzing egocentric vision and activity recognition. In section 3, we introduce our feature extraction and semi-supervised learning method. We report our experimental setup and results in section 4. Finally, the conclusions and future work follow in section 5.

2. **Related work.** Research in summarizing and segmenting egocentric videos recorded by body-worn cameras dates back to the early 2000s [2]. Since then,

this has been an active research area due to the advancements in computer vision and machine learning [9]; here we review work most relevant to our own.

The task of activity recognition in body-worn video can be categorized into three lines of research: (1) one relies on object-hand interactions and video content (i.e. what objects and people are in the video), (2) one uses the motion of the camera, and (3) one uses a combination of the previous two. Typically, neither object-hand interactions nor the motion of the camera is directly available as metadata in egocentric vision data sets, so all three lines of research start with inferring respective pieces of information from raw video footage.

Works following the first approach rely on object detection and tracking to classify the camera-wearer's activities, for instance, [12, 13, 35, 24, 42]. Popular benchmark data sets used to validate methods focusing on hand-object interactions are the GTEA and GTEA Gaze+ data sets, provided by [13], and ADL-short in [41] and ADL-long in [35]. The GTEA and GTEA Gaze+ data sets are recorded by Tobii eye-tracking glasses when wearers are cooking in a natural setting, so these two data sets contain eye-gaze direction information not typically available in other body-worn video data sets. Both ADL data sets are recorded with a chest-mounted camera when the wearers are performing various daily tasks indoors. The aforementioned data sets are different from our police body-worn video data set, which was recorded outdoors and usually did not capture police officers' hand movement, so we do not pursue this thread of research.

The second line of research is to recognize activities based on motion analysis. A wide variety of motion features have been proposed in the literature. [21] uses a histogram-based motion feature to classify sports activities in videos recorded by head-mounted GoPro cameras. [39] proposes a motion descriptor that inspired our feature selection method. [30] uses inferred camera movement signals and their dominant frequencies. Many ways of incorporating temporal information in motion analysis are proposed; for instance, [40] proposes to apply multiple temporal pooling operators to any per-frame motion descriptor. Deep convolutional neural networks are also used to extract motion features; for instance, [1] learns a motion representation by using 2D convolution neural network on stacked spectrograms and a Long Short-Term Memory (LSTM) network. With multiple available features extracted, [34] proposes a multiple kernel learning method to combine local and global motion features. A benchmark data set for this line of research is the HUJI EgoSeg data set provided by [36], which is recorded when the wearer is performing a variety of activities in both indoor and outdoor settings. As in the HUJI EgoSeg data set, we observe that many activities of interest in our police body-worn video data set induce distinctive camera movement patterns, and so we focus on a motion-based approach. The proposed approach differs from the aforementioned methods in that it is semi-supervised; we demonstrate in section 4.3 that it achieves comparable performance to the supervised methods on the HUJI EgoSeg data set using a fraction of training data.

For the third line of research, methods that utilize both appearance (i.e. object recognition and tracking) and motion cues are often combined with deep learning. Both [37] and [27] use a two-stream deep convolution neural network, one stream for images and another stream for optical flow fields, to discover long-term activities in body-worn video. Both [6] and [46] use an auto-encoder network to extract motion and appearance features in an unsupervised fashion. We note that features extracted from appearance cues using a convolutional neural network may be used
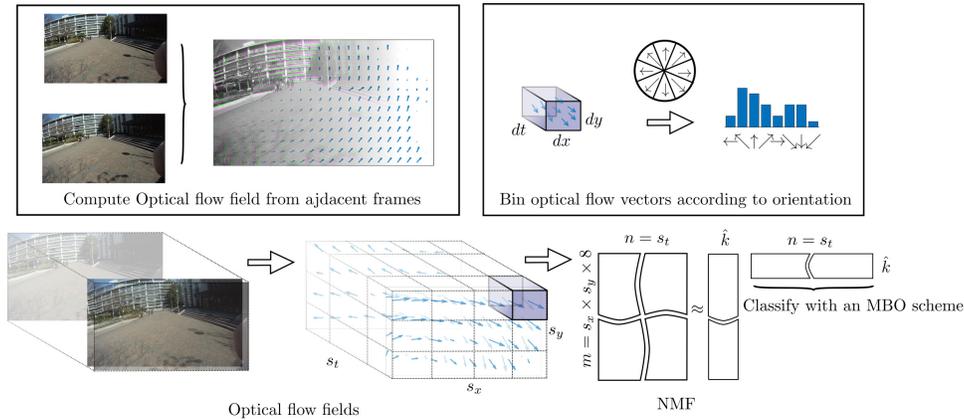
FIGURE 1. A summary of the proposed method. First, we compute a dense optical flow field for each pair of consecutive frames. We then divide each optical flow field into $s_x \times s_y$ spatial regions, where each region consists of $dx \times dy$ pixels, and divide the video into s_t temporal segments, where each segment consists of dt frames. For each $dx \times dy \times dt$ cuboid, we count the number of flow vectors with direction lying within in each octant, yielding a $s_x \times s_y$ histogram for each segment of video. We reshape and concatenate each histogram into a single feature vector of dimension $s_x \times s_y \times 8$ describing the motion that occurs within the video segment. The dimension of the feature vectors is reduced with NMF and we smooth them with a moving-window average operator. Finally, we classify the smoothed features with a semi-supervised MBO scheme.

to reconstruct the original frame, which can potentially be used to recover personally identifiable information, so we do not pursue this line of approach.

In [30], the authors also study ego-activity recognition in police body-worn video. We improve upon their work by choosing a more sophisticated feature than theirs to handle the increased diversity of activities in our much larger data set. We demonstrate the improved performance of the proposed method in section 4. To the best of our knowledge, no other experiment results on real-world police body-worn video have been reported in the literature.

3. **Method.** We start with extracting features based on motion cues from the video. The extracted motion features are potentially high-dimensional, so they are compressed to a lower-dimensional representation to alleviate computational burden. Finally, we classify the video footage with the low-dimensional representation using a PDE-based semi-supervised learning method that only requires 10% training data from each class of activity. The flowchart fig. 1 summarizes the proposed system, which we detail below.

3.1. **Motion descriptor.** Our motion descriptor is similar to the one presented in [39] except for the final dimension reduction step: [39] uses the principle component analysis (PCA) whereas we choose the non-negative matrix factorization (NMF) because the features are inherently non-negative. Before we compute any feature,

we resize all video frames to have a resolution of 576×1024 and hence an aspect ratio of $16 : 9$, allowing us to choose a uniform set of video parameters across all data sets.

3.1.1. *Dense optical flow fields.* Dense optical flow fields [18, 25, 3, 10], which describe relative motion between objects in the scene and the camera, form the basis of our motion analysis. Optical flow fields are fields of two-dimensional vectors (u, v) defined on the two-dimensional domain of images. In the discrete setting, an optical flow field associates each pixel in an image with an optical flow vector which consists of a horizontal and vertical component. An optical flow field is calculated from a pair of consecutive frames under the assumption that pixels displaced according to the optical flow field should preserve their intensities after the displacement. Formally, let $x(t), y(t)$ be the pixel location of a particular pixel that is displaced according to the optical flow field,

$$\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u\left(x(t), y(t)\right) \\ v\left(x(t), y(t)\right) \end{bmatrix}.$$

Then the intensity constancy assumption can be formulated by

$$\frac{d}{dt} I\left(x(t), y(t), t\right) = 0, \tag{1}$$

which yields the following identity [18],

$$\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} = 0.$$

The well-known Horn-Schunk method then seeks the optical flow field (u, v) by minimizing

$$\iint \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right)^2 + \lambda \left(\|\nabla u\|_2^2 + \|\nabla v\|_2^2 \right) dx\, dy, \tag{2}$$

where $\| \cdot \|_2$ denote the 2-norm of a vector. Note that the first term of eq. (2) encourages the flow fields to satisfy the intensity constancy assumption eq. (1) while the second term regularizes this ill-posed problem by promoting smooth vector fields. In their original paper, the authors of [18] solve the Euler-Lagrange equation of eq. (2) to determine the optical flow fields; myriads of optical flow methods have been proposed in the past three decades and we refer readers to the survey paper [14] for this subject.

Assuming that the objects recorded in a pair of frames are static, the optical flow field encodes the movement of the camera and hence the movement of the camera-wearer. Although this assumption does not necessarily hold perfectly for real-world body-worn video footage, static background objects often cover the majority of frames, and thus we can use optical flow fields to estimate the movement of the camera-wearer. Even when this assumption is not true, we have found that optical flow fields induced by the movement of objects instead of the camera-wearer are still helpful in certain situations. For instance, they characterize driving a car by the static interior of the vehicle and the movement in the windshield region. This is also observed in the experiments conducted by [37]; the authors find distinctive patterns of optical flow fields in the windshield region that correspond well to the camera-wearer driving a car.

3.1.2. *Histograms on dense optical flow fields.* Using optical flow fields is common in classifying ego-activities. Different motion features are effectively different ways of aggregating them. For instance, authors of [21, 39, 40] bin optical flow vectors to construct features in the form of concatenated histograms, [37, 46, 6] aggregate them via convolution kernels, and [30, 36] infer camera movement using unaggregated optical flow fields as input. In our case, we compute the motion descriptors, proposed in [39], as histograms of extracted dense optical flow vectors. We bin the vectors according to their locations in the frames and orientations, and then count the number of vectors in each bin. Note that we lose magnitude information in this process because the bins only correspond with locations and orientations. The features proposed in [21] retain magnitude information by further grouping optical flow vectors according to their magnitudes, but in our experiments we observe comparable performance using the simplified features.

To compute the motion descriptors from the optical flow fields, we consider a video as a 3D volume with frames (optical flow fields) stacked along the time axis. We spatially divide each frame into s_x by s_y rectangular regions of fixed width dx and height dy pixels; the choice of dx and dy determines the spatial resolution of the final features. We have found that choosing dx and dy that are divisible by the total number of pixels in length and height, respectively — yielding $s_x = 16$ and $s_y = 9$ — gives good performance on all data sets tested. We also divide the video into s_t video segments, each with a fixed time duration ΔT, that is, dt frames. We choose ΔT depending on the time scale of the ego-activities that we wish to classify. For instance, we choose $\Delta T = 0.2$ second for videos containing a mix of long term and short term ego-activities, whereas we choose $\Delta T = 4$ seconds if we wish to classify relatively long-term activities. The choice of ΔT also determines the computation cost of the subsequent analysis. A finer time resolution, i.e. a smaller ΔT, yields more video segments for a given video and hence results in more computations.

Consider the optical flow vectors in each $dx \times dy \times dt$ volume. We place each of them into one of the pre-defined eight histogram bins based on its orientation. Formally, a vector with a directional angle θ is placed in bin $\lfloor \theta / \frac{\pi}{4} \rfloor$. Repeating the above steps for every $dx \times dy \times dt$ volumes in each video segment of duration ΔT, we obtain a feature vector with a dimension of $s_x \times s_y \times 8$ for each segment, which we reshape into a single column vector. By repeating the above procedures for every video segments of length ΔT and stacking obtained feature vectors, we obtain a data matrix X with the number of columns equal to the number of segments in the video. A detailed description of this procedure is presented in algorithm 3 in the appendix.

3.1.3. *Non-negative matrix factorization.* The concatenated histograms for each video segment can have $9 \cdot 16 \cdot 8 = 1152$ entries, resulting in a $1152 \times s_t$ matrix, which can potentially be expensive to compute with. To alleviate this problem, we employ dimension reduction techniques. In [39], the authors use the principal component analysis (PCA) to perform dimension reduction. However, we use non-negative matrix factorization (NMF) [23] because the concatenated histograms are inherently non-negative. NMF is widely used in the context of topic modeling, where users want to learn topics, a collection of words that often co-occur in textual documents, each of which is represented by a histogram of words. In our case, each video segment is represented by a histogram of "motion words"; each motion word is the movement of a specific orientation in a specific region of the frame. Analogously, a topic — a collection of motion words — describes a global movement pattern. We

then model the concatenated histogram of motion words of each video segment as a non-negative linear combination of the topics.

NMF factorizes a non-negative $m \times n$ matrix X (in our case, $m = s_x \cdot s_y \cdot 8$ and $n = s_t$) into the product of two low rank non-negative $m \times \hat{k}$ and $\hat{k} \times n$ matrices V and H. The number $\hat{k}$ is chosen by the users according to their computation resources and tuned based on the resulting performance. We have found that $\hat{k} = 50$ works well for all considered data sets. Formally, this is achieved by solving the following constrained minimization problem,

$$\min_{V,H} \|X - VH\|_F^2, \text{subject to } V \geq 0, H \geq 0, \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Each column in V represents a basis vector (a topic), and each entry in H represents the non-negative linear combination coefficients. Each column in the matrix H is the feature vector for a single video segment, which will be passed into our classification algorithm after a post-processing step (detailed in section 3.1.4).

We also note that we do not necessarily need to perform NMF every time we obtain a new video. We may choose to fix V which we obtain by applying NMF to the initial data set. Then we only need to compute the combination coefficients H^{new} for the new videos X^{new} by solving a non-negative least squares problem

$$H^{\text{new}} = \arg\min_{H} \|X^{\text{new}} - VH\|_F^2, \tag{4}$$

which can be solved very efficiently using methods such as ones proposed in [22].

The dimension reduction step also helps to secure personally identifiable information. Since we do not make use of V in the classification algorithm, there is no need to save it. Without the basis, it is impossible to reconstruct the data matrix X and let alone the content of the videos.

3.1.4. *Post-processing.* We assume a certain degree of temporal regularity of the extracted features: the duration of activities is typically much longer than transitions between them, and so transitions are relatively rare. We note that none of our feature extraction procedure takes advantage of this temporal regularity. Each optical flow field is computed from only two adjacent frames, motion descriptors are aggregated within non-overlapping video segments, and NMF treats columns in the data matrix X (motion descriptors of video segments) independently. Methods exploiting temporal regularity have been proposed before. In [40], for instance, the authors apply multiple temporal pooling operators to the extracted per-frame motion and visual features and use the outputs as additional features. We choose a simpler approach, in which we apply a single moving-window average operator on each row of H and then pass these averaged features to the classification method. We determine the window size of the moving-window average operator experimentally for each data set. Choosing a large window size may eliminate distinct features of short-term activities, so the choice depends on the types of activities in the data set as well as the chosen value of ΔT.

3.2. **Classification method.** Recently, PDE-inspired graph-based semi-supervised and unsupervised learning methods have been successfully applied to image processing [32] and classification of high-dimensional data such as hyperspectral images [33, 19, 48, 29] and body-worn videos [30]. In this section, we outline one of these methods based on minimizing the graph Total Variation, which has been studied in

[4, 16, 31]. We consider each data point (i.e. video segment) as a node in a weighted graph. The edge weight between a pair of nodes i and j is given by the similarity

$$w_{ij} = \exp\left(-\frac{\|H_i - H_j\|_2^2}{\tau_{ij}}\right), \tag{5}$$

where $\|\cdot\|_2$ denotes the 2-norm of a vector and τ_{ij}'s are scaling constants. Here H_i is the ith column of matrix H obtained from NMF. The scaling constants can either be the same chosen τ for all pairs of i and j, or chosen locally for each individual pair [47]. We choose the local scaling constants $\tau_{ij} = \tau_i \tau_j$ where τ_i is the distance between i and its Kth nearest neighbor.

We aim to partition n nodes into c classes (i.e. ego-activities) such that

1. similar nodes between which edge weights are large (i.e. w_{ij}'s are close to 1 should be in the same class, and
2. fidelity nodes (i.e. manually labeled nodes) should be classified according to their labels.

To achieve 1), we optimize the graph Total Variation (TV) defined as follows. Let u be an $\{0,1\}^c$-valued assignment function on the set of nodes, that is $u_\ell(i) = 1$ meaning we assign the ith data point to class ℓ. We can then define the graph Total Variation

$$|u|_{TV} = \frac{1}{2} \sum_{i,j=1}^n w_{ij} \|u(i) - u(j)\|_1, \tag{6}$$

which is referred to as anisotropic Total Variation by [17]. We observe that (6) admits a trivial minimizer that is constant across all nodes. To avoid this problem and to incorporate the training data, we introduce a least-squares data fidelity term

$$F(u) = \frac{1}{2} \sum_{i=1}^n M(i) \|u(i) - f(i)\|_2^2, \tag{7}$$

where $M(i) = 1$ if node i is chosen as fidelity and 0 otherwise, and $f(i) \in \{0,1\}^c$ encodes the known label of node i. We weight the fidelity term by a positive parameter η to balance the graph TV term and the fidelity term in the objective function,

$$\frac{1}{2}|u|_{TV} + \eta F(u). \tag{8}$$

Instead of minimizing (8) directly, which is discrete and combinatorial, we solve the Ginzburg-Landau relaxation [4] for $u(i) \in \mathbb{R}^c$. Namely, we replace the graph Total Variation $|u|_{TV}$ with

$$GL_\epsilon(u) = \frac{1}{4} \sum_{i,j=1}^n w_{ij} \|u(i) - u(j)\|_2^2 + \frac{1}{\epsilon} \sum_{i=1}^n P\left(u(i)\right), \tag{9}$$

where ϵ is a small positive constant, and P is a multi-well potential with minima at the corners of the unit simplex, for instance

$$P\left(u(i)\right) = \prod_{\ell=1}^c \frac{1}{4} \|u(i) - e_\ell\|_1^2, \tag{10}$$

where e_ℓ is the unit vector in $\mathbb{R}^c$ in the ℓth direction. The authors of [44] prove the following Γ-convergence

$$GL_\epsilon(u) \xrightarrow{\Gamma} \begin{cases} |u|_{TV} & \text{if } u \text{ is binary} \\ +\infty & \text{otherwise} \end{cases} \tag{11}$$

as $\epsilon \to 0$ in the case of $c = 2$. The Γ-convergence ensures that the minimizers of $GL_\epsilon(u)$ approach the minimizers of $|u|_{TV}$ and hence justifies the Ginzburg-Landau relaxation of the Total Variation. After the Ginzburg-Landau relaxation, we arrive at the objective function

$$GL_\epsilon(u) + \eta F(u), \tag{12}$$

which we minimize with respect to u.

To formulate (12) in terms of matrices, we first identify u and f by a $n \times c$ matrix where $u_{i\ell} = u_\ell(i)$ and $f_{i\ell} = f_\ell(i)$. We let W be the matrix of w_{ij}'s, and D be an $n \times n$ diagonal matrix with the ith entry d_i being the strength of node i, i.e. $d_i = \sum_{j=1}^n w_{ij}$, and then define the graph Laplacian

$$L = D - W. \tag{13}$$

We also let M be an $n \times n$ diagonal matrix of which the ith entry is $M(i)$ indicating whether node i is chosen as fidelity. If we define L and M this way, we can write (12) in the matrix form

$$\frac{1}{2}\text{trace}\left(u^T L u\right) + \frac{1}{\epsilon}\sum_{i=1}^n P\left(u_i\right) + \frac{\eta}{2}\|M(u-f)\|_2^2. \tag{14}$$

In graph clustering, unsupervised learning, and community detection literature, the graph Laplacian is often normalized to guarantee convergence to a continuum differential operator with a large number of data points (see, for instance, [4]). One popular version of normalized graph Laplacian is the symmetric Laplacian

$$L_s = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

If we substitute L for L_s, the first quadratic term of (14) becomes

$$\text{trace}\left(u^T L_s u\right) = \frac{1}{2}\sum_{i,j=1}^n w_{ij} \left\|\frac{u(i)}{\sqrt{d_i}} - \frac{u(j)}{\sqrt{d_j}}\right\|_2^2.$$

The methods described in the remainder of this paper carry over regardless of which graph Laplacian is used, and the notation L is a placeholder for any choice of graph Laplacian. In our experiments, we choose to use the symmetric graph Laplacian L_s because it permits the use of efficient and simple computation routines to approximate its eigenvalues and eigenvectors.

3.2.1. *Optimization scheme.* Minimizing (14) using the standard gradient descent method yields

$$\frac{\partial u}{\partial t} = -Lu - \frac{1}{\epsilon}\nabla \hat{P}(u) - \eta M(u - f), \tag{15}$$

where $\hat{P}(u) = \sum_{i=1}^n P(u_i)$. The steady-state solution of (15) is a stationary point of (14). This is known as the graph Allen-Cahn equation. In the continuum, the The Allen-Cahn equation converges to the mean curvature flow and an analogous convergence for the graph case has been established in [26]. We follow [28] to use a variant of the Merriman-Bence-Osher (MBO) scheme to approximate and solve (15). We note that, in the continuum, the MBO scheme is known to approximate

the mean curvature flow, just as the Allen-Cahn equation. An explicit connection between the graph Allen-Cahn equation and the MBO scheme has been explored in the recent artical [7]. In short, we first randomly initialize u^0, which we use as the initial condition for (15). We then alternate between the following two steps:

1. *Diffusion*: for given u^k, we obtain $u^{k+\frac{1}{2}}$ by solving a force-driven heat equation

$$\frac{\partial u}{\partial t} = -Lu - \eta M(u - f),\qquad(16)$$

for $t_k \leq t \leq t_k + \frac{1}{2}\Delta t$, where Δt is a parameter.

2. *Threshold*: we threshold $u^{k+\frac{1}{2}}$ to obtain u^{k+1}, i.e.

$$u^{k+1}(i) = e_\ell \ , \text{ where } \ell = \arg\max_{\hat{\ell}} u_{i\hat{\ell}}^{k+\frac{1}{2}}.\qquad(17)$$

For a small ϵ, this approximates solving

$$\frac{\partial u}{\partial t} = -\frac{1}{\epsilon}\nabla\hat{P}(u)\qquad(18)$$

for $t_k + \frac{1}{2}\Delta t \leq t \leq t_{k+1} = t_k + \Delta t$.

Choosing Δt is delicate. If it is too small, $u^{k+1} = u^k$ after thresholding, whereas if it is too large, u converges to the steady-state solution of (16),

$$(L + \eta M)^{-1}Mf,$$

in one diffusion step, independent of the initial condition u^k. Either way, extreme Δt results in a "freezing" scheme. In [45], the authors give guidance on how to choose Δt in the case of unnormalized graph Laplacian, $c = 2$ (i.e. binary classification), and $\eta = 0$. Currently, there is no analogous result for a symmetric graph Laplacian, multi-class classification, and nonzero η. We have found, however, that $\Delta t = 0.1$ gives nontrivial dynamics (i.e. convergent and not "freezing") on all data sets used in testing.

3.2.2. *Numerical methods.* We follow [4, 16] to employ a semi-implicit ordinary differential equation solver to solve (16), and use a pseudo-spectral method coupled with the Nyström extension to make the ordinary differential equation solver efficient. We note that the graph Laplacian matrix L is large, with n^2 entries where n is the number of data points; it is also not inherently sparse, which makes approximation techniques such as the Nyström extension necessary.

For the ordinary differential equation solver, we take N_{step} time steps to reach $u^{k+\frac{1}{2}}$ from u^k, where N_{step} is a parameter to choose. Formally, we let $u^{k,s}, s = 0, 1, \cdots, N_{\text{steps}}$ denote the numerical solutions of (16) at intermediate time $t_k + s\delta t$, where $\delta t = \Delta t/2N_{\text{step}}$. We solve

$$\frac{u^{k,s+1} - u^{k,s}}{\delta t} = -Lu^{k,s+1} - \eta M(u^{k,s} - f)\qquad(19)$$

for $u^{k,s+1}$. We use $N_{\text{step}} = 10$ to ensure convergence of the ordinary differential equation solver when $\eta < 500$ and $\Delta t = 0.1$.

We use a pseudo-spectral method to solve Equation (19). We project the solution u onto an orthonormal eigenbasis of the graph Laplacian L, or an eigen-subbasis that consists of N_{eig} eigenvectors corresponding to the smallest N_{eig} eigenvalues. We detail how we compute the spectrum of L with the Nyström extension in section 3.3. Choosing a modest $N_{\text{eig}} \ll n$ will greatly improve the efficiency of the

algorithm because solving (19) only requires $O(nN_{\text{eig}})$ operations if the eigenvectors and eigenvalues of L are provided. Suppose Φ is an $n \times N_{\text{eig}}$ eigenvector matrix, of which the jth column ϕ_j is the eigenvector of L corresponding to the jth smallest eigenvalue λ_j, and Λ is the diagonal matrix containing all N_{eig} smallest eigenvalues λ_j's. We let a denote the coordinates we obtain by projecting columns of u onto the eigen-subspace spanned by columns of Φ, i.e. $a = \Phi^T u$. Solving (19) in the eigen-subspace is simply

$$
\begin{aligned}
a^{k,s+1} &= (I + \delta t \Lambda)^{-1} a^{k,s} - \delta t \cdot \eta \Phi^T M(u^{k,s} - f), \\
u^{k,s+1} &= \Phi a^{k,s+1}.
\end{aligned}
$$

Algorithm 1 Graph MBO scheme [4]

1: **Input:** $\Phi, \Lambda, M, f, \eta$, and initial guess u^0.
2: **Output:** u.
3: **Initialize** $u^{0,0} = u^0, a^{0,0} = \Phi^T u^0$.
4: **for** $k = 1, 2, \cdots$, MaxIter or u^k has converged **do**
5:     a. Diffusion:
6:     **for** $s = 0, 1, \cdots, N_{\text{step}} - 1$ **do**
7:        $a^{k,s+1} = (I + \delta t \Lambda)^{-1} a^{k,s} - \delta t \cdot \eta \Phi^T M(u^{k,s} - f)$.
8:        $u^{k,s+1} = \Phi a^{k,s+1}$.
9:     **end for**
10:    b. Threshold $u^{k+1/2} := u^{k,N_{\text{step}}}$:
11:    **for** $i = 1, 2, \cdots, n$ **do**
12:       $u^{k+1,0}(i) = e_\ell$, where $\ell = \arg\max_{\hat{\ell}} u_{i\hat{\ell}}^{k,N_{\text{step}}}$
13:    **end for**
14: **end for**

3.3. Nyström extension.

We employ the Nyström extension [15], which approximates the eigenvectors and eigenvalues of L with $O\left(nN_{\text{eig}}^3\right)$ computation complexity and $O(nN_{\text{eig}})$ memory requirement. In practice, we choose N_{eig} with respect to the physical memory available to us and we also ensure that it is sufficiently large so that the first N_{eig} eigenvectors encapsulate enough geometric information of the similarity graph. With $N_{\text{eig}} \ll n$, the computation complexity and memory scales linearly with respect to the number of data points. The idea of the Nyström extension is to uniformly randomly sample a smaller set of data points $A \subset \{1, 2, \cdots, n\}$ with $|A| = N_{sample} \ll n$, perform spectral decomposition on an $N_{sample} \times N_{sample}$ system calculated from the set of data points A, and then interpolate the result to obtain an approximation to the spectral decomposition of the entirety of L. Let B be the complement of A, i.e. $A \cup B = \{1, 2, \cdots, n\}$ and $A \cap B = \emptyset$. Let W_{AA} denote the weights associated with nodes in set A, and similarly, let $W_{AB} = W_{BA}^T$ denote weights between nodes in set A and B. If we reorder the nodes so that $A = \{1, 2, \cdots, N_{sample}\}$ and $B = \{N_{sample} + 1, N_{sample} + 2, \cdots, n\}$, we can rewrite

$$
W = \begin{bmatrix} W_{AA} & W_{AB} \\ W_{BA} & W_{BB} \end{bmatrix}. \tag{20}
$$

It can be shown [15] that the matrix W_{BB} can be approximated by $W_{BB} \approx W_{BA} W_{AA}^{-1} W_{AB}$ in the context of approximating the spectral decomposition. The

Nyström extension uses this property to approximate the spectrum of W, and henceforth L. We summarize the Nyström extension algorithm to approximate the spectrum of symmetric graph Laplacian in Algorithm 2. An analogous algorithm for unnormalized graph Laplacian can be found in [4]. In Algorithm 2, **1** denotes a vector of one's that is used to compute the strength of each nodes, i.e. the sum of weights, and let $X./Y$ denote component-wise division between two matrices X and Y of the same size. We let $\sqrt{X}$ denote the non-negative square root of each component of any non-negative matrix X, and if X is positive definite with the spectral decomposition $X = Q\Gamma Q^T$, we let $X^{1/2} = Q\Gamma^{1/2}Q^T$ and similarly $X^{-1/2} = Q\Gamma^{-1/2}Q^T$.

Algorithm 2 Nyström Extension for symmetric graph Laplacian[4][15]

1: **Input:** $\{H_i\}_{i=1}^n$ and $\{\tau_{ij}\}_{ij=1}^n$.
2: **Output:** $\Phi, \{\lambda_j\}_{j=1}^{N_{\text{eig}}}$.
3: Randomly sample $A \subset \{1, 2, \cdots, n\}$ with $|A| = N_{sample} \geq N_{\text{eig}}$ and B such that $A \cup B = \{1, 2, \cdots, n\}$.
4: Compute W_{AA} and W_{AB} using (5).
5: Compute the strength of nodes in A, $d_A = W_{AA}\mathbf{1}$.
6: Approximate the strength of nodes in B, $d_B = W_{BA}\mathbf{1} + W_{BA}W_{AA}^{-1}W_{AB}\mathbf{1}$.
7: Normalize $W_{AA} = W_{AA}./\sqrt{d_A d_A^T}$.
8: Normalize $W_{AB} = W_{AB}./\sqrt{d_A d_B^T}$.
9: Perform spectral decomposition on $W_{AA} + W_{AA}^{-1/2}W_{AB}W_{AB}^T W_{AA}^{-1/2}$ to obtain the N_{eig} largest eigenvalues $\{\xi_i\}_{i=1}^{N_{\text{eig}}}$ and the corresponding eigenvectors $\{\psi_i\}_{i=1}^{N_{\text{eig}}}$. We let Ψ denote the matrix of the eigenvectors and Ξ be a diagonal matrix with ξ_i's on the diagonal.
10: Output $\lambda_i = 1 - \xi_i$, and $\Phi = \begin{bmatrix} W_{AA}^{1/2} \\ W_{BA}W_{AA}^{-1/2} \end{bmatrix} \Psi\Xi^{-1/2}$.

---

4. **Experiments.** We apply our method on two publicly available data sets, the QUAD data set [21], and the HUJI EgoSeg data set [36], and compare our results to those reported in [21, 40, 43, 37, 36]. We also apply both our method and the one proposed in [30][1] on a police body-worn video data set provided by the LAPD. Our experimental procedures and parameters are summarized in TABLE 1. The measures of success we use are precision

$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

and recall

$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

within each class, mean precision and recall directly averaged over all classes, and the overall accuracy, i.e. the percentage of correctly classified data points.

The feature extraction is done on an offline machine to ensure the security of the LAPD video. Subsequent analysis, including the Nyström extension and the graph MBO scheme, is performed on a 2.3GHz machine with Intel Core i7 and 4 GB of

---

[1]with the implementation kindly provided by the authors of [30]

Table 1. Experimental Setup

	Motion feature				NMF	Spectrum of the Graph Laplacian				MBO		
	ΔT (sec)	FPS	Number of segments	Window size (segment)	$\hat{k}$	N_{eig}	τ_{ij}	N_{sample}	Batch size (segment)	η	Δt	N_{step}
QUAD	1/60	60	14,399	-	50	500	$\tau = 1$	1000	-	300	0.1	10
LAPD	1/5	30	274,443	5	50	2000	$K = 100$	2000	30000	400	0.1	10
LAPD [30]	1/5	30	274,443	-	-	2000	$K = 100$	2000	30000	400	0.1	10
HUJI	4	15	36,421	20	50	400	$K = 40$	400	-	300	0.1	10

memory. Both experiments on the QUAD data set and the HUJI EgoSeg data set can be finished within a minute after extracting features; each batch of the LAPD body-worn video data set (see section 4.2 for details) takes around two minutes.

4.1. **QUAD data set.** The authors of [21] choreographed and made public the QUAD data set, which is about four minutes long and filmed at 60 frames per second. The footage was recorded with a head-mounted Go-Pro Camera while the camera-wearer was undergoing nine ego-activities (reported in TABLE 2), such as walking, jumping, and climbing up stairs[2]. The authors of [21] and [30] tested their ego-activity classification methods on this data set; we follow the same experimental protocol as [30]. Each video "segment" is chosen to be an individual frame and we uniformly sample 10% segments within each category as fidelity in agreement with the protocol employed in [30]. Such choice of one frame per segment yields 14,399 segments.

In TABLE 2, we report precision within each category and the mean precision, directly averaged over nine classes; the authors of [21] have also reported the mean precision and the authors of [30] provided detailed precision per class. Both our method and the method in [30] use 10% of the video, sampled uniformly, as fidelity. The method in [21] is unsupervised and the reported mean precision is calculated after matching the discovered ego-activity categories to the ground-truth categories in a way that the best match gives the highest harmonic mean of the precision and recall (i.e. the best F-measure). Our result is overall an improvement upon [30] in terms of precision.

The QUAD data set only consists of a short choreographed video, in which activities of interest have a relatively balanced proportion, and the challenges we observe in the field data sets are absent. However, the experiment on the choreographed data set validates the baseline ability of our method in recognizing ego-activities in body-worn videos. We further test our method and showcase the applicability of our method to data sets consist of multiple videos of different lengths that are not choreographed and recorded in a variety conditions.

4.2. **LAPD BWV data set.** The LAPD body-worn video data set consists of 100 videos with a total length of 15.25 hours recorded at 30 frames per second. The video footage is recorded by cameras mounted on police officers' chests when they are performing a variety of law enforcement activities. The data set consists of videos recorded both inside vehicles and outdoors and under a variety of illumination conditions. We manually annotated each frame of all 100 videos with one of 14 class labels. Although we train on and classify video footage in all 14 categories, we exclude five insignificant classes, such as "exiting car" and "obscured camera", from performance evaluations of the ego-activity recognition algorithms. We report

---

[2]The reported categories of ego-activities are the same ones used in [30] and are different from [21].

TABLE 2. Class proportion and precision of the QUAD data set

		Precision		
Class	**Proportion**	[21]	[30]	**Ours**
Jump	14.54%	-	92.51%	99.07%
Stand	13.74%	-	87.90%	87.11%
Walk	12.75%	-	84.52%	98.37%
Step	12.65%	-	93.98%	98.54%
Turn Left	11.25%	-	89.43%	96.96%
Turn Right	10.16%	-	92.80%	96.21%
Run	9.00%	-	92.38%	96.17%
Look Up	8.85%	-	80.36%	90.02%
Look Down	7.06%	-	84.59%	89.00%
Mean	11.11%	**95%**	88.74%	94.49%

activity proportions of the selected classes in TABLE 3 and, for completeness, all 14 classes in TABLE 5 of the Appendix.

We apply the method in [30] with the provided implementation on the LAPD body-worn video data set. [30] computes a feature vector per frame instead of per short video segment, which consists of 6 frames (0.2 seconds). The average of the frame-wise features over a segment is used as the feature vector of the segment. By doing so, the numbers of video segments to classify in both methods are the same. We apply a moving window average operator with a window size of one second (five segments) to our features. The features of [30] inherently incorporate temporal information, so we use the aggregated segment-wise features as they are without further smoothing.

We divide the 274,443 segments into 9 disjoint batches, each of which consists of approximately 30,000 segments. As each segment has a duration of 0.2 seconds, each batch therefore consists of 100-minutes of footage spanning multiple videos. We perform the classification on each batch independently and concatenate the classification results. We note that both our method and the method proposed in [30] make use of the Nyström extension and the MBO scheme described in section 3.3 and section 3.2 respectively, so they share the same set of parameters. We choose $N_{sample} = 2000$ and $N_{eig} = 2000$ to be the same for both methods for each batch so that they share the same computation cost and both give good performance relatively to other choices of parameters. We have tuned parameters η ranging from 0.01 to 1000 and found that $\eta = 400$ and τ selected automatically according to [47] with $K = 100$ work well for both methods.

With regards to sampling fidelity points, we use the same protocol as the one used in [30] where we uniformly sample 10% segments within each class. Consequently, we have many more samples of common activities than rare activities.

In TABLE 3, we report the precision and recall within each class and their respective means averaged over the selected nine classes. We refer readers to TABLE 5 in the Appendix for a full table of all 14 classes as well as the overall accuracy, which is the proportion of video segments that are correctly classified. We also present a sample of the color-coded classification results in fig. 2 and the confusion matrices in fig. 3.

Our method outperforms [30] in most of the categories in terms of precision and is a major improvement according to recall. We theorize that the features proposed

TABLE 3. Class proportion, precision, and recall of the selected nine classes in the LAPD body-worn video data set

		Precision		Recall	
Class	**Proportion**	**[30]**	**Ours**	**[30]**	**Ours**
Stand still	62.57%	73.10%	89.44%	85.42%	95.24%
In stationary car	16.84%	41.83%	93.69%	43.18%	89.73%
Walk	9.04%	38.36%	70.53%	19.54%	59.41%
In moving car	5.76%	70.71%	91.03%	25.08%	84.40%
At car window	0.64%	17.23%	71.45%	10.94%	45.28%
At car trunk	0.58%	73.78%	71.79%	11.09%	51.78%
Run	0.33%	96.15%	75.94%	11.03%	53.35%
Bike	0.33%	85.71%	86.49%	14.37%	75.44%
Motorcycle	0.08%	100%	92.49%	10.76%	71.75%
Mean	10.68%	66.32%	82.54%	25.71%	69.60%
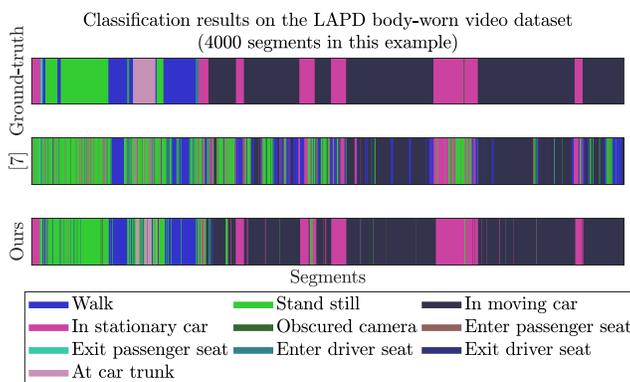
FIGURE 2. Classification results on a contiguous sample of 4000 segments (approximately 13 minutes) from the LAPD body-worn video data set. The results are obtained by running both methods with the parameters described in section 4.2.

in [30] are too simple to distinguish among the increased variety of ego-activities in the larger LAPD body-worn video data set. The features they propose do not make use of the locality of motion within each frame, which we consider crucial in order to differentiate, for instance, driving a car and walking forward. Both activities feature forward motion, but the motion is localized within the windshield region only in the former case. We also note that frequency is a significant component of the features proposed in [30]; however, we do not observe much periodic motion in many ego-activities.

4.3. **HUJI EgoSeg data set.** We also evaluate the performance of our method on the HUJI EgoSeg data set [36] [37]. This data set contains 65 hours of egocentric videos including 44 videos shot using a head-mounted GoPro Hero3+, the Disney

(a) Method proposed in [30]

Actual class \ Classified as	Stand still	In stationary car	Walk	In moving car	At car window	At car trunk	Run	Bike	Motorcycle
Stand still	129916	14607	3047	1109	648	19	2	9	0
In stationary car	19149	16395	1711	206	66	1	0	2	0
Walk	14069	2877	4214	68	31	16	2	0	0
In moving car	6242	2420	1439	3573	26	1	0	0	0
At car window	1072	127	71	0	162	0	0	0	0
At car trunk	770	166	26	1	0	121	0	0	0
Run	423	362	6	1	0	0	100	0	0
Bike	322	65	14	0	0	0	0	72	0
Motorcycle	141	3	53	0	2	0	0	0	24

(b) Ours

Actual class \ Classified as	Stand still	In stationary car	Walk	In moving car	At car window	At car trunk	Run	Bike	Motorcycle
Stand still	159576	1364	4684	391	177	162	74	46	7
In stationary car	3237	41480	264	592	50	50	10	30	6
Walk	9290	233	14598	109	51	36	62	13	0
In moving car	1400	638	251	13346	0	24	4	9	0
At car window	707	41	160	11	786	0	3	9	0
At car trunk	567	34	87	18	0	827	0	0	0
Run	240	9	170	1	5	0	486	0	0
Bike	150	5	50	0	8	0	1	685	0
Motorcycle	29	0	27	6	1	0	0	0	160

FIGURE 3. Confusion matrices for the LAPD Body-worn video data set. The background intensity in cell (k, ℓ) corresponds to the number of data points in class k that are classified as class ℓ by the algorithm.

data set [11] and other YouTube videos[3]. The data set contains 7 ego-action categories: *Walking, Driving, Riding Bus, Biking, Standing, Sitting, and Static*. We normalize the frame rate of each video to 15 frames per second to match with the normalized frame rate in [37]. We divide each video sequence into segments of 4 seconds ($\Delta T = 4$ seconds, 60 frames), which also matches the length of each video segment in [37]. The activities present in the HUJI EgoSeg data set are all relatively long-term activities compared to the LAPD Body-worn video data set, so using longer video segments reduces the number of data points without the risk of missing short-term activities. With such choice of ΔT, we have 36,421 segments. For the Nyström extension and the MBO scheme, we have found that the combination of $N_{sample} = 400$, $N_{\mathrm{eig}} = 400$, $\eta = 300$, and $K = 40$ gives satisfactory results.

We follow the same experimental protocol of [36, 37] to divide the entire data set into a training set and a testing set. We randomly pick video sequences until we have 1300 segments (approximately 90 minutes of video) per class as the training set, and we uniformly sample 10% of the training set as fidelity points, which is about 10% of the training data used in [37][4]. In this experiment, we use recall to evaluate the performance since it is the common measure of success in [36, 40, 43, 37]. TABLE 4 details the classification results on the testing set. The classification performance of methods other than ours are reported in [37]. We also report the confusion matrix in fig. 4 and a color-coded sample of the classification result in fig. 5.

We observe that the recalls of Sitting, Standing, and Riding Bus are typically lower than other activities across all five methods, so we believe that these activities are inherently difficult to recognize with motion-based features. According to TABLE 4, our method outperforms — using recall as a measure of success — other methods that use handcrafted motion and/or appearance features with or without

---

[3]The HUJI EgoSeg data set can be downloaded at http://www.vision.huji.ac.il/egoseg/videos/dataset.html.

[4]The authors of [37] do not explicitly mention the fidelity percentage; we estimate the percentage according to their released code at http://www.vision.huji.ac.il/egoseg/.

FIGURE 4. Confusion matrix for the HUJI EgoSeg data set. The background intensity in cell (k, ℓ) corresponds to the number of data points in class k that are classified as class ℓ by the algorithm.
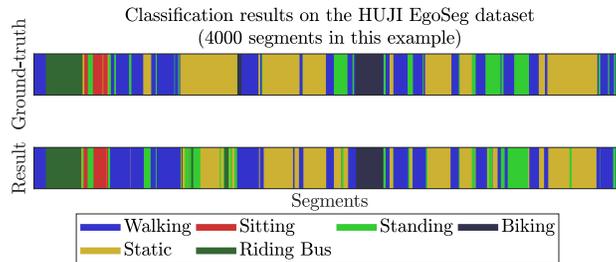
FIGURE 5. Classification results on a contiguous sample of 4000 segments (approximately 4 hours) from the testing set of HUJI EgoSeg data set. The recall of the same experiment is reported in TABLE 4.

deep convolution neural networks, with the exception of [37]. We emphasize that our method uses a fraction of the training data of the supervised methods and still achieves comparable results. When we use the entire training set as fidelity, the mean recall only sees a slight increase.

5. **Conclusion and future work.** In this paper, we study ego-activity recognition in first-person video with an emphasis on the application to real-world police body-worn video. We propose a system for classifying ego-activities in body-worn video footage using handcrafted features and a graph-based semi-supervised learning method. These features based on motion cues do not identify people or objects in the scene and hence secure any personally identifiable information within the video. Our experiments also illustrate that the features are able to differentiate a variety of ego-activities and yield better classification results than an earlier work [30]. The

TABLE 4. Class proportion and recall of the HUJI EgoSeg data set

		Recall				
Class	**Proportion**	**[36]**	**[40]**	**[43]**	**[37]**	**Ours**
Walking	34%	83%	91%	79%	89%	91%
Sitting	25%	62%	70%	62%	84%	71%
Standing	21%	47%	44%	62%	79%	47%
Biking	8%	86%	34%	36%	91%	88%
Driving	5%	74%	82%	92%	100%	95%
Static	4%	97%	61%	100%	98%	96%
Riding Bus	4%	43%	37%	58%	82%	84%
Mean	14%	70%	60%	70%	89%	82%
Training		∼60%	∼60%	∼60%	∼60%	6%

semi-supervised classification method addresses the challenge of insufficient training data; it achieves comparable performance to supervised methods on two publicly available benchmark data sets using only a fraction of training data. Despite using a smaller fraction of training data, our classification results are comparable to or better than those in prior works, which include both classical and deep-learning methodologies. The proposed system also demonstrates promising results on field data from body-worn cameras used by the Los Angeles Police Department.

We note that the MBO-based classification method can be used with any feature design, not only the global motion descriptor as presented here. The general graphical setting of the classification method even allows features that cannot be represented by a vector in the Euclidean space so long there exists a way to measure similarity between the features of two data points. Also, the Nyström extension is still applicable with new features or similarity measures to efficiently approximate the spectrum of the graph Laplacian.

Recent developments in unsupervised convolution neural networks [6, 46] might be used to improve and extend the current feature selection method, although caution must be taken to prevent the recovery of personal identifiable information from the learned features. Better incorporation of temporal information is another way to move forward. We observe that the police body-worn video data set contains a mix of long-term and short-term activities, making it difficult to select a single time scale to design the features around. In the present experiment, we chose the length of each segment to be 0.2 seconds in order to capture short-term activities, but this was redundant for recognizing long-term activities. We chose ΔT to be four seconds for the HUJI EgoSeg data set, which significantly reduced the computation cost without sacrificing accuracy, but this was only possible because all activities in the HUJI EgoSeg data set have long durations. Designing features that efficiently handle a mix of long-term and short-term activities is another challenge to be addressed.

Future work will also be directed towards improving the proposed classification method. For instance, [20] recently proposed to incorporate the knowledge of the proportions of classes as an extra input in the semi-supervised classification method described in section 3.2. Considering the heterogeneity in the class distribution that we observe in the police body-worn video data set (see table 3), we expect to see an improvement in the classification performance with the class proportion information.

Despite our best effort to develop an accurate system for the classification of police body-worn videos, the variability of the data leads to imperfect classification. Our classification method is naturally paired with uncertainty quantification (UQ) [5]. Besides giving a video segment an ego-activity label, we may use this technology to estimate a measure of uncertainty, which identifies hard-to-classify video segments that require further investigation. Moreover, the measure of uncertainty can suggest footage for police analysts to label to train classification algorithms making an efficient use of human labeling effort. We expect that further development of the feature selection, classification, and uncertainty quantification methodologies will facilitate an implementation of the proposed system to be used by law enforcement agencies to summarize a large volume of body-worn video footage.

Algorithm 3 Global Motion Descriptor

1: **Input:** Optical flow fields matrix $O \in \mathbb{R}^{n_f \times n_x \times n_y \times 2}$
2: **Output:** Matrix $X \in \mathbb{R}^{s_t \times (s_x \cdot s_y \cdot 8)}$
3: **Initialize** $dt = 60, dx = dy = 64, s_x = \frac{n_x}{dx}, s_y = \frac{n_y}{dy},$
   $s_t = \lfloor \frac{n_f}{dt} \rfloor$, histogram count matrix $C \in \mathbb{R}^{s_t \times s_x \times s_y \times 8}$
4: **for** $i = 0 : s_t$ **do**
5:   **for** $j = 0 : s_x$ **do**
6:     **for** $k = 0 : s_y$ **do**
7:       % Step 1. Partition:
8:       cuboid $= O[idt : (i+1)dt, jdx : (j+1)dx,$
             $kdy : (k+1)dy, :]$
9:       % reshape: $\mathbb{R}^{dt \times dx \times dy \times 2} \mapsto \mathbb{R}^{(dt \cdot dx \cdot dy) \times 2}$
10:      cuboid $=$ reshape(cuboid)
11:      % Step 2. Histogram count:
12:      **for** $l = 0, 1, \cdots, (dt \cdot dx \cdot dy)$ **do**
13:        $v = $ cuboid$[l, :]$
14:        $\theta = $ phase(v)
15:        $bin = \lfloor \theta / \frac{\pi}{4} \rfloor$
16:        $C[i, j, k, bin]{++}$
17:      **end for**
18:     **end for**
19:   **end for**
20: **end for**
21: % reshape: $\mathbb{R}^{s_t \times s_x \times s_y \times 8} \mapsto \mathbb{R}^{(s_x \cdot s_y \cdot 8) \times s_t}$
22: $X = $ reshape(C)

Appendix. We report the classification results of the entire 14 ego-activity categories in the LAPD body-worn video data set in TABLE 5 as well as the full confusion matrices in fig. 6.

Actual class \ Classified as	Stand still	In stationary car	Walk	In moving car	Obscured camera	At car window	At car trunk	Exit driver seat	Exit passenger seat	Run	Bike	Enter passenger seat	Enter driver seat	Motorcycle
Stand still	129916	14607	3047	1109	440	648	19	1100	19	2	9	683	489	0
In stationary car	19149	16395	1711	206	97	66	1	57	1	0	2	215	72	0
Walk	14069	2877	4214	68	80	31	16	121	1	2	0	41	43	0
In moving car	6242	2420	1439	3573	447	26	1	16	2	0	0	66	15	0
Obscured camera	3833	1927	263	81	1173	2	0	31	1	0	0	36	15	0
At car window	1072	127	71	0	5	162	0	44	0	0	0	0	0	0
At car trunk	770	166	26	1	4	0	121	0	0	0	0	3	0	0
Exit driver seat	629	86	19	6	9	1	0	102	2	0	1	7	1	0
Exit passenger seat	596	82	61	5	12	2	6	8	102	0	0	4	2	0
Run	423	362	6	1	0	0	0	12	0	100	0	0	3	0
Bike	322	65	14	0	0	0	0	28	0	0	72	0	0	0
Enter passenger seat	349	40	34	3	3	0	0	6	0	0	0	67	3	0
Enter driver seat	215	34	26	0	1	0	0	1	0	0	0	1	39	0
Motorcycle	141	3	53	0	0	2	0	0	0	0	0	0	0	24

(a) Method proposed in [30]

Actual class \ Classified as	Stand still	In stationary car	Walk	In moving car	Obscured camera	At car window	At car trunk	Exit driver seat	Exit passenger seat	Run	Bike	Enter passenger seat	Enter driver seat	Motorcycle
Stand still	159576	1364	4684	391	753	177	162	83	134	74	46	59	41	7
In stationary car	3237	41480	264	592	167	50	50	84	113	10	30	74	71	6
Walk	9290	233	14598	109	121	51	36	24	18	62	13	8	7	0
In moving car	1400	638	251	13346	113	0	24	6	0	4	9	13	8	0
Obscured camera	1747	194	162	111	5399	4	37	3	3	0	0	3	0	0
At car window	707	41	160	11	15	786	0	0	0	3	9	1	3	0
At car trunk	567	34	87	18	61	0	827	1	0	0	0	0	2	0
Exit driver seat	530	96	89	19	11	9	4	203	0	0	0	0	0	0
Exit passenger seat	488	75	73	37	13	9	2	0	250	0	0	4	0	0
Run	240	9	170	1	0	5	0	0	0	486	0	0	0	0
Bike	150	5	50	0	7	8	0	0	2	1	685	0	0	0
Enter passenger seat	276	64	46	18	12	0	6	0	0	0	0	137	0	0
Enter driver seat	171	39	37	2	8	0	4	0	0	0	0	0	69	0
Motorcycle	29	0	27	6	0	1	0	0	0	0	0	0	0	160

(b) Ours

FIGURE 6. Confusion matrices for the LAPD police Body-worn video data set. The background intensity of cell (k, ℓ) corresponds to the number of data points in class k that are classified as class ℓ by the algorithm.

REFERENCES

[1] G. Abebe and A. Cavallaro, A long short-term memory convolutional neural network for first-person vision activity recognition, in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 1339–1346.

TABLE 5. Class proportion, precision , recall, and accuracy on the LAPD body-worn video data set

		Precision		Recall	
Class	**Proportion**	**[30]**	**Ours**	**[30]**	**Ours**
Stand still	62.57%	73.10%	89.44%	85.42 %	95.24%
In stationary car	16.84%	41.83%	93.69%	43.18%	89.73%
Walk	9.04%	38.36%	70.53%	19.54%	59.41%
In moving car	5.76%	70.71%	91.03%	25.08%	84.40%
Obscured camera	2.80%	51.65%	80.82%	15.93%	70.46%
At car window	0.64%	17.23%	71.45%	10.94%	45.28%
At car trunk	0.58%	73.78%	71.79%	11.09%	51.78%
Exit driver	0.35%	6.68%	50.25%	11.82%	21.12%
Exit passenger	0.34%	79.69%	48.08%	11.59%	26.29%
Run	0.33%	96.15%	75.94%	11.03%	53.35%
Bike	0.33%	85.71%	86.49%	14.37%	75.44%
Enter passenger	0.20%	5.97%	45.82%	13.27%	24.51%
Enter driver	0.12%	5.72%	34.33%	12.3%	20.91%
Motorcycle	0.08%	100%	92.49%	10.76%	71.75%
Mean	7.14%	53.33%	71.58%	21.17%	56.41%
Accuracy		65.03%	88.15%		

[2] K. Aizawa, K. Ishijima and M. Shiina, Summarizing wearable video, in *Proceedings to 2001 International Conference on Image Processing*, vol. 3, IEEE, 2001, 398–401.

[3] J. L. Barron, D. J. Fleet and S. S. Beauchemin, Performance of optical flow techniques, *International Journal of Computer Vision*, **12** (1994), 43–77.

[4] A. L. Bertozzi and A. Flenner, Diffuse interface models on graphs for classification of high dimensional data, *SIAM Review*, **58** (2016), 293–328.

[5] A. L. Bertozzi, X. Luo, A. M. Stuart and K. C. Zygalakis, Uncertainty quantification in graph-based classification of high dimensional data, *SIAM/ASA Journal on Uncertainty Quantification*, **6** (2018), 568–595.

[6] B. L. Bhatnagar, S. Singh, C. Arora, C. Jawahar and K. CVIT, Unsupervised learning of deep feature representation for clustering egocentric actions, in *IJCAI*, 2017, 1447–1453.

[7] J. Budd and Y. V. Gennip, Graph Merriman–Bence–Osher as a semi-discrete implicit euler scheme for graph Allen–Cahn flow, *SIAM Journal on Mathematical Analysis*, **52** (2020), 4101–4139.

[8] T. F. Chan and L. A. Vese, Active contours without edges, *IEEE Transactions on image processing*, **10** (2001), 266–277.

[9] A. G. del Molino, C. Tan, J.-H. Lim and A.-H. Tan, Summarization of egocentric videos: a comprehensive survey, *IEEE Transactions on Human-Machine Systems*, **47** (2017), 65–76.

[10] G. Farnebäck, Two-frame motion estimation based on polynomial expansion, in *Scandinavian Conference on Image Analysis*, Springer, 2003, 363–370.

[11] A. Fathi, J. K. Hodgins and J. M. Rehg, Social interactions: A first-person perspective, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, 1226–1233.

[12] A. Fathi, A. Farhadi and J. M. Rehg, Understanding egocentric activities, in *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, 407–414.

[13] A. Fathi, Y. Li and J. M. Rehg, Learning to recognize daily actions using gaze, in *European Conference on Computer Vision*, Springer, 2012, 314–327.

[14] D. Fortun, P. Bouthemy and C. Kervrann, Optical flow modeling and computation: A survey, *Computer Vision and Image Understanding*, **134** (2015), 1–21.

[15] C. Fowlkes, S. Belongie, F. Chung and J. Malik, Spectral grouping using the Nyström method, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26** (2004), 214–225.

[16] C. Garcia-Cardona, E. Merkurjev, A. L. Bertozzi, A. Flenner and A. G. Percus, Multiclass data segmentation using diffuse interface methods on graphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36** (2014), 1600–1613.

[17] G. Gilboa and S. Osher, Nonlocal operators with applications to image processing, *Multiscale Modeling & Simulation*, **7** (2008), 1005–1028.

[18] B. K. Horn and B. G. Schunck, Determining optical flow, *Artificial Intelligence*, **17** (1981), 185–203.

[19] G. Iyer, J. Chanussot and A. L. Bertozzi, A graph-based approach for feature extraction and segmentation of multimodal images, in *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, 3320–3324.

[20] M. Jacobs, E. Merkurjev and S. Esedoglu, Auction dynamics: A volume constrained MBO scheme, *Journal of Computational Physics*, **354** (2018), 288–310.

[21] K. M. Kitani, T. Okabe, Y. Sato and A. Sugimoto, Fast unsupervised ego-action learning for first-person sports videos, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, 3241–3248.

[22] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, SIAM, Philadelphia, PA, 1995.

[23] D. D. Lee and H. S. Seung, Algorithms for non-negative matrix factorization, in *Advances in Neural Information Processing Systems*, 2001, 556–562.

[24] Y. Li, Z. Ye and J. M. Rehg, Delving into egocentric actions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, 287–295.

[25] B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, in *Proceedings of the 1981 DARPA Image Understanding Workshop*, 1981, 121–130.

[26] X. Luo and A. L. Bertozzi, Convergence of the graph Allen–Cahn scheme, *Journal of Statistical Physics*, **167** (2017), 934–958.

[27] M. Ma, H. Fan and K. M. Kitani, Going deeper into first-person activity recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 1894–1903.

[28] Z. Meng, A. Koniges, Y. H. He, S. Williams, T. Kurth, B. Cook, J. Deslippe and A. L. Bertozzi, OpenMP parallelization and optimization of graph-based machine learning algorithms, in *International Workshop on OpenMP*, Springer, 2016, 17–31.

[29] Z. Meng, E. Merkurjev, A. Koniges and A. L. Bertozzi, Hyperspectral image classification using graph clustering methods, *Image Processing On Line*, **7** (2017), 218–245.

[30] Z. Meng, J. Sánchez, J.-M. Morel, A. L. Bertozzi and P. J. Brantingham, Ego-motion classification for body-worn videos, in *Imaging, Vision and Learning Based on Optimization and PDEs* (eds. X.-C. Tai, E. Bae and M. Lysaker), Springer International Publishing, Cham, 2018, 221–239.

[31] E. Merkurjev, C. Garcia-Cardona, A. L. Bertozzi, A. Flenner and A. G. Percus, Diffuse interface methods for multiclass segmentation of high-dimensional data, *Applied Mathematics Letters*, **33** (2014), 29–34.

[32] E. Merkurjev, T. Kostic and A. L. Bertozzi, An MBO scheme on graphs for classification and image processing, *SIAM Journal on Imaging Sciences*, **6** (2013), 1903–1930.

[33] E. Merkurjev, J. Sunu and A. L. Bertozzi, Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video, in *Image Processing (ICIP), 2014 IEEE International Conference on*, IEEE, 2014, 689–693.

[34] F. Özkan, M. A. Arabaci, E. Surer and A. Temizel, Boosted multiple kernel learning for first-person activity recognition, in *Signal Processing Conference (EUSIPCO), 2017 25th European*, IEEE, 2017, 1050–1054.

[35] H. Pirsiavash and D. Ramanan, Detecting activities of daily living in first-person camera views, in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, 2847–2854.

[36] Y. Poleg, C. Arora and S. Peleg, Temporal segmentation of egocentric videos, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, 2537–2544.

[37] Y. Poleg, A. Ephrat, S. Peleg and C. Arora, Compact CNN for indexing egocentric videos, in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, IEEE, 2016, 1–9.

[38] L. I. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D: Nonlinear Phenomena*, **60** (1992), 259–268.

[39] M. S. Ryoo and L. Matthies, First-person activity recognition: What are they doing to me?, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, 2730–2737.

[40] M. S. Ryoo, B. Rothrock and L. Matthies, Pooled motion features for first-person videos, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, 896–904.

[41] S. Singh, C. Arora and C. Jawahar, Trajectory aligned features for first person action recognition, *Pattern Recognition*, **62** (2017), 45–55.

[42] E. H. Spriggs, F. De La Torre and M. Hebert, Temporal segmentation and activity classification from first-person sensing, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, 17–24.

[43] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, Learning spatiotemporal features with 3D convolutional networks, in *Computer Vision (ICCV), 2015 IEEE International Conference on*, IEEE, 2015, 4489–4497.

[44] Y. Van Gennip and A. L. Bertozzi et al., Γ-convergence of graph Ginzburg-Landau functionals, *Advances in Differential Equations*, **17** (2012), 1115–1180.

[45] Y. Van Gennip, N. Guillen, B. Osting and A. L. Bertozzi, Mean curvature, threshold dynamics, and phase field theory on finite graphs, *Milan Journal of Mathematics*, **82** (2014), 3–65.

[46] X. Wang, L. Gao, J. Song, X. Zhen, N. Sebe and H. T. Shen, Deep appearance and motion learning for egocentric activity recognition, *Neurocomputing*, **275** (2018), 438–447.

[47] L. Zelnik-Manor and P. Perona, Self-tuning spectral clustering, in *Advances in Neural Information Processing Systems*, 2005, 1601–1608.

[48] W. Zhu, V. Chayes, A. Tiard, S. Sanchez, D. Dahlberg, A. L. Bertozzi, S. Osher, D. Zosso and D. Kuang, Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm, *IEEE Transactions on Geoscience and Remote Sensing*, **55** (2017), 2786–2798.

E-mail address: lihao0809@math.ucla.edu
E-mail address: chenhonglin@ucla.edu
E-mail address: mhaberla@calpoly.edu
E-mail address: bertozzi@math.ucla.edu
E-mail address: branting@ucla.edu