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

Sriram Narayanan[1] Ramin Moslemi[1] Francesco Pittaluga[1] Buyu Liu[1]
Manmohan Chandraker[1,2]
[1]NEC Labs America, [2]UC San Diego

Abstract

In the supplementary, we provide further details about ALAN architecture and anchor selection. Additionally, we show quantitative and qualitative comparisons for proposed DAC and ALAN framework.

1. Further Details for ALAN

1.1. Network Architecture

In this section, we provide more details about our network architecture.

Centerline Encoder: It takes lane anchor for each individual agent in the form of $p \times 2$ points and reshapes them as $2p \times 1 \times 1$ vector. We then pass it through a series of 1×1 convolutions with 256,128,64 output filters.

Past Trajectory Encoder: It contains an embedding layer and an LSTM encoder. The embedding layer takes a 5 dimensional vector containing $X_i^t, N_{i,k}^t$ for every timestep along with a boolean mask indicating if trajectory information is available for the timestep and produces an embedded vector of 16 dimensions. Then the embedded past trajectory is fed through a LSTM of hidden size 64 to produce a past state vector of 64 dimensions.

Multi-Agent Convolutional Encoder: The past state vector (64 dim) and embedded centerline vector (64 dim) are concatenated to form a 128 dimensional vector for every agent. The agent specific information is then encoded at its respective location to form a $H \times W \times 128$ vector. This along with the BEV map of size $H \times W \times 3$ is provided as input to this module. The module uses a ResNet-18[6] encoder backbone where we replace the first layer with a convolutions of 3×3 kernel size and stride 2. We then pass the features through ResNet[6] base layers from 1 to 7.

Hypercolumn Trajectory Decoder: First we extract hypercolumn descriptors for every agent based on the agent's location in the BEV map. Specifically, we extract hypercolumn features from layers 0,2,4,5,6,7 and pass them through a series of 1×1 convolutions containing 2048,1024,1024 filters. Finally, a 1×1 output convolution then produces primary and auxiliary outputs.

Ranking Module: It takes in 1024 features vectors from the Hypercolumn Trajectory Decoder before the final output layer and passes them through a 1×1 convolution with 1024 filters and finally produces M trajectory scores as output.

1.2. Learning

Our input BEV map is of size 256×256 dimensions at a resolution of $0.5m$ per pixel. The models are trained with Adam[8] optimizer with an initial learning rate of 1e-4 and batch size 8. We use exponential lr decay with gamma value 0.95 called after every epoch. The hypotheses are split in case of DAC after every 2000 iterations and the models are trained for 150k iterations (approximately 38 epochs). The ALAN is implemented in pytorch[11] and trained on NVIDIA RTX 2080Ti GPU.

1.3. Nuscenes Dataset

Approximately 2.5% of validation set contains bad anchors, such as some due to either unconnected lanes or places without lane centerlines. In such cases, for the benchmark evaluation we use the nearest lane centerline that is closest to the trajectory. Further, we evaluate our method by approximately removing examples which have average normal distance of the past trajectory to nearest lane greater than a threshold distance (3 meters) and compare it with baselines [12, 3]. As seen from Table 1 our proposed ALAN provides even better performance when bad anchors are removed.

2. Anchor Retrieval

Nuscenes[1] provides HD map data containing lane centerline information represented a sequence of points. We follow steps similar to [7] in order to retrieve plausible lanes.

Identify Closest Lanes Given a position of the vehicle in city coordinates we first identify a set of closest lane

Table 1: Nuscenes Evaluations. Removing bad anchors based on normal distance to lane (threshold = 3m). Total validation examples after removing = 8823.

Model	mADE_1	mADE_5	mADE_10	Miss_2_5	Miss_2_10	mFDE_1	mFDE_5	mFDE_10	OffRoadRate
CoverNet[12]	6.81	3.09	2.39	87	76	13.9	5.92	4.27	0.13
MTP [3]	4.14	2.80	1.74	70	53	9.91	6.42	3.71	0.11
ALAN (top-M)	4.49	1.79	1.14	58	47	9.76	3.41	1.76	0.003
ALAN (Oracle)	4.48	1.70	1.08	57	46	9.72	3.16	1.60	0.004
ALAN (BofA)	4.54	1.69	1.03	55	43	9.84	3.20	1.56	0.006

segments to the vehicle within a radius d.

Retrieve Candidate Anchors We identify plausible lane anchors by traversing through successor and predecessor lanes till a threshold distance. Several connected lane segments from an anchor.

Prune Duplicate Anchors We then filter candidate anchors by removing lanes, which are a subset of others.

Heuristic based Pruning Based on the vehicle's velocity we identify a look ahead point on the lane for a future timestep T and remove duplicate anchors which pass through the same point. This is done to further reduce the number of plausible anchors and remove duplicates which only diverge after a sufficient distance from the vehicle's position.

Distance Along Lane Score Then we rank the candidates anchors based on distance travelled along the lane by the vehicle. First, we calculate the corresponding nt coordinates for the trajectory along every plausible anchor. The score is determined as the absolute sum of normal values for every timestep in the trajectory. Anchors are then ranked based on their scores.

Centerline Yaw Score Further, we rank candidate anchors which have same distance along lane score based on centerline yaw score. It is calculated as the absolute difference between yaw angle of the vehicle and lane yaw angle at a point closest to the vehicle.

Learning Every plausible anchor divided into p equally spaced points, in our case $p = 150$. For training, we use the oracle anchor where oracle is determined based on the trajectory information from $1...T$. During inference, for ALAN (top-M) we rank trajectories using observed locations from $1...t_{obs}$.

3. Additional Results

In this section, we show some additional quantitative and qualitative comparisons for the proposed DAC and ALAN framework.

3.1. DAC Qualitative Comparisons

We evaluate our proposed DAC through additional simulations including modes with non-uniform probabilities and compare it with previous WTA objectives [9, 13, 10] (shown in Figure 1). As observed, WTA[9] leaves many hypothesis untrained. While RWTA[13] solves the convergence problem in WTA it brings non-winner hypothesis to an equilibrium position due to residual constraints. Further, EWTA[10] captures the data distribution better but still suffers from the problem of spurious modes as hypothesis can be attracted towards many ground truths and finally left untrained as only top k hypotheses are penalized. Finally, proposed DAC captures the data distribution as good or better even when modes have such low likelihoods and solves the problem of spurious modes by making use of all hypotheses. In every stage, DAC reaches close to a Centroidal Voronoi Tessellation[4] with effective number of outputs increasing at every stage, leading to hypotheses capturing some probability mass and thus avoiding the spurious mode problem.

Method (mIoU)	DeepLabV3	DeepLabV3 + WTA	DeepLabV3 + DAC
Pascal2012 (val set)	76.83	82.14	83.44

Table 2: mIoU score for DeepLabV3[2] on Pascal2012[5] dataset with no multimodality, WTA(k=3) and DAC on Pascal[5] val set.

3.2. DAC on Other Networks

Further, we implement our proposed DAC on other popular networks such as DeepLabV3[2] where we train and test on PascalVOC2012[5] dataset for semantic segmentation task and calculate the IoU scores for 21 classes. Our mIoU scores across all clases is reported in Table 2. Here, we modify the final layer of DeepLabV3[2] to contain multiple segmentation heads in order to implement DAC and WTA.

3.3. ALAN Qualitative Comparisons

Figure 2 shows qualitative comparison of ALAN with other baselines [12, 3].

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 1

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017. 2

[3] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. *2019 International Conference on Robotics and Automation (ICRA)*, May 2019. 1, 2, 5

[4] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM review*, 41(4):637–676, 1999. 2

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1

[7] Siddhesh Khandelwal, William Qi, Jagjeet Singh, Andrew Hartnett, and Deva Ramanan. What-if motion prediction for autonomous driving, 2020. 1

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 1

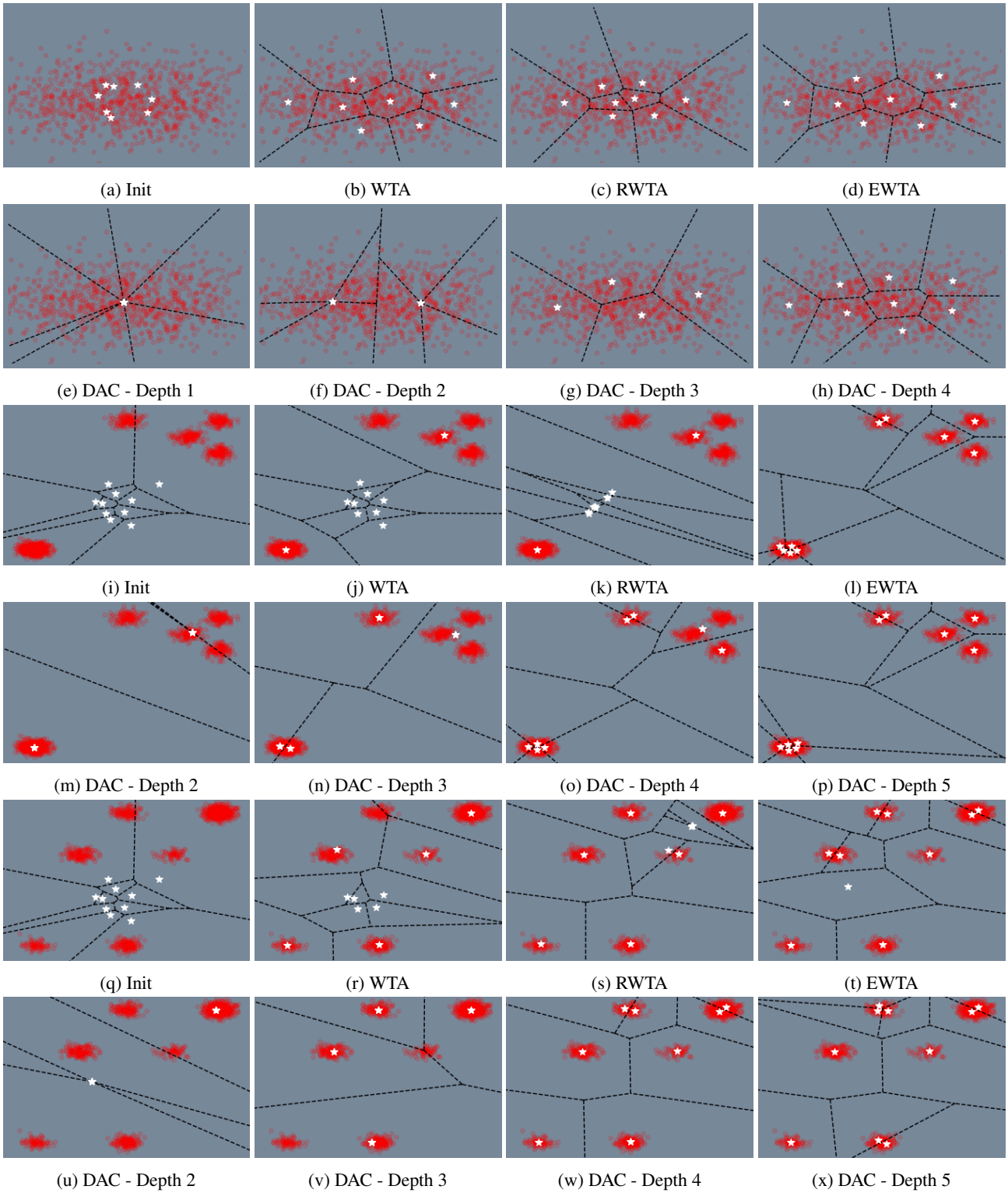
[9] Stefan Lee, Senthil Purushwalkam, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles, 2016. 2

[10] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 2

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[12] Tung Phan-Minh, Elena Corina Grigore, Freddy A. Boulton, Oscar Beijbom, and Eric M. Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 5

[13] Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D. Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses, 2017. 2



(a) Init (b) WTA (c) RWTA (d) EWTA

(e) DAC - Depth 1 (f) DAC - Depth 2 (g) DAC - Depth 3 (h) DAC - Depth 4

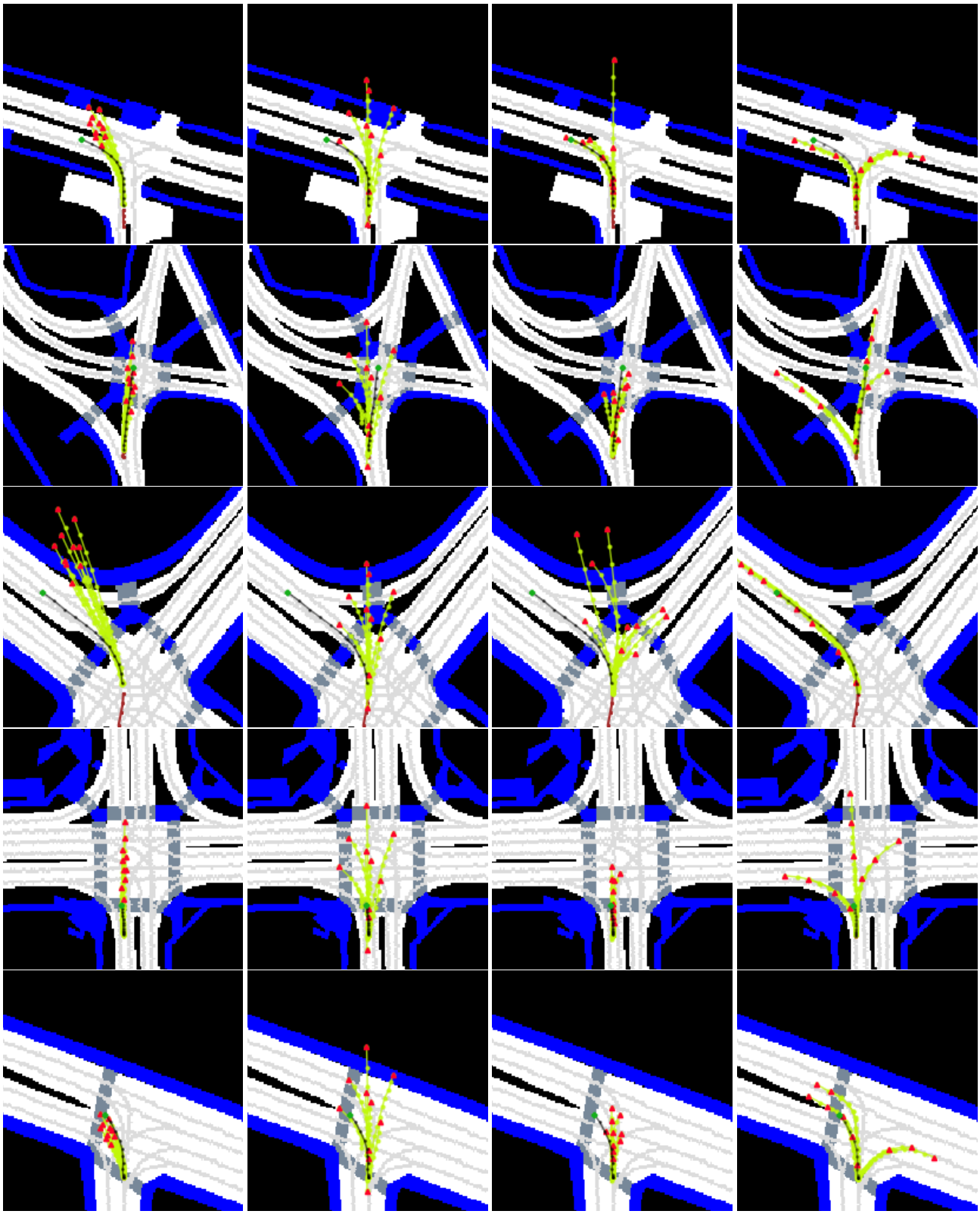
(i) Init (j) WTA (k) RWTA (l) EWTA

(m) DAC - Depth 2 (n) DAC - Depth 3 (o) DAC - Depth 4 (p) DAC - Depth 5

(q) Init (r) WTA (s) RWTA (t) EWTA

(u) DAC - Depth 2 (v) DAC - Depth 3 (w) DAC - Depth 4 (x) DAC - Depth 5

Figure 1: Additional toy examples including modes with non-uniform likelihoods comparing DAC with other WTA family of objectives. The first two rows shows a mixture distribution with high variance and second two rows show a well separated gaussian mixture model. Other examples contain modes with $\pi = \{5, 7.5, 12.5, 50\}\%$. As seen from Row 3, WTA captures the distribution poorly by either leaving many hypotheses untrained or by introducing spurious modes that do not correspond to the data distribution. On the other hand, every hypothesis in proposed DAC captures some part of the data as seen from its voronoi space.



(a) MTP[3] (b) CoverNet[12] Epsilon=8 (c) CoverNet[12] Epsilon=2 (d) ALAN (Ours)

Figure 2: Shows comparison of ALAN predictions with baselines. The past trajectory is in brown and the GT is shown in black. The endpoint of GT is shown as a green dot. The predicted trajectories are shown in green and their endpoints as triangles. The final trajectories are chosen based on the predicted IOC score of each trajectory. As observed ALAN predictions are more semantically aligned in comparison.