TMCOSS: Thresholded Multi-Criteria Online Subset Selection for Data-Efficient Autonomous Driving

Soumi Das
IIT Kharagpur

Harikrishna Patibandla
IIT Kharagpur

Suparna Bhattacharya
Hewlett Packard Labs, HPE Enterprise

Kshounis Bera
HPE Enterprise

Niloy Ganguly
IIT Kharagpur

Sourangshu Bhattacharya
IIT Kharagpur

Abstract

Training vision-based Autonomous driving models is a challenging problem with enormous practical implications. One of the main challenges is the requirement of storage and processing of vast volumes of (possibly redundant) driving video data. In this paper, we study the problem of data-efficient training of autonomous driving systems. We argue that in the context of an edge-device deployment, multi-criteria online video frame subset selection is an appropriate technique for developing such frameworks. We study existing convex optimization based solutions and show that they are unable to provide solution with high weightage to loss of selected video frames. We design a novel multi-criteria online subset selection algorithm, ***TMCOSS****, which uses a thresholded concave function of selection variables. Extensive experiments using driving simulator CARLA show that we are able to drop 80% of the frames, while succeeding to complete 100% of the episodes. We also show that TMCOSS improves performance on the crucial affordance "Relative Angle" during turns, on inclusion of bucket-specific relative angle loss (****BL****), leading to selection of more frames in those parts. TMCOSS also achieves an 80% reduction in number of training video frames, on real-world videos from the standard BDD and Cityscapes datasets, for the tasks of drivable area segmentation, and semantic segmentation.*

1. Introduction

Many A.I.-based autonomous driving applications e.g. affordance-based driving models [21], semantic segmentation models [5], drivable area detection [24], etc., need to collect a large amount of video data from edge devices for training machine learning models. However, much of the input video contains redundant information from the task point of view. For example, in the case of affordance-based driving models, training using many frames on straight road

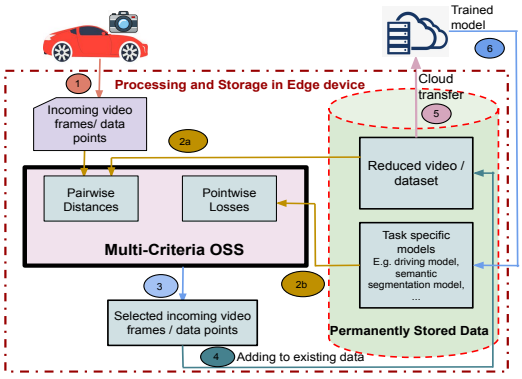

Figure 1: **Data subset selection framework on edge devices for reduced training dataset collection.**

sections may not be necessary; while one may need a lot of frames in the turns for training. In this paper, we are interested in developing an online subset selection (OSS) algorithm [13] which can be deployed on edge devices, and can select the most informative video frames used for training one or more models. Figure 1 shows an overview of our application scenario.

The problem of video frame subset selection has been studied in various contexts, including video summarization [13, 19], video recognition[23], video fast forwarding [17], etc. Deep reinforcement learning based methods, which learn a frame skipping network, with reward for better performance [17, 22], or more confidence [23] on the end task are infeasible for deployment on edge devices since they require multiple evaluations of the end task objective (e.g. error on a validation set or fraction of test episodes completed), for learning the parameters of the skipping network. The external criteria based methods which optimize a global criterion between selected frames and whole video, e.g. perceptual similarity [13], manifold spanning [15], etc are appropriate for our setting, but do not incorporate signals from A.I. tasks, e.g. loss from the currently trained

models. [12] ensures compatibility between the consecutive selected video segments through a Markov Model. The OSS framework [13] was extended to incorporate pointwise loss of the selected frames in a composite criteria [9], and is the most appropriate for our setting.

In this paper, we build on the multi-criteria OSS framework (MCOSS) [9], where at each step, an existing set of selected frames is supplemented by the most relevant frames from an incoming set, based on the sum of total dissimilarity between selected frames and incoming frames, and the pointwise loss incurred by incoming selected frames. However, through a rigorous analysis, we show that, additive incorporation of pointwise loss criteria in [9] suffers from selection of fewer frames from incoming set as we provide higher weightage to the pointwise criteria. This is due to the fact that additive incorporation suffers from multiple counting of loss of selected points thus leading to selection of fewer frames. We propose a novel thresholded MCOSS formulation (**TMCOSS**) which alleviates this problem while retaining the convexity of the optimization problem. We also propose **SubMCOSS**, which to the best of our knowledge, is the first submodular set function-based formulation for online subset selection incorporating pointwise criteria.

We evaluate the video frame selection performance of TMCOSS using the autonomous driving simulator CARLA [11] for the CAL driving model [21] task, as well as on real-world datasets of Berkeley Deep-drive (BDD) [24] for drivable area segmentation task and Cityscape dataset [8] for semantic segmentation tasks, using DeepLabv3+ model [5].

For the CAL model task, we define a novel bucket-specific loss (**BL**) for the crucial relative angle affordance. We show that **TMCOSS** along with bucket-specific loss (**BL**), called **TMCOSS-BL**, can achieve a 100:20 compression (selecting 1 in 5 frames) while completing 10 out of 10 episodes in 4 different driving conditions, though models trained on data collected from MCOSS can can only complete 7 episodes for the same compression ratio. Empirically, we also show that both on synthetic data, as well as data from selection of video frames for autonomous driving, TMCOSS performs better than approximation algorithms for submodular maximization [1, 2] using SubMCOSS. On the semantic segmentation task, TMCOSS achieves a 100:20 compression with a 1% decrease in overall mIoU score, while MCOSS suffers a decrease 8% in mIOU. Finally, on drivable area segmentation task, we achieve a compression 100:20 with only 1% decrease mIoU, compared to 5% decrease for MCOSS. In all cases, datasets selected using TMCOSS are more informative than any of the individual criteria. To summarize, our main contributions are:

- We introduce the problem of designing *data-efficient autonomous driving* platforms, with a key challenge being multi-criteria OSS.
- We propose two novel multi-criteria OSS methods. The convex relaxation based method, **TMCOSS** is theoretically motivated and empirically superior to multiple recent state-of-the-art baselines.
- We propose a novel bucket specific relative angle loss (**BL**), which when used with TMCOSS (**TMCOSS-BL**), provide state-of-the-art compression on CAL driving model task. **TMCOSS** also demonstrates significant savings in training data requirements on benchmark real-world datasets.

1.1. Related Work

We describe two broad classes of relevant prior works: (a) Applications related to self-driving tasks and (b) Video frame Subset selection. CARLA [11] is an well-established driving simulator for the task of autonomous driving. Codevilla et al. [6] proposed imitation learning based approaches, which were then bettered by the Conditional Affordance Learning (CAL) model [21]. Recently, [7] proposed a conditional imitation learning model, CILRS, and a reinforcement learning approach for collecting better on-policy data [20], both of which reportedly perform better than CAL [21]. For the purpose of demonstrating the effectiveness of TMCOSS, we have used the CAL [21] as a driving model which in our opinion is still a good model for training driving models on CARLA simulator. We reiterate that TMCOSS can be used with any imitation learning approach including [7]. We also studied the effect of subset selection on tasks of drivable area detection [24] [18] which is essentially segmenting drivable roads and alternate drivable roads, and semantic segmentation [4] [5] [16] on driving datasets. We used DeepLabV3+ [5] for both tasks for its consistent robust performance over other existing models.

Video frame subset selection:
Recent online subset selection approaches can be divided into two broad classes (1) deep learning based, and (2) based on optimization of some input criteria. The first class of techniques [17], [22] [23] depend on selection networks added to the pipeline of existing tasks and are trained jointly. They are able to learn complex selection criteria through deep models, but do not come with any stated selection criteria, w.r.t. which the selection is optimal. These methods are jointly trained with the end objective, e.g. video recognition for [22] and [23], and are typically too expensive to be deployed on edge devices.

The second class of techniques, which are also closest to our approach, selects the data points based on different defined criteria. These criteria include reconstruction error [3], linear dependency [15], perceptual similarity [13] or criteria based on end tasks such as distinctiveness and uncertainty [14] [9]. [15] recently proposed an online approach based on linear dependence criteria. However for the current problem, we build on the pairwise criteria based

approach proposed in [13], and extend to include multiple pairwise and pointwise criteria [9]. While these approaches were used for the problems of video summarization [13] and semantic segmentation [9], their setting provides flexibility of incorporating multiple different types of criteria, which is relevant to our application. We provide detailed comparison with these approaches.

2. Data Efficient Autonomous Driving

In this section, we describe the problem of data efficient training of autonomous driving models, with the core idea being selection of relevant video segments in an online setting. We formulate this problem as an online subset selection problem (OSS) for selecting subsets of video frames, given input signals from existing selected video frames, and the trained model, which involve multiple input criteria - both for pairs of input frames and for single input frame. Section 2.1 describes the setup of data efficient autonomous driving and its connection to multi-criteria OSS problem formulation. Section 2.2 points out a drawback with existing multi-criteria OSS formulation. Sections 2.3 and 2.4 describe our new formulations for the multi-criteria OSS problem.

2.1. Problem setup and OSS

Training of vision based autonomous driving models [21, 20] requires processing of large amounts of annotated video data. In many cases, the videos are collected in episodes over a period of time, leading to processing and training of models in batches which are ordered in time. Hence, the batchwise OSS scheme discussed in [13], is an ideal setting for selection of video frames in this context. We denote a complete dataset as $\mathcal{D} = \{(x_i, y_i),\ i = 1, \ldots, n\}$ where n is the total number of datapoints (annotated video frames) in the dataset, x_i are the features extracted from video frames, and labels y_i corresponding to various learning tasks, e.g. affordances [21]. Let $X_t = \{(x_i, y_i),\ i = 1, \ldots, m\},\ t = 1, \ldots, T$ denote the t^{th} batch of episodes collected, where m denotes the number frames in a batch [1]. Hence $mT = n$. We also define the cumulative sets $C_t = \cup_{i=1}^{t} X_i$ denoting all data collected till batch t. We are interested in constructing representative set $R_t \subseteq C_t$, which consists of a representative set of frames till batch t. The intention here is that an autonomous driving model M_t trained on the cumulative set C_t, should perform similar to another model M'_t trained on the representative set of frames R_t in terms of an end performance metric, e.g. the episode completion metric used in [21]. Furthermore, the size of R_t should be small so that R_t uses lower storage space and communication bandwidth, and training of M'_t potentially takes lower time. An algorithm for selection of R_{t+1} from X_{t+1}, given C_t, and M'_t thus constitutes a *data efficient scheme* for training of autonomous driving models, since we are only storing and processing R_t's. Note that, this scheme can also be used in the reinforcement learning schemes for improvement of driving policies such as the one described in [20], where X_t can be taken from the replay buffer at iteration t.

For the OSS formulation, we focus on an input batch of episodes X_{t+1}. The selection algorithm uses two input sets of frames R_t and X_{t+1}, here referred to as the old set (superscript o) and new set (superscript n) respectively, following notation used in [13]. Let d^o_{ij} denote a dissimilarity measure between new frame i (from X_{t+1}) and old frame j (from R_t), and d^n_{ij} denote the dissimilarity between new frames i and j (both from X_{t+1}). The OSS formulation minimizes the composite criteria with two parts: (1) total dissimilarity between the "representative frames" (either from new or old set) and the incoming frame it represents, and (2) number of representative frames from the new set. Let z^o_{ij}, z^n_{ij} be the relaxed binary assignment variables ($Z_{ij} \in [0, 1]$), where $z^o_{ij} = 1$ denotes that the representative of i^{th} new example ($(x_i, y_i) \in X_{t+1}$) is j^{th} old example ($(x_j, y_j) \in R_t$), and $z^n_{ij} = 1$ denotes that representative of i^{th} new example ($(x_i, y_i) \in X_{t+1}$) is the j^{th} new example ($(x_j, y_j) \in X_{t+1}$). Otherwise, $z_{ij} = 0$. Hence any solution for optimal representative allocation should satisfy the constraint: $\sum_{j=1}^{|R_t|} z^o_{i,j} + \sum_{j=1}^{m} z^n_{i,j} = 1$, asserting that every frame $i \in X_{t+1}$ has exactly one representative. The objective function can be written as:

$$L(z^o_{ij}, z^n_{ij}) = \sum_{i=1}^{m}\sum_{j=1}^{|R_t|} z^o_{ij} d^o_{ij} + \sum_{i,j=1}^{m} z^n_{ij} d^n_{ij} + \lambda \sum_{j=1}^{m} \|[z^n_{1,j} \ldots z^n_{m,j}]\|_p$$

Das et al. [9] has incorporated both pairwise scores (e.g. distance d_{ij} between pairs of frames i, j) and pointwise scores (e.g. negative loss $-L_i$ for the frame i). The modified *cumulative dissimilarity* function Q_{ij} is a weighted sum of d_{ij} and L_j - the loss incurred by the representative point. Thus $Q_{ij} = \rho d_{ij} - (1 - \rho) L_j$. Let L^n_i denote the pointwise attribute (here loss value) for datapoint i in X_{t+1} and analogously for L^o_i (denoting loss for datapoint i in R_t). The final formulation is:

$$\begin{aligned}
&\min_{z^o_{ij}, z^n_{ij}} \sum_{i=1}^{m}\sum_{j=1}^{|R_t|} z^o_{ij} Q^o_{ij} + \sum_{i,j=1}^{m} z^n_{ij} Q^n_{ij} + \lambda \sum_{j=1}^{m} \|[z^n_{1,j} \ldots z^n_{m,j}]\|_p \\
&s.t. \sum_{j=1}^{|R_t|} z^o_{i,j} + \sum_{j=1}^{m} z^n_{i,j} = 1,\ \forall i \in X_{t+1} \\
&z^n_{i,j}, z^o_{i,j} \in [0, 1],\ \forall i, j
\end{aligned} \tag{1}$$

where $Q^n_{ij} = \rho d^n_{ij} - (1 - \rho) L^n_j$ and $Q^o_{ij} = \rho d^o_{ij} - (1 - \rho) L^o_j$. This is a convex optimization problem which can be solved efficiently for moderate sizes of sets X_{t+1} and R_t using off-the-shelf solvers, e.g. CVXPY [10]. We call this formulation *multi-critria OSS* (MCOSS).

[1] Equal batch size is for simplicity of exposition, not a requirement

2.2. Analysis of Multi-Criteria OSS

While applying MCOSS to our problem, we noticed that as we give higher weightage to the pointwise component by choosing lower value of ρ, the number of selected points decreases. From an application point of view, this allows the pointwise score to have a limited impact on the set of points selected. This might be tolerable in certain applications, e.g. semantic segmentation where the perceptual dissimilarity measure contains sufficient information for frame subset selection. However for the application of autonomous driving, we find that task-wise and situation-wise losses have much more impact on the quality of frames selected.

To understand the mechanism through which this problem arises, we observe that using only pointwise metric yields a maximum of one representative for all images belonging to incoming set X_{t+1}. This is the setting when ρ = 0. We have $Q^o_{ij} = -(1-\rho)L^o_j$ and $Q^n_{ij} = -(1-\rho)L^n_j$, both of them are constant across i. The representative j of any instance $i \in X_{t+1}$ $(i = 1, ..., m)$ will be from X_{t+1} if $L^n_j > L_{j'} \forall j' \in X_{t+1}$ in which case only one point will be selected (see Corollary 1.1). Otherwise the representative will be from R_t, in which case no points are selected.

While the above intuitions are motivated for special case of $\rho = 0$, the ideas also apply to more general values of $0 < \rho \leq 1$. We further illustrate this by characterising the solution of formulation 1 in the following theorem.

Theorem 1 *Let z^o_{ij} and z^n_{ij} be the optimal solution for formulation 1. A new frame $j \in X_{t+1}$ is selected as a representative frame for at least one incoming frame $i \in X_{t+1}$, i.e. $z^n_{ij} = 1$, only if BOTH these conditions hold:*

- *For some incoming frame $i \in X_{t+1}$, $Q^n_{ij} < Q^n_{ij'}$, for all $j' \in X_{t+1}$ and $j' \neq j$*
- *For some incoming frame $i \in X_{t+1}$, $Q^n_{ij} < \frac{\sum_{i'=1}^m z^o_{i',k} Q^o_{i'k} + \lambda \|[z^n_{1,j} ... z^n_{m,j}]\|_p}{\|\mathbf{z}^n_j\|_1}$*

where $k = argmin_j \sum_{i=1}^m z^o_{i,j} Q^o_{i,j}$, and $\|\mathbf{z}^n_j\|_1 = \sum_{i'=1}^m z^n_{i'j}$

Due to space constraints, we provide the formal proof in the supplementary material. Note that the first condition states that there is at least one frame i in the incoming set whose cumulative dissimilarity Q^n_{ij} is lower than all other points. The second condition signifies that cumulative dissimilarity Q^n_{ij} between a representative j and the point it is representing i is lower than minimal contribution from a potential representative k from existing set of selected examples $k \in R_t$. Next, we provide two corollaries to illustrate our point. Corollary 1.1 illustrates the conditions in Theorem 1 for the special case of $\rho = 0$. Since, the dependence on i is removed, it is easy to see that at most one $j \in X_{t+1}$ will satisfy the condition.

Corollary 1.1 *Let z^o_{ij} and z^n_{ij} be the optimal solution for formulation 1. A new frame $j \in X_{t+1}$ is selected as a representative frame for at least one incoming frame $i \in X_{t+1}$, i.e. $z^n_{ij} = 1$, only if BOTH these conditions hold:*

- *$L^n_j > L^n_{j'}$ for all $j' \in X_{t+1}$ and $j' \neq j$*
- *$L^n_j > \frac{\sum_{i=1}^m z^o_{i,k} L^o_k - \lambda \|[z^n_{1,j} ... z^n_{m,j}]\|_p}{\|\mathbf{z}^n_j\|_1}$*

where $k = argmin_j \sum_{i=1}^m z^o_{i,j} Q^o_{i,j}$, and $\|\mathbf{z}^n_j\|_1 = \sum_{i'=1}^m z^n_{i'j}$

Corollary 1.2 *Let $\Delta_d(i,j) = \|\mathbf{z}^n_j\|_1 d^n_{ij} - \sum_{i'=1}^m z^o_{i'k} d^o_{i'k}$ and $\Delta_L(j) = \|\mathbf{z}^n_j\|_1 L^n_j - \sum_{i'=1}^m z^o_{i'k} L^o_k$. If $\Delta_d(i,j) < -\Delta_L(j)$ for all z^n_{ij}, z^o_{ij}, and for $\rho = 0$, $j \in X_{t+1}$ is not a representative frame, then for some $\rho \geq 0$, Theorem 1 will not be satisfied by any pair $i, j \in X_{t+1}$.*

Corollary 1.2 states that if a frame $j \in X_{t+1}$ is not a representative, and satisfies the conditions on $\Delta_d(i,j)$ and $\Delta_L(j)$, then it will stop being a representative for some value of $\rho \geq 0$. By rearranging the terms in second condition of theorem 1, we get: $\rho\Delta_d(i,j) - (1-\rho)\Delta_L(j) \leq \lambda \frac{\|\mathbf{z}^n_j\|_p}{\|\mathbf{z}^n_j\|_p}$. For $p = 1$ the RHS is constant, but LHS decreases with ρ. Hence the second condition of Theorem 1 is not satisfied by any $i \in X_{t+1}$ for the given candidate representative frame $j \in X_{t+1}$. These results motivate us to look for better formulations of multi-criteria OSS problem.

2.3. Submodular Multi-Criteria OSS

In this section, we describe an algorithm for multi-criteria OSS problem based on submodular optimization. The problem can be posed as a set function incorporating both pairwise and pointwise attributes and can be solved using submodular optimisation. The natural criteria used for selection is the pre-defined modified *cumulative dissimilarity* function Q_{ij}.

For every set S, the set function $f(S)$ can be defined as:

$$f(S) = \sum_{i \in X} min\{min_{j \in R} Q_{ij}, min_{j \in S} Q_{ij}\} \tag{2}$$

Here, the problem is solved by selecting a representative j which contributes the least dissimilarity value Q_{ij} to the incoming instances $i \in X$. By definition, we can say

Remark 1 *-f(S) is submodular.*

For proof, see supplementary. We can thus pose it as a submodular maximisation problem by solving the problem $min_{S \subseteq X} f(S)$. We call this formulation *submodular multi-criteria OSS* (SubMCOSS). We define a greedy submodular maximisation approach for solving the optimisation problem in Algorithm 1. The algorithm is a randomised greedy algorithm that examines the dataset k times to select the representatives for incoming data. We then show a thresholded convex approach for solving multi-criteria OSS problem.

Algorithm 1 : Submodular Multi-Criteria OSS

Input:
S_0: Initial representative set = ϕ
X: Incoming Set of Instances
k: Subset cardinality , $f(S)$: Objective function
Process:
for i = 1,2, . . ., k **do**
  for each x $\in$ X $\setminus S_{i-1}$ **do**
    $fv_x \leftarrow f(S_{i-1} \cup x)$
  end for
  Let $M_i \in X \setminus S_{i-1}$ be subset of top k elements maximising $\sum_{m \in M_i} fv_m$
  Let u_i be randomly sampled from M_i
  $S_i \leftarrow S_{i-1} \cup u_i$
end for
Output:
S_k : Subset of size k

2.4. Thresholded Convex multi-criteria OSS

SubMCOSS , described in previous section uses the natural formulation of weighted linear aggregation of pointwise and pairwise loss function. However, the algorithm for submodular optimization is a randomized approximation algorithm, and also computationally expensive due to multiple sampling runs required for a good optimal subset. In this section, we describe a novel convex formulation of multi-criteria OSS which alleviates the problems of MCOSS (Equation 1) as well as SubMCOSS (Algorithm 1).

The key observation which helps us in designing the novel algorithm is that in MCOSS (Equation 1), it is possible for a frame $j \in X_{t+1}$ to contribute $-m(1-\rho)L^n_j$ by becoming a representative for every point $i \in X_{t+1}$ (see that the terms involving pointwise loss add up to $-(1-\rho)(\sum_{i=1}^m z^n_{ij})L^n_j$). However, in reality it only adds one data point to the training set with the pointwise score of $-L^n_j$. This problem is alleviated by using a coefficient of L^n_j which is an indicator of whether j is a representative point or not, rather than $(\sum_{i=1}^m z^n_{ij})$ which counts the number of points represented by j. This is achieved by using a concave function S_j of z_{ij}: $S_j = \frac{1}{\epsilon}\min(\epsilon, \sum_{i=1}^m z_{ij})$ where ϵ is an input parameter. The modified objective function becomes: $\mathcal{G}(z^o_{ij}, z^n_{ij}) = \rho(\sum_{i=1}^m \sum_{j=1}^{|R_t|} z^o_{ij} d^o_{ij}(t) + \sum_{i,j=1}^m z^n_{ij} d^n_{ij}(t)) - (1-\rho)(\sum_{j=1}^{|R_t|} S^o_j * L^o_j + \sum_{j=1}^m S^n_j * L^n_j)$, where, $S^o_j = \frac{1}{\epsilon}\min(\epsilon, \sum_{i=1}^m z^o_{ij})$, $S^n_j = \frac{1}{\epsilon}\min(\epsilon, \sum_{i=1}^m z^n_{ij})$. Note that $\mathcal{G}$ is a convex function of z^o_{ij}, z^n_{ij} since S is a concave function. Also note that, each potential representative $j \in X_{t+1}$ can contribute a maximum of its own pointwise score L_j, since S_j can take a maximum value of 1. For $\rho = 0$, and under representativeness constraint $\sum_j z_{ij} = 1$; $\sum_j L_j S_j$ is highest when $z_{ij} = z_{i'j'} = 1 \implies j \neq j'$ if $i \neq i'$. Hence, S_j's will also provide a non-trivial solution.

Another drawback of MCOSS (Equation 1) is that compression ratio has no direct relation with the parameter λ. We use a constraint based cardinality criteria in order to have more precise control over the number of representative selected. The user provided parameter *frac* specifies and upper bound over the fraction of incoming frames to be selected as representatives. Overcoming these drawbacks our final convex optimisation based multi-criteria OSS problem formulation is:

$$\min_{z^o_{ij}, z^n_{ij}} \mathcal{G}(z^o_{ij}, z^n_{ij}) \tag{3}$$
$$s.t. \sum_{j=1}^{|R_t|} z^o_{i,j} + \sum_{j=1}^{m} z^n_{i,j} = 1$$
$$z^n_{i,j}, z^o_{i,j} \in [0,1]$$
$$\sum_{j=1}^{m} \|[z^n_{1,j} \dots z^n_{m,j}]\|_p \leq frac * m$$

This can be efficiently solved using any modelling language for solving convex problems, e.g. CVXPY [10]. We call this formulation *thresholded multi-criteria OSS* (TMCOSS). ϵ is a user input which is designed to be the maximum value taken by the variable $\sum_i z_{ij}$, when none of the z_{ij} denote a representative relation to be true. In an ideal situation (when we achieve a $\{0,1\}$ solution to z_{ij}), any positive value for ϵ is sufficient. In practise, we set ϵ to a value less than 1, e.g. $\epsilon = 0.9$. Next, we experimentally demonstrate the utility of our method.

3. Experiments

In this section, we describe experimental results comparing the proposed TMCOSS and SubMCOSS algorithms with MCOSS [9], OSS [13], and only loss (OL) based subset selections. We compare the frame subset selection methods on both driving simulator and real-world driving videos from Cityscapes [8] and Berkeley DeepDrive [24] datasets. Section 3.1 describes the simulator setup, data from which is used for driving model based comparison (Section 3.3) and objective function value based comparison (Section 3.2) of proposed methods. Section 3.4 compares the proposed methods using two real world tasks: (1) drivable area segmentation on the standard BDD dataset [24] and (2) semantic segmentation task on the Cityscapes dataset [8].

3.1. Experimental Setup - Simulator

Dataset: We use the open-source driving simulator CARLA[11] for generating our driving dataset. The collected data comprises of 262 driving episodes and a total of 100,000 video frames, collected using the CAL controller [21] with the ground truth affordances as input. For each video frame, we collect: (1) front center camera image,

and (2) six affordances (Discrete: Red Light, Hazard Stop, Speed Sign ; Continuous: Relative Angle, Centerline Distance, Vehicle Distance). We use approximately 85% of the video frames as the training data, and the remaining as test set. We use Conditional Affordance Learning [21] (CAL) model as the driving model for the experiments involving simulated driving data.

Frame selection methods We compare the following baseline video frame selection methods with the proposed methods **TMCOSS** and **SubMCOSS**:

- **WS**: Entire collected set of video frames.
- **US**: Uniform sampling, frames sampled at regular intervals depending on the compression ratio.
- **OL**: Only loss, subset with the highest total loss.
- **OSS**[13]: OSS based only on pairwise dissimilarity.
- **MCOSS**[9]: Multi-Criteria OSS based on additive pairwise and pointwise dissimilarities.

We use SIFT dissimilarity as the pairwise dissimilarity metric (d_{ij}) for all selection methods and two variants of losses (L_j) as pointwise metric - total loss (**TL**), bucket specific relative angle loss (**BL**). **TL** is defined as the summation of loss (L) over each task/affordance t for a frame j. $TL_j = \sum_t L_{tj}$. **BL** is defined as the weighted summation of relative angle loss and other task losses. $BL_j = (w_{rb} * L_{rb,j}) + \frac{1-w_{rb}}{t}(\sum_t L_{t,j})$ where w_{rb} = weight for the relative angle bucket to which the frame j belongs.

3.2. Comparison of TMCOSS and SubMCOSS

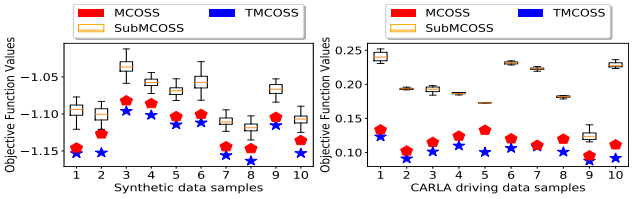

Figure 2: **Objective function values for MCOSS, SubMCOSS and TMCOSS for (left) Synthetic data and (right) CARLA driving data samples.**

In this section, we compare the optimal subsets reported by the baseline method MCOSS [9], SubMCOSS and the proposed convex optimization based method TMCOSS in terms of final objective function values. Figure 2 (*Left*) shows the objective function values for the three approaches, for 10 randomly synthesized problem instances (d matrix of dimension 100×100 and L vector of dimension 100). For the submodular method, we report a box plot of results over 100 runs of the algorithm to capture the randomness. While all three methods find approximate solutions, TMCOSS consistently finds lower values of objective function, followed by MCOSS, and SubMCOSS. We report the same in Figure 2 (*Right*) for 10 episodes in our collected driving data using CARLA. We can clearly observe that function values attained by our proposed convex method, TMCOSS lies below that of the other approximate methods, thus proving it to be an efficient approach.

Next, we discuss the application of the subset selection methods on simulated and real world driving data.

3.3. Driving Simulator based Comparison of OSS

In this section, we will study the performance of different selection techniques on the basis of episode completion and affordance accuracies using simulated driving data. We consider four tasks under episode completion which had been originally defined in [11]: (1) *Straight*: All waypoints lie on a straight road. (2) *One-Turn*: The waypoints pass through 1-turn. (3) *Straight Dynamic* and (4) *One-Turn Dynamic*: similar to *Straight* and *One-Turn* tasks, but in the presence of other vehicles and pedestrians.

We show in Table 1 the performance of CAL model[21] trained on subsets, obtained by various selection techniques, by simulating it with the CARLA [11] simulator. We report number of successfully completed episodes (out of a total of 10 episodes) for each subtask in training and test conditions. We find that TMCOSS performs the best among all frame selection methods by completing all episodes for 100:20 compression ratio and at least 8 episodes out of 10 for compression ratio of 100:7. We observe that the tasks - *Straight* and *Straight Dynamic* are fairly easy to accomplish. The completion of episodes in *One-turn* and *One turn dynamic*, depends on the affordance *Relative Angle*.We observe that Uniform Skip (US) performs poorly in turns since it does not sample adequate number of important frames near turns. While OSS and OL perform better than US, they only complete 7 out of 10 episodes for 100:20 compression. Suprisingly, MCOSS also performs similarly to OSS despite using additional information from the model. This may be attributed to the low importance given to pointwise component of the criteria as explained in Section 2.

Figure 3 analyzes a typical example of episode with turn towards the end. The left plot shows the ground truth and predicted relative angles as a function of distance. The location of the turn is clearly visible. It can be seen that model trained by MCOSS starts to turn early, while the other models shown in the figure start at the appropriate time. The center plot shows the error in prediction, again clearly suggesting that the MCOSS model starts making early errors, and recovers from it very late. Finally, the right plot shows fraction of selected frames for different distance buckets. Note that MCOSS selects a lot of frames much earlier before the turn, whereas TMCOSS selects more frames during

Table 1: **Episode completion for models trained using data from different OSS methods for various tasks.**

Compression Ratio	Methods	Training Conditions				Test Conditions			
		Straight	One-Turn	Straight Dynamic	One-Turn Dynamic	Straight	One-Turn	Straight Dynamic	One-Turn Dynamic
	WS	10	10	10	10	10	10	10	10
100:20	US	9	3	8	3	9	5	9	5
	OL	10	6	10	6	10	7	9	7
	OSS	10	7	10	7	10	7	9	6
	MCOSS	9	8	9	8	7	7	7	7
	SubMCOSS	9	7	9	7	9	7	9	7
	TMCOSS-TL	10	8	10	7	10	9	10	9
	TMCOSS-BL	10	**10**	10	**10**	10	**10**	10	**10**
100:7	MCOSS	9	5	9	5	7	4	7	4
	SubMCOSS	9	3	9	3	9	2	9	2
	TMCOSS-TL	10	7	10	7	10	**9**	10	**9**
	TMCOSS-BL	10	**8**	10	**8**	10	**9**	10	**9**

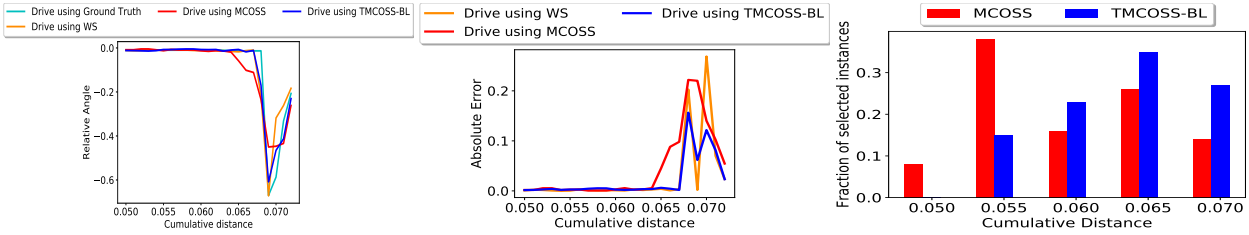

Figure 3: **Analysis of episode failed by MCOSS but completed successfully by TMCOSS. Left: relative angle vs distance travelled, center: error in relative angle, right: fraction of selected instances.**

Table 2: **Prediction accuracies for discrete affordances (macro) and mean absolute errors for continuous affordances for all OSS methods.**

Method	Hazard Stop	Red Light	Vehicle Distance	Centerline Distance
WS	99.72	97.41	0.03	0.05
US	99.47	97.03	0.06	0.05
OL	99.6	91.08	0.09	0.08
OSS	99.57	96.07	0.09	0.08
MCOSS	98.39	89.73	0.05	0.06
SubMCOSS	99.60	93.91	0.05	0.06
TMCOSS-TL	99.61	96.05	0.05	0.05
TMCOSS-BL	99.71 (0.01%)	92.83 (4.58%)	0.04 (33.33%)	0.05 (0%)

the turn. This conclusively demonstrates that TMCOSS selects more informative frames compared to MCOSS.

Table 2 compares the prediction performance of CAL driving model trained using subsets obtained by different OSS techniques for 100:20 compression ratio. We report the prediction accuracies of two discrete affordances and error for two continuous affordances. We neglect speed sign, since its prediction performance is not crucial to episode completion. We also report the % difference in the performance metric for TMCOSS-BL w.r.t. that of WS. We notice that TMCOSS predicts the crucial Hazard stop affordance satisfactorily with very little difference from WS. Performance in prediction of vehicle distance and centerline distance are also close to WS. Curiously, TMCOSS-TL performs better than TMCOSS-BL on the Red light affordance predictiondue to its absence in center camera during turns.

We observe that among all 6 affordances, relative angle, which provides the steering angle of the car, is the most essential affordance for episode completion. Hence, we study it in greater detail in Figure 4. The entire range of Relative Angle affordance can be divided into 20 buckets (ranging from -1.0 to +1.0. in steps of 0.1). Buckets corresponding to (-1.0 to -0.1), (-0.1 to 0.1) and (0.1 to 1.0) indicate the **left turn**, **straight road** and **right turn** respectively. We observe that the MAE for all OSS methods in straight road buckets lie in a narrow range. This is due to the skewness in data distribution for Relative Angle affordance (4 % *Left*, 92% *Straight*, 4% *Right*). We note that TMCOSS-TL and TMCOSS-BL outperform all other methods in turn buckets, which have lesser number of datapoints. We also observe that TMCOSS-BL selects comparatively higher fraction of instances for both Left and Right turns. We find that the difference in MAE becomes more evident with increase in compression ratio (see supplementary material). Next, we study the effectiveness of the proposed method on real driving benchmark datasets for the task of segmentation which

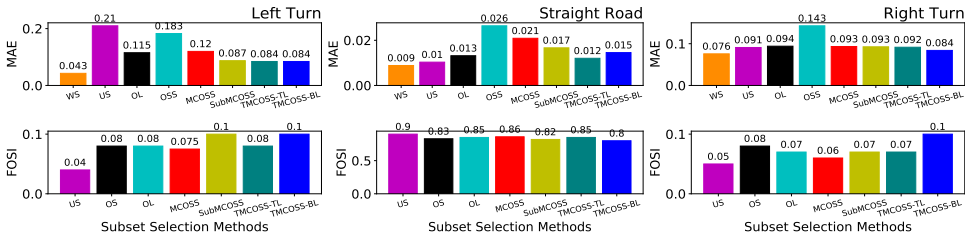

Figure 4: **Fraction of selected instances (FOSI) and Mean Absolute Error (MAE) for 3 relative angle buckets.**

is an important sub-task in autonomous driving.

3.4. Real-world Driving Tasks based Comparison of OSS

In this experiment, we show the usefulness of the proposed method in two other tasks which are equally significant for autonomous driving - drivable area segmentation and semantic segmentation. We use the two benchmark datasets - Berkeley DeepDrive dataset (BDD)[24] for the first task and Cityscapes[8] for the second. We use DeepLabV3+[5] for both the tasks and report the standard metric (IoU) for the important classes.

Table 3 compares performances of various OSS techniques for compression ratio of 100:20. TMCOSS-TL performs better than all baselines and achieves performance close to WS. From Table 3-top, we observe that alternate drive area is a harder task for OSS compared to drivable area. We also note from Table 3 (*bottom*) that TMCOSS-TL is able to segment the important classes better than the baselines. For e.g., in the task of pedestrian detection (*Person* class), TMCOSS-TL performs better than MCOSS.

Table 3: **Prediction performances for Drivable Area segmentation using BDD (*top*) and semantic segmentation using Cityscapes (*bottom*) for various OSS techniques.**

Method	Drivable Area IoU (%)	Alternate Drive Area IoU (%)	MIoU (%)
WS	81.0	69.0	75.0
OL	77.0	62.0	69.5
OSS	75.0	59.0	67.0
MCOSS	76.0	59.0	67.5
TMCOSS	**80.0**	**65.0**	**72.5**

Method	Road IoU (%)	Wall IoU (%)	Side walk IoU(%)	Person IoU(%)	Car IoU (%)	Bicycle IoU (%)	Mean IoU (%)
WS	98.0	50.0	83.0	81.0	94.0	76.0	80.33
OL	96.0	35.0	75.0	73.0	90.0	68.0	72.83
OSS	96.0	31.0	75.0	74.0	90.0	71.0	72.83
MCOSS	96.0	29.0	75.0	74.0	90.0	70.0	72.33
TMCOSS	**98.0**	**50.0**	**82.0**	**79.0**	**93.0**	**74.0**	**79.33**

Figure 5 analyses the selected frames by TMCOSS and MCOSS, by reporting the fraction of selected instances (Figure 5-left), as well as fraction of selected pixels (Figure 5-right). The fraction of pixels are important because selecting a frame with a person in prominent visibility is more useful than one with a person in a far corner. While both the images will be marked as instances containing the class *Person*, the former image will have more pixels, and hence will be more useful for the person segmentation task. We can see that the proposed method selects higher fraction of instances as well as pixels for the difficult classes, thus justifying its better performance in those classes in Table 3. Hence, we show that the proposed method performs better than the baselines not only in simulated scenario (using CARLA) but also in tasks involving real driving data (using BDD and Cityscapes).

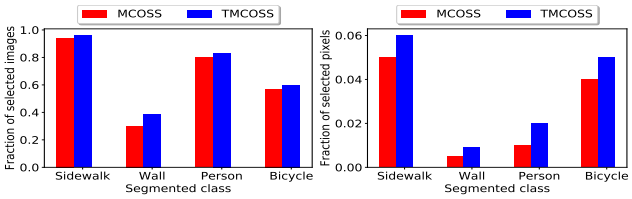

Figure 5: **Fraction of selected instances (*left*) and pixels (*right*) for semantic segmentation on Cityscapes.**

4. Conclusion

In this paper, we propose TMCOSS, novel thresholded convex optimization based online video frame subset selection technique incorporating pairwise dissimilarities between video frames and pointwise loss of video frames on current models for a task. We study the effectiveness of TMCOSS on tasks of driving model training measured by episode completion on CARLA simulator, and semantic segmentation in real world driving datasets of BDD and Cityscape. We find that TMCOSS is effective for selection of relevant video frames, where even after dropping 80% of frames, we succeed in maintaining a performance close to that of the whole set. We also compare TMCOSS to a submodular set-function formulation proposed here called SubMCOSS, concluding that TMCOSS outperforms SubMCOSS on episode completion in CARLA.

References

[1] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1433–1452. SIAM, 2014.

[2] Niv Buchbinder, Moran Feldman, Joseph Seffi, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing*, 44(5):1384–1402, 2015.

[3] Chongyu Chen, Jianfei Cai, Weisi Lin, and Guangming Shi. Surveillance video coding via low-rank and sparse decomposition. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 713–716, 2012.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[6] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[7] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9329–9338, 2019.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[9] Soumi Das, Sayan Mandal, Ashwin Bhoyar, Madhumita Bharde, Niloy Ganguly, Suparna Bhattacharya, and Sourangshu Bhattacharya. Multi-criteria online frame-subset selection for autonomous vehicle videos. *Pattern Recognition Letters*, 2020.

[10] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

[12] Ehsan Elhamifar. Sequential facility location: Approximate submodularity and greedy algorithm. In *International Conference on Machine Learning*, pages 1784–1793. PMLR, 2019.

[13] Ehsan Elhamifar and M Clara De Paolis Kaluza. Online summarization via submodular and convex optimization. In *CVPR*, pages 1818–1826, 2017.

[14] Sheng-Jun Huang, Jia-Wei Zhao, and Zhao-Yang Liu. Cost-effective training of deep cnns with active model adaptation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1580–1588. ACM, 2018.

[15] Mohsen Joneidi, Saeed Vahidian, Ashkan Esmaeili, Weijia Wang, Nazanin Rahnavard, Bill Lin, and Mubarak Shah. Select to better learn: Fast and accurate deep learning using data selection from nonlinear manifolds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7819–7829, 2020.

[16] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[17] Shuyue Lan, Rameswar Panda, Qi Zhu, and Amit K Roy-Chowdhury. Ffnet: Video fast-forwarding via reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6771–6780, 2018.

[18] Ziyi Liu, Siyu Yu, and Nanning Zheng. A co-point mapping-based approach to drivable area detection for self-driving cars. *Engineering*, 4(4):479–490, 2018.

[19] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 202–211, 2017.

[20] Aditya Prakash, Aseem Behl, Eshed Ohn-Bar, Kashyap Chitta, and Andreas Geiger. Exploring data aggregation in policy learning for vision-based urban autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11763–11773, 2020.

[21] Axel Sauer, Nikolay Savinov, and Andreas Geiger. Conditional affordance learning for driving in urban environments. *arXiv preprint arXiv:1806.06498*, 2018.

[22] Wenhao Wu, Dongliang He, Xiao Tan, Shifeng Chen, and Shilei Wen. Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[23] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S. Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[24] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.