Learning Clustering for Motion Segmentation

Xun Xu, *Senior Member, IEEE,* Le Zhang, Long-Fah Cheong, Zhuwen Li, and Ce Zhu*, *Fellow, IEEE*

***Abstract*—Subspace clustering has been extensively studied from the hypothesis-and-test, algebraic, and spectral clustering-based perspectives. Most assume that only a single type/class of subspace is present. Generalizations to multiple types are non-trivial, plagued by challenges such as choice of types and numbers of models, sampling imbalance and parameter tuning. In many real world problems, data may not lie perfectly on a linear subspace and hand designed linear subspace models may not fit into these situations. In this work, we formulate the multi-type subspace clustering problem as one of learning non-linear subspace filters via deep multi-layer perceptrons (mlps). The response to the learnt subspace filters serve as the feature embedding that is clustering-friendly, i.e., points of the same clusters will be embedded closer together through the network. For inference, we apply K-means to the network output to cluster the data. Experiments are carried out on synthetic data and real world motion segmentation problems, producing state-of-the-art results[1].**

***Index Terms*—Motion Segmentation, Deep Learning, Subspace Clustering**

I. Introduction

Subspace clustering aims to cluster data points into separate subspaces, with the dimension of the subspaces typically much smaller than the ambient space. Examples include vanishing point detection [1], rigid motion segmentation [2], [3], [4] and face clustering [5]. To make the problem tractable, traditional subspace clustering approaches tend to make various assumptions, such as data lying on a linear manifold, independence between subspaces, data drawn from a single type of subspace, known number of models, etc.

Despite the considerable amount of effort, there are still major lacunae in this research. Firstly, many real-world problems consist of data drawn from a union of multiple types of subspaces. We term this problem multi-type subspace clustering. Fig. 1 shows some examples: a toy example of line, circle and ellipses co-existing together, and two real-world motion segmentation scenarios. In the latter two scenarios, the appropriate model to fit the foreground object motions can waver between affine motions, homography, fundamental matrix [4], and even non-rigid motion, with no clear dividing boundary between them. With few exceptions [6], [2], [3], none of the existing works have considered this realistic scenario. Even if one attempts to fit multiple types of model sequentially like in [2], it is non-trivial to decide the type when the dichotomy of the models is unclear in the first place, e.g. when is the rotation dominant enough so that homography becomes a better model than fundamental matrix? For non-rigid motions, an analytic subspace model can be hard to define, thus neither the hypothesize-and-test nor the algebraic approach could be easily applied.

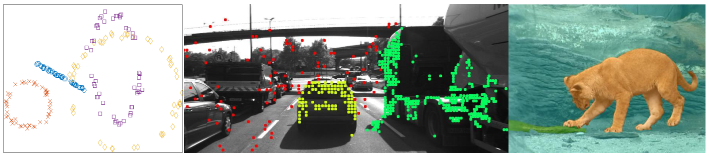


Fig. 1: Multi-type subspace clustering examples.

Secondly, for problems where there are a significant number of models, the traditional hypothesis-and-test approach is often overwhelmed by sampling imbalance, i.e. points from the same subspace represent only a minority, rendering the probability of hitting upon the correct hypothesis very small. This problem becomes severe when a large number of data samples are required for hypothesizing a model (e.g., eight points are needed for a linear estimation of the fundamental matrix and 5 points for fitting an ellipse). Moreover, for optimal performance, there is inevitably a lot of manipulation of parameters needed, among which the most sensitive include those for deciding what constitutes an inlier for a model [7], [8], for sparsifying the affinity matrices [9], [4], and for selecting the model type [3]. Often, dataset-specific tuning is required, with very little theory to guide the tuning.

Another open challenge in subspace clustering is to automatically determine the number of models, also referred to as model selection in the literature [10], [11], [12], [9]. Traditional methods are based upon the statistical analysis of the residual of the clustering [10], [13]. Other methods approach the problem using various heuristics including analyzing eigen values [14], [15], over-segment and merge [12], [9], soft thresholding [16] or adding penalty terms [17]. Most of the above works require extensive parameter tuning and have never been tested on data drawn from mixed-type of models. Lastly, hypothesis-and-test methods have to go through expensive sampling step, whereas analytic approaches have to contend with solving complex optimization problems. Thus, both approaches suffer from slow inference (as evidenced by our experimental comparisons), which is a serious qualification for real-time applications.

With the above considerations, we propose the SubspaceNet, a deep network that learns appropriate feature embeddings from input feature points without having to manually

* indicates corresponding author.

X. Xu and L. Zhang are with I2R, A-STAR, Singapore. e-mail: alex.xun.xu@gmail.com.

L.F. Cheong is with ECE, NUS, Singapore.

Z. Li is with Nuro, CA, USA.

C. Zhu is with University of Electronic Science and Technology of China. e-mail: eczhu@uestc.edu.cn.

This work was supported by the Natural Science Foundation of China under Grant U19A2052 and 62020106011.

[1] Tensorflow implementations and corresponding data will be released on https://github.com/alex-xun-xu/LearnSubspaceMoSeg.

design similarity metric nor to know the subspace model a priori. The learnt feature representation allows clusters to be readily identified using off-the-shelf methods, even when the underlying data are drawn from a union of mixed types of models, with the dividing boundary between these multiple types of subspaces being unclear (e.g. the transitions from a circle to an ellipse), or the underlying subspace is not analytically expressible (e.g. non-rigid motion). Our network consists mainly of stacked multi-layer perceptions (mlps). Each of the mlps has output in the form of $y = \mathbf{w}^\top x{+}b$, which describes a linear subspace. For each layer of mlp (m,n) (m and n indicate the number of input and output neurons respectively), we have up to n different subspaces and they could be stacked together to define convex polytopes delimited by multiple linear cuts in the original space. More importantly, by coupling mlps with non-linear activations functions and stacking the resultant nonlinear features into a hierarchy, we can approximate very complex non-linear subspaces in the ambient space. At each layer of mlp, feature points are represented as responses (distances) to the subspaces. This is analogous to the concept of Ordered Residual Kernel (ORK) in [11]: feature points of the same model display similar responses to the set of subspaces hypothesized and these responses can be regarded as a new form of feature representation. Here, given labelled data (inlier points for each model and outliers), the network learns the appropriate subspace filters (mlps) that produce the feature embeddings (responses to mlps) amenable for grouping into the respective, possibly mixed models. The preference for the various mixed types of models is also decided by the network in a data-driven manner without having to tune a lot of system parameters.

We summarize our contributions as follows. (i) First, we address multi-type subspace clustering, i.e. data drawn from mixed types of (possibly non-analytic) models. (ii) Our solution naturally affords the ability to handle model selection and sampling imbalance. (iii) We propose a subspace clustering network (SubspaceNet) by stacking multi-layer perceptrons and achieved state-of-the-art performance on three datasets. The SubspaceNet is more effective than alternative networks designed for sparse set of feature points. (iv) We proposed a more effective metric learning loss optimizing the distribution of learnt feature embedding.

II. Related Work

Subspace Fitting: Early approaches address this in a sequential RANSAC fashion [3], [18], [19] by iteratively fitting and removing inliers. The J-Linkage [20] and T-Linkage [7] simultaneously consider the interactions between all points and hypotheses. The final partition is achieved by clustering. The above greedy algorithms often do not perform well under high noise level. Global algorithms have also been proposed to minimize an energy with various regularization terms, including spatial regularization (PEaRL) [21] and label count penalty [22]. To eschew the problem of having to set thresholds, the ORK approach [11], [23] ranked the hypothesis according to data preference rather than absolute residuals. Analytic approaches are characterized by elegant mathematical formulation, including those based on the sparsity [24] and low-rank [16] assumptions and their variants. Many of the preceding works adopt spectral clustering for final grouping and assume known number of models, but only a few considered the model selection problem, e.g., [25], [12], [16], [26]. Even fewer works [6], [27], [2], [3] considered the problem of fitting multiple model of various types, and in these few works, the types are assumed to be known a priori and well-defined which is often not realistic.

Deep Learning for Geometric Modeling Problems: Using deep learning to solve geometric model fitting has received growing considerations. The dense approaches use raw image to model the transformation between image pairs as homography [28] or non-rigid transformation [29]. [30] proposed to estimate the camera pose directly from image sequences. DSAC [31] learns to extract from sparse feature correspondences a geometric model in a manner akin to RANSAC. The ability to learn representations from sparse points was also developed recently [32], [33]. This ability was exploited by [34] to fit essential matrix from noisy correspondences. Despite the promising results, none of the existing works have considered generic model fitting and, more importantly, fitting data drawn from multiple models and even multiple types. In our work, we formulate the generic multi-type fitting problem as one of learning good representations for clustering.

Deep Learning for Clustering: Unsupervised approaches tackle the problem by finding a latent embedding that minimizes the reconstruction loss of an autoencoder [35], [36], [37]. They are further combined with various losses for clustering objectives [38], [39], [40], [41]. Among these, the k-means loss was proposed by [39] optimizing the points-to-center distance. The subspace self-expressiveness objective was considered for discovering linear subspaces in the latent space [40]. In our tasks, there is a multiplicity of geometric models that are equally valid and subtly differentiated. For instance, given images of a rigidly moving cube, are we supposed to group the trajectory features by a single fundamental matrix or by multiple homographies? It would be difficult for the unsupervised networks to know the preference without any form of supervision.

In supervised approach, labelled data are used to learn feature embedding amenable for clustering [42]. With the advent of deep neural network, works in metric learning focus on designing losses amenable to clustering labelled data [43], [44], [45], [46], [47]. Among these, [46] minimizes the L2 distance between the predicted and ground-truth affinities and provides a competitive baseline. To further take into account the global distribution of the data points, we propose the clustering-specific loss MaxInterMinIntra, which optimizes the inter-cluster separation and intra-cluster variance and is proven to be more effective than existing alternatives.

III. Methodology

A. Network Architecture

We denote the input sparse data with N points as $\mathbf{X} = \{\mathbf{x}_i\}_{i=1\cdots N} \in \mathcal{R}^{D\times K}$ where each individual point is $\mathbf{x}_i \in \mathcal{R}^D$. The input sparse data could be 2D point cloud representing geometric shapes, or feature trajectories in multiple

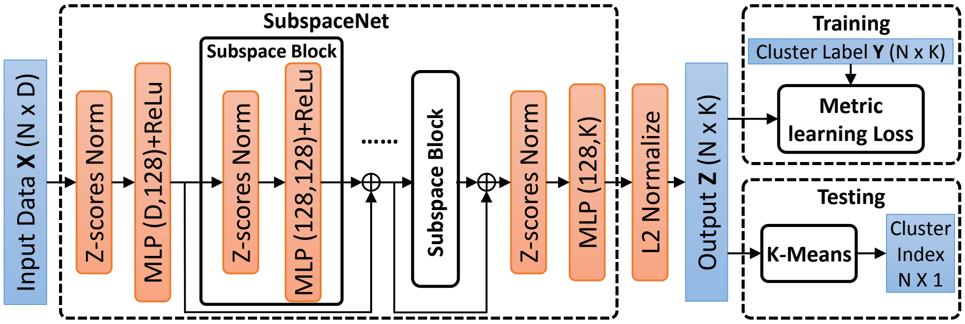


Fig. 2: Our Subspace clustering network. The metric learning loss is defined to learn good feature representation.

frames. For 2D point cloud $D = 2$. For motion segmentation task, we stack the feature trajectories' x and y coordinates and thus $D = 2F$ where F is the number of frames. We further denote the one-hot key encoded labels accompanying the input data as $\mathbf{Y} = \{\mathbf{y}_i\} \in \{0,1\}^{K\times N}$ where $\mathbf{y}_i \in \{0,1\}^K$ and K is the number of clusters or partitions of the input data.

Our subspace network consists mainly of stacked multi-layer perceptions (mlps) as shown in Fig. 2. It resembles the correspondence network [34] in that both exploit the power of mlps. We have noted that each layer of mlp works as multiple linear subspaces and the response to each layer of mlp serves as the new feature representation of the input feature points. Since the mlps are not scale invariant, a normalization layer is thus necessary before each mlp layer to center all feature points at origin with unit variance. The is realized by a standard z-score normalization on each input dimension, denoted as *Z-score Norm* layer in Fig. 2. We note that this step resembles the context norm (CN) proposed in [34]. However, the role of CN was ascribed to capturing the relation between feature points by [34] whereas here, we believe the role of *Z-score Norm* is more specifically that of ensuring uniform scale. We adopt the same ResNet [48] structure with CorresNet for training deeper network and the depth, number of *Subspace Blocks* is fixed at 50 for all experiments. For the output layer, we do not apply any activation but instead conduct L2 normalization on each sample. The output embedding is denoted as $\mathbf{Z} = \{f(\mathbf{X};\Theta)\} \in \mathcal{R}^{K\times N}$. To make the output $\mathbf{Z}$ clustering-friendly, we apply a differentiable, clustering-specific loss function $\mathcal{L}(\mathbf{Z},\mathbf{Y})$, measuring the match of the output feature representation with the ground-truth labels. The problem now becomes that of learning a CorresNet backbone $f(\mathbf{X};\Theta)$ that minimizes the loss $\mathcal{L}(\mathbf{Z},\mathbf{Y};\Theta)$.

B. Clustering Loss

We expect our clustering loss function to have the following characteristics. First, it should be invariant to permutation of models, e.g. the order of these models are exchangeable. Second the loss must be adaptable to varying number of groups. Lastly, the loss should enable good separation of data points into clusters. We consider the following loss functions.

L2Regression Loss: Given the ground-truth labels $\mathbf{Y}$ and the output embeddings $\mathbf{Z} = f(\mathbf{X};\Theta)$, the ideal and reconstructed affinity matrices are respectively,

$$\mathbf{K} = \mathbf{Y}^\top\mathbf{Y}, \quad \hat{\mathbf{K}} = \mathbf{Z}^\top\mathbf{Z} \tag{1}$$

The training objective is to minimize the difference between $\mathbf{K}$ and $\hat{\mathbf{K}}$ measured by element-wise L2 distance. In such way, the learned feature embedding $\mathbf{Z}$ will encode the cluster structure and can be used for cluster inference [46]

$$\begin{aligned} L(\Theta) &= ||\mathbf{K} - \hat{\mathbf{K}}||_F^2 \\ &= ||\mathbf{Y}^\top\mathbf{Y} - \mathbf{Z}^\top\mathbf{Z}||_F^2 \\ &= ||f(\mathbf{X};\Theta)^\top f(\mathbf{X};\Theta)||_F^2 - 2||f(\mathbf{X};\Theta)\mathbf{Y}^\top||_F^2 \end{aligned} \tag{2}$$

The above L2 Regression loss is obviously differentiable w.r.t. $f(\mathbf{X};\Theta)$. Since the output embedding $\mathbf{Z}$ is L2-normalized, the inner product between two point representations is $\mathbf{z}_i^\top\mathbf{z}_j \in [-1,1]$.

Cross-Entropy Loss: As alternative to the L2 distance, one could measure the discrepancy between $\mathbf{K}$ and $\hat{\mathbf{K}}$ as KL-Divergence. Since $D_{kl}(\mathbf{K}||S(\hat{\mathbf{K}})) = H(\mathbf{K}, S(\hat{\mathbf{K}})) - H(\mathbf{K})$, where $H(\cdot)$ is the entropy function and $S(\cdot)$ is the sigmoid function, with fixed $\mathbf{K}$, we simply need to minimize the cross-entropy $H(\mathbf{K}, S(\hat{\mathbf{K}}))$ which yields the following element-wise cross-entropy loss,

$$\begin{aligned} L(\Theta) &= \sum_{i,j} H\left(\mathbf{y}_i^\top\mathbf{y}_j, S\left(\mathbf{z}_i^\top\mathbf{z}_j\right)\right) \\ &= \sum_{i,j} H(\mathbf{y}_i^\top\mathbf{y}_j, S(f(\mathbf{x}_i;\Theta)^\top f(\mathbf{x}_i;\Theta))) \end{aligned} \tag{3}$$

The cross-entropy loss is more likely to push points i and j of the same cluster together faster than L2Regression, i.e. inner product $\mathbf{z}_i^\top\mathbf{z}_j \rightarrow 1$ and those of different clusters apart, i.e. inner product $\mathbf{z}_i^\top\mathbf{z}_j \rightarrow -1$.

MaxInterMinIntra Loss: Both the above losses consider the pairwise relation between points; the overall point distribution in the output embedding is not explicitly considered. We now propose a new loss which takes a more global view of the point distribution rather than just the pairwise relations. Specifically, we are inspired by the classical Fisher LDA [49]. LDA discovers a linear mapping $z = \mathbf{w}^\top\mathbf{x}$ that maximizes the distance between class centers/means $\mu_i = 1/N\sum_j z_j$ and minimizes the scatter/variance within each class $s_i = \sum_j(z_j - \mu_i)^2$. Formally, the objective for a two-class problem is written as,

$$J(\mathbf{w}) = \frac{|\mu_1 - \mu_2|^2}{s_1^2 + s_2^2} \tag{4}$$

which is to be maximized over $\mathbf{w}$. For linearly non-separable problem, one has to design kernel function to map the input features before applying the LDA objective. Equipped now with more powerful nonlinear mapping networks, we adapt

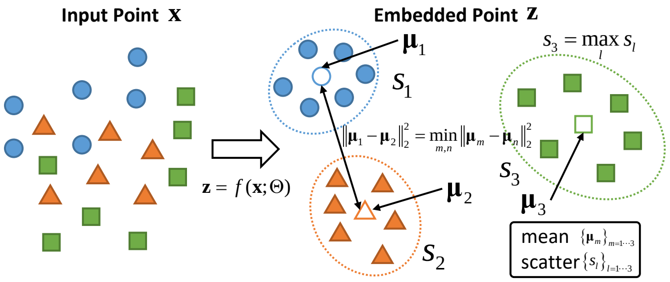

Input Point $\mathbf{x}$
Embedded Point $\mathbf{z}$
$\mathbf{z} = f(\mathbf{x};\Theta)$
$\boldsymbol{\mu}_1$
S_1
$\boldsymbol{\mu}_2$
S_2
$\boldsymbol{\mu}_3$
S_3
$s_3 = \max_l s_l$
$\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2^2 = \min_{m,n} \|\boldsymbol{\mu}_m - \boldsymbol{\mu}_n\|_2^2$
mean $\{\boldsymbol{\mu}_m\}_{m=1\cdots3}$
scatter $\{s_l\}_{l=1\cdots3}$

Fig. 3: Illustration of MaxInterMinIntra loss for point representation metric learning. The objective considers the minimal distance $\min_{m,n} ||\boldsymbol{\mu}_m - \boldsymbol{\mu}_n||_2^2$ between clusters and maximal scatter $\max_l s_l$ within clusters.

the LDA objective—for the multi-class scenarios—to perform these mappings automatically as below,

$$J(\Theta) = \frac{\min\limits_{m,n\in\{1\cdots K\},m\neq n} ||\boldsymbol{\mu}_m - \boldsymbol{\mu}_n||_2^2}{\max\limits_{l\in\{1\cdots K\}} s_l} \tag{5}$$

where $\boldsymbol{\mu}_m = \frac{1}{|\mathcal{C}_m|}\sum_{i\in\mathcal{C}_m} \mathbf{z}_i$, $s_l = \sum_{i\in\mathcal{C}_l} ||\mathbf{z}_i - \boldsymbol{\mu}_l||_2^2$ and $\mathcal{C}_l$ indicating the set of points belonging to cluster l. We use the extrema of the inter-cluster distances and intra-cluster scatters (see Fig. 3) so that the worst case is explicitly optimized. Hence, we term the loss as MaxInterMinIntra (MIMI). By applying log operation on the objective,we arrive at the following loss function to be minimized:

$$L(\Theta) = -\log\min_{m,n} ||\boldsymbol{\mu}_m - \boldsymbol{\mu}_n||_2^2 + \log\max_l s_l \tag{6}$$

One can easily verify that the MaxInterMinIntra loss is differentiable w.r.t. $\mathbf{z}_i$. We provide the gradient in the supplementary material.

Optimization: The Adam optimizer [50] is used to minimize the loss $L(\Theta)$. The learning rate is fixed at $1e-3$ and mini-batch at one frame pair or sequence. For all tasks, we train the network for 300 epochs.

C. Inference

During testing, we apply standard K-means to the output embeddings $\{\mathbf{z}_j\}_{j=1\cdots N_{te}}$. We also notice that any off-the-shelf cluster inference algorithm, e.g. DBSCAN [51] and Spectral Clustering [52] can be applied. This step is applicable to both multi-model and multi-type clustering problems, as we do not need to specify explicitly the type of model to fit. If there is a need to estimate the number of models K, we examine the K-means residuals defined by,

$$r(K) = \sum_{m=1\cdots K}\sum_{i\in\mathcal{C}_m} ||\mathbf{z}_i - \boldsymbol{\mu}_m||_2^2 \tag{7}$$

Good estimate of K often yields low $r(K)$ and further increasing K does not significantly reduce $r(K)$. Thus we find the K at the 'elbow' position. We adopt two off-the-shell approaches for this purpose: second order difference (SOD) [53] and silhouette analysis [13]. Both are parameter-free.

D. Discussion

We expect the proposed method to perform better than existing hypothesis-and-test and algebraic approaches. First, the high capacity of mlps allows the network to simultaneously lear nmultiple types of subspaces. While the existing approaches often have to sequentially fit one type of a model, e.g. RPA does sequentially fit line, circle and ellipses. Second, the nonlinearity of mlps allows fitting arbitrary nonlinear subspaces, a.k.a. manifold. Therefore, fitting non-rigid motion segmentation, which is not a linear subspace, can be formulated in a data driven manner. In contrast, existing model-based motion segmentation approach has to assume a linear model, e.g. homography, for segmenting non-rigid motions. Therefore, our data-driven approach would naturally outperform on non-rigid motions.

IV. Experiment

We demonstrate the performance of our network on both synthetic and real world data, with extensive comparisons with traditional geometric model fitting algorithms.

A. Datasets

Synthesized Lines, Circles and Ellipses (LCE): Fitting ellipses has been a fundamental problem in computer vision [54]. We synthesize for each sample four different types of conic curves in a 2D space, specifically, one straight line, two ellipses and one circle. We randomly generate 8,000 training samples, 200 validation samples and 200 testing samples. Each point is perturbed by adding a gaussian noise with $\sigma = 0.05$. The synthetic dataset provides insight into segmentation under mixed type of underlying models because line, circle and ellipse are represented by different degrees of conic equation. This poses great challenge to existing hypothesis-and-fit or subspace clustering based approaches where often a single type of model is assumed.

KT3DMoSeg [4]: This benchmark consists of 22 sequences from the KITTI dataset [55]. Each sequence contains two to five rigid motions. As analyzed by [4], the geometric model for each individual motion can range from an affine transformation, a homography, to a fundamental matrix, with no clear dividing line between them. We evaluate this benchmark to demonstrate our network's ability to tackle clustering under multiple type of motion models, namely homography and fundamental matrix. For fair comparison, we only crop the first 5 frames of each sequence for evaluation, so that the broken trajectory does not give undue advantage to certain methods.

FBMS59 [56]: This dataset was proposed for analyzing video object segmentation based on point trajectories, with 59 sequences in total, of which 29 are for training and 30 for testing. It covers a wide variety of scenes and the ground-truth is defined over semantic objects with dense mask. Most of the moving objects involve moderate non-rigidity, for which analytic geometric models are hard to define. We evaluate the first-10-frame setting as reported in [56] for fair comparison. The ground-truth for training is constructed by assigning the trajectories to the nearest label mask and the evaluation metric is the standard F-measure [56]. The main challenge of

FBMS59 lies in the non-rigidity of motion and mixture of type of motions.

Adelaide RMF Dataset [57]: We are concerned with the two-view motion segmentation task of this dataset. This task consists of 19 frame pairs each comprising of 2 to 5 independent motions. Though it is nominally a single-type multiple fundamental matrix fitting problem and has been treated as such by the community, we observe moderate degeneracies, i.e. near planar rigid objects, present in this dataset. Hence, we treat it as another multi-type (homography and fundamental matrix) clustering problem.

B. Multi-Type Curve Fitting

There is no clear dividing boundary between lines, circles, and ellipses as they can be all explained by the general conic equation (with the special cases of lines and circles obtained by setting some coefficients to 0):

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{8}$$

There are two ways to adapt the traditional multi-model fitting methods for this multi-type setting. One approach formulates the problem as fitting multiple models parameterized by the same conic equation in Eq (8), which is termed *HighOrder* (H.O.) fitting. Alternatively, one could sequentially fit three types of models, which is termed *Sequential* (Seq.) fitting. For ellipse-specific fitting, the direct least square approach [54] is adopted. For our model, we evaluate the various metric learning losses introduced in Section 3.2 and present the results in Tab. I. The results are reported with the optimal setting determined by the validation set. We evaluate the performance by two clustering metrics, Classification Error Rate (Error Rate), i.e. the best classification results subject to permutation of clustering labels, and Normalized Mutual Information (NMI). Comparisons are made with state-of-the-art multi-model fitting algorithms including T-linkage [7], RPA [8] and RansaCov [58]. We notice that T-linkage returns extremely over-segmented results in the sequential setting, e.g. more than 10 lines, making classification error evaluation intractable. For our model, we evaluate the three loss variants, the L2 Regression loss (L2), Cross Entropy loss (CE) and MaxInterMinIntra loss (MIMI).

TABLE I: Evaluations on synthetic multi-model and multi-type fitting dataset. ↑ and ↓ indicate the number is the higher or lower the better respectively. − indicates evaluation intractable.

Mdl.	T-Linkage [7]		RPA [8]		RansaCov [58]		SubspaceNet		
	H.O.	Seq.	H.O.	Seq.	H.O.	Seq.	L2	CE	MIMI
Err↓	52.14	-	39.43	23.17	40.57	24.04	18.49	18.32	**18.04**
NMI↑	0.340	-	0.464	0.667	0.394	0.604	0.713	0.720	**0.727**

We make the following observations about the results. First, all our metric learning variants outperform the *HighOrder* and *Sequential* multi-type fitting approaches. Second, the all-encompassing model used in the *HighOrder* approach suffers from ill-conditioning when fitting simpler models. Thus, the performance is much inferior to that of *Sequential* fitting. However, it is worth noting that despite the *Sequential* approach being given the strong a priori knowledge of both the model type and the number of model for each type, its performance is still significantly worse off than ours. For qualitative comparison, we visualize the ground-truth and segmentation results of each method in Fig. 4. Our clustering results on the bottom row show success in discovering all individual shapes with mistakes made only at the intersections of individual structures. The RPA failed to discover ellipses as sampling all 5 inliers amidst the large number of outliers and fitting an ellipse from even correct 5 support points with noise (noise in coordinate) are both very difficult, the latter demonstrated in [54].

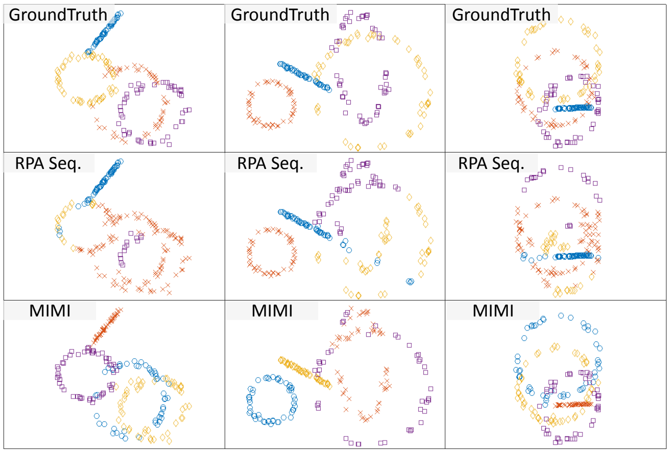


Fig. 4: Examples of multi-type clustering on synthetic dataset. We only show the RPA results based on the *Sequential* fitting approach.

TABLE II: Motion segmentation performance on KT3DMoSeg 5-frame task. Performances without ('Vanilla') and with augmentation ('Augment') are separated by a /. All error numbers are in % and inference time (Inf. Time) is in seconds.

	Model	Mean Err.	Median Err.	Inf. Time
Non-Deep	GPCA [59]	36.46	33.93	1.51
	ALC [60]	15.17	16.42	582.48
	LSA [61]	36.34	40.31	30.08
	LRR [16]	22.00	18.16	4.63
	MSMC [62]	32.74	36.48	125.73
	SSC [24]	26.62	29.14	3254.99
	MVC [4]	**10.99**	**6.57**	143.52
Deep Approaches	Unsupervised			
	DSCN [63]	28.14	30.00	1.85
	DCN [64]	48.45	48.16	1.85
	Supervised (Different Losses)			
	SHT [65]	9.93	9.11	1.85
	LIFT [66]	28.06	28.34	1.85
	NMI [67]	16.91	11.65	1.85
	L2 [46]	10.95/6.89	7.84/3.83	1.85
	CE	11.12/7.81	**7.04**/5.41	1.85
	MIMI	**10.62/5.83**	8.44/**3.58**	1.85

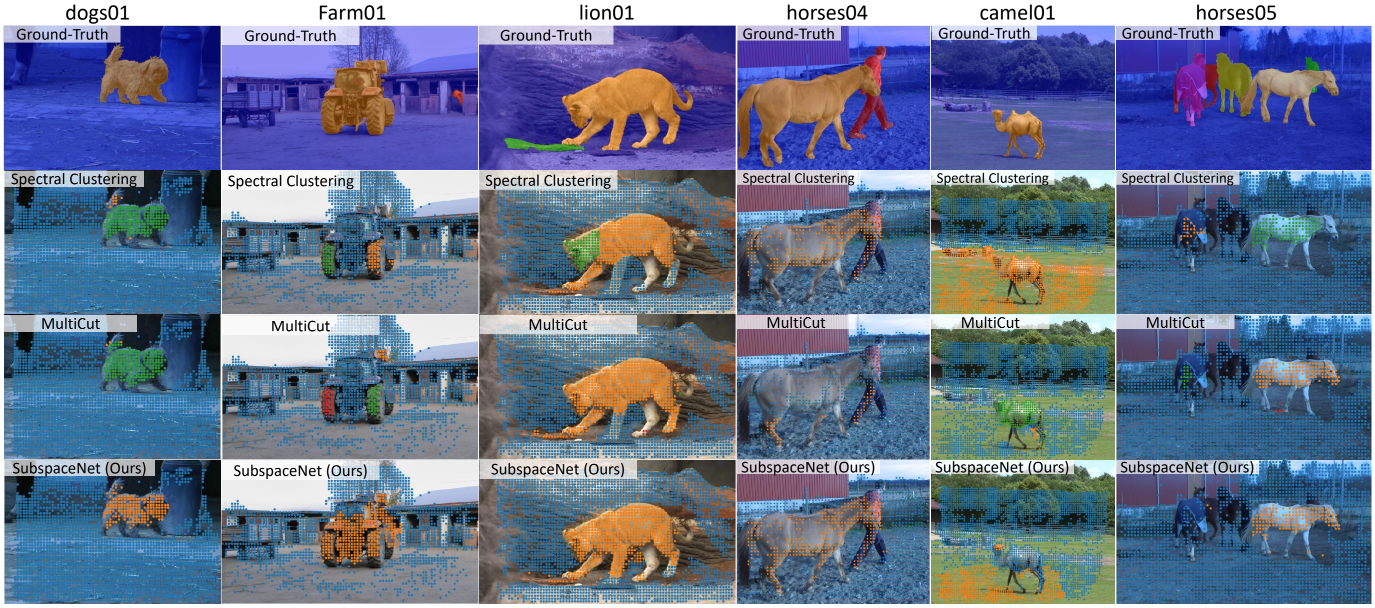


Fig. 5: Qualitative comparisons on FBMS59 test set using the first 10 frames.

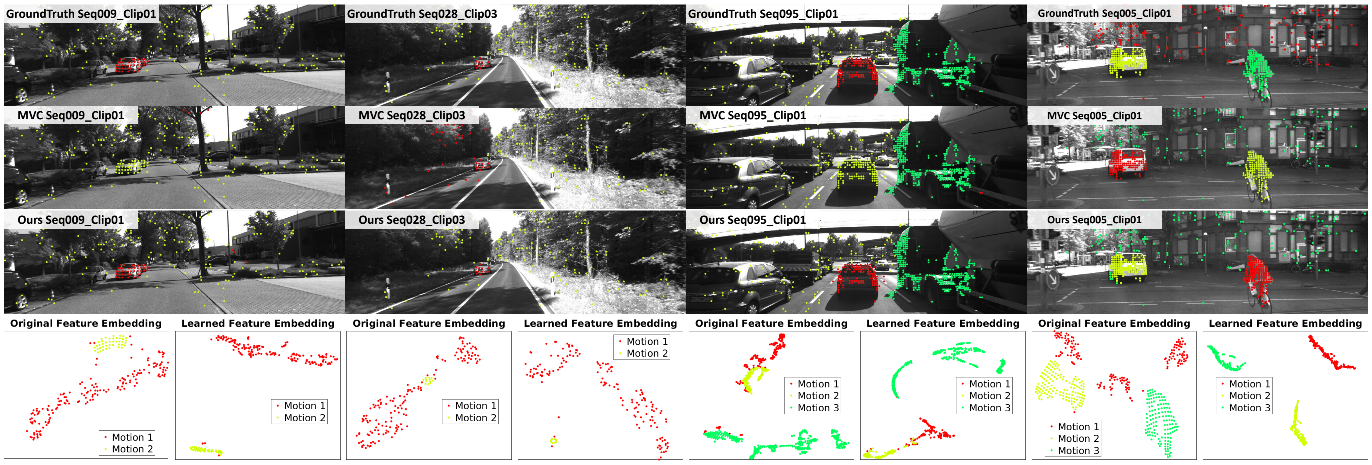


Fig. 6: Qualitative comparisons on 4 sequences from KT3DMoSeg. First row is the ground-truth. Second and third rows are the results of Multi-View Clustering [4] and our multi-type network respectively. The last row is the point feature embeddings before and after learning.

C. Multi-Type Motion Segmentation

Each sequence of the KT3DMoSeg benchmark [4] often consists of a background whose motion can be explained by a fundamental matrix while the models for the foreground motions can sometimes be ambiguous due to the limited spatial extent of the objects, thus giving rise to mixed types of models. For example, in Fig. 6, the vehicles in 'Seq009_Clip01' and 'Seq028_Clip03' can be roughly explained by an affine transformation or homography while the oil tanker in 'Seq095_Cip01' should be modeled by a fundamental matrix. When the background is dominated by a plane, for instance, the quasi-planar row of trees on the right side of the road in 'Seq028_Clip03', it is likely to lead to degeneracies in the fundamental matrix estimation. For this dataset, we apply leave-one-out cross-validation; we dubbed this the 'Vanilla' setting. Each sequence has between 10-20 frames, so we could further increase the training data by augmenting with all the remaining five-frame clips from each sequence, termed as the 'Augment' setting. The testing clips (first five frames of each sequence) are kept the same for both settings. We compare with conventional non-deep subspace clustering approaches, GPCA [59], LSA [61], ALC [60], LRR[16], MSMC [62] and SSC [24] and the multi-view clustering (MVC) methods in [4]. For the unsupervised deep clustering approaches, we include the Deep Subspace Clustering Network (DSCN) [63] and Simultaneous Deep Learning and Clustering (DCN) [64] for comparison. For the supervised setting, we compare with semi-hard triplet loss (SHT) [65], lifted structured feature embedding (LIFT) [66] and clustering quality metric (NMI) [67] with the same network architecture. Results are presented in Tab. II.

Our vanilla approach achieved very competitive performance on all 22 sequences in KT3DMoSeg. In the 'Augment' setting, our approach even outperforms the state-of-the-art multi-view clustering approaches (MVC) [4]. Of all benchmarked methods, only MVC has considered the multi-type fitting issue. Furthermore, we notice that our proposed MIMI metric is the best among all alternative losses considered. The unsupervised deep approaches lag behind by a large margin corroborating our earlier argument about the necessity to exploit labelled information for complex multi-type subspace clustering problem.

We also report the inference time for each method, however it is worth noting that the non-deep approaches are implemented in CPU-based Matlab while deep approaches are implemented with GPU-based Tensorflow. All methods are evaluated on a workstation with Intel i7 with GTX1080Ti. Therefore, the non-deep approaches could be further optimized. It is obvious the deep approaches are very efficient in inference, costing only 1.85 seconds to process all sequences (from trajectory input to clustering output).

Finally, we present qualitative comparisons in Fig. 6. The SubspaceNet surpasses our expectations in how it performs in 'Seq009_Clip01'. Here the independently moving car (the yellow group in the ground truth image) has a flow field that is consistent with the epipolar constraint associated with the background motion (due to them both translating in the same direction) [4]. Without resorting to reconstructing the depth of the car, it would be impossible to separate it from the background. However, criteria involving depth would be very unwieldy to specify analytically in the existing approaches. Here, without having any preconceived notion of the geometrical model, our network has learnt the requisite criteria to separate the independent motion.

D. Non-Rigid Motion Segmentation

We demonstrate the ability to learn non-rigid motion segmentation which is hard to be modeled by analytic geometric models. We train our model on the training set with 29 unique sequences and evaluate on the test set with 30 sequences following the rule established by [56]. For our method, the number of motion is estimated via SOD [53] with candidate cluster range from 1 to 10. SSC [24], ALC [60], Spectral Clustering (SC) [56] and MultiCut [58] are compared and the results are presented in Tab. V. We observe that our SubspaceNet, for both losses, is superior in performance compared with all three baseline methods. We further present qualitative comparisons with [56], [58] in Fig. 5. It is evident that the translational model employed in [56] with spatial and color information [58] detects the whole background but at the cost of over-segmenting the non-rigid foreground, e.g. the lion's head and the tractor's wheels. In contrast, our SubspaceNet detects the whole non-rigid foreground while keeping the background segmentation intact. Some objects are missed by all methods, e.g. the horse in stable of "Farm01", since it does not move significantly in the first 10 frames. We also notice that SubspaceNet performs quite bad on 'camel01' because it involves significant camera side-way translation and large scene depth variation. The lack of similar sequence in the training data makes it hard for data-driven approach to generalize to out-of-sample test sequence. The poor performance on 'horse05' is partly attributed to the insignificant motion of horse on the left. It suggests the weakness of SubspaceNet's insensitivity to subtle motion.

Furthermore, we compare against existing deep learning based optical flow estimation approaches [68], [69], [70], [71] to provide a context, even though they rely on RGB image sequence as input and do not generalize to motion segmentation with abstracted features as input only. We find the FlowNet2.0 [68] provided a benchmark on FBMS59 and we present the comparison in Tab. III. The 'Eval. Proto.' indicates evaluation on full trajectory training set (Train Set (Full)), full trajectory testing set (Test Set (Full)) and first 10 frames testing set (Test Set (10 frames)). 'Deep' indicates whether the method is deep learning based or not. We make the following observations. First, all deep methods except ours are networks designed for optical flow estimation, thus they all depend on full RGB image sequence and do not generalize to segmentation problems with abstracted trajectories only. Second, MultiCut [72] does almost as good as all alternative deep learning based methods on the training set of FBMS59. While there is a gap between training set and testing set on FBMS59, as suggested by MC on Train Set (Full) v.s. Test Set (Full). Moreover, the performance of MC on the first 10 frames on testing set (Test Set (10 frames)) is much worse than SubspaceNet. Based on all these observations, the SubspaceNet is a very competitive method even compared against existing state-of-the-art deep learning based approaches on FBMS59.

E. Two-View Motion Segmentation

We evaluate the motion segmentation task in the Adelaide RMF dataset [57]. We carry out a leave-one-out cross-validation. For comparability, we report the classification error rate (ErrorRate). The state-of-the-art models being compared include J-Linkage (J-Lnk)[20], T-Linkage (T-Lnk) [7], RPA [8], RCMSA [74], ILP-RansaCov (ILP) [58], DGSAC [73] and NMU [75]. The comparisons are presented in Tab. IV. We observe that our SubspaceNet gives competitive results; in particular, our model with MIMI loss gives a mean error of 5.17%. We note the performance is achieved by training on only a very small amount of data (18 sequences) and without any dataset-specific parameter tuning. We also notice that our SubspaceNet is efficient at the inference stage (1.4 seconds) where the experiment setting is the same with Sect. IV-C.

F. Transfer Learning

In this section, we investigate the ability of proposed network to transfer beyond training datasets. In specific, we first carry out a cross country motion segmentation experiment by transferring the model trained on KT3DMoSeg (collected from Germany) to the Berkley DeepDrive 100k dataset (BDD) (collected from USA) [76]. Due to the lack of motion segmentation ground-truth, we only present qualitative results. We further evaluate transferring the model trained on FBMS59 to the Densely Annotated VIdeo Segmentation dataset 2017 unsupervised task (DAVIS). DAVIS has 90 sequences in total and is divided into train set (60 sequences) and validation set (30 sequences). We take notice that network is trained

TABLE III: Comparison against deep optical flow estimation methods on FBMS59.

Eval. Proto.	Train Set (Full)					Test Set (Full)	Test Set (10 frames)	
Method	MultiCut [72]	DeepFlow [69]	EpicFlow [70]	FlowFields [71]	FlowNet2 [68]	MultiCut	MultiCut	SubspaceNet
Deep	X	✓	✓	✓	✓	X	X	✓
F-measure	79.51	80.18	78.36	79.7	79.92	76.24	72.97	76.57

TABLE IV: AdelaideRMF two-view motion segmentation classification error (%). Inf. Time is inherited from [73] (- indicates unreported).

	State-of-the-Arts							SubspaceNet		
	J-Lnk [20]	T-Lnk [7]	RCMSA [74]	RPA [8]	ILP [58]	DGSAC [73]	NMU [75]	L2	CE	MIMI
Average	16.43	9.36	12.37	**5.49**	6.04	6.95	5.72	6.13	6.92	**5.17**
Median	14.29	7.80	9.87	4.57	4.27	4.98	**3.64**	4.50	5.21	**3.00**
Inf.Time	-	5.31	-	967.2	145.9	114.72	499.6	1.4	1.4	**1.4**

TABLE V: FBMS testset first 10 frames performance (%).

Model	SC [56]	SSC [24]	ALC [60]	MC [72]	SubspaceNet	
					L2	MIMI
Precision	87.44	53.11	**91.67**	89.05	85.62	85.70
Recall	60.77	56.40	50.57	61.81	69.09	**69.20**
Fmeasure	71.71	54.70	65.18	72.97	76.47	**76.57**

on FBMS59 only and inference is implemented on all 90 sequences of DAVIS. Moreover, DAVIS is focused on general video segmentation with more diverse foreground objects, e.g. airplanes and trains, that have never appeared in FBMS59. Thus it is able to validate that our network is agnostic to foreground semantic type. The qualitative results are provided for both DAVIS and BDD100K in Fig. 7. Thanks to the ground-truth annotation, a quantitative evaluation is available for DAVIS unsupervised tasks, we compared SubspaceNet with MIMI loss (Ours) with Spectral Clustering (SC) [56] and MultiCut (MC) [58]. The precision, recall and fmeasure are measured in the same way as FBMS59 for fair comparison. The results are presented in Tab. VI. Both the superior quantitative performance on DAVIS 2017 and the successful examples in BDD100K suggest the network is able to learn motion structures from known scenes and generalize to unknown scenes.

TABLE VI: Quantitative comparison of motion segmentation performance on DAVIS 2017 unsupervised task. Prec., Rec. and Fm. indicate precision, recall and fmeasure respectively. Numbers are in (% higher the better).

		Prec.	Rec.	Fm.		Prec.	Rec.	Fm.
Ours	TrainSet	**83.52**	64.01	**72.47**	ValSet	**83.06**	**69.12**	**75.45**
SC [56]		77.17	60.77	67.99		78.85	65.70	71.67
MC [72]		76.36	**64.27**	69.80		82.84	68.25	74.84

G. Model Selection

As can be seen from Fig. 6, the point distribution in the learned feature embedding is amenable for model selection. We evaluate the ability of both Second Order Difference (SOD) [16] and Silhouette Analysis (Silh.) [13] to estimate the number of motions. We also compare with alternative subspace clustering approaches with built-in model selection, namely, LRR [16], MSMC [62], SSC [24], GPCA [59], ALC [60] and additionally apply self-tuning spectral clustering(S.T.) [14] to the affinity matrix obtained in MVC [4]. Among the above competitors, the model selection for GPCA and SSC are implemented with SOD. Performances are evaluated in terms of mean classification error (Mean Err) and correct rate (Correct), i.e. the percentage of samples/sequences with correctly estimated number of cluster (higher the better). Comparisons are presented in Tab. VII. Thanks to the deep feature learning, both SOD and Silh. applied to our method yield substantially better performance without the need to tune any parameter.

TABLE VII: Comparison of model selection on KT3DMoSeg. Numbers are in %.

Method	**MIMI Loss**		S.T. [14]	LRR [16]	MSMC [62]	ALC [60]	GPCA [59]	SSC [24]
	SOD [16]	Silh. [13]						
Mean Err ↓	7.36	**7.25**	18.16	25.08	48.29	34.72	47.35	64.82
Correct ↑	**86.36**	81.82	40.91	54.55	22.73	45.45	18.18	18.18

V. Further Study

A. Sampling Imbalance

In this section, we further demonstrate the ability of our network to robustly handle sampling imbalance, i.e. the inlier points represent a minority. We demonstrate via a synthetic single-type multi-model fitting problems. Specifically, we synthesize 8,000 training samples and 200 testing samples for each of the type, line, circle and ellipses, and compare with RPA [8]. The results are presented in Fig. 8. We conclude that, first, our multi-model network performs comparably with RPA on multi-line segmentation task while outperforming RPA with large margin on the more challenging multi-circle and multi-ellipse tasks. The performance drops sharply from multi-line (blue) to multi-ellipse (green) fitting for RPA, with the drop getting more acute as the number of model increases. This suggests that the increasing size of the minimal support set (2 points for line, 3 points for circle and 5 points for ellipse) poses great challenge for the RANSAC-based approaches due to sampling imbalance. More precisely, in a noiseless N-model experiment, the chance of hitting the true model in a single sampling reduces from $(1/N)^2$ for straight line to $(1/N)^5$ for ellipse. It is evident that our multi-model network

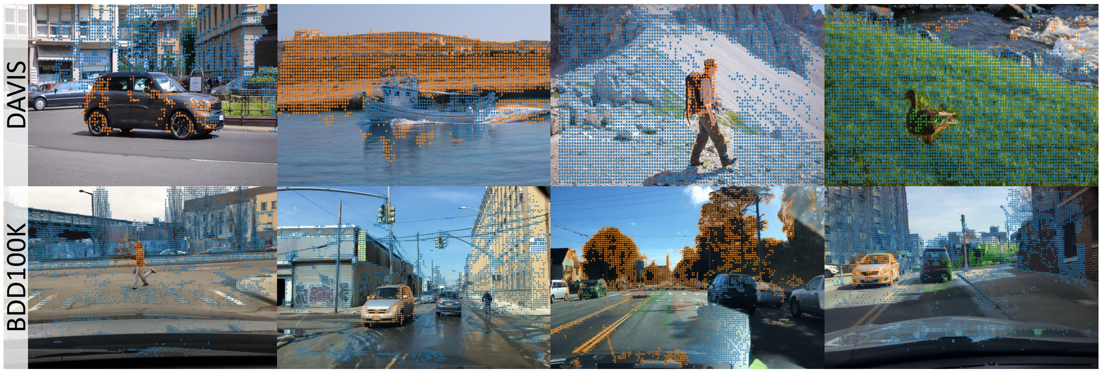


Fig. 7: Qualitative examples of transfer learning evaluations on DAVIS 2017 (top row) and Berkeley DeepDrive 100K (bottom row).

is less sensitive to the complexity of the model, as the drop in performance (purple and cyan bars) is less significant.

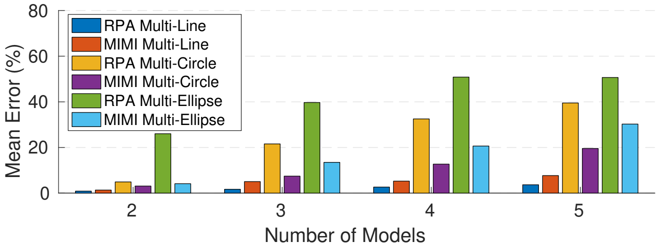


Fig. 8: Performance v.s. the number of models for synthetic multi-model fitting.

B. Feature Embedding

We provide direct visualization of the learnt representations. We use T-SNE [77] to project both the KT3DMoSeg raw feature points (of dimension ten for 5 frames) and network output embeddings to a 2-dimensional space. Three example sequences are presented in the last row of Fig. 6. We conclude from the figure that: (i) the original feature points are hard to be grouped by K-means correctly; and (ii) after our network embedding, feature points are more likely to be grouped according to the respective motions, regardless of the underlying types of motions.

C. Network Comparison

We compare SubspaceNet with alternative networks that are able to learn from sparse set of data. In particular, we compare with the correspondence network (CorresNet) [34] and PointNet [32] on KT3DMoSeg(KT3D.) and AdelaideRMF MoSeg(Adel.), both of which are experimented with L2 Loss and our MIMI loss. The results are presented in Fig. 9(a). We observe a significant performance gap between our SubspaceNet and the two alternatives. The proposed MIMI loss is also effective with alternative networks.

D. Dimension of Output Embedding

We investigate the impact of the dimension of the output embedding $\mathbf{z}$. We vary the size of the embedding dimension from 3 to 7 for three tasks and present the resulting error rates against the dimension in Fig 9 (b). As can be seen, the errors are relatively stable w.r.t. the output embedding dimension from 4 to 7 for all three tasks, with optimal dimension between 5 to 6 coinciding with the maximal number of clusters for each task (5 motions for KT3DMoSeg and 4 structures for Synthetic). Thus the maximal number of clusters serves as a good heuristic for the dimension of the network output embedding.

E. Weak Supervision

The SubspaceNet is trained on labelled data points which is often very costly to obtain compared with image category labels. In this section, we investigate the interaction between weaker supervision, i.e. fewer labelled data points and performance. In specific, we randomly subsample 20% to 80% labelled data points for each sequence in KT3DMoSeg and AdelaideRMF MoSeg and train the model with reduced labelled data while keeping the same evaluation protocol as normal. The results averaged over 10 trials are presented in Fig. 9 (c). We observe very stable error rate from 40% subsample rate, suggesting the SubspaceNet is robust to fewer annotated data. This discovery also opens new research opportunities regarding exploiting fewer labeled data, a.k.a. semi-supervised learning.

F. Network Design

In this section, we make more ablation study of the proposed SubspaceNet. In particular, we are concerned with the depth of the network, the necessity of the L2 normalization layer.

a) Network Depth: We evaluate the impact of the depth of SubspaceNet. The depth is varied from 20 to 80 with step of 10 and both the mean error and median error on KT3DMoSeg (KT3D.) and AdelaideRMF MoSeg (Adel.) are reported in Fig. 9 (d). We observe relatively stable performance w.r.t the depth of network thanks to the ResNet structure. In particular, the optimal range is between 40 to 60.

b) L2 Normalization Layer: We introduced a L2 normalization layer at the output of SubspaceNet. This layer normalizes the scales of all feature embeddings so that all feature points lie on a unit sphere, thereby benefitting the metric

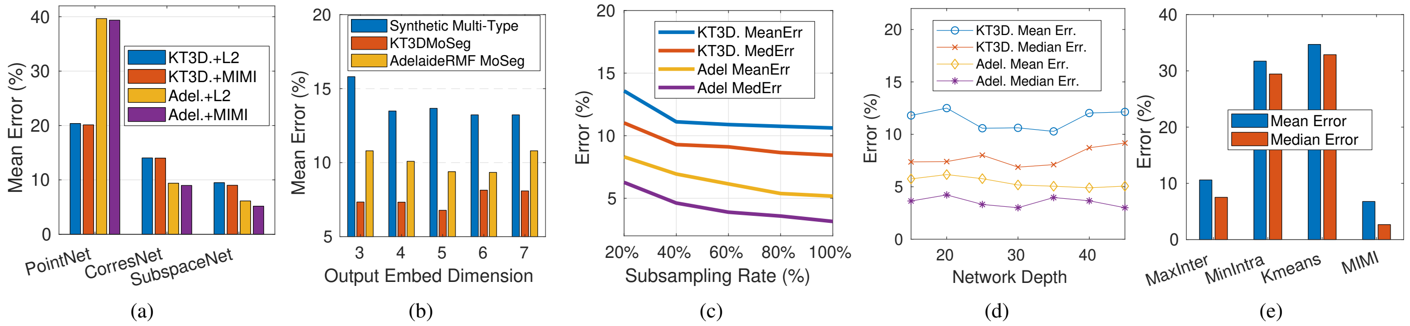


Fig. 9: (a) Comparison with alternative networks. (b) Performance v.s. the network output dimension. (c) Weak supervision. (d) Network depth v.s. the performance. (e) Different variants of MIMI loss.

learning procedure. We specifically evaluate the necessity of this layer by comparing the results on motion segmentation with and without the L2 normalization layer. As can be seen from Tab. VIII, the performance is consistently better with the L2norm layer for both the KT3DMoSeg and AdelaideRMF MoSeg datasets, suggesting that the L2norm layer is beneficial for learning better feature embeddings.

TABLE VIII: Comparison of with (w) L2norm layer and without (w/o) L2norm layer. The numbers are in $\%$.

	w L2norm		w/o L2norm	
Dataset	Mean	Med.	Mean	Med.
KT3DMoSeg	10.62	8.44	11.56	8.78
AdelaideMoSeg	5.17	3.00	6.68	3.81

G. *MIMI Loss Components*

Here we investigate the necessity of both maximizing inter cluster distance and minimizing intra cluster variance. Specifically, we compare the following variants. (i) **MaxInter**: only maximizing the inter cluster distance is considered, equivalent to the first term in Eq (9). (ii) **MinIntra**: only minimizing the intra cluster variance is considered, the second term in Eq (9). (iii) **K-means loss**: we further note the k-means loss [64] proposed for unsupervised deep clustering shares the same objective with **MinIntra**. We therefore adapt the k-means loss to supervised learning with fixed point-to-cluster assignment during training. We compare the three variants with our final MIMI loss on KT3DMoSeg and present the results in Fig. 9 (e). The MIMI loss is consistently better (lower error) than all three variants. In particular, the **MinIntra** and **K-means loss** produce large errors. This indicates that pushing points of different clusters away is vital to feature embedding for clustering.

$$L(\Theta) = -\log \min_{m,n} ||\boldsymbol{\mu}_m - \boldsymbol{\mu}_n||_2^2 + \log \max_{l} s_l \tag{9}$$

H. *Training*

Due to the limited size of existing subspace clustering datasets, exhibiting a low diversity of motion-scene types for motion segmentation, one might suspect the risk of overfitting. In this section, we investigate this issue by visualizing both the training/validation loss and errors. The results on both KT3DMoSeg and AdelaideRMF MoSeg are shown in Fig. 10. We observe both training and validation loss converging after 100 epochs as does the prediction accuracy. There is still a gap between training and validation accuracy suggesting the challenge of generalization gap.

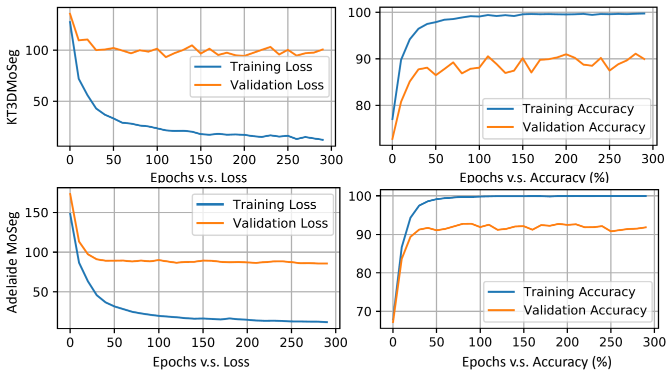


Fig. 10: Training procedure: epochs v.s. loss and accuracy.

VI. Conclusion

In this work, we investigate training a deep neural network for general multi-type subspace clustering. We formulate the problem as learning non-linear feature embeddings that maximize the distance between points of different clusters and minimize the variance within clusters. For inference, the output features are fed into a K-means to obtain the grouping. Model selection is easily achieved by just analyzing the K-means residual in a parameter free manner. Experiments are carried out on both synthetic and real motion segmentation tasks. Comparison with state-of-the-art approaches proves that our network can better deal with multiple types of models simultaneously. Our method is also less sensitive to sampling imbalance brought about by the increasing number of models, and it is highly efficient at inference stage. As future works, one could consider including additional texture and color information and adopting sliding window technique to handle arbitrary long sequences.

References

[1] C. Rother, "A new approach to vanishing point detection in architectural environments," *Image and Vision Computing*, 2002.

[2] Y. Sugaya and K. Kanatani, "Geometric structure of degeneracy for multi-body motion segmentation," in *In Workshop on Statistical Methods in Video Processing*, 2004.

[3] P. H. Torr, "Geometric motion segmentation and model selection," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 1998.

[4] X. Xu, L. F. Cheong, and Z. Li, "Motion segmentation by exploiting complementary geometric models," in *CVPR*, 2018.

[5] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *CVPR*, 2003.

[6] D. Barath and J. Matas, "Multi-class model fitting by energy minimization and mode-seeking," in *CVPR*, 2018.

[7] L. Magri and A. Fusiello, "T-linkage: A continuous relaxation of J-linkage for multi-model fitting," in *CVPR*, 2014.

[8] ——, "Robust Multiple Model Fitting with Preference Analysis and Low-rank Approximation," in *BMVC*, 2015.

[9] T. Lai, H. Wang, Y. Yan, T. J. Chin, and W. L. Zhao, "Motion Segmentation Via a Sparsity Constraint," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[10] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2001.

[11] T. Chin, H. Wang, and D. Suter, "The ordered residual kernel for robust motion subspace clustering," in *NIPS*, 2009.

[12] Z. Li, J. Guo, L.-F. Cheong, and S. Zhiying Zhou, "Perspective motion segmentation via collaborative clustering," in *ICCV*, 2013.

[13] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, 1987.

[14] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *NIPS*, 2005.

[15] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, 2007.

[16] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[17] Z. Li, L. F. Cheong, and S. Z. Zhou, "SCAMS: Simultaneous clustering and model selection," in *CVPR*, 2014.

[18] E. Vincent and R. Laganiére, "Detecting planar homographies in an image pair," in *the 2nd International Symposium on Image and Signal Processing and Analysis*, 2001.

[19] Y. Kanazawa and H. Kawakami, "Detection of planar regions with uncalibrated stereo using distributions of feature points," in *BMVC*, 2004.

[20] R. Toldo and A. Fusiello, "Robust multiple structures estimation with j-linkage," in *ECCV*, 2008.

[21] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *International Journal of Computer Vision*, 2012.

[22] H. Li, "Two-view motion segmentation from linear programming relaxation," in *CVPR*, 2007.

[23] T.-J. Chin, J. Yu, and D. Suter, "Accelerated hypothesis generation for multi-structure robust fitting," in *ECCV*, 2010.

[24] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[25] C. Alzate and J. A. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel pca," *IEEE transactions on pattern analysis and machine intelligence*, 2010.

[26] M. Soltanolkotabi, E. J. Candes *et al.*, "A geometric analysis of subspace clustering with outliers," *The Annals of Statistics*, 2012.

[27] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *CVPR*, 2007.

[28] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," *arXiv preprint arXiv:1606.03798*, 2016.

[29] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *CVPR*, 2017.

[30] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Relative camera pose estimation using convolutional neural networks," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, 2017.

[31] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *CVPR*, 2017.

[32] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *CVPR*, 2017.

[33] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NIPS*, 2017.

[34] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *CVPR*, 2018.

[35] F. J. Huang, Y.-L. Boureau, Y. LeCun *et al.*, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *CVPR*, 2007.

[36] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*, 2008.

[37] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *ICML*, 2009.

[38] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering." in *AAAI*, 2014.

[39] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *CVPR*, 2016.

[40] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *NIPS*, 2017.

[41] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, 2016.

[42] E. Xing, M. Jordan, S. Russell, and A. Ng, "Distance metric learning with application to clustering with side-information," in *NIPS*, 2003.

[43] S. Chopra, R. Hadsell, and L. Y., "Learning a similiarty metric discriminatively, with application to face verification," in *CVPR*, 2005.

[44] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *CVPR*, 2015.

[45] K. Sohn, "Improved Deep Metric Learning with Multi-class N-pair Loss Objective," in *NIPS*, 2016.

[46] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *ICASSP*, 2016.

[47] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy, "Deep metric learning via facility location," in *CVPR*, 2017.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[49] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[51] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *ACM KDD*, 1996.

[52] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, 2000.

[53] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *International Journal of Computer Vision*, 2012.

[54] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 21, no. 5, pp. 476–480, 1999.

[55] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, 2013.

[56] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE transactions on pattern analysis and machine intelligence*, 2014.

[57] H. S. Wong, T.-J. Chin, J. Yu, and D. Suter, "Dynamic and hierarchical multi-structure geometric model fitting," in *ICCV*, 2011.

[58] L. Magri and A. Fusiello, "Multiple model fitting as a set coverage problem," in *CVPR*, 2016.

[59] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE transactions on pattern analysis and machine intelligence*, 2005.

[60] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[61] J. Yan and M. Pollefeys, "A General Framework for Motion Segmentation : Degenerate and Non-degenerate," in *ECCV*, 2006.

[62] R. Dragon, B. Rosenhahn, and J. Ostermann, "Multi-scale clustering of frame-to-frame correspondences for motion segmentation," in *ECCV*, 2012.

[63] P. Ji, M. Salzmann, and H. Li, "Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data," in *ICCV*, 2016.

[64] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *ICML*, 2017.

[65] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015.

[66] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *CVPR*, 2016.

[67] H. Oh Song, S. Jegelka, V. Rathod, and K. Murphy, "Deep metric learning via facility location," in *CVPR*, 2017.

[68] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *CVPR*, 2017.

[69] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *ICCV*, 2013.

[70] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *CVPR*, 2015.

[71] C. Bailer, B. Taetz, and D. Stricker, "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation," in *ICCV*, 2015.

[72] M. Keuper, B. Andres, and T. Brox, "Motion trajectory segmentation via minimum cost multicuts," in *ICCV*, 2015.

[73] L. Tiwari and S. Anand, "Dgsac: Density guided sampling and consensus," in *WACV*, 2018.

[74] T. T. Pham, T.-J. Chin, J. Yu, and D. Suter, "The random cluster model for robust geometric fitting," *IEEE transactions on pattern analysis and machine intelligence*, 2014.

[75] M. Tepper and G. Sapiro, "Nonnegative matrix underapproximation for robust multiple model fitting," in *CVPR*, 2017.

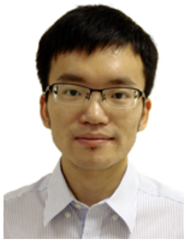
[76] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[77] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, 2008.

Xun Xu received the PhD degree from Queen Mary University of London in 2016. He was currently a research fellow with National University of Singapore from 2016 to 2019. He is now with Institute of Infocomm Research (I2R), A*STAR. His research interests include semi-supervised learning, active learning, adversarial learning, 3D point cloud, motion segmentation, etc.

Le Zhang received the BEng degree from the University of Electronic Science and Technology of China, in 2011, and the MSc and PhD degrees form Nanyang Technological University (NTU), in 2012 and 2016, respectively. Currently, he is a scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. He served as TPC member in several conferences such as AAAI, IJCAI. He has served as a guest editor of the Pattern Recognition and the Neurocomputing. His current research interests include deep learning and computer vision.

Loong-Fah Cheong received the BEng degree from the National University of Singapore, and the PhD degree from the University of Mary- land at College Park, Center for Automation Research, in 1990 and 1996, respectively. In 1996, he joined the Department of Electrical and Computer Engineering, National University of Singapore, where he is an associate professor currently. His research interests include the processes in the perception of three-dimensional motion, shape, and their relationship, as well as the 3D motion segmentation and the change detection problems.



Zhuwen Li received the B.E. in Computer Science from Tianjin University in 2008, the Master's degree in Computer Science from Zhejiang University in 2011, and the Ph.D. degree in Electrical and Computer Engineering from National University of Singapore in 2014. Currently, he is a Postdoc Researcher at Intel Intelligent Systems Lab. His research interests include motion analysis, 3D vision and graph neural networks.

Ce Zhu received the B.S. degree from Sichuan University, Chengdu, China, in 1989, and the M.Eng and Ph.D. degrees from Southeast University, Nanjing, China, in 1992 and 1994, respectively, all in electronic and information engineering. He held a post-doctoral research position with the Chinese University of Hong Kong in 1995, the City University of Hong Kong, and the University of Melbourne, Australia, from 1996 to 1998. He was with Nanyang Technological University, Singapore, for 14 years from 1998 to 2012, where he was a Research Fellow, a Program Manager, an Assistant Professor, and then promoted to an Associate Professor in 2005. He has been with University of Electronic Science and Technology of China, Chengdu, China, as a Professor since 2012. His research interests include video coding and communications, video analysis and processing, 3D video, visual perception and applications. He has served on the editorial boards of a few journals, including as an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON BROADCASTING, IEEE SIGNAL PROCESSING LETTERS, an Editor of IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and an Area Editor of SIGNAL PROCESSING: IMAGE COMMUNICATION. He has also served as a Guest Editor of a few special issues in international journals, including as a Guest Editor in the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING. He is an APSIPA Distinguished Lecturer (2021-2022), and was also an IEEE Distinguished Lecturer of Circuits and Systems Society (2019-2020).