# TSC-DL: Unsupervised Trajectory Segmentation of Multi-Modal Surgical Demonstrations with Deep Learning

Adithyavairavan Murali*, Animesh Garg*, Sanjay Krishnan*,
Florian T. Pokorny, Pieter Abbeel, Trevor Darrell, Ken Goldberg

***Abstract*— The growth of robot-assisted minimally invasive surgery has led to sizeable datasets of fixed-camera video and kinematic recordings of surgical subtasks. Temporal segmentation of these trajectories into meaningful contiguous sections is an important first step to facilitate human training and the automation of subtasks. Manual, or supervised, segmentation can be prone to error and impractical for large datasets. We present Transition State Clustering with Deep Learning (TSC-DL), a new unsupervised algorithm that leverages video and kinematic data for task-level segmentation, and finds regions of the visual feature space that mark transition events using features constructed from layers of pre-trained image classification Convolutional Neural Networks (CNNs). We report results on five datasets comparing architectures (AlexNet and VGG), choice of convolutional layer, dimensionality reduction techniques, visual encoding, and the use of Scale Invariant Feature Transforms (SIFT). TSC-DL matches manual annotations with up-to 0.806 Normalized Mutual Information (NMI). We also found that using both kinematics and visual data results in increases of up-to 0.215 NMI compared to using kinematics alone. We also present cases where TSC-DL discovers human annotator errors. Supplementary material, data and code is available at: http://berkeleyautomation.github.io/tsc-dl/**

## I. Introduction

Kinematic and fixed-camera video demonstrations from robot-assisted minimally invasive procedures can be used for surgical skill assessment [6], development of finite state machines [12, 25], learning from demonstration (LfD) [30], and calibration [23]. Surgical tasks are often multi-step procedures that have complex interactions with the environment, and as a result, demonstrations are noisy and may contain superfluous or repeated actions [15]. Temporal segmentation of the demonstrations into meaningful contiguous sections facilitates local learning from demonstrations and salvaging good local segments from inconsistent demonstrations.

Manual annotation provides one approach to segmentation (e.g., [7]); however, as datasets grow, this can become impractical. Human annotators can also be prone to error by missing segments or applying segmentation criteria inconsistently across a dataset. There are a number of recent proposals to algorithmically extract segments [3, 27, 15]. Such algorithms fall into two broad categories: (1) dictionary-based, (2) and unsupervised. Dictionary-based algorithms require a pre-defined vocabulary of primitives and decompose new trajectories in terms of the primitives. However, specific primitives may not cover all of the actions seen in a set of demonstrations, while broad primitives may overlook important transitions.

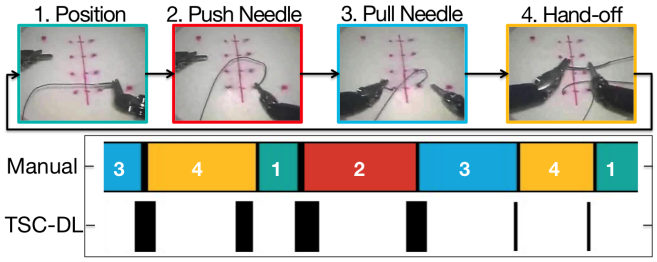


Fig. 1: We apply TSC-DL to a suturing task. Each "throw" of suturing repeats between four steps, and figure illustrates that TSC-DL extracts a segmentation that closely aligns with the manual annotation without supervision.

Unsupervised techniques avoid dependence on a pre-defined set of primitives using generative mixture models for the data, e.g., locally Gaussian segments, and fit trajectories to these models by clustering together locally similar points [3, 15, 17] . A clustering model allows us to detect outliers and inconsistencies (segmentation points that lie in small clusters) without annotations susceptible to human error. It can also give us a natural notion of confidence, through the goodness-of-fit, which can guide the acquisition of demonstrations of segments that require more data. While unsupervised segmentation has been widely studied in the context of kinematic data, increasingly, fixed-camera video is also available.

Visual features can provide information about the state of manipulated objects [15, 27], and trajectory information when there is state-dependent sensor noise in kinematic data [23]. Existing work uses hand-tuned features [15], poses for all objects in the workspace via AR markers [27], or motion capture markers [18]. We explore relaxing these constraints by applying recent results in Deep Learning to learn a visual representation that generalizes across tasks.

In this paper, we significantly extend our prior segmentation work, Transition State Clustering [15], with automatically constructed visual features using *deep* convolutional neural networks (CNNs). Computer vision frameworks such as CAFFE [10] can leverage recent advances in CNNs [16, 10, 22] through pre-trained models (on large corpora of natural images). CNNs learn expressive but general feature representations that transfer across domains allowing us to take advantage of the models without having to acquire a large number of examples. Transition State Clustering segments demonstrations by learning switched linear dy-

* These authors contributed equally to the paper
EECS & IEOR, University of California, Berkeley CA USA; {adithya_murali, animesh.garg, sanjaykrishnan, ftpokorny, pabbeel, trevor, goldberg}@berkeley.edu

Preprint submitted to 2016 IEEE International Conference on Robotics and Automation. Recieved September 1, 2016.

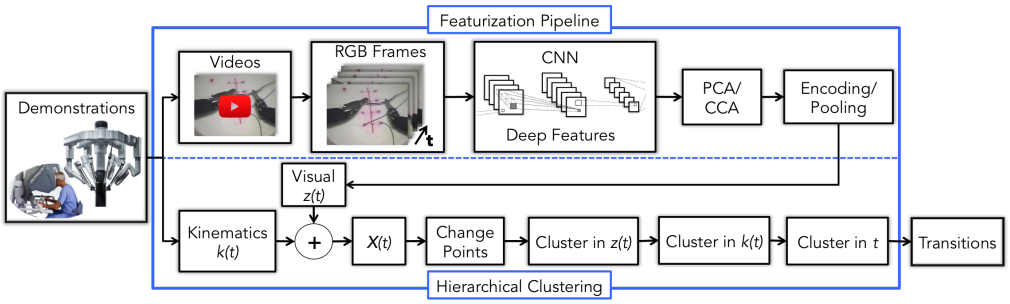


Figure 2: We use a visual processing pipeline with deep features to construct a trajectory of high-dimensional visual states $z(t)$. We concatenate encoded versions of these features with kinematics and apply hierarchical clustering to find segments.

namical systems and using clustering to identify regions of the state-space associated with switching events. TSC applies a Dirichlet Process Gaussian mixture hierarchically first clustering transition states spatially and then temporally. Our prior results suggested that augmenting the state-space with hand-tuned visual features could significantly improve accuracy. In the present paper, we extract visual features using filters derived from layers of pre-trained CNNs applied to frames of video recordings and call this new algorithm Transition State Clustering with Deep Learning (TSC-DL). This constructs a high-dimensional trajectory that augments the kinematic state-space (Figure 2).

We study CNN features for unsupervised temporal trajectory segmentation on five datasets: (1) a synthetic 4 segment example, (2) JIGSAWS surgical needle passing, (3) JIGSAWS surgical suturing, (4) toy plane assembly by the PR2, and (5) Lego assembly by the PR2. On the synthetic example, we find that TSC-DL recovers the 4 underlying segments in the presence of partial state observation (one kinematic state hidden), control noise, and sensor noise. TSC-DL is an unsupervised algorithm that consistently applies segmentation criteria derived from linear dynamical regimes. We compare this criteria with manual annotations when available. On real datasets, we find that TSC-DL matches the manual annotation with up to 0.806 normalized mutual information. Our results also suggest that including kinematics and vision results in increases of up-to 0.215 NMI over kinematics alone. We demonstrate the benefits of using an unsupervised approach by presenting examples where TSC-DL discovers unlabeled segments due to human annotator error (as shown in Figure 5), and can learn across demonstrations with widely varying operator skill levels (as shown in Table II).

## II. Related Work

### A. Learning From Demonstrations

One model for learning from demonstrations uses segmentation to discretize action spaces (skill-learning) which allows for efficient learning of complex tasks [9, 28]. This line of work largely focuses on pre-defined primitives. Niekum et al. [26] proposed an unsupervised extension to the motion primitive model by learning a set of primitives using the Beta-Process Autoregressive Hidden Markov Model (BP-AR-HMM). The work by Niekum et al. does incorporate visual information, however, it does not use visual information to actually find segments. Post segmentation, Niekum et al. uses AR markers to estimate poses of all of the objects in the workspace. The segments, discovered with kinematics alone, are then specified in each objects reference frame. When the objects are then moved, the trajectory can be transferred using a Dynamic Motion Primitive model.

Calinon et al. [2, 4] characterizes segments from demonstrations as skills that can be used to parametrize imitation learning. In this line work, the authors apply Gaussian Mixture Models (GMMs) to cluster observations from the same mixture component. A number of other works have leveraged this model for segmentation e.g., [13, 14, 33]. As we will later describe, Gaussian Mixture Models have a duality with switched linear dynamical systems [24]. Calinon et al. [4] uses segmentation to teach a robot how to hit a moving ball. They use visual features through a visual trajectory tracking of a ball. The visual sensing model in Calinon et al. is tailored to the ball task, and in this paper, we use a set of general visual features for all tasks.

### B. Surgical Robotics

Other surgical robotics works have largely studied the problem of supervised segmentation using either segmented examples or a pre-defined dictionary of motions (similar to motion primitives) [36, 35, 19, 42, 29].

### C. Visual Gesture Recognition

A number of recent works, attempt to segment human motions from videos [8, 34, 40, 11, 38, 37]. Tang et al. and Hoai et al. proposed supervised models for human action segmentation from video. Building on the supervised models, there are a few unsuperivsed models for segmentation of human actions: Jones et al.[11], Yang et al. [40], Di Wu et al. [38] , and Chenxia Wu et al. [37]. Jones et al. [11] restricts their segmentation to learning from two views of the dataset (i.e., two demonstrations). Yang et al. [40] and Wu et al. [37] use k-means to learn a dictionary of primitive motions, however, in prior work, we found that transition state clustering outperforms a standard k-means segmentation approach. In fact, the model that we propose is complementary to these works and would be a robust drop-in-replacement for the k-means dictionary learning step [15]. The approach taken by Di Wu et al. is to parametrize human actions using a skeleton model, and they learn the parameters to this skeleton model using a deep neural network. In this work, we explore using generic deep visual features for robotic segmentation without requiring task-specific optimization such as skeleton or action models using in human action recognition.

Preprint submitted to 2016 IEEE International Conference on Robotics and Automation. Recieved September 1, 2016.

### D. Deep Features in Robotics

Robotics is increasingly using deep features for visual sensing. For example, Lenz et al. uses pre-trained neural networks for object detection in grasping [20] and Levine et al. [21] fine-tune pre-trained CNNs for policy learning. For this reason, we decide to explore methodologies for using deep features in segmentation as well. We believe that segmentation is an important first step in a number of robot learning applications, and the appropriate choice of visual features is key to accurate segmentation. We present an initial exploration of different visual featurization strategies and segmentation accuracy.

## III. Prior Work: Transition State Clustering

The Transition State Clustering algorithm (TSC), learns clusters of states that mark dynamical regime transitions.

### A. Learning Transition States

Let $\mathcal{D} = \{d_1, ..., d_k\}$ be the set of demonstrations where each $d_i$ is a trajectory of fully observed robot states and each state is a vector in $\mathbb{R}^d$. TSC-DL finds a set of transition states clusters, which are states across demonstrations associated with the same transition event, reached by a fraction of at least $\rho \in [0,1]$ of the demonstrations. We assume that demonstrations are recorded in a global fixed coordinate frame and visually from a fixed point of view and they are *consistent* with at least one transition state cluster.

Transitions are defined in terms of switched linear dynamical systems (SLDS). We model each demonstration as a SLDS:

$$\mathbf{x}(t+1) = A_i\mathbf{x}(t) + W(t) \; : A_i \in \{A_1, ..., A_k\}$$

In this model, transitions between regimes $\{A_1, ..., A_k\}$ are instantaneous where each time $t$ is associated with exactly one dynamical system matrix $1, ..., k$. *Transition state* is defined as the last state before a dynamical regime transition in each demonstration. A Transition state is the state $\mathbf{x}(t)$ at time $t$, such that $A(t) \neq A(t+1)$.

Suppose there was only one regime, then we obtain a linear regression problem:

$$\arg\min_A \|AX_t - X_{t+1}\|$$

where $X_t$ and $X_{t+1}$ are matrices with $T-1$ columns of $[\mathbf{x}(1), \mathbf{x}(2), ..., \mathbf{x}(T-1)]$, and $[\mathbf{x}(2), \mathbf{x}(3), ..., \mathbf{x}(T)]$ respectively. Moldovan et al. [24] proves that fitting a Jointly Gaussian model to $n(t) = \binom{\mathbf{x}(t+1)}{\mathbf{x}(t)}$ is equivalent to Bayesian Linear Regression. We use Dirichlet Process Gaussian Mixture Models (DP-GMM) to learn the regimes without have to set the number of regimes in advance. Each cluster learned signifies a different regime, and co-linear states are in the same cluster. To find transition states, we move along a trajectory from $t = 1, ..., t_f$, and find states at which $n(t)$ is in a different cluster than $n(t+1)$. These points mark a transition between clusters (i.e., transition regimes).

### B. Learning Transition State Clusters

A *transition state cluster* is defined as a clustering of the set of transition states across all demonstrations; partitioning these transition states into $m$ non-overlapping similar groups: $\mathcal{C} = \{C_1, C_2, ..., C_m\}$ We model the states at transition states as drawn from a GMM model: $x(t) \sim N(\mu_i, \Sigma_i)$. Then, we can apply the DP-GMM again to cluster the state vectors at the transition states. Each cluster defines an ellipsoidal region of the state-space space.

Each of these clusters will have constituent vectors where each $n(t)$ belongs to a demonstration $d_i$. Clusters whose constituent vectors come from fewer than a fraction $\rho$ demonstrations are *pruned*.

*Given a consistent set of demonstrations, the algorithm finds a sequence of transition state clusters reached by at least a fraction $\rho$ of the demonstrations.*

## IV. Transition State Clustering With Deep Learning

We extend our prior work with states defined with visual features, and present the TSC-DL algorithm in Algorithm 1.

### A. Visual Features

We define an augmented state space $\mathbf{x}(t) = \binom{k(t)}{z(t)}$, where $k(t) \in \mathbb{R}^k$ are the kinematic features and $z(t) \in \mathbb{R}^v$ are the visual features. We use layers from a pre-trained Convolutional Neural Network (CNNs) to derive the features frame-by-frame. CNNs are increasingly popular for image classification and as a result a number of image classification CNNs exist that are trained on millions of natural images. Intuitively, CNNs calssify based on aggregations (pools) of hierarchical convolutions of the pixels. Removing the aggregations and the classifiers, results in convolutional filters which can be used to derive generic features.

We found that use of these features requires a number of pre-processing and post-processing steps; in addition to a number of design choices within the CNN such as which convolutional layer(s) to use for composing the visual featurization.

*1) Pre-processing:* CNNs are trained on static images for image classification, and as a result their features are optimized for identifying salient edges and colors. However, they do not capture temporal features and do differentiate the between robot and workspace features. Furthermore, since we aggregate across demonstrations, we need to ensure that these features are largely consistent. To reduce variance due to extraneous objects and lighting changes, we crop each video to capture the only the relevant workspace where robot manipulation occurs. Then, the videos are rescaled to 640x480 along with down-sampling to 10 frames per second for computational efficiency. All frames in the videos are normalied to a *zero* mean in all channels (RGB) individually [16, 32]. All of pre-processing were preformed with open source `ffmpeg` library.

**Algorithm 1: TSC-DL:** Transition State Clustering with Deep Learning

**Data**: Set of demonstrations:$\mathcal{D}$
**Parameters**: pruning factor ($\rho$), time window ($w$), PCA dim ($d_p$), hyperparams ($\alpha_1, \alpha_2, \alpha_3, \alpha_4$)
**Result**: Set of Predicted Transitions $\mathcal{T}_i,\ \forall d_i \in \mathcal{D}$

1 **foreach** $d_i \in \mathcal{D}$ **do**
2 $\quad z_i \leftarrow$ VisualFeatures $(d_i, w, d_p)$
3 $\quad k_i \leftarrow$ KinematicFeatures $(d_i, w)$
4 $\quad \mathbf{x}_i(t) \leftarrow \binom{\mathbf{k_i}(t)}{\mathbf{z_i}(t)}\ \forall t \in \{1, \ldots, T_i\}$
5 $\quad \bar{x}_i(t) \leftarrow \left[\mathbf{x}(t+1)^T,\ \mathbf{x}(t)^T,\ \mathbf{x}(t-1)^T\right]^T,\ \forall t$
6 $\quad N \leftarrow \left[N^T, \bar{x}_i(1)^T, \ldots, \bar{x}_i(T_i)^T\right]^T$
`// Cluster to get Change Points`
7 $\mathcal{C}^{CP} = \text{DPGMM}(N, \alpha_0)$ `// $\alpha_0$ is hyperparameter`
8 **foreach** $N(n) \in \mathcal{C}_i^{CP},\ N(n+1) \in \mathcal{C}_j^{CP}, i \neq j$ **do**
9 $\quad CP \leftarrow CP \cup \{N(n)\}$
`// Cluster over Kinematic Feature Subspace`
10 $\mathcal{C}_1 = \text{DPGMM}(CP, \alpha_1)$ `// $\mathcal{C}_1$: set of clusters`
11 **foreach** $C_k \in \mathcal{C}_1$ **do**
12 $\quad CP(C_k) \leftarrow \{CP(n) \in C_k, \forall n \in \{1, \ldots, |CP|\}\}$
$\quad$ `// Cluster over Visual Feature Subspace`
13 $\quad \mathcal{C}_{k2} \leftarrow \text{DPGMM}(CP(C_k), \alpha_2)$
14 $\quad$ **foreach** $C_{kk'} \in \mathcal{C}_{k2}$ **do**
15 $\quad\quad$ **if** $\sum_{d_i} \mathbf{1}\Big(\sum_{n:N(n)\in d_i} \mathbf{1}(CP(n) \in C_{kk'}) \geq 1\Big) \leq \rho|\mathcal{D}|$ **then**
16 $\quad\quad\quad \mathcal{C}_{k2} \leftarrow \mathcal{C}_{k2} \setminus \{C_{kk'}\}$ `// Cluster Pruning`
$\quad\quad$ `// collect intra-cluster transitions` $\forall\ d_i$
17 $\quad\quad \forall\ d_i \in \mathcal{D}$ **do** $T_i \leftarrow T_i \cup \{CP(n) \in C_{kk'}, \forall n : N(n) \in d_i\}$
`// Cluster over time to predict Transition Windows`
18 **foreach** $d_i \in \mathcal{D}$ **do**
19 $\quad$ Repeat steps 1-17 for $\mathcal{D}' = \mathcal{D} \setminus d_i$
20 $\quad \mathtt{T}_j \leftarrow \mathtt{T}_j \cup T_j^{(i)},\ \{\forall\ j : d_j \in \mathcal{D}'\}$ `// $T_j^{(i)}$: ith iteration`
21 **foreach** $d_i \in \mathcal{D}$ **do** $\mathcal{T}_i \leftarrow DPGMM(\mathtt{T}_i, \alpha_4)$
22 **return** $\mathcal{T}_i,\ \forall d_i \in \mathcal{D}$

*2) Visual Featurization:* Once the images were pre-processed, we applied the convolutional filters from the pre-trained neural networks. Yosinski et al. note that CNNs trained on natural images exhibit roughly the same Gabor filters and color blobs on the first layer [41]. They established that earlier layers in the hierarchy give more general features while later layers give more specific ones. In our experiments, we explore the level of generality of features required for segmentation. In particular, we explore two architectures designed for image classification task on natural images: (a) **AlexNet:** Krizhevsky et al. proposed multilayer (5 in all) a CNN architecture [16], and (b) **VGG:** Simoyan et al. proposed an alternative architecture termed VGG (acronym for Visual Geometry Group) which increased the number of convolutional layers significantly (16 in all) [32]. We also compare these features to other visual featurization techniques such as SIFT and SURF for the purpose of task segmentation using TSC-DL.

*3) Post-Processing: Encoding:* After constructing these features, the next step is encoding the results of the convolutional filter into a vector $z(t)$. We explore three encoding techniques: (1) Raw values, (2) Vector of Locally Aggregated Descriptors (VLAD) [1], and (3) Latent Concept Descriptors (LCD) [39].

*4) Post-Processing: Dimensionality Reduction:* After encoding, we feed the CNN features $z(t)$, often in more than 50K dimensions, through a dimensionality reduction process to boost computational efficiency. This also balances the visual feature space with a relatively small dimension of kinematic features ($< 50$). Moreover, GMM-based clustering algorithms usually converge to a local minima and very high dimensional feature spaces can lead to numerical instability or inconsistent behavior. We explore multiple dimensionality reduction techniques to find desirable properties of the dimensionality reduction that may improve segmentation performance. In particular, we analyze Gaussian Random Projections (GRP), Principal Component Analysis (PCA) and Canonical Correlation Analysis (CCA) in Table I. GRP serves as a baseline, while PCA is used based on widely application in computer vision [39]. We also explore CCA as it finds a projection that maximizes the visual features correlation with the kinematics.

### B. Robust Temporal Clustering

To reduce over-fitting and build a confidence interval as a measure of accuracy over the temporal localization of transitions, we use a Jackknife estimate. It is calculated by aggregating the estimates of each $N-1$ estimate in the sample of size $N$. We iteratively hold out one out of the $N$ demonstrations and apply TSC-DL to the remaining demonstrations. Then, over $N-1$ runs of TSC-DL, $N-1$ predictions are made $\forall d_i \in \mathcal{D}$. We temporally cluster the transitions across $N-1$ predictions, to estimate final transition time mean and variance $\forall d_i \in \mathcal{D}$. This step is illustrated in step 20-21 of Algorithm 1.

### C. Sliding Window States

To better capture hysteresis and transitions that are not instantaneous, in this current paper, we use rolling window states where each state $\mathbf{x}_{(t)}$ is a concatenation of $T$ historical states. We varied the length of temporal history $T$ and evaluated performance of the TSC-DL algorithm for the suturing task using metric defined in Section V-A. We empirically found a sliding window of size 3, i.e., $\mathbf{x}_{(t)} = \binom{\mathbf{k}(t)}{\mathbf{z}(t)}$, as the state representation led to improved segmentation accuracy while balancing computational effort.

### D. Skill-Weighted Pruning

Demonstrators may have varying skill levels leading to increased outliers, and so we extend our outlier pruning to include weights. Let, $w_i$ be the weight for each demonstration $d_i \in \mathcal{D}$, such that $w_i \in [0,1]$ and $\hat{w}_i = \frac{w_i}{\sum w_i}$. Then a cluster $C_{kk'}$ is pruned if it does not contain change points $CP(n)$ from at least $\rho$ fraction of demonstrations. This converts to:

$$\sum_{d_i} \hat{w}_i \mathbf{1}\Big(\sum_{n:N(n)\in d_i} \mathbf{1}(CP(n) \in C_{kk'}) \geq 1\Big) \leq \rho$$

| | GRP | PCA | CCA |
|---|---|---|---|
| SIFT | - | 0.443±0.008 | - |
| AlexNet conv3 | 0.559±0.018 | 0.600±0.012 | 0.494±0.006 |
| AlexNet conv4 | 0.568±0.007 | 0.607±0.004 | 0.488±0.005 |
| AlexNet pool5 | 0.565±0.008 | 0.599±0.005 | 0.486±0.012 |
| VGG conv5_3 | 0.571±0.005 | **0.637±0.009** | 0.494±0.013 |
| VGG LCD-VLAD | 0.506±0.001 | 0.534±0.011 | 0.523±0.010 |
| AlexNet LCD-VLAD | 0.517±0.001 | 0.469±0.027 | 0.534±0.018 |

TABLE I: The silhouette score for each of the techniques and dimensionality reduction schemes on a subset of suturing demonstrations (5 expert examples). We found that PCA (100 dims) applied to VGG conv5_3 maximizes silhouette score

## V. Experiments

### A. Evaluation Metrics

It is important to note that TSC-DL is an unsupervised algorithm that does not use labeling. Therefore, we evaluate TSC-DL both intrinsically (without labels) and extrinsically (against human annotations).

**Intrinsic metric:** The goal of the intrinsic metric is compare the performance of different featurization techniques, encodings, and dimensionality reduction within TSC-DL without reference to external labels. This score is not meant to be an absolute metric of performance but rather a relative measure. The intrinsic metric we use measures the "tightness" of the transition state clusters. This metric is meaningful since we require that each transition state cluster contains transitions from a fraction of at least $\rho$ of the demonstrations, the tightness of the clusters measures how well TSC-DL discovers regions of the state space where transitions are grouped together. This is measured with the mean *Silhouette Score* (denoted by ss), which is defined as follows for each transition state $i$:

$$\text{ss}(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}, \quad \text{ss}(i) \in [-1, 1]$$

if transition state $i$ is in cluster $C_j$, $a(i)$ is defined the average dissimilarity of point $i$ to all points in $C_j$, and $b(i)$ the dissimilarity with the closest cluster measured as the minimum mean dissimilarity of point $i$ to cluster $C_k$, $k \neq j$. We use $L_2$-norm as the dissimilarity metric and rescale ss $\in [0, 1]$ for ease of comparison.

**Extrinsic metric:** To calculate an absolute measure of similarity of TSC-DL predictions $\mathcal{T}$ with respect to manual annotations $\mathcal{L}$, we use *Normalized Mutual Information* (NMI) which measures the alignment between two label

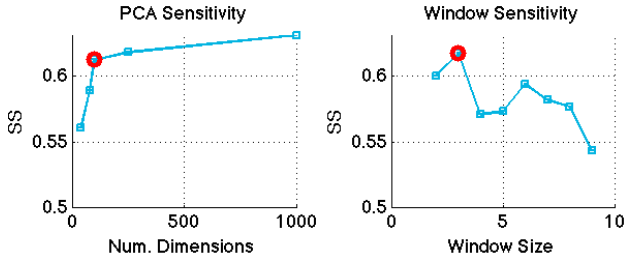


Fig. 3: We evaluate the sensitivity of two hyperparameters set in advance: number of PCA dimensions and sliding window size. The selected value is shown in red double circles.

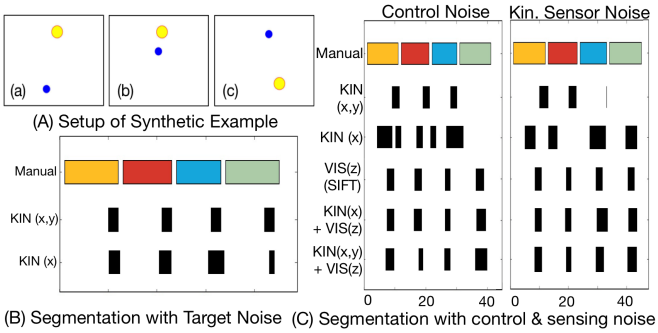


Fig. 4: (A) The figure shows a 2D synthetic example with a moving point in blue and target in yellow. The robot moves to the target in a straight line in discrete steps, and a new target appears. (B) Segmentation results for repeated demonstrations with variance in target position. (C) Segmentation under control noise, sensor noise, and partial observeration.

assignments. NMI is equal to the KL-divergence of the joint distribution with the product distribution of the marginals; intuitively, the distance from pairwise statistical independence. NMI score lies in $[0,\ 1]$, where 0 indicates independence while 1 is perfect matching. It is defined as,

$$NMI(\mathcal{T}, \mathcal{L}) = \frac{I(\mathcal{T}, \mathcal{L})}{\sqrt{H(\mathcal{T})H(\mathcal{L})}}, \quad NMI(\mathcal{T}, \mathcal{L}) \in [0, 1]$$

### B. Evaluation of Visual Featurization

In our first experiment, we explore different visual featurization, encoding, and dimensionality reduction techniques. We applied TSC-DL to our suturing experimental dataset, and measured the silhouette score of the resulting transition state clusters. Table I describes the featurization techniques on the vertical axis and dimensionality reduction techniques on the horizontal axis. Our results suggest that on this dataset features extracted from the pre-trained CNNs resulted in tighter transition state clusters compared to SIFT features with a 3% lower ss than the worst CNN result. Next, we found that features extracted with the VGG architecture resulted in the highest ss with a 3% higher ss than the best AlexNet result. We also found that PCA for dimensionality reduction gave the best ss performance 7% higher than the best GRP result and 10% higher than best CCA result. Because CCA finds projections of high correlation between the kinematics and video, we believe that CCA discard features informative features resulting in reduced clustering performance. We note that neither of the encoding schemes, VLAD or LCD significantly improve the ss.

There are two hyper-parameters for TSC-DL which we set empirically: sliding window size (T = 3), and the number of PCA dimensions (k = 100). In Figure 3, we show a sensitivity plot with the ss as a function of the parameter. We calculated the ss using the same subset of the suturing dataset as above and with the VGG conv5_3 CNN. We found that T = 3 gave the best performance. We also found that PCA with k = 1000 dimensions was only marginally better than k = 100 yet required >30 mins to run. For computational reasons, we selected k = 100.

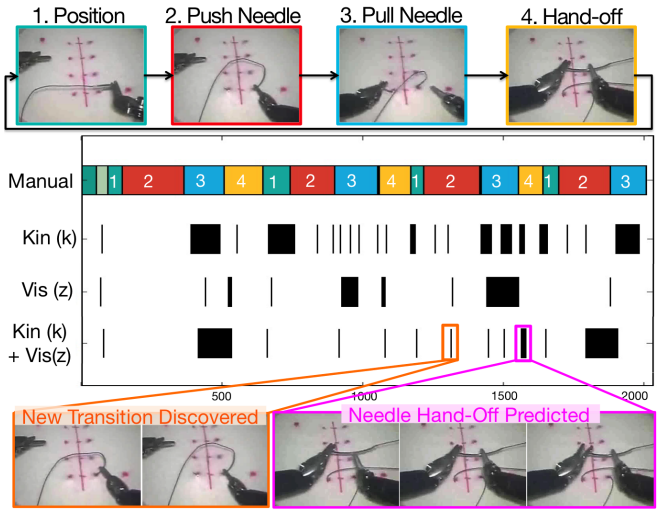


Fig. 5: The first row shows a manual segmentation of the suturing task in 4 steps: (1) Needle Positioning, (2) Needle Pushing, (3) Pulling Needle, (4) Hand-off. TSC-DL extracts many of the important transitions without labels and also discovers un-labled transition events.

### C. End-to-End Evaluation

For all subsequent experiments on real data, we use a pre-trained VGG CNN conv5_3 and encoded with PCA with 100 dimensions.

**1. Synthetic Example:** We first evaluate TSC-DL on a synthetic example consisting of 4 linear segments (Figure Figure 4). A point robot on a plane moves towards a target in a straight line, once it reaches the target, the target moves to a new location. This process is repeated four times. We use the simulation to generate image data and kinematics data. Figure 4 (b) shows the results of unsupervised segmentation using only kinematics data ($\binom{x(t)}{y(t)}$). When the state is full observed (i.e., we have both x and y positions), we accurately recover 4 segments with kinematics alone. If we hide one of the states, we see that we can still recover the 4 segments. In this example, when there is no noise on the kinematics, one dimension alone is enough to learn the segmentation.

Next in Figure 4, we make this scenario more complex by introducing control noise: $x(t+1) = x(t) + u(t) + v$, where $v \sim \mathcal{N}(0, d_1)$ where $d_1 = 0.25$ We find that when there is control noise, partial observed kinematics can lead to erroneous segments even in this synthetic example. We use this example to demonstrate the importance of visual features. If we add visual features (using SIFT since these are not natural images), we find that we can mitigate the problems caused by noise and partial observability. Finally, we repeat the above experiment for kinematic sensor noise in the system $\hat{x}(t) = x(t) + v$, where $v \sim \mathcal{N}(0, d_2)$ where $d_2 = d_2 = 0.25$. We note that only the kinematics is corrupted with noise, while the vision sees a straight trajectory.

**2. Suturing:** We apply our method to a subset of JIGSAWS dataset[6] consisting of surgical task demonstrations under tele-operation using the da Vinci surgical system. The dataset was captured from eight surgeons with different levels of skill, performing five repetitions each of suturing and needle passing. We use 39 demonstrations of a 4 throw suturing task (Figure 5) and we manually annotate these demonstrations for reference. We apply TSC-DL to kinematics and vision alone respectively and then the combination. With combined kinematics and vision, TSC-DL learns many of the important segments identified by annotation in [6]. After learning the segmentation, we apply it to a representative trajectory and show that we accurately recover 10/15 transitions annotated by our manual labeling.

Upon further investigation of the false positives, we found that they corresponded to crucial actions missed by our labeling. For example, TSC-DL discovers that a crucial needle repositioning step where many demonstrators penetrate and push-through the needle in two different motions. TSC-DL finds segments that correspond to linear dynamical systems, and applies this criterion consistently. Human annotators may miss subtle transitions such as quick two-step motions.

| | | K | Z | K+Z |
|---|---|---|---|---|
| Silhouette Score – Intrinsic Evaluation | | | | |
| Suturing | E | 0.630±0.014 | 0.576±0.018 | 0.654±0.065 |
| | E+I | 0.550±0.014 | 0.548±0.015 | 0.716±0.046 |
| | E+I+N | 0.518±0.008 | 0.515±0.021 | 0.733±0.056 |
| Needle Passing | E | 0.524±0.004 | 0.609±0.010 | 0.716±0.097 |
| | E+I | 0.521±0.006 | 0.536±0.013 | 0.666±0.067 |
| | E+I+N | 0.513±0.007 | 0.552±0.011 | 0.557±0.010 |
| NMI Score – Extrinsic evaluation against manual labels | | | | |
| Suturing | E | 0.516 ± 0.026 | 0.266 ± 0.025 | 0.597 ± 0.096 |
| | E+I | 0.427 ± 0.053 | 0.166 ± 0.057 | 0.646 ± 0.039 |
| | E+I+N | 0.307 ± 0.045 | 0.157 ± 0.022 | 0.625 ± 0.034 |
| Needle Passing | E | 0.287 ± 0.043 | 0.222 ± 0.029 | 0.565 ± 0.037 |
| | E+I | 0.285 ± 0.051 | 0.150 ± 0.048 | 0.471 ± 0.023 |
| | E+I+N | 0.272 ± 0.035 | 0.186 ± 0.034 | 0.385 ± 0.092 |

TABLE II: Comparison of TSC-DL performance on Suturing and Needle Passing Tasks. We compare the prediction performance by incrementally adding demonstrations from Experts (E), Intermediates (I), and Novices (N) respectively to the dataset.

**3. Needle Passing:** Next, we applied TSC-DL to 28 demonstrations of the needle passing task. These demonstrations were annotated in [6]. Table II lists quantitative results for both needle passing and suturing with both ss and NMI agreement with the human labels. Demonstrations from the JIGSAWS dataset were annotated with the skill-level of the demonstrators (Expert (E), Intermediate (I), and Novice (I)). In our surgical datasets, where a mix of skill levels were used, we applied weighted outlier pruning to account for increased outliers amongst novice demonstrators. We used a weight of 5 for experts, 2 for intermediates, and 1 for novices, and these weights were determined empirically using analysis of task time in the datasets (the max novice time was 5x slower than the expert time). We present results with weighting on the mixed groups and without weighting on experts only. We find that in both surgical datasets, kinematics and vision gives improved performance (intrinsically and extrinsically) than either set of features alone. This emphasizes the benefits of using TSC-DL as it takes advantage of multimodal trajectory. Also, we see a strong dependence on the operator's skill level. The results are very different when applied to just experts compared to

Preprint submitted to 2016 IEEE International Conference on Robotics and Automation. Recieved September 1, 2016.

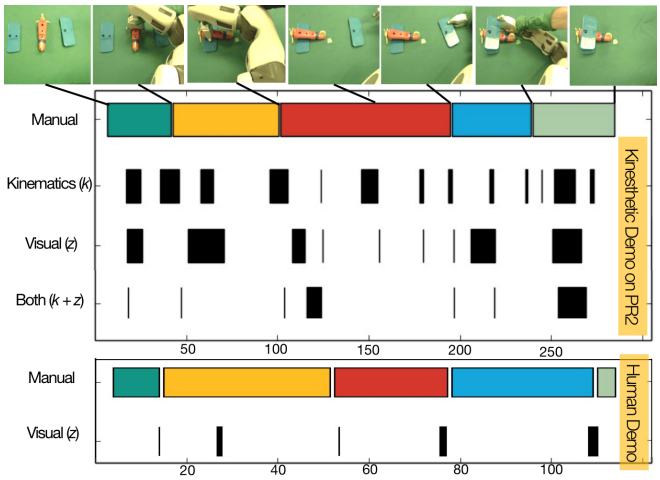


Fig. 6: We compare TSC-DL on 12 kinesthetic demonstrations (top) and 8 human demonstrations (bottom). No kinematics were available for the human demonstrations. We illustrate the segmentation for an example demonstration of each. Our manual annotation of the task has 5 steps and TSC-DL recovers this structure separately for both Kinesthtic demos on PR2 and Human demos with the same visual features.

all skill levels. For kinematics and vision alone, the intrinsic metric drops as we add less skilled demonstrations. However, when we include kinematics and vision we see that the metric increases fur the suturing dataset. We will investigate this in future work, but we speculate this has to do with sampling error, i.e., adding more data makes the segmentation more accurate.

**4. PR2: Legos and Toy Plane Assembly:** In our next experiment, we explore segmenting a multi-step assembly task using (1) large *Lego* blocks and (2) toy *Plane* from the YCB dataset [5]. We demonstrate that TSC-DL applies generally outside of surgical robotics. We collect 8 kinesthetic demonstrations for each task through kinesthetic demonstrations of the task on the PR2 robot. Figure 6 illustrates the segmentation for the plane assembly task. We find the plane assembly task using kinematics or vision alone results in a large number of segments. The combination can help remove spurious segments restricting our segments to those tranistions that occur in most of the demonstrations–agreeing in similarity both kinematically and visually.

**5. Human Demonstration of Toy Plane Assembly:** We extend the toy plane assembly experiment to collect 8 demonstrations each from two human users. These examples only have videos and no kinematic information. We note that there was a difference between users in the grasping location of fuselage. The results of TSC-DL performance are summarised in Table III. An example of toy plane assembly by both robot and human is qualitatively compared in Figure 6. This emphasizes on the benefits of TSC-DL, namely, that we do not tune the features to any specific robot or task. These are general visual features that can apply broadly even when a human is performing demonstrations. We omit a visualization of the results for the Lego assembly, however, we summarize the results quantitatively in Table III.

| | K | Z | K+Z |
|---|---|---|---|
| Silhouette Score – Intrinsic Evaluation | | | |
| Lego (Robot) | 0.653±0.003 | 0.644±0.026 | 0.662±0.053 |
| Plane (Robot) | 0.741±0.011 | 0.649±0.007 | 0.771±0.067 |
| Plane (Human 1) | – | 0.601 ±0.010 | – |
| Plane (Human 2) | – | 0.628 ±0.015 | – |
| NMI Score – Extrinsic evaluation against manual labels | | | |
| Lego (Robot) | 0.542 ± 0.058 | 0.712 ± 0.041 | 0.688 ± 0.037 |
| Plane (Robot) | 0.768 ± 0.015 | 0.726 ± 0.040 | 0.747 ± 0.016 |
| Plane (Human 1) | – | 0.726 ± 0.071 | – |
| Plane (Human 2) | – | 0.806 ± 0.034 | – |

TABLE III: Plane and Lego Assembly Tasks. Both tasks show improvements in clustering and prediction accuracy using multi-modal data as compared to either modality. Further, only vision ($Z$) is available for human demos of the plane assembly task. Comparable segmentation results are obtained using only video input for human demos. *Higher* Silhoutte Scores and NMI scores are better, respectively.

## VI. Conclusion

In this work, we propose TSC-DL extending the Transition State Clustering algorithm to include visual feature extraction from pre-trained CNNs. In our experiments, we apply TSC-DL to five datasets: (1) a synthetic 4 segment example, (2) JIGSAWS surgical needle passing, (3) JIGSAWS surgical suturing, (4) toy plane assembly by the PR2, and (5) Lego assembly by the PR2. On the synthetic example, we find that TSC-DL recovers the 4 underlying segments in the presence of partial state observation (one kinematic state hidden), control noise, and sensor noise. On real datasets, we find that TSC-DL matches the manual annotation with up to 0.806 NMI. Our results also suggest that including kinematics and vision results in increases of up-to 0.215 NMI over kinematics alone. We demonstrated the benefits of an unsupervised approach with examples in which TSC-DL discovers inconsistencies such as segments not labeled by human annotators, and apply TSC-DL to learn across demonstrations with widely varying operator skill levels. We also validated surgical results in a different domain with demonstrations of assembly tasks with the PR2 and human-only demonstrations.

## VII. Future Work

Our results suggest a number of important directions for future work. First, we plan to apply the results from this paper to learn transition conditions for finite state machines for surgical subtask automation. The CNNs applied in this work are optimized for image classification of natural images and not the images seen in surgery. In future work, we will explore training CNNs from scratch to identify features directly from both pixels and kinematics, or fine-tuning existing networks. Next, our current visual featurization is applied frame-by-frame. This approach misses transient events that span frames. We will explore applying convolutional features that capture temporality such as 3D convolutional layers and optical flow [31]. We are also interested in exploring using recurrent neural networks and variational autoencoders to perform and end-to-end neural network implementation of TSC-DL.

Preprint submitted to 2016 IEEE International Conference on Robotics and Automation. Recieved September 1, 2016.

**Acknowledgement:** This work is supported in part by a seed grant from the UC Berkeley CITRIS, and by the U.S. NSF Award IIS-1227536: Multilateral Manipulation by Human-Robot Collaborative Systems. NVIDIA for computing equipment grants; Andy Chou and Susan Lim for developmental grants; Greg Hager for datasets; Sergey Levine, Katerina Fragkiadaki, Greg Kahn, Yiming Jen, and Zhongwen Xu for discussions; Jeff Mahler and Michael Laskey for draft reviews. Supplementary material is available at: http://berkeleyautomation.github.io/tsc-dl/

## REFERENCES

[1] R. Arandjelovic and A. Zisserman, "All about vlad," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1578–1585.

[2] S. Calinon, "Skills learning in robots by interaction with users and environment," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2014 11th International Conference on*. IEEE, 2014, pp. 161–162.

[3] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation," *Robotics & Automation Magazine, IEEE*, vol. 17, no. 2, pp. 44–54, 2010.

[4] S. Calinon, E. L. Sauser, A. G. Billard, and D. G. Caldwell, "Evaluation of a probabilistic approach to learn and reproduce gestures by imitation," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2671–2676.

[5] B. Çalli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols," *CoRR, abs/1502.03143*, 2015.

[6] Y. Gao, S. Vedula, C. Reiley, N. Ahmidi, B. Varadarajan, H. Lin, L. Tao, L. Zappella, B. Bejar, D. Yuh, C. Chen, R. Vidal, S. Khudanpur, and G. Hager, "The jhu-isi gesture and skill assessment dataset (jigsaws): A surgical activity working set for human motion modeling," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2014.

[7] W. Han, S. Levine, and P. Abbeel, "Learning compound multi-step controllers under unknown dynamics," in *International Conference on Intelligent Robots and Systems 2016*. IEEE, 2016.

[8] M. Hoai, Z.-Z. Lan, and F. De la Torre, "Joint segmentation and classification of human actions in video," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011.

[9] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Neural Information Processing Systems (NIPS)*, 2002, pp. 1523–1530.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.

[11] S. Jones and L. Shao, "Unsupervised spectral dual assignment clustering of human actions in context," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014.

[12] B. Kehoe, G. Kahn, J. Mahler, J. Kim, A. Lee, A. Lee, K. Nakagawa, S. Patil, W. Boyd, P. Abbeel, and K. Goldberg, "Autonomous multilateral debridement with the raven surgical robot," in *Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[13] G. Konidaris and A. G. Barto, "Efficient skill learning using abstraction selection." in *IJCAI*, vol. 9, 2009, pp. 1107–1112.

[14] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot Learning from Demonstration by Constructing Skill Trees," *Int. Journal of Robotics Research*, vol. 31, no. 3, pp. 360–375, 2011.

[15] S. Krishnan*, A. Garg*, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg (*denotes equal contribution), "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning," in *International Symposium of Robotics Research*. Springer STAR, 2015.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[17] V. Kruger, D. Herzog, S. Baby, A. Ude, and D. Kragic, "Learning actions from observations," *Robotics & Automation Magazine, IEEE*, vol. 17, no. 2, pp. 30–43, 2010.

[18] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *Int'l Journal of Robotics Research*, 2011.

[19] C. Lea, G. D. Hager, and R. Vidal, "An improved model for segmentation and recognition of fine-grained activities with application to surgical training tasks." in *WACV*, 2015.

[20] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, 2015.

[21] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *arXiv preprint arXiv:1504.00702*, 2015.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *arXiv preprint arXiv:1411.4038*, 2014.

[23] J. Mahler, S. Krishnan, M. Laskey, S. Sen, A. Murali, B. Kehoe, S. Patil, J. Wang, M. Franklin, P. Abbeel, and G. K., "Learning accurate kinematic control of cable-driven surgical robots using data cleaning and gaussian process regression," in *Int. Conf. on Automated Sciences and Engineering (CASE)*, 2014, pp. 532–539.

[24] T. Moldovan, S. Levine, M. Jordan, and P. Abbeel, "Optimism-driven exploration for nonlinear systems," in *Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[25] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," in *Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[26] S. Niekum, S. Osentoski, G. Konidaris, and A. Barto, "Learning and generalization of complex tasks from unstructured demonstrations," in *Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2012.

[27] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto, "Learning grounded finite-state representations from unstructured demonstrations," *Int'l Journal of Robotic Research*, 2015.

[28] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 763–768.

[29] G. Quellec, M. Lamard, B. Cochener, and G. Cazuguel, "Real-time segmentation and recognition of surgical tasks in cataract surgery videos," *Medical Imaging, IEEE Transactions on*, Dec 2014.

[30] C. E. Reiley, E. Plaku, and G. D. Hager, "Motion generation of robotic surgical tasks: Learning from expert demonstrations," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 967–970.

[31] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[32] ——, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[33] K. Subramanian, C. Isbell, and A. Thomaz, "Learning options through human interaction," in *2011 IJCAI Workshop on Agents Learning Interactively from Human Teachers (ALIHT)*. Citeseer, 2011.

[34] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012.

[35] L. Tao, L. Zappella, G. D. Hager, and R. Vidal, "Surgical gesture segmentation and recognition," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*. Springer, 2013.

[36] B. Varadarajan, C. Reiley, H. Lin, S. Khudanpur, and G. Hager, "Data-derived Models for Segmentation with Application to Surgical Assessment and Training," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2009, pp. 426–434.

[37] C. Wu, J. Zhang, S. Savarese, and A. Saxena, "Watch-n-patch: Unsupervised understanding of actions and relations," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2015.

[38] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2014.

[39] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative cnn video representation for event detection," *arXiv:1411.4006*, 2014.

[40] Y. Yang, I. Saleemi, and M. Shah, "Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 7, pp. 1635–1648, 2013.

[41] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, 2014.

[42] L. Zappella, B. Bejar, G. Hager, and R. Vidal, "Surgical gesture classification from video and kinematic data," *Medical image analysis*, vol. 17, no. 7, pp. 732–745, 2013.