Divide-and-Conquer for Lane-Aware Diverse Trajectory Prediction

Sriram Narayanan\textsuperscript{1} Ramin Moslemi\textsuperscript{1} Francesco Pittaluga\textsuperscript{1} Buyu Liu\textsuperscript{1} Mannohan Chandraker\textsuperscript{1,2}

\textsuperscript{1}NEC Labs America, \textsuperscript{2}UC San Diego

Abstract

Trajectory prediction is a safety-critical tool for autonomous vehicles to plan and execute actions. Our work addresses two key challenges in trajectory prediction, learning multimodal outputs, and better predictions by imposing constraints using driving knowledge. Recent methods have achieved strong performances using Multi-Choice Learning objectives like winner-takes-all (WTA) or best-of-many. But the impact of those methods in learning diverse hypotheses is under-studied as such objectives highly depend on their initialization for diversity. As our first contribution, we propose a novel Divide-And-Conquer (DAC) approach that acts as a better initialization technique to WTA objective, resulting in diverse outputs without any spurious modes. Our second contribution is a novel trajectory prediction framework called ALAN that uses existing lane centerlines as anchors to provide trajectories constrained to the input lanes. Our framework provides multi-agent trajectory outputs in a forward pass by capturing interactions through hypercolumn descriptors and incorporating scene information in the form of rasterized images and per-agent lane anchors. Experiments on synthetic and real data show that the proposed DAC captures the data distribution better compared to other WTA family of objectives. Further, we show that our ALAN approach provides on par or better performance with SOTA methods evaluated on Nuscenes urban driving benchmark.

1. Introduction

Prediction of diverse multimodal behaviors is a critical need to proactively make safe decisions for autonomous vehicles. A major challenge lies in predicting not only the most dominant modes but also accounting for the less dominant ones that might arise sporadically. Hence, there is need for models that can disentangle the plausible output space and provide diverse futures for any given number of samples. Further, a vast majority of actors execute socially acceptable maneuvers that adhere with the underlying scene structure. Predicting socially non-viable outputs can lead to unsafe planning decisions with some more dangerous than the others \[7\]. For example, a method that provides close enough predictions that does not follow road semantics is more dangerous compared to similar performing method that adheres to the scene structure.

Traditionally, generative models have been widely adapted to capture the uncertainties related to trajectory prediction problems \[25, 22, 37, 21, 39\]. However, generative methods may suffer from mode collapse issues, which reduces their applicability for safety critical applications such as self-driving cars. Recent methods \[32, 28\] use Multi-Choice Learning objectives \[26\] like winner-takes-all (WTA) but suffer from instability associated with network initialization \[30, 34\]. As a first contribution, we propose a Divide and Conquer (DAC) approach that provides a better initialization to the WTA objective. Our method solves issues related to spurious modes where some hypotheses are either untrained in the training process, or reach equilibrium positions that do not represent any part of the data. We show that the proposed DAC captures the data distribution better on both real and synthetic scenes with multi-modal ground truth, compared to baseline WTA objectives \[30, 34\].

Figure 1: Depicts trajectory prediction problem in an intersection scenario with possible lane anchors for agents shown as coloured dashed lines.
Further, trajectory prediction methods incorporate driving knowledge using scene context either in the form of rasterized images [25, 37, 39, 32, 33, 8] or by exploiting HD map data structure [28, 15] as inputs. Usually, this information is represented as a feature given as input to the network and does not guarantee strong semantic coupling. Our second contribution addresses this by proposing ALAN, a novel trajectory prediction framework that uses lane centerlines as anchors to predict trajectories (Figure 1). Our outputs provide accurate predictions with strong semantic alignment demonstrated by FDE and OffRoadRate values and validated using our qualitative visualizations.

Specifically, we use a single representational model [39] for multi-agent inputs and encode interactions through novel use of hypercolumn descriptors [2] that extracts information from features at multiple scales. Moreover, we transform the prediction problem to normal-tangential (nt) coordinates with respect to input lanes. This is critical in order to use lane centerlines as anchors. Further, we regularize anchor outputs through auxiliary xy predictions to make them less susceptible to bad anchors and rely on agent dynamics. Finally, we rank our predictions through an Inverse Optimal Control based ranking module [25].

In summary, our contributions are the following:

- A novel Divide and Conquer approach as a better initialization to WTA objective that captures data distribution without any spurious modes.
- A new anchor based trajectory prediction framework called ALAN that uses existing centerlines as anchors to provide context-aware outputs with strong semantic coupling.
- Strong empirical performance on the Nusscenes urban driving benchmark.

2. Related Work

Multi-Choice Learning: Multi-modal predictions have been realized in different domains through Multi-choice learning (MCL) [17, 12, 26] objectives in the past. Several works have shown use cases of MCL to provide diverse hypotheses in classification [26, 34], segmentation [26, 34], captioning [26], pose estimation [34], image synthesis [10] and trajectory proposals [40]. Convergence issue related to WTA objectives have been shown in [34, 30]. Following this work, [34] proposed a relaxed winner-takes-all objective (RWTA) to solve the convergence problem but this method itself suffers from the problem of hypotheses incorrectly capturing the data distribution. [30] proposed an evolving winner-takes-all (EWTA) loss that captures the distribution better compared to [34]. Despite the aforementioned improvements, these methods still can’t capture the data distribution accurately due to spurious modes at equilibrium or hypotheses untrained during the training process. Alternatively, we propose a Divide and Conquer approach where we exponentially increase the effective number of outputs during training with set of hypotheses capturing some part of the data at every stage.

Forecasting Methods: The future trajectory prediction has been investigated broadly in the literature using both classical [42, 27, 23] and deep learning based methods [16, 1, 39]. Deterministic models [1, 29, 36] predict most likely trajectory for each agent in the scene while neglecting the uncertainties inherited in the trajectory prediction problem. To capture the uncertainties and create diverse trajectory predictions, stochastic methods have been proposed which encode possible modes of future trajectories through sampling random variables. Non-parametric deep generative models such as Conditional Variational Autoencoder (CVAE) [25, 3, 22, 20, 39] and Generative Adversarial Networks (GANs) [24, 16, 35] have been widely used in this domain. However, these methods fail to capture all underlying modes due to imbalance in the latent distribution [43]. Recent methods predict a fixed set of diverse trajectories [32, 28] for the same input context. Our method uses a similar approach to predict a set of M hypothesis.

Representation: HD map rasterization have been widely used in the literature to encode and process map information by neural networks [3, 46, 13, 6, 39]. Some methods [38, 31] construct top view map using semantics and depth information from perspective images. Some [44, 6] use a combination rasterized HD maps and sensor information. Several recent works [28, 15] utilize map information directly by representing the vectorized map data as a graph data structure. Our work uses a hybrid map input combining both rasterized map and vectorized lane data provided as input for every agent at its location on the spatial grid [39].

Trajectory Prediction: Traditionally, several works [39, 28, 15, 32] formulate trajectory prediction problem as a regression over cartesian coordinates. [38] poses it as a classification of future locations over a spatial grid. Chang et. al [9] use a normal-tangential coordinate similar to ours but is only limited to classical nearest neighbor and vanilla LSTM [19] approaches. Related to our work, some methods tackle the multi-modality problem by quantizing the output space into several predefined diverse anchors and then reformulating the original trajectory problem into sequential anchor classification (selection) and offset regression sub-problems [45, 33, 8, 44]. However, Anchors usually are pre-clustered into a fixed set as a priori or are calculated in real-time based on kinematic heuristics [45]. Hence, the process of creating anchors may add computational complexity in the inference time, also it could be highly scenario dependent and hard to generalize. In contrast, our method uses HD map centerline information as anchors which is consistent for diverse scenarios and also readily available at inference.
3. Divide and Conquer

In this section, we provide detailed description of our method in training Multi-Hypothesis prediction networks where our approach acts as an initialization technique for winner-takes-all [26] objective. Let $\mathcal{X}$ denote the vector space of inputs and $\mathcal{Y}$ denotes the vector space of output variables. Let $\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}$ be a set of $N$ training tuples and $p(x, y) = p(y|x)p(x)$ be the joint probability density. Our goal is to learn a function $f_\theta: \mathcal{X} \rightarrow \mathcal{Y}^M$ that maps every input in $\mathcal{X}$ to a set of $M$ hypotheses. Mathematically, we define:

$$f_\theta(x) = (f_1^\theta(x), ..., f_M^\theta(x)).$$

(1)

As shown by Rupprecht et al. [34], winner-takes-all objective minimizes the loss with the closest of $M$ hypotheses:

$$\int_\mathcal{X} \sum_{j=1}^M \int_{\mathcal{Y}_j(x)} \mathcal{L}(f_j^\theta(x), y)p(x, y)dydx,$$

(2)

where $\mathcal{Y}_j$ is the Voronoi tessellation of label space with $\mathcal{Y} = \cup_{j=1}^M \mathcal{Y}_j$. This objective leads to Centroidal Voronoi tessellation [14] of outputs where each hypothesis minimizes to the probabilistic mass centroid of the Voronoi label space $\mathcal{Y}_j$ enclosed by it. In practice, to obtain diverse hypotheses WTA objective can be written as a meta loss [30, 34, 26, 17],

$$\mathcal{L}_{WTA} = \sum_{k=1}^K \delta_k(k = \arg \min_i \mathcal{L}(f_i^\theta(x))) \mathcal{L}(f_k^\theta(x), y),$$

(3)

where $\delta(\cdot)$ is the Kronecker delta function with value 1 when condition is True and 0 otherwise.

Initialization difficulties for WTA As mentioned by Makansi et al. [30] Equation 3 can be compared to EM algorithm and K-means clustering where they depend mainly on initialization for optimal convergence. As shown in 2b this makes training process very brittle as the Voronoi region of only few hypotheses encloses the data distribution, leaving most of the hypotheses untrained due to winner-takes-all objective. The alternative solution proposed by Ruppercht et al. [34] to solve the convergence problem by assigning $\epsilon$ weight to the non-winners does not work as every ground truth associates with atmost one hypothesis making other non-winners to reach the equilibrium as shown in 2c. Makansi et al. [30] then proposed evolving winner-takes-all (EWTA) objective where they update top $k$ winners. The $k$ varies starting from $k = M$ to $k = 1$ leading to winner takes all objective in training process. This method captures the data distribution better compared to RWTA and WTA but still produces hypothesis with incorrect modes as shown in the Figure 2d.
We now describe our method in detail. We propose an novel initialization technique called Divide and Conquer that alleviates the problem of spurious modes, leaving the Voronoi region of every output hypothesis to capture some part of the data, as shown in Figure 2h. We divide M hypotheses into k sets and update the set with argmin outputs to match the ground truth. The value of k starts with 1 and increases exponentially as every set is broken down into two halves as we progress through the training. This creates a binary tree with the depth of the tree dependent on the number of output hypotheses M. Algorithm 1 shows pseudo-code of the proposed Divide and Conquer technique. Here depth specifies the maximum depth that can be reached in the current training stage and we define list as variable containing set of hypotheses at any stage in the training. Further, we define newly formed sets from $k^{th}$ set as set$k_1$ and set$k_2$. Set from the list that produces argmin output is denoted as mSet. Finally we take mean loss of all hypotheses in mSet to get $L_{DAC}$.

From Figure 2e, with $k = 1$ and list containing a single set, all M hypotheses reach towards the equilibrium. As the number of sets in the list increases from 2e to 2f the hypotheses divide the distribution space based on the Voronoi region to capture different parts of the data. The effective number of outputs grows at every stage, with the data captured by the $k^{th}$ set in the previous stage split across two newly formed sets in the next stage. Finally, as we reach the leaf nodes, every set contains one hypothesis leading to a winner-takes-all objective similar to Equation 3.

DAC starts with all hypotheses fitting the whole data and at every stage DAC ensures some data to be enclosed in the Voronoi space. During split, hypotheses divide the data enclosed within their Voronoi space to reach new equilibrium. Although, DAC does not guarantee equal number of hypotheses capturing different modes of the data it ensures convergence. Further we would like to note that DAC does not have any significant computational complexity as only dividing into sets and min calculations are involved. In Section 5, we show benefits of DAC in capturing multimodal distributions better, producing diverse set of hypotheses compared to other WTA objectives.

4. Trajectory Prediction with Lane Anchors

In this section, we introduce a single representation model called ALAN that produces lane aware trajectories for multiple agents in a forward pass. We formulate the problem as one shot regression of diverse hypotheses across time steps. We now describe our method in detail.

4.1. Problem Statement

Our method takes scene context input in two forms: a) rasterized birds-eye-view (BEV) representation of the scene denoted as $I$ of size $H \times W \times 3$ and b) per-agent lane centerline information as anchors. We define lane anchors $L = \{L_1,...,L_p\}$ as a sequence of p points with coordinates $L_p = (x, y)$ in the BEV frame of reference. We denote $X_i = \{X^1_i,...,X^T_i\}$ as trajectory coordinates containing past and future observations of the agent $i$ in Cartesian form, where $X^t_i = (x^t_i, y^t_i)$. For every agent $i$, we identify a set of candidate lanes that the vehicle may take based on trajectory information like closest distance, yaw alignment and other parameters (see supplementary). We denote this as a set of plausible lane centerlines $A = \{L_1,...,L_k\}$, where $k$ represents total number of lane centerlines along which the vehicle may possibly travel. We then define vehicle trajectories $X_i$ along these centerlines in a 2d curvilinear normal-tangential (nt) coordinate frame. We denote $N_{i,k} = \{N^1_{i,k},...,N^T_{i,k}\}$ as the nt coordinates for the agent $i$ along the centerline $L_k$, where $N^t_{i,k} = (n^t_{i,k}, l^t_{i,k})$ denotes normal and longitudinal distance to the closest point along the lane. Use of nt coordinates is crucial to capture complex road topologies and associated dynamics to provide predictions that are semantically aligned and has been studied in our experiments (Section 5).

We then define trajectory prediction problem as the task of predicting $nt_{Y_{i,k}} = \{N^1_{i,k},...,N^T_{i,k}\}$ for the given lane anchor $L_k$ provided as input to the network. We follow an input representation similar to [39], where we encode agent specific information at their respective $X^t_{i,obs}$ locations on the spatial grid. Finally, to get trajectories in BEV frame of reference we convert our output predictions to cartesian form along these centerlines in a 2d curvilinear normal-tangential (nt) coordinate frame. We denote $A$ as

\begin{algorithm}[h]
\begin{algorithmic}[1]
\Procedure{DAC}{\textbf{loss}, \textbf{depth}}
\State $set_1 = \{\textbf{loss}\}$ \Comment{All M hypotheses}
\State \textbf{list} = $[set_1]$ \\
\For{$i \leftarrow 2$ \textbf{to} \textbf{depth}}
\For{$set_k$ \textbf{in} list}
\State // Divide set$k$ into halves
\State \textbf{list} += $\{[set_{k1}1],[set_{k2}]\}$
\EndFor
\State $mSet = \{set_k : \min(set_k) < \min(set_j) ; \forall j \in \{1,...,\textbf{len}([\textbf{list}])\}, j \neq k\}$
\State $L_{DAC} = \text{mean}(mSet)$
\EndFor
\Return $L_{DAC}$
\EndProcedure
\end{algorithmic}
\caption{Divide and Conquer technique}
\end{algorithm}

An overview of our framework is shown in Figure 3. Our method consists of five major components: a) a centerline encoder b) a past trajectory encoder c) a multi-agent convolutional interaction encoder d) hypercolumn [2] trajectory decoder and e) an Inverse Optimal Control (IOC) based ranking module [25].
Centerline Encoder: We encode our input lane information $L_{i,k}$ for every agent through a series of 1D convolutions to produce an embedded vector $C_{i,k} = c_{enc}(L_{i,k})$ for every agent in the scene.

Past Trajectory Encoder: Apart from $nt$ coordinates $N_{i,k}$ for the lane anchor, we provide additional $X_{i}$ as input to the past encoder. We first embed the temporal inputs through a $MLP$ and then pass it through a LSTM[19] network to provide a past state vector $h_{i,obs}^{s}$. Formally,  

$$s_{i}^{t} = MLP(X_{i}^{t}, N_{i,k}^{t}) \quad (4)$$  

$$h_{i,obs}^{t} = LSTM(s_{i}^{1:t-obs}) \quad (5)$$

Multi-Agent Convolutional Encoder: We realize multi-agent prediction of trajectories in a forward pass through a convolutional encoder module [39]. First, we encode agent specific information $C_{i,k}, h_{i,obs}^{s}$ at their respective locations $X_{i}^{t-obs}$ in the BEV spatial grid. This produces a scene state map $S$ of size $H \times W \times 128$ containing information of every agent in the scene. We then pass this through a convolutional encoder along with the rasterized BEV map $I$ to produce activations at various feature scales. In order to calculate feature vectors of each individual agent, we adapt a technique from Bansal et al. [2] to extract hypercolumn descriptors $D_{i}$ from their locations. The hypercolumn descriptor contains features extracted at various scales by bi-linearly interpolating $X_{i}^{t-obs}$ for different feature dimensions. Thus,  

$$D_{i} = [c_{1}(X_{i}^{t}), ..., c_{k}(X_{i}^{t})] \quad (6)$$

where $c_{k}$ is the feature extracted at $k^{th}$ layer by bilinearly interpolating the input location to the given dimension. The intuition is to capture interactions at different scales when higher convolutional layers capturing the global context and low-level features retaining the nearby interactions. In Section 5, we show using hypercolumn descriptors in trajectory prediction task can be beneficial compared to just using global context vectors.

Hypercolumn Trajectory Decoder: The hypercolumn descriptor $D_{i}$ of every agent is then fed through a decoder containing a series of 1x1 convolutions to output $M$ hypotheses at once. Here we investigate two variants of ALAN prediction. ALAN-$nt$ where we predict $nt$ trajectories $\hat{Y}_{i}$ in the direction of the lane and ALAN-$nt,xy$ which also provides an auxiliary $xy$ predictions $\hat{xy}_{i}$. Linear values in $nt$ can correspond to trajectories of higher degrees based on the input anchor. Moreover, two trajectories having same $nt$ values can have completely different dynamics. Thus we make use of the auxiliary predictions to regularize anchor based outputs to make the network aware of agent dynamics and less susceptible to bad anchors. The $M$ hypotheses predicted from our network is given as:  

$$\hat{nt}_{i}^{xy}, \hat{xy}_{i} = C N N_{1 \times 1}(D_{i}) \quad (7)$$

$$\hat{nt}_{i}^{xy} = \{\hat{nt}_{i,1}, \hat{nt}_{i,2}, ..., \hat{nt}_{i,M}\} \quad (8)$$

$$\hat{xy}_{i} = \{\hat{xy}_{i,1}, \hat{xy}_{i,2}, ..., \hat{xy}_{i,M}\} \quad (9)$$

Ranking Module: We use the technique from Lee et al. [25] to generate scores $\hat{Y}_{i} = \{\hat{Y}_{i,1}, \hat{Y}_{i,2}, ..., \hat{Y}_{i,M}\}$ for the $M$ output hypotheses. It measures the goodness $\hat{Y}_{i,k}$ of predicted hypotheses by assigning rewards that maximizes towards their goal[41]. The module uses predictions $\hat{nt}_{i}$ to obtain the target distribution $q_{t}$, where
where network trained with DAC objective can be inferred that the proposed DAC captures the data distribution better with hypothesis spread out across the crosswalk resembling the ground truth distribution of points.

\[ q = \text{softmax}(-d(nt \hat{Y}_i, nt \hat{Y}_i)) \] and \( d \) being the L2 distance between the ground truth and predicted outputs. Thus, the score loss is given as \( L_{\text{score}} = \text{Cross-Entropy}(\ast Y_i, q) \).

4.3. Learning

We supervise the network outputs \( \{nt \hat{Y}_i, xy \hat{Y}_i\} \) as the L2 distance with their respective ground truth labels \( nt Y \) for the input lane anchor \( L_k \) and \( xy Y \). We use the proposed Divide and Conquer technique to train our Multi-Hypothesis prediction network. Hence, the reconstruction loss for both primary and auxiliary predictions is given by:

\[ nt L_{DAC} = DAC(nt \hat{Y}_i), \quad (10) \]
\[ xy L_{DAC} = DAC(xy \hat{Y}_i). \quad (11) \]

Additionally, we penalize our anchor based predictions based on \( xy \hat{Y}_i \) by transforming the predictions to \( nt \) coordinates \( xy \hat{Y}_i \) along the input lane. We also add the regularization other way to penalize \( xy \hat{Y}_i \) predictions based on the anchor outputs \( nt \hat{Y}_i \) by converting them to \( xy \) coordinates \( nt \hat{Y}_i \). We add the regularization as L2 distance between the converted primary and auxiliary predictions for all hypotheses:

\[ nt L_{xy} = L2(nt \hat{Y}_i, xy \hat{Y}_i), \quad (12) \]
\[ xy L_{nt} = L2(xy \hat{Y}_i, nt \hat{Y}_i). \quad (13) \]

The total learning objective for the network to minimize can be given by,

\[ L \leftarrow nt L_{DAC} + xy L_{DAC} + \lambda_1 nt L_{xy} + \lambda_2 xy L_{nt} + L_{\text{score}}. \quad (14) \]

5. Experiments

We first evaluate our proposed Divide and Conquer technique on the synthetic Car Pedestrian dataset[30]. Further, we show evaluations of DAC and the proposed anchor based prediction technique on Nuscenes[5] prediction dataset.

Table 1: Comparison of Methods on CPI dataset based on FDE and EMD metrics, where p - pedestrian and c - car

<table>
<thead>
<tr>
<th>Method</th>
<th>pFDE</th>
<th>cFDE</th>
<th>Avg FDE</th>
<th>pEMD</th>
<th>cEMD</th>
<th>Avg EMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAC</td>
<td>5.56</td>
<td>5.61</td>
<td>5.58</td>
<td>1.14</td>
<td>1.48</td>
<td>1.31</td>
</tr>
<tr>
<td>EWTA[30]</td>
<td>5.8</td>
<td>5.63</td>
<td>5.76</td>
<td>1.09</td>
<td>1.59</td>
<td>1.34</td>
</tr>
<tr>
<td>RWT A[34]</td>
<td>4.90</td>
<td>9.56</td>
<td>7.23</td>
<td>1.02</td>
<td>1.64</td>
<td>1.33</td>
</tr>
<tr>
<td>WT A[26]</td>
<td>5.32</td>
<td>6.32</td>
<td>5.82</td>
<td>1.17</td>
<td>2.41</td>
<td>1.79</td>
</tr>
<tr>
<td>CVAE</td>
<td>15.9</td>
<td>19.2</td>
<td>17.6</td>
<td>1.72</td>
<td>2.74</td>
<td>2.23</td>
</tr>
</tbody>
</table>

5.1. Car Pedestrian Dataset

Unlike real world settings where only a single outcome is observed, CPI dataset consists of interacting agents with multi-modal ground truths. We aim to evaluate how well our multi-hypothesis predictions capture the true distribution of samples in the test set. We use a similar training strategy from [30] using a ResNet-18 [18] encoder backbone where we train a two-stage mixture density network [4]. The first stage takes past observations of the car and pedestrian as the inputs and predicts \( k \) output hypotheses containing future goals of both actors after \( \Delta t \) timestep. We train the first stage using different variants of the winner-takes-all loss function. The second stage then fits a mixture distribution with \( M \) modes over the hypothesis by predicting soft-assignments for the outputs. We refer readers to Equations 7, 8 and 9 from [30] for more details about calculating the parameters for the mixture distribution. We use evaluation metrics such oracle error (FDE) and Earth Mover’s Distance (EMD) used in [30].

Oracle error (FDE) measures the diversity of our outputs predictions by choosing the closest hypothesis with the ground truth.

EMD distance quantifies the amount of probability mass that has to be moved from the predicted distribution to match the true distribution.

From Table 1 it can be inferred that the proposed DAC method outperforms the other variants of WTA objective showing that DAC captures the data distribution better compared to EWTA, RWT A and WTA. This can also been seen in Figure 4 where network trained with DAC objective captures the ground truth distribution of actors better compared to other variants. The average EMD of the proposed DAC is significantly better than WTA and comparable to EWTA and RWTA objective. DAC better captures goals for the cars that spread across compared to pedestrian goals. Moreover, as shown by Table 1, the average oracle error (FDE) for the DAC method is significantly lower compared to other variants confirming that DAC WTA produces diverse hypotheses.

5.2. Nuscenes Dataset

Nuscenes[5] contains a large collection of complex road scenarios from cities of Boston and Singapore. Approx-
Figure 5: Shows example predictions from ALAN. The past trajectory is shown in brown and the GT is shown in black. The endpoint of GT is shown as a green dot. The input lane anchor is shown in cyan, with predicted trajectories in green and their endpoints as triangles. (a) and (b) shows predictions that follow a complex lane structure. Anchor based predictions can be beneficial especially for a longer prediction horizon as the complexity of the trajectory increases anchors can be helpful in following the semantics. (c) predicts a U-turn with appropriate dynamics when the lane of interest is in opposite direction and (d) shows a multi-agent prediction scenario.

Table 2: Nuscenes Trajectory Prediction Benchmark

<table>
<thead>
<tr>
<th>Model</th>
<th>mADE_1</th>
<th>mADE_5</th>
<th>mADE_10</th>
<th>Miss_2.5</th>
<th>Miss_2.10</th>
<th>mFDE_1</th>
<th>mFDE_5</th>
<th>mFDE_10</th>
<th>OffRoadRate</th>
</tr>
</thead>
<tbody>
<tr>
<td>cxx</td>
<td>-</td>
<td>1.63</td>
<td>1.29</td>
<td>69</td>
<td>60</td>
<td>8.86</td>
<td>-</td>
<td>-</td>
<td>0.08</td>
</tr>
<tr>
<td>pq</td>
<td>-</td>
<td>2.23</td>
<td>1.68</td>
<td>69</td>
<td>56</td>
<td>9.56</td>
<td>-</td>
<td>-</td>
<td>0.12</td>
</tr>
<tr>
<td>CoverNet[33]</td>
<td>-</td>
<td>2.62</td>
<td>1.92</td>
<td>76</td>
<td>64</td>
<td>11.36</td>
<td>-</td>
<td>-</td>
<td>0.13</td>
</tr>
<tr>
<td>MTP [11]</td>
<td>4.42</td>
<td>2.22</td>
<td>1.74</td>
<td>74</td>
<td>67</td>
<td>10.36</td>
<td>4.83</td>
<td>3.54</td>
<td>0.25</td>
</tr>
<tr>
<td>MultiPath [8]</td>
<td>4.43</td>
<td>1.78</td>
<td>1.55</td>
<td>78</td>
<td>76</td>
<td>10.16</td>
<td>3.62</td>
<td>2.93</td>
<td>0.36</td>
</tr>
<tr>
<td>Trajectron++[37]</td>
<td>-</td>
<td>1.88</td>
<td>1.51</td>
<td>70</td>
<td>57</td>
<td>9.52</td>
<td>-</td>
<td>-</td>
<td>0.25</td>
</tr>
<tr>
<td>MHA_JAM [32]</td>
<td>3.69</td>
<td>1.81</td>
<td>1.24</td>
<td>59</td>
<td>45</td>
<td>8.57</td>
<td>3.72</td>
<td>2.21</td>
<td>0.07</td>
</tr>
<tr>
<td>ALAN (top-M)</td>
<td>4.62</td>
<td>1.87</td>
<td>1.22</td>
<td>60</td>
<td>49</td>
<td>9.98</td>
<td>3.54</td>
<td>1.87</td>
<td>0.01</td>
</tr>
<tr>
<td>ALAN (Oracle)</td>
<td>4.61</td>
<td>1.78</td>
<td>1.16</td>
<td>59</td>
<td>48</td>
<td>9.95</td>
<td>3.29</td>
<td>1.70</td>
<td>0.01</td>
</tr>
<tr>
<td>ALAN (BofA)</td>
<td>4.67</td>
<td>1.77</td>
<td>1.10</td>
<td>57</td>
<td>45</td>
<td>10.00</td>
<td>3.32</td>
<td>1.66</td>
<td>0.01</td>
</tr>
</tbody>
</table>

5.2.1 Baselines

We show comparisons of our ALAN predictions with several baseline methods evaluated on Nuscenes benchmark. MTP [11] uses rasterized image as input to predict trajectories. CoverNet [33] uses fixed set of trajectories to solve the prediction as a classification over the trajectory set. Multipath [8] is the closest baseline that uses time parameterized anchor trajectories obtained from the train set and formulates the problem as regression of offset values with respect to their anchor heads. MHA_JAM [32] is recent method that uses joint agent-map representation to produce outputs with multi-head attentions. Trajectron++ [37] is graph recurrent model that predicts trajectories incorporating agent dynamics and semantics. We utilize the numbers for [11] and [8] from [32].

5.2.2 Metrics

We use standard evaluation metrics such as Average Displacement Error (mADE_M) and Final Displacement Error (mFDE_M). Further, we compute miss rate (Miss_{d,M}) of top M likely trajectories with the GT. A set of predictions is considered to be a miss if there’s no hypothesis across predictions having maximum displacement point less than the threshold d. OffRoadRate computes percentage of output trajectories that fall outside the drivable region. We use the example API provided by Nuscenes to compute our metrics.

5.2.3 Quantitative Results

We first show that ALAN can achieve on par or better performance compared to our baseline approaches. Here we evaluate ALAN with different anchor sampling strategies, top-M, oracle and best-of-all (BofA). In ALAN (top-M) we pick top M trajectory outputs from different anchors based on predicted IOC scores for each trajectory. ALAN (oracle) uses oracle anchor with highest centerline score (see
supplementary) and ALAN (BoFA) picks best from top-k hypothesized lane anchors. Results represented by Table 2 demonstrate that all our ALAN evaluations either show on par performance or significantly outperform other baselines on several metrics with at least 11% improvements in terms of mADE$_{10}$ and 25% boost in terms mFDE$_{10}$ from our BoFA method. Moreover, all our ALAN predictions provide an OffRoadRate of 0.01 showing only 1% of the predicted trajectories fall outside the road. This is significantly lower compared to other baselines where they have 7% or higher OffRoadRate’s. This strong coupling of output predictions with the semantics can be attributed to the anchor lanes that help in providing output predictions in the lane direction. Other approaches like [8, 33] use trajectories extracted from the train set, either as anchors or to perform classification, this can lead to poor generalization of outputs to unseen scenarios and trajectories with complex lane structure. Moreover, we would like to note that our ALAN performance is understated due issues such as unconnected lanes and places without lane centerlines in the data leading to bad anchors. We talk about such situations in supplementary but have not removed these here for benchmark purposes.

Ablation Study: Further, we perform ablation studies of our ALAN along with the proposed DAC and other variants in Table 3. We first introduce hypercolumn descriptors [2] to extract multi-scale features and compare it with using a global context vector fed as input to the decoder. Then we investigate several variants of our ALAN predictions. First, we add reference centerline as input and predict trajectories in $xy$ coordinate space (MCL + Poly). This improved the performance significantly. Using lane centerlines as anchors and predicting trajectories in $nt$ space (MCL+LA-$nt$) performed a little worse but we attribute this to networks difficulty in figuring out agent dynamics from anchor based inputs. For example, two trajectories with the same $nt$ coordinates can have different dynamics based on the lane that they’re traveling. So we further add $xy$ coordinates as input and predict auxiliary trajectories in cartesian space (MCL+LA-$ntxy$). As it is shown in Table 3, making such auxiliary predictions improved the primary anchor based outputs. Further, we regularize our anchor outputs using auxiliary predictions and vice-versa. The intuition is that anchor outputs can benefit from auxiliary predictions when there’s a bad input anchor since auxiliary predictions are not constrained to provide trajectories along the lane direction. Adding a regularizer to match our primary and auxiliary trajectories significantly improved our anchor output performance as seen in Table 3 from MCL+LA-$ntxy$+Reg values.

Comparing variations of ALAN in Table 3, it can be inferred that network trained with DAC beats the EWT A and RWTA objectives confirming the ability of the proposed DAC method to produce diverse hypotheses and capture the data distribution better. Please note that although we perform evaluations for DAC in trajectory prediction setting, MCL[26] techniques are applicable in a wide range of problems where our DAC method can be used as a better initialization strategy for WTA objectives.

5.2.4 Qualitative Results

Figure 5 shows qualitative results from ALAN. In general, using lane as anchors and transforming the prediction problem to $nt$ space can be helpful to guide the prediction and follow semantics. As we predict trajectories for a longer time horizon the executed trajectories become complex with more than just one straight or turn maneuvers where using lane as anchors can simplify the problem.

6. Conclusion

In this paper we addressed issues related to learning multi-modal outputs using WTA objectives and using driving knowledge to impose constraints on output predictions. First, we introduced a novel DAC approach that learns diverse hypotheses to capture the data distribution better. Please note that although we perform evaluations for DAC in trajectory prediction setting, MCL[26] techniques are applicable in a wide range of problems where our DAC method can be used as a better initialization strategy for WTA objectives.

<table>
<thead>
<tr>
<th>Model</th>
<th>mADE$_{1}$</th>
<th>mADE$_{5}$</th>
<th>mADE$_{10}$</th>
<th>Miss$_{2,5}$</th>
<th>Miss$_{2,10}$</th>
<th>mFDE$_{1}$</th>
<th>mFDE$_{5}$</th>
<th>mFDE$_{10}$</th>
<th>OffRoadRate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVAE</td>
<td>5.51</td>
<td>2.12</td>
<td>1.55</td>
<td>76</td>
<td>62</td>
<td>12.03</td>
<td>4.45</td>
<td>2.85</td>
<td>0.03</td>
</tr>
<tr>
<td>MCL + Global</td>
<td>8.45</td>
<td>2.85</td>
<td>1.88</td>
<td>87</td>
<td>75</td>
<td>17.52</td>
<td>5.34</td>
<td>3.05</td>
<td>0.16</td>
</tr>
<tr>
<td>MCL + Hyper</td>
<td>5.55</td>
<td>1.99</td>
<td>1.33</td>
<td>72</td>
<td>58</td>
<td>12.11</td>
<td>3.81</td>
<td>2.26</td>
<td>0.12</td>
</tr>
<tr>
<td>MCL + Poly</td>
<td>6.50</td>
<td>2.03</td>
<td>1.27</td>
<td>77</td>
<td>57</td>
<td>13.6</td>
<td>3.88</td>
<td>2.01</td>
<td>0.05</td>
</tr>
<tr>
<td>MCL + LA - nt</td>
<td>4.69</td>
<td>2.62</td>
<td>1.45</td>
<td>78</td>
<td>59</td>
<td>9.86</td>
<td>4.83</td>
<td>2.26</td>
<td>0.05</td>
</tr>
<tr>
<td>MCL + LA - ntxy</td>
<td>6.65</td>
<td>2.14</td>
<td>1.41</td>
<td>75</td>
<td>53</td>
<td>13.86</td>
<td>3.97</td>
<td>2.18</td>
<td>0.01</td>
</tr>
<tr>
<td>MCL + LA - ntxy + Reg + WTA</td>
<td>7.45</td>
<td>3.91</td>
<td>1.72</td>
<td>82</td>
<td>71</td>
<td>13.5</td>
<td>6.49</td>
<td>2.37</td>
<td>0.01</td>
</tr>
<tr>
<td>MCL + LA - ntxy + Reg + RWTA</td>
<td>4.41</td>
<td>2.55</td>
<td>1.21</td>
<td>64</td>
<td>45</td>
<td>9.22</td>
<td>5.03</td>
<td>1.77</td>
<td>0.01</td>
</tr>
<tr>
<td>MCL + LA - ntxy + Reg + EWT A</td>
<td>4.38</td>
<td>2.06</td>
<td>1.20</td>
<td>64</td>
<td>52</td>
<td>9.16</td>
<td>3.83</td>
<td>1.76</td>
<td>0.01</td>
</tr>
<tr>
<td>MCL + LA - ntxy + Reg + DAC</td>
<td>4.31</td>
<td>2.10</td>
<td>1.17</td>
<td>63</td>
<td>50</td>
<td>9.06</td>
<td>3.98</td>
<td>1.73</td>
<td>0.01</td>
</tr>
</tbody>
</table>
References


[10] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks, 2017. 2


[26] Stefan Lee, Senthil Piransewakam, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles, 2016. 1, 2, 3, 6, 8


[28] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph


