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 sizable datasets of fixed-camera video and kinematic recordings of surgical subtasks. Segmentation of these trajectories into locally-similar contiguous sections can facilitate learning from demonstrations, skill assessment, and salvaging good segments from otherwise inconsistent demonstrations. 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 correlate with transition events using features constructed from layers of pre-trained image classification Deep Convolutional Neural Networks (CNNs). We report results on three datasets comparing Deep Learning architectures (AlexNet and VGG), choice of convolutional layer, dimensionality reduction techniques, visual encoding, and the use of Scale Invariant Feature Transforms (SIFT). We find that the deep architectures extract features that result in up-to a 30.4% improvement in Silhouette Score (a measure of cluster tightness) over the traditional "shallow" features from SIFT. We also present cases where TSC-DL discovers human annotator omissions. Supplementary material, data and code is available at: http://berkeleyautomation.github.io/tsc-dl/

I. Introduction

Inspired by the recent success of deep neural networks in reinforcement learning [12, 11], this paper explores how visual features extracted from Convolutional Neural Networks can be used for task segmentation. We are motivated by examples in robot-assisted surgery, where there are a growing number of datasets with kinematic and video recordings of surgical procedures. While these datasets have the potential to facilitate learning and autonomy, the variability of surgical data poses a unique challenge. Extracting common segments shared across multiple demonstrations of the same surgical task is an important pre-processing step before using this data [9, 3, 16]. Segmentation can facilitate learning from demonstrations, skill assessment, and salvaging segments from otherwise inconsistent demonstrations.

There are several recent proposals to learn segmentation criteria with minimal supervision (i.e., no dictionaries or labels) [3, 16]. Inherently, the success of these approaches depends on the state representation, which is particularly challenging for visual features. Visual perception pipelines often require hand-coding of essential features (e.g., object tracking and pose estimation), and thus, have to be modified for each new task. The recent results in Deep Learning, especially with Convolutional Neural Networks (CNNs), show that it is possible to use pre-trained CNNs to extract task-agnostic features [28]. These features have been shown empirically to perform well in recent work in robot visual perception [12, 11].

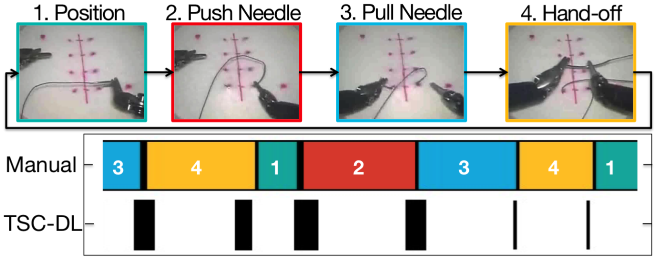

Fig. 1: Illustrative TSC-DL result for a suturing procedure. TSC-DL extracts a segmentation that closely aligns with the manual annotation without supervision. The width of the black segments illustrates a confidence interval on the predicted segment endpoint.

We propose Transition State Clustering with Deep Learning (TSC-DL), which extends our previous work [9] with automatically constructed visual features from pre-trained CNNs (i.e., trained on large libraries of images [10]). In the Transition State Clustering mode, each demonstration is a switching linear dynamical system (SLDS). We identify the modes of the SLDS and infer regions of the state-space at which mode transitions likely occur. We model these transitions as generated from a nonparametric Bayesian model, where the number of regions is determined by a Dirichlet Process and the shape of the regions are determined by a mixture of multivariate Gaussian random variables. The key insight of this paper is that pre-trained CNNs are effective for extracting relevant features from videos for Transition State Clustering.

The primary contributions are: (1) exploring the effectiveness of Deep Learning methods to extract visual features for segmentation, (2) a hierarchical multi-modal clustering algorithm combining visual and kinematic trajectory data, and (3) a resampling-based estimator to predict segmentation times with confidence intervals. We report results on three datasets, two Deep Learning architectures (AlexNet and VGG), different convolutional layers, and varying dimensionality reduction techniques, to study the performance when compared with standard implementations of Scale Invariant Feature Transforms (SIFT). Comparing the performance of a pre-trained Deep Neural Network against SIFT on

* 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

extracting visual features for segmentation into a sequence of segments with distinct linear dynamical system parameters, the former produced a significant (up to 30.4%) improvement in Silhouette Score (a standard measure of cluster tightness). We also compare TSC-DL with manual annotations when available using Normalized Mutual Information (NMI, a measure of sequence alignment). On real surgical datasets from JHU JIGSAWS, we find that TSC-DL matches the manual annotation with up to 0.806 NMI. Our results also suggest that applying TSC-DL to both kinematic and visual states results in increases of up to 0.215 NMI over just using the kinematics alone.

II. Related Work

A. Deep Features in Robotics

Neural networks have demonstrated empirical success in end-to-end robotic control problems, where robots learn policies directly from images [11, 12]. The success of convolutional features in learning control policies, suggests that these features may also have other properties related to the underlying dynamical system. In this paper, we explore methodologies for leveraging deep features in segmentation in combination with the Transition State model which is motivated by dynamical system theory. We believe that segmentation is an important first step in many robot learning applications, and the appropriate choice of visual features is key to accurate segmentation.

B. Visual Gesture Recognition

A highly relevant line of work is visual activity recognition, and many recent works attempt to segment human motion primitives from videos [6, 21, 27, 7, 25, 24]. There are a few unsupervised models for segmentation of human actions: Jones et al. [7], Yang et al. [27], Di Wu et al. [25], and Chenxia Wu et al. [24]. TSC-DL studies a broader problem of robot task segmentation where states may be represented by kinematics, vision, or both. Jones et al. [7] studied the problem of segmentation with two temporally aligned views of the same action, and they proposed an algorithm called Dual Assignment k-Means (DAKM) to relate the segments in the two views. It is not clear how this would support multiple demonstrations (>2) with temporal inconsistencies. Other algorithms derived from k-means have also been popular. Yang et al. [27] and Wu et al. [24] 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 could provide a robust drop-in-replacement for the k-means dictionary learning step [9].

C. Learning From Demonstrations (LfD)

The motion primitive model is a popular LfD framework that learns to control by composing a discretized set of actions [17]. This line of work mostly focuses on pre-defined primitives. Niekum et al. [15] 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 is not used to identify segments.

Calinon et al. [2, 4] characterize segments from demonstrations as skills that can be used to parametrize imitation learning. A number of other works have leveraged a similar model for segmentation, e.g., [8, 20]. As we describe in Section III, learning Gaussian Mixture Models is closely related to learning switching linear dynamical systems [14]. While Calinon et al. [4] have explored the use of visual features, the visual sensing model is tailored to a specific task (i.e., tracking the trajectory of a ball to catch), and our paper focuses on general visual features used for all tasks.

D. Surgical Robotics

Surgical robotics has largely studied the problem of supervised segmentation using either segmented examples or a pre-defined dictionary of motions (similar to motion primitives). For example, given manually segmented videos, Zappella et al. [29] use features from both the videos and kinematic data to classify surgical motions. Similarly, Quellec et al. [18] use manually segmented examples as training for segmentation and recognition of surgical tasks based on archived cataract surgery videos. The dictionary-based approaches utilize a pre-defined set of motion primitives for surgery called *surgemes* to bootstrap learning of temporal segmentation [13, 23, 22]. This work does not assume prior knowledge of motion primitives.

III. Model Overview

Each demonstration is modeled as a realization of an unknown time-varying linear dynamical system with a discrete number of dynamical modes and zero-mean process noise. Switching events, i.e., when $A(t) \neq A(t+1)$, happen stochastically as a function of the current state. Thus, the observed transitions from repeated demonstrations induce a probability density f over the state space $\mathcal{X}$ (including both kinematic and visual states). The modes of the density, which intuitively represent a propensity of a state $\mathbf{x} \in \mathcal{X}$ to trigger a switch, are called *Transition States*. The goal will be to use a Gaussian Mixture Model to approximate f.

1) 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 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. We further assume that the demonstrations are *consistent*, that is, there exists a non-empty sequence of common transitions that respects the partial order of transition events in all demonstrations (for details see [9]).

Transitions are defined as mode switches in a switching 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 in $\{A_1, ..., A_k\}$. *Transition state* is a state $\mathbf{x}(t)$ at time t, such that $A(t) \neq A(t+1)$.

2) 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 sets:

$$\mathcal{C} = \{C_1, C_2, ..., C_m\}.$$

The model $\mathcal{C}$ can be used to infer the structure of the task. When transition states are drawn from a GMM model, this clustering is the Maximum Likelihood Assignment:

$$x(t) \sim N(\mu_i, \Sigma_i).$$

Therefore, associated with each cluster C_i, there is a tuple (μ_i, Σ_i), resulting in the following GMM:

$$\{(\mu_1, \Sigma_1), (\mu_2, \Sigma_2), ..., (\mu_l, \Sigma_l)\}.$$

Finally, with each GMM state cluster, we cluster the transitions temporally. Each cluster then has a time-interval defined by the temporal cluster, in addition to the state-space region:

$$\mathcal{C} = \{(\mu_1, \Sigma_1, [\tau_1^- \tau_1^+]), (\mu_2, \Sigma_2, [\tau_2^- \tau_2^+]), ..., (\mu_m, \Sigma_m, [\tau_m^- \tau_m^+])\}.$$

IV. Problem Setup

The goal of Transition State Clustering is to learn $\mathcal{C}$ from a set of demonstrations of a task. There are two sub-problems related to this goal: (1) learning the parameters of the model $\mathcal{C}$ from all demonstrations, and (2) for each demonstration d, identifying states that most align with the segments defined by $\mathcal{C}$.

Problem 1. Task Segmentation

A set of demonstrations is consistent if there exists a clustering model $\mathcal{C}$ that respects the partial order of every demonstration (see [9] for a precise definition). Given a consistent set of demonstrations, the problem is to find a sequence of transition state clusters $\mathcal{C}$ reached by at least a fraction ρ of the demonstrations.

Problem 2. Temporal Segmentation

The set of clusters $\mathcal{C}$ define regions of the state-space and times where transitions occur common to multiple demonstrations of a task. For each demonstration d, we would like to know which states are transitions that correspond to the clusters in $\mathcal{C}$. Due to the pruning, there may be transitions that are present in some demonstrations but not in others. Furthermore, a demonstration may have multiple transitions within the same cluster. Hence, we also need a measure of confidence on when the transition occurs.

Given $\mathcal{C}$, the problem is to find a set of *predicted* transitions *for each demonstration* d_i. For every d_i, there will be some subset of transition state clusters $C^i \subseteq \mathcal{C}$ that are relevant to the individual demonstration. For each $c \in C^i$, we would like to identify the time t_c of the transition event in d_i.

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 to 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. This measures "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 ρ 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:

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

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 $b(i)$ is the dissimilarity with the closest cluster measured as the minimum mean dissimilarity of point i to cluster C_k, $k \neq j$. We use the L_2-norm as the dissimilarity metric and rescale $\mathsf{ss} \in [0, 1]$ for ease of comparison.

Extrinsic metric: For every time t, we will have a TSC-DL prediction τ_t and a manual annotation l_t. To calculate a measure of similarity between τ and l we use the *Normalized Mutual Information* (NMI), which measures the alignment between two label assignments irrespective of index choice. NMI is equal to the KL-divergence between the joint distribution and the product distribution of the marginals; intuitively quantifying the distance from pairwise statistical independence. The NMI score lies in [0, 1], where 0 indicates independence while 1 corresponds to a perfect matching. It is defined as:

$$NMI(\tau, l) = \frac{I(\tau, l)}{\sqrt{H(\tau)H(l)}}, \quad NMI(\tau, l) \in [0, 1].$$

V. Transition State Clustering With Deep Learning

In this section, we describe algorithms to learn the solutions to problem 1 and problem 2.

Identifying Transitions: Suppose there was only one regime, then following from the Gaussian assumption, we obtain a linear regression problem:

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

where $X_t = [\mathbf{x}(1), \ldots, \mathbf{x}(T)] \in \mathbb{R}^{n \times T}$ with each column as the state at time t: $\mathbf{x}(t) \in \mathbb{R}^n$. Generalizing to multiple regimes, Moldovan et al. [14] showed that fitting a Jointly Gaussian model to $n(t) = \binom{\mathbf{x}(t+1)}{\mathbf{x}(t)}$ is equivalent to Bayesian Linear Regression–and thus fitting a GMM finds locally linear regimes.

A. Task Segmentation as a Sequence of Transition Clusters

Over all of the demonstration, TSC-DL clusters the states at which these transitions occur. The key challenge is that we have a state-space composed of multiple sensing modalities such as kinematics and visual state. Such states may not be directly comparable due to differences in cardinality (many more visual states than kinematics states), in semantics (distances between kinematic states may be more significant), and in stochasticity (kinematic measurements are likely less noisy than visual ones). We address this problem by constructing a hierarchy of GMM clusters, where each hierarchy only clusters over a single sensing modality.

Visual Features: Transition State Clustering with Deep Learning (TSC-DL) utilizes domain independent visual features from pre-trained CNNs. CNNs are increasingly popular for image classification and with existing models trained on millions of natural images. Intuitively, CNNs classify based on aggregations (pools) of hierarchical convolutions of the pixels. Yosinski et al. noted that CNNs trained on natural images exhibit roughly the same Gabor filters and color blobs on the first layer for various datasets [28]. They established that earlier layers in the hierarchy learn more general features while later layers learn more specific ones. Hence, removing the aggregations and the classification layers results in convolutional filters which can be used to derive generic features across datasets.

We use layers from a pre-trained Convolutional Neural Network (CNNs) to derive the features frame-by-frame. 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 [10], 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) [19]. In our experiments, we explore the level of generality of features required for segmentation. We also compare these features to other visual featurization techniques such as SIFT for the purpose of task segmentation using TSC-DL.

Visual Feature Encoding and Dimensionality Reduction

1) *Feature 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) [26].

2) *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 minimum 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 its wide application in computer vision [26]. We also explore CCA as it finds a projection that maximizes the correlation between the visual features and the kinematics features.

Algorithm 1: TSC-DL: Transition Learning

```
  Data: Set of demonstrations:$\mathcal{D}$
1 foreach $d_i \in \mathcal{D}$ do
      // concatenate kinematic & visual features
2     $x_i(t) \leftarrow \left[\binom{\mathbf{k_i}(t-1)}{\mathbf{z_i}(t-1)}, \binom{\mathbf{k_i}(t)}{\mathbf{z_i}(t)}, \binom{\mathbf{k_i}(t+1)}{\mathbf{z_i}(t+1)}\right]^T \ \forall t \in \{1,\ldots,T_i\}$
3     foreach $t \in \{1,\ldots,T_i\}$ do $\mathbf{X} \leftarrow \mathbf{X} \cup x_i(t)$
  // $C_i(t)$ is Index of cluster containing $x_i(t)$
4 $\{C_i(t), \forall\ x_i(t) \in \mathbf{X}\} \leftarrow$ DP-GMM(X)
5 $\Theta \leftarrow \emptyset$            // $\Theta$: set of all transition states in $\mathcal{D}$
6 foreach $d_i \in \mathcal{D}$ do
7     $\Theta \leftarrow \Theta \cup x_i(t),\ \forall t$, s.t. $C_i(t) \neq C_i(t+1)$
  Result: The set of transitions $\Theta$
```

Algorithm 2: TSC-DL: Task Segmentation Learning

```
  Data: The set of transitions $\Theta$, data $\mathbf{X}$      // line #3 Alg ??
  // Cluster over Visual Features of Transitions
1 $\mathcal{C}^z : \{z_i(t), \forall\ x_i(t) \in \Theta\} \leftarrow$ DP-GMM($\Theta$)  $z_i(t)$:cluster index
2 foreach $z_i(t) \in \mathcal{C}^z$ do
3     $\Theta^z \leftarrow \{x_i(t) \in \Theta,$ s.t. $\hat{z}_i(t) = z_i(t)\}$   $\hat{z}_i(t)$: index of $x_i(t)$
      // Cluster over Kinematic Features of Transitions
4     $\mathcal{C}^z_k : \{k_i(t), \forall\ x_i(t) \in \Theta^z\} \leftarrow$ DP-GMM($\Theta^z$)
5     foreach $k_i(t) \in \mathcal{C}^z_k$ do
6         $\Theta^z_k \leftarrow \{x_i(t)$ s.t. $x_i(t) \in \Theta^z, \hat{k}_i(t) = k_i(t)\}$
7         if $\sum_{d_i} \mathbf{1}\left(\sum_{t \in T_i} \mathbf{1}(x_i(t) \in \Theta^z_k) \geq 1\right) \leq \rho|\mathcal{D}|$ then
8             $\mathcal{C}^z_k \leftarrow \mathcal{C}^z_k \setminus \{k_i(t)\}$          // Cluster Pruning
  Result: The set of transitions $\Theta^z_k, \forall z, k\}$
```

Algorithm Overview: 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. The augmented state for each demonstration $d_i \in \mathcal{D}$ is collected in a state vector $\mathbf{X}$. GMM clustering over the sequence of states in $\mathbf{X}$, results in the identification of the set of transitions Θ, or switching events where $A(t) \neq A(t+1)$ as outlined in Algorithm 1.

Subsequent hierarchical clustering uses state representations only at transitions in set Θ. Intuitively, the Transition Learning results in an over-segmentation of the trajectory in state space, while subsequent clustering steps retain only a small subset of transition states that are consistent across the data set.

After that, we cluster in sub-spaces of each of the modalities – perception and kinematics. We start with clustering over subspace of visual feature to obtain a set clusters:Θ^z,

Algorithm 3: TSC-DL: Temporal Segmentation

```
Data: Set of demonstrations: $\mathcal{D}$
foreach $d_i \in \mathcal{D}$ do
    $\Theta_{[i]} \leftarrow$ Transition-Learning($\mathcal{D}'$) for $\mathcal{D}' = \mathcal{D} \setminus d_i$
    $\{\ddot{\Theta}_k^z, \forall z, k\} \leftarrow$ Task-Seg-Learning($\Theta_{[i]}, \mathbf{X}_{[i]}$)
    foreach $c \in \mathcal{C}_{[i]}$ do
        foreach $d_i \in \mathcal{D}'$ do
            $T_i \leftarrow T_i \cup \{t : \hat{k}_i(t) = k_i(t), x_i(t) \in d_i\}$
    $\mathtt{T}_j \leftarrow \mathtt{T}_j \cup T_j^{(i)}, \ \{\forall \ j : d_j \in \mathcal{D}'\}$    // $T_j^{(i)}$: ith iteration
// Cluster over time to predict Transition Windows
foreach $d_i \in \mathcal{D}$ do
    $(\mathcal{T}_i, \sigma_i) \leftarrow DPGMM(\mathtt{T}_i, \alpha_4)$
Result: Set of Predicted Transitions Times $\mathcal{T}_i \pm \sigma_i, \ \forall d_i \in \mathcal{D}$
```

indexed by z_i. Within each visual feature space cluster (Θ^z), we model the kinematics change points to be drawn from a GMM: $k \sim N(\mu_i, \sigma_i)$, and fit a GMM to the kinematic subspace of the transition states in Θ_k^z as outlined in Algorithm 2.

Similarly, time can also be modeled as a separate sensing modality. Without consideration of time, the transitions may be ambiguous. For example, in a "Figure 8" trajectory, the robot may pass over a point twice in the same task. Within a state cluster, we model the times at which change points occur as drawn from a GMM: $t \sim N(\mu_i, \sigma_i)$. This groups together events that happen at similar times during the demonstrations. The result is clusters of states and times. Thus, a transition state m_k defines a GMM over the state-space and a time interval.

Skill-Weighted Pruning: After the second stage of clustering, we perform a consistency check in recovered transition state clusters by pruning clusters which do not have change points from at least a ρ-fraction of the dataset. This accounts for outliers and identifies inconsistent demonstrations.

However, 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 $k_i(t)$ is pruned if it does not contain transitions Θ_k^z from at least a ρ fraction of demonstrations:

$$\sum_{d_i} \hat{w}_i \mathbf{1}\Big(\sum_{t \in T_i} \mathbf{1}(x_i(t) \in \Theta_k^z) \geq 1\Big) \leq \rho.$$

This criterion enforces that the task segmentation contains transition states from highly weighted demonstrations even if the data set is unbalanced, i.e., it contains many more noisy data points than good ones. In our experiment, the choose the weights as inversely proportional to average time of each example: $\hat{w}_i = 1/T_i$.

State Memory: To better capture 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 consecutive states starting at t. We varied the length of temporal history T and evaluated the performance of the TSC-DL algorithm for the suturing task using a metric defined in Section IV. We empirically found a sliding window of size 3, i.e., $\mathbf{x}_t = \left[\binom{\mathbf{k_i}(t-1)}{\mathbf{z_i}(t-1)}, \binom{\mathbf{k_i}(t)}{\mathbf{z_i}(t)}, \binom{\mathbf{k_i}(t+1)}{\mathbf{z_i}(t+1)}\right]^T$, as the state representation that led to improved segmentation accuracy while balancing computational effort.

TABLE I: The Table lists the silhouette scores 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

	GRP	PCA	CCA
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
SIFT		0.443±0.008	

B. Temporal Segmentation of Each Demonstration

Once we have learned the model parameters for the entire task, the next step is to identify which states in each demonstration correspond to transition events. In a single demonstration, we may have missing transitions and transitions with multiple candidate states, and so there is some ambiguity about which state best represents a particular transition cluster. Our criteria for disambiguating the assignments is *robustness*, where we want to identify those assignments that are most likely to persist even if the rest of the demonstrations are slightly different.

We iteratively hold out one of the N demonstrations and apply TSC-DL to the remaining demonstrations. For each demonstration $d_i \in \mathcal{D}$, there are $N-1$ predictions in each of the runs where d_i is in the sample. We aggregate the predictions using another clustering step, and output cluster means ($\mathcal{T}_i$) and variances (σ_i) as temporal segment predictions with standard deviations as outlined in Algorithm 3. This style of estimation has been well studied in non-parametric statistics (e.g., Bootstrapping, and Jackknife estimators).

VI. Experiments

A. Pre-processing

Once the images were pre-processed, we applied the convolutional filters from the pre-trained neural networks frame by frame. To reduce variance due to extraneous objects and lighting changes, we crop each video to capture 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 normalized to a *zero* mean in each RGB-channel. individually [10, 19]. All pre-processing was performed with the open source `ffmpeg` library.

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

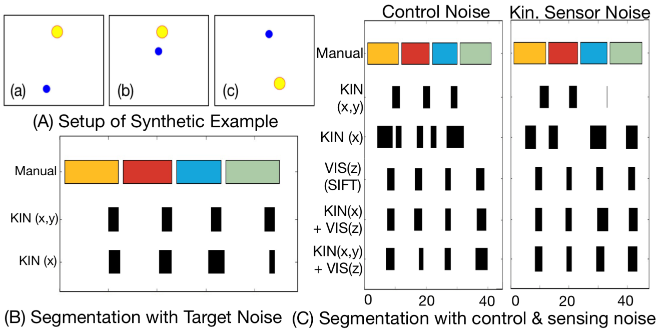

Fig. 2: (A) The figure shows a 2D synthetic example with a moving point in blue and target in yellow. The robot moves to the target in a straight line in discrete steps, and a new target appears. (B) Segmentation results for repeated demonstrations with variance in target position. (C) Segmentation under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

clusters. Table I describes the featurization techniques on the vertical axis and dimensionality reduction techniques on the horizontal axis. On this dataset, our results suggest that 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 achieved a SS performance of 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 discards informative features resulting in reduced clustering performance. We note that neither of the encoding schemes, VLAD or LCD significantly improves 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 4, 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.

C. t-SNE visualization of visual features

One of the main insights of this study is that features from pre-trained CNNs exhibit locally-linear behavior which allows application of a switching linear dynamical system model. We experimentally tested this by applying dimensionality reduction to trajectories of features from different video featurization techniques. Figure 3 shows t-SNE embeddings of visual features extracted for a single demonstration of suturing. The deep features display clear locally-linear properties and can be more easily clustered than SIFT features extracted for the corresponding frames. We speculate that SIFT breaks up trajectory structure due to its natural scale and location invariance properties. We also compared to using the raw RGB image pixel values and discovered that the deep features result in more well-formed locally linear trajectories. However, it is important to note that unlike spatial trajectories there are discrete jumps in the convolutional trajectories. We hope to explore this problem in more detail in future work.

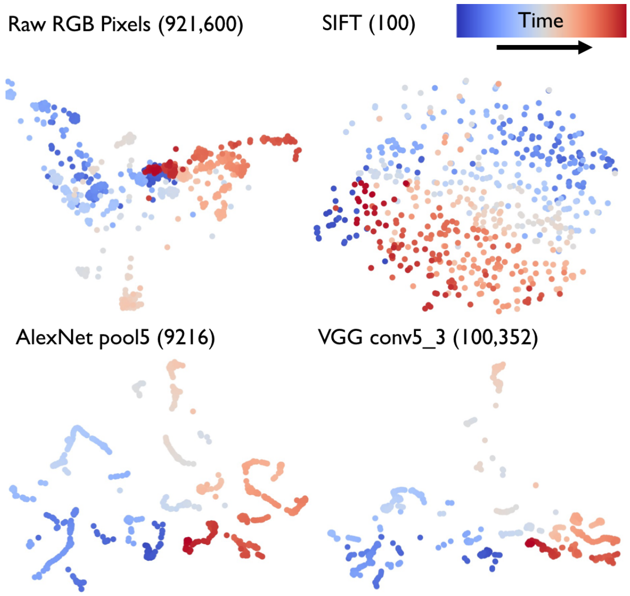

Fig. 3: Each data point in the figure corresponds to a t-SNE visualization of features of a single frame in the video. (a) RGB pixel values of original image (b) shallow SIFT features (c) CNN features from AlexNet pool5 (d) CNN features from VGG Conv5_3.

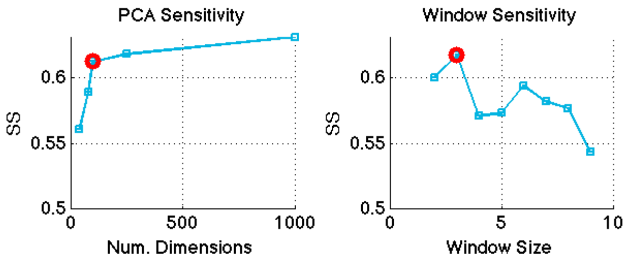

Fig. 4: We evaluate the sensitivity of two hyperparameters set in advance: number of PCA dimensions and sliding window size. The selected value is shown in red double circles.

D. End-to-End Evaluation

For all subsequent experiments on real data, we used 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 2). 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 2 (b) shows the results of unsupervised segmentation using only kinematics component of the data $\left(\binom{x(t)}{y(t)}\right)$. When the state is fully observed (i.e., we have both x and y positions), we accurately recover four segments with kinematics

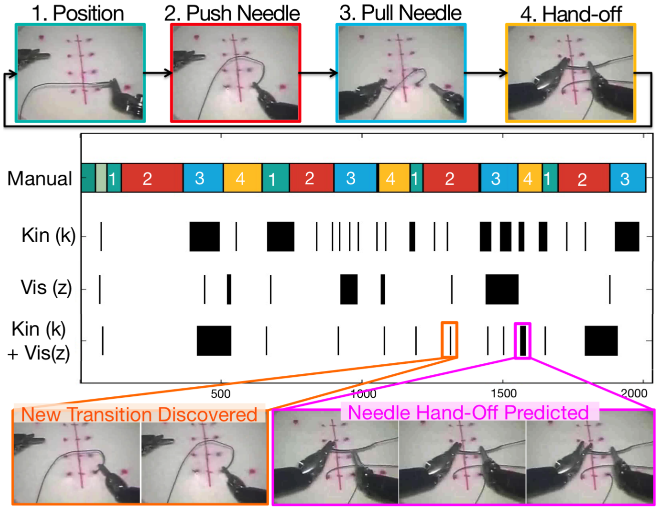

Fig. 5: The first row shows a manual segmentation of the suturing task in 4 steps: (1) Needle Positioning, (2) Needle Pushing, (3) Pulling Needle, (4) Hand-off. TSC-DL extracts many of the important transitions without labels and also discovers un-labled transition events.

alone. If one of these dimensions is unobserved, we find that we can still recover the four segments. In this example, when there is no noise on the kinematics, one dimension alone is enough to learn the segmentation.

Next, in Figure 2, we make this scenario more complex by introducing control noise: $x(t+1) = x(t) + u(t) + \nu$, with $\nu \sim \mathcal{N}(0, d_1)$ 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 $\hat{x}(t) = x(t) + \nu$, where $\nu \sim \mathcal{N}(0, d_2)$ with $d_2 = 0.25$. We note that only the kinematics is corrupted with noise while the vision sees a straight trajectory.

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.

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

2. Suturing: We apply our method to a subset of the JIGSAWS dataset [5] consisting of surgical task demonstrations under teleoperation 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. 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)). For the suturing dataset, we find that using both kinematics and video gives up-to 30.1% improvement in SS and 52.3% improvement in NMI over using kinematics alone. Not surprisingly, we also find that the expert demonstrations, which are usually smoother and faster, lead to improved segmentation performance when using only the kinematic data. However, when we incorporate the visual data, the trend is not as clear. We speculate this has to do with the tradeoff between collecting more data (denser clusters and more accurate modeling) versus inconsistencies due to novice errors, and this tradeoff is evident in higher dimensional data.

We visualize the results of the segmentation on one representative trajectory (Figure 5). With combined kinematics and vision, TSC-DL learns many of the important segments identified by annotation in [5]. Upon further investigation of the false positives, we found that they corresponded to meaningful actions missed by human annotators. TSC-DL discovers that a repositioning step where many demonstrators penetrate and push-through the needle in two different motions. While this is largely anecdotal evidence, we were able to find some explanations for some of the false positives found by TSC-DL.

3. Needle Passing: Next, we applied TSC-DL to 28 demonstrations of the needle passing task. These demonstrations were annotated in [5]. In this task, the robot passes a needle through a loop using its right arm, then its left arm to pull the needle through the loop. Then, the robot hands the needle off from the left arm to the right arm. This is repeated four times. Similar to the suturing dataset, we find that the combination of the features gives the best results. For the needle passing dataset, we find that using both kinematics and video gives up to 22.2% improvement in SS and 49.7% improvement in NMI over using the best of either kinematics or vision alone.

We found that the learned segments for the needle passing task were less accurate than those learned for the suturing task. We speculate that this is due to the multilateral nature of this task. This task uses both arms more than the suturing task, and as a result, there are many visual occlusions for a fixed camera. Important features such as the needle pose and the thread may be obscured at different points during the task. Furthermore, we constructed the state-space using the states of both arms. For such a task, it may be better to segment each of the arms independently.

VII. Conclusion

This paper explored how task segmentation can be learned from visual state representations extracted from *deep* convolutional neural networks (CNNs) with a new algorithm called

TSC-DL. We were surprised to find that "off-the-shelf" visual filters derived from Deep Learning CNNs trained on non-surgical images can yield valuable features for clustering and segmentation. This suggests that our previous segmentation method [9] can be extended to eliminate manual intervention. However, this required several novel contributions including hierarchical clustering, dimensionality reduction, and temporal clustering. On real datasets, we find that TSC-DL matches the manual annotation with up to 0.806 NMI, and our results also suggest that including kinematics and vision results in increases of up to 0.215 NMI over kinematics alone.

For these experiments, we used “off-the-shelf" pre-trained deep learning architectures trained on large image libraries that do not include surgical images. We intend to investigate if the performance improves when we train the CNNs with surgical images [10]. We will explore how to extract consistent structure across inconsistent demonstrations. We find that some surgical demonstrations have loops, i.e., repetitive motions where the surgeon repeats a subtask until success. Consolidating these motions into a single primitive is an important priority for us. The next step is to apply this and future automated segmentation methods to skill assessment and policy learning.

Acknowledgement: This work is supported in part by a seed grant from the UC Berkeley CITRIS, by the U.S. NSF Award IIS-1227536: Multilateral Manipulation by Human-Robot Collaborative Systems, NSF CISE Expeditions Award CCF-1139158, DOE Award SN10040 DE-SC0012463, and DARPA XData Award FA8750-12-2-0331, and gifts from Amazon Web Services, Google, IBM, SAP, The Thomas and Stacey Siebel Foundation, Adatao, Adobe, Apple Inc., Blue Goji, Bosch, Cisco, Cray, Cloudera, Ericsson, Facebook, Fujitsu, Guavus, HP, Huawei, Intel, Microsoft, Pivotal, Samsung, Schlumberger, Splunk, State Farm, Virdata, VMware, and the Knut and Alice Wallenberg Foundation. We acknowledge Greg Hager and Colin Lea for datasets and discussions. Additionally: NVIDIA for computing equipment grants; Andy Chou and Susan Lim for developmental grants; Yiming Jen 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] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

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

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

[13] H. Lin, I. Shafran, T. Murphy, A. Okamura, D. Yuh, and G. Hager, “Automatic detection and segmentation of robot-assisted surgical motions,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2005, pp. 802–810.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv:1409.1556*, 2014.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] 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.