Supplementary Material for: Overcoming Limitations of Mixture Density Networks: A Sampling and Fitting Framework for Multimodal Future Prediction

Osama Makansi, Eddy Ilg, Özgün Çiçek and Thomas Brox
University of Freiburg
makansio,ilge,cicek,brox@cs.uni-freiburg.de

1. CPI Dataset

For evaluating multimodal future predictions, we present a simple toy dataset. The dataset consists of a car and a pedestrian and we name it Car Pedestrian Interaction (CPI) dataset. It is targeted to predicting the future conditioned on this interaction. In the evaluation one can see whether methods just predict independent possible futures for both actors or if they actually constrain these predictions, taking the interactions into account (visible in Figure 4 of the main paper). We show more examples of the data in Figure 1. The dataset and code to generate it will be made available upon publication. We will now describe the policy used to create the dataset.

Let $\mathbf{x}_{P,t}$, $\mathbf{x}_{C,t}$ denote the locations of car and pedestrian at time t. For the car we define a bounding box of size 40×40 pixels and for the pedestrian of size 20×20. We denote the pixel regions covered by these boxes by $\mathbf{r}_P$ and $\mathbf{r}_C$ respectively. We furthermore define the areas of the scene shown in Figure 2. In the beginning of a sequence (for $t = 0$), we use rejection sampling to sample valid positions for both actors (such that the pedestrian is contained completely in $R_P \cup R_S$ and the car contained completely in R_V). We define a sets of possible displacements for pedestrian and car as:

$$\begin{aligned} \alpha_P &= \{\mathbf{v}(0°), \mathbf{v}(45°), \mathbf{v}(90°), ..., \mathbf{v}(360°)\} \text{ and} \\ \alpha_C &= \{\mathbf{v}(0°), \mathbf{v}(90°), \mathbf{v}(180°), \mathbf{v}(270°)\}, \end{aligned}$$

where $\mathbf{v}(\gamma) = 10.0(sin(\gamma), cos(\gamma))$. With adding a displacement to a bounding box $\mathbf{r}$, we indicate that the whole box is shifted. We furthermore define a set of helper functions given in Table 1. For pedestrian and car, we define the following states:

$$\begin{aligned} s_{P,t} &\in \{\text{TC}, \text{SC}, \text{C}, \text{FC}, \text{AC}\} \quad (\text{see Table 2}) \text{ and} \\ s_{C,t} &\in \{\text{C}, \text{SC}, \text{FC}, \text{OC}\} \quad (\text{see Table 3}), \end{aligned}$$

and the world state as:

$$\mathbf{w}_t = (\mathbf{x}_{P,t}, \mathbf{x}_{C,t}, E),$$

Name	Description
$\text{argmin}_i(x)$	i-th smallest argument
$\text{argmax}_i(x)$	i-th largest argument
$\text{dtc}(\mathbf{x})$	distance of $\mathbf{x}$ to the closest corner of the pedestrian area R_P
$\text{ad}(\mathbf{a_1}, \mathbf{a}_2)$	Angle difference between actions $\mathbf{a_1}$ and $\mathbf{a}_2$
$\text{ov}(\mathbf{a}_x, R)$	Number of pixels overlapping from the bounding box $\mathbf{r}_x$ of actor x and region R, after action $\mathbf{a}$ was taken

Table 1: Helper functions.

where E is the given environment (in this case the crossroad). We define the history of states for pedestrian and car as:

$$\begin{aligned} \mathbf{h}_{P,t} &= (s_{P,0}, ..., s_{P,t}) \text{ and} \\ \mathbf{h}_{C,t} &= (s_{C,0}, ..., s_{C,t}). \end{aligned}$$

The current state of pedestrian and car are then determined from their respective histories and the world state:

$$\begin{aligned} s_{P,t} &= F_P(\mathbf{h}_{P,t-1}, w_t) \quad (\text{see Table 4}) \text{ and} \\ s_{C,t} &= F_C(\mathbf{h}_{C,t-1}, w_t) \quad (\text{see Table 5}). \end{aligned}$$

Given the states, we then define distributions over possible actions and sample from these to update locations:

$$\begin{aligned} \mathbf{a}_{P,t} &\sim \sum_k \pi_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k),\ (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k) \in A_P(s_{P,t}, s_{C,t}), \\ \mathbf{a}_{C,t} &\sim \sum_k \pi_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k),\ (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k) \in A_C(s_{P,t}, s_{C,t}), \\ \mathbf{x}_{P,t+1} &= \mathbf{x}_{P,t} + \mathbf{a}_{P,t} \text{ and} \\ \mathbf{x}_{C,t+1} &= \mathbf{x}_{C,t} + \mathbf{a}_{C,t}, \end{aligned}$$

where $A_P(\cdot)$ and $A_C(\cdot)$ are the parameter mapping functions described in Tables 6 and 7. We then use this policy to

State	Description
TC	Moving **T**owards **C**rossing
SC	**S**tart **C**rossing
C	**C**rossing
FC	**F**inish **C**rossing
AC	**A**lready **C**rossed

Table 2: List of possible pedestrian states.

State	Description
SC	**S**tart **C**rossing
C	**C**rossing
FC	**F**inish **C**rossing
OC	**O**ut of **C**rossing

Table 3: List of possible car states.

$F_P(\mathbf{h}_{P,t-1}, w_t) =$	
TC	if $\neg$inter$(\mathbf{r}_{P,t}, R_S)$ and C $\notin \mathbf{h}_{P,t}$
SC	else if inter$(\mathbf{r}_{P,t}, R_S)$ and C $\notin \mathbf{h}_{P,t}$
FC	else if inter$(\mathbf{r}_{P,t}, R_P)$ and C $\in \mathbf{h}_{P,t}$
C	else if inter$(\mathbf{r}_{P,t}, R_S)$
AC	else if $\neg$inter$(\mathbf{r}_{P,t}, R_S)$ and C $\in \mathbf{h}_{P,t}$

Table 4: List of possible pedestrian states determined by the history and world state. inter$(A, B) = [A \cap B \neq \emptyset]$. For region definitions (R_S, R_P) see Figure 2.

$F_C(\mathbf{h}_{C,t-1}, w_t) =$	
C	if within$(\mathbf{r}_{C,t}, R_X)$ and C $\notin \mathbf{h}_{C,t}$
SC	else if inter$(\mathbf{r}_{C,t}, R_S)$ and C $\notin \mathbf{h}_{C,t}$
FC	else if inter$(\mathbf{r}_{C,t}, R_S)$ and C $\in \mathbf{h}_{C,t}$
OC	else if $\neg$within$(\mathbf{r}_{C,t}, R_X)$

Table 5: List of possible pedestrian states determined by the history and world state. inter$(A, B) = [A \cap B \neq \emptyset]$. within$(A, B) = [A \cap B = B]$. For region definitions (R_X, R_S) see Figure 2.

generate $20k$ sequences with three image frames. For each sequence, we generate 10 different random futures resulting in $200k$ samples for training in total.

$s_{P,t}, s_{C,t}$	$\{(\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1), ..., (\pi_n, \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n)\} = A_P(s_{P,t}, s_{C,t})$
TC,*	$\mu_1 = \underset{\mathbf{a}\in\alpha_\mathbf{P}}{\text{argmin}_1}(\text{dtc}(\mathbf{x}_{P,t} + \mathbf{a}))$ $\mu_2 = \underset{\mathbf{a}\in\alpha_\mathbf{P}}{\text{argmin}_2}(\text{dtc}(\mathbf{x}_{P,t} + \mathbf{a}))$ $\pi = (0.7, 0.3)$ $\sigma_i = 2.0$
SC,{SC,C,FC}	$\mu_1 = (0, 0)$ $\pi_1 = 1.0$ $\sigma_1 = 0$
SC,OC	$\mu_1 = \underset{\mathbf{a}\in\alpha_\mathbf{P}}{\text{argmin}_1}(\text{ad}(\mathbf{a}, \mathbf{a}_{P,t-1}) - \text{ov}(\mathbf{a}, R_S))$ $\pi_1 = 1.0$ $\sigma_1 = 2.0$
C,*	$\mu_1 = \underset{\mathbf{a}\in\alpha_\mathbf{P}}{\text{argmin}_1}(2 * \text{ad}(\mathbf{a}, \mathbf{a}_{P,t-1}) - \text{ov}(\mathbf{a}, R_S))$ $\pi_1 = 1.0$ $\sigma_1 = 2.0$
FC,*	$\mu_1 = \underset{\mathbf{a}\in\alpha_\mathbf{P}}{\text{argmin}_1}(\text{ad}(\mathbf{a}, \mathbf{a}_{P,t-1}) - \text{ov}(\mathbf{a}, R_P))$ $\pi_1 = 1.0$ $\sigma_1 = 2.0$
AC,*	$\mu_i = \underset{\mathbf{a}\in\alpha_\mathbf{P}}{\text{argmax}_i}(\text{dtc}(\mathbf{x}_{P,t} + \mathbf{a}))$ for $i = 1..4$ $\pi = (0.4, 0.2, 0.2, 0.2)$ $\sigma_i = 2.0$

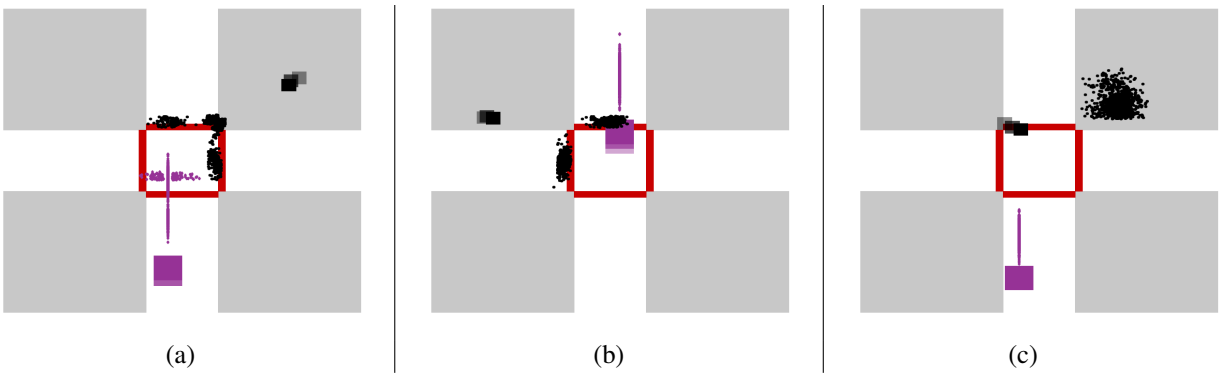
Table 6: State to distribution parameter mapping for the pedestrian.

$s_{P,t}, s_{C,t}$	$\{(\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1), ..., (\pi_n, \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n)\} = A_C(s_{P,t}, s_{C,t})$
{C, SC, FC}, *	$\mu_1 = (0, 0)$ $\pi_1 = 1.0$ $\sigma_1 = 0$
*, C	$\mu_i = \underset{\mathbf{a}\in\alpha_\mathbf{C}}{\text{argmin}_i}(\text{ad}(\mathbf{a}, \mathbf{a}_{C,t-1}))$ for $i = 1..3$ $\pi_i = 1/3$ $\sigma_i = 2.0$
*, {FC,SC}	$\mu_1 = \underset{\mathbf{a}\in\alpha_\mathbf{C}}{\text{argmin}_1}(\text{ad}(\mathbf{a}, \mathbf{a}_{C,t-1}))$ $\mu_2 = (0, 0)$ $\pi = (0.7, 0.3)$ $\sigma = (2.0, 0)$
*, {OC}	$\mu_1 = \underset{\mathbf{a}\in\alpha_\mathbf{C}}{\text{argmin}_1}(\text{ad}(\mathbf{a}, \mathbf{a}_{C,t-1}))$ $\mu_2 = (0, 0)$ $\pi = (0.8, 0.3)$ $\sigma = (2.0, 0)$

Table 7: State to distribution parameter mapping for the car.

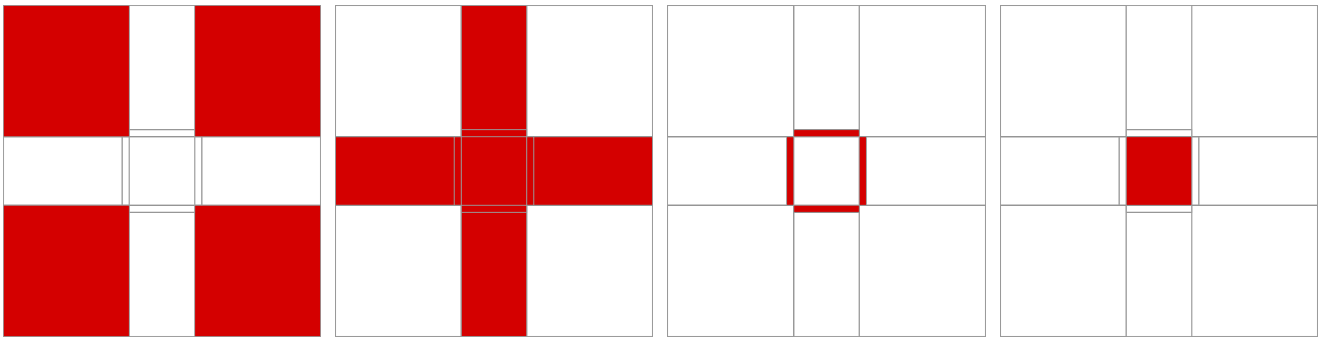
2. Architecture

We base our architecture on the encoder of FlowNetS [1]. Architecture details are given in Table 8.



(a) (b) (c)

Figure 1: Examples from our CPI dataset. Black rectangles denote the current and past locations of the pedestrian, while black dots indicate its future locations ($\Delta t = 20$). Same applies to the car but colored in pink. **(a)** Pedestrian and car are heading toward the crossing area. The pedestrian must stop at the corner if the car reaches the crossing before, otherwise he can cross over one of the two crossing areas. The car must also stop before the crossing if the pedestrian is crossing or can enter otherwise. **(b)** The car is leaving the crossing area and therefore only one direction is possible, while the pedestrian does not need to wait and will cross from one of the two possible areas. **(c)** The pedestrian is in the middle of crossing and the future is unimodal in the destination area. The car needs to wait for the pedestrian to finish crossing.



(a) Pavement area R_P (b) Vehicle area R_V (c) Shared area R_S (d) Crossing area R_X

Figure 2: Definition of regions of the CPI dataset.

Name	Ch I/O	InpRes	OutRes	Input
fc7	1024/1024	8×8	1×1	conv6a
fc8	1024/1024	1×1	1×1	fc7
fc9	1024/$N1$	1×1	1×1	fc8
fc10	N_{CH1}/500	1×1	1×1	fc9
droup-out	500/500	1×1	1×1	fc10
fc11	500/$N2$	1×1	1×1	drop-out

Table 8: The top part of the table indicates our base architecture used for MDNs and our first stage. Outputs $N1$ depend on the number of possible output parameters. The bottom part shows the proposed the Mixture Density Fitting (MDF) stage. Outputs $N2$ depend on the number of possible output parameters. Drop-out is performed with dropping probability of 0.5.

3. Baselines

3.1. Kalman Filter

The Kalman filter is a linear filter for time series observations, which contains process and observation noise [2]. It aims to get better estimates of a dynamic process. It is applied recursively. At each time step there are two phases: predict and update.

In the predict phase, the future prediction for $t+1$ is calculated given the previous prediction at t. For this purpose, a model of the underlying process needs to be defined. We define our process over the vector $\mathbf{x}$ of $(\text{location}, \text{velocity})$ and uncertainties $\mathbf{P}$. The equations integrating the predictions are then:

$$\begin{aligned} \mathbf{x}'_{t+1} &= \mathbf{F} \cdot \mathbf{x}_t \,, \\ \mathbf{P}'_{t+1} &= \mathbf{F} \cdot \mathbf{P}'_t \cdot \mathbf{F}^T + \mathbf{Q} \,, \end{aligned}$$

σ_{NP}	EMD
1.0	2.39
3.0	**2.35**
4.0	3.32
10.0	5.07

Table 9: Comparison study on the kernel width of the non-parametric baseline.

where $\mathbf{F}$ is defined as the matrix $(1, \Delta_t; 0, 1)$ and $\mathbf{Q}$ is the process noise. We do not assume any control from outside and assume constant motion. We compute this constant motion as the average of 2 velocities we get from our history of locations.

In the update phase, the future prediction is computed using the observation $\mathbf{z}_{t+1}$ as follows:

$$\begin{aligned} \mathbf{K} &= \mathbf{P}'_{t+1} \cdot (\mathbf{P}'_{t+1} + \mathbf{R}_{t+1})^{-1}, \\ \mathbf{x}_{t+1} &= \mathbf{x}'_{t+1} + \mathbf{K} \cdot (\mathbf{z}_{t+1} - \mathbf{x}'_{t+1}), \\ \mathbf{P}_{t+1} &= \mathbf{P}'_{t+1} - \mathbf{K} \cdot \mathbf{P}'_{t+1}, \end{aligned}$$

where $\mathbf{R}$ is the observation noise.

For our task we can iterate predict and update only 3 times, since we are given 2 history and 1 current observation. However, since our task is future prediction at $t + \Delta t$ and we assume to not have any more observations until (and including) the last time point, we perform the predict phase at the last iteration k times with the constant motion we assumed. This can be seen as extrapolation by constant motion on top of Kalman filtered observations. In this manner the Kalman filter is a robust linear extrapolation to the future with an additional uncertainty estimate. In our experiments the process and the observation noises are both set to 2.0.

3.2. Single Point

For the single point prediction, we apply the first stage of the architecture from Table 8, but we only output a single future position. We train this using the Euclidean Distance loss l_{ED} (Equation (2) of the main paper).

3.3. Distribution Prediction

For the distribution prediction, we apply the first stage of the architecture from Table 8, but we output only mean and variance for a unimodal future distribution. We train this using the NLL loss l_{NLL} (Equation (3) of the main paper).

3.4. Non-parametric

In this variant we use the FlowNetS architecture [1]. The possible future locations are discretized into pixels and a probability $q_{\mathbf{y}}$ for each pixel $\mathbf{y}$ is output through a softmax from the encoder/decoder network.

Variant	EMD-CPI	NLL-SDD
EWTAP-MDF	1.70	9.56
EWTAD-MDF	**1.57**	**9.33**

Table 10: Comparison between the two proposed variants of our sampling-fitting framework.

This transforms the problem into a classification problem, for which a one-hot encoding is usually used as ground truth, assigning a probability of 1 to the true location and 0 to all other locations. However, in this case such an encoding is much too peaked and would only update a single pixel. In practice we therefore blur the one-hot encoding by a Gaussian with variance σ_{NP} (also referred to as soft-classification [3]).

We then minimize the cross-entropy between the output $q_{\mathbf{y}}$ and the distribution $\phi(\mathbf{y}|\hat{\mathbf{y}}, \sigma_{NP})$ (proportional to the KL-Divergence):

$$L_{NP} = \min[-\sum_{\mathbf{y}} \phi(\mathbf{y}|\hat{\mathbf{y}}, \sigma_{NP}) \log(q_{\mathbf{y}})] .$$

We try three different values for σ_{NP} as shown in Table 9 and use $\sigma_{NP} = 3.0$ in practice.

4. Training Details

Training details for our networks are given in Figure 3. To stabilize the training, we also implement an upper bound for σ by passing it through a scaled sigmoid function, the slope in the center scaled to 1.

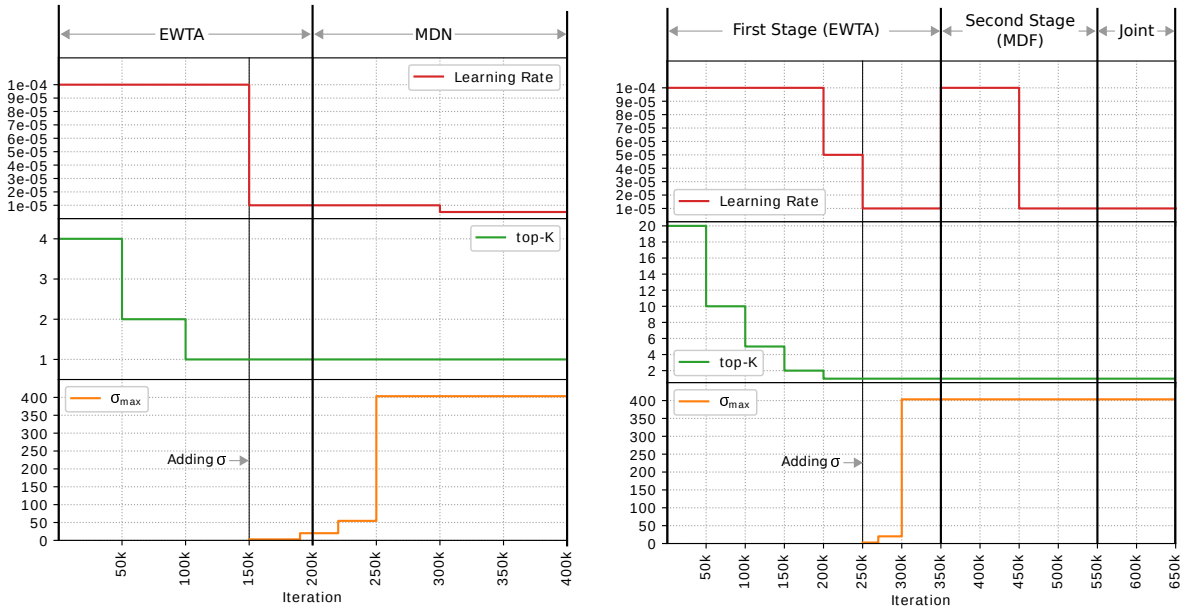
5. Ablation Studies

5.1. Variants of Sampling-Fitting Framework

We show a comparison between the two proposed variants of our framework namely EWTAP-MDF and EWTAD-MDF. We observe that the latter leads to better results on both CPI and SDD datasets (see Table 10). This shows that using WTA with l_{NLL} (Equation 3 of main paper) and using the predicted uncertainties in the MDF stage is in general better than WTA with l_{ED} (Equation 2 of the main paper).

5.2. Effect of History

We conduct an ablation study on the length of the history for the past h frames. Table 11 shows the evaluation on both, SDD and CPI. Intuitively, observing longer history into the past improves the accuracy of our proposed framework on CPI. However, when testing on SDD, a significant improvement is observed when switching from no history ($h = 0$) to one history frame ($h = 1$), while only slight difference is observed when using a longer history ($h = 2$). This indicates that for SDD only observing one previous



(a) Training schedule for MDNs. We first train for 150k iterations using EWTA, optimizing only the means with l_{ED} (Equation 2 of main paper). At 150k, we switch the loss and optimize l_{NLL} (Equation 3 of the main paper) to obtain also variances. At 200k we switch from EWTA to the full mixture density NLL loss. To stabilize the training we set an upper bound on σ, which we increase during training.

(b) Training schedule for EWTAD-MDF. We first train only the first stage for 250k iterations using EWTA, optimizing only the means with l_{ED} (Equation 2 of main paper). At 350k, we add the second stage and train it with the first stage fixed until 550k iterations. After 550k, we remove the loss in the middle and finetune both stages jointly. To stabilize the training we set an upper bound on σ, which we increase during training.

Figure 3: Details of training schedules.

h	EMD-CPI	NLL-SDD
0.0	2.92	15.30
1.0	2.13	**9.13**
2.0	**1.63**	9.30

Table 11: Evaluation of different lengths of the history used in our EWTAD-MDF.

Δt (frames)	EMD-CPI	Δt (sec)	NLL-SDD
10	**1.30**	2.5	**6.94**
20	1.74	5	8.46
40	1.84	10	12.21

Table 12: Evaluation of different time horizons of the future (Δt) on the proposed framework EWTAD-MDF.

frame is sufficient. While one past frame allows to estimate velocity, two past frames allow also to estimate also acceleration. This does not seem to be of importance for SDD.

5.3. Effect of Time Horizon

We conduct an ablation study to analyze the effect of different time horizons in predicting the future. Table 12 shows the evaluation on both CPI and SDD. Clearly predicting longer into the future is a more complex task and therefore the error increases.

5.4. Effect of Number of Hypotheses

We conduct an ablation study on the number of hypotheses generated by our sampling network EWTAD. Table 13 shows the comparison on CPI and SDD. We observe that generating more hypotheses by the sampling network usually leads to better predictions. However, increasing the number of hypotheses is limited by the capacity of the fitting network to fit a mixture modal distribution, thus explaining the slightly worse results for $K = 80$. A deeper and more complex fitting network architecture can be investigated in the future to benefit from more hypotheses.

K	EMD-CPI	NLL-SDD
20	1.63	9.33
40	**1.57**	**9.17**
80	1.65	9.22

Table 13: Evaluation of different number of hypotheses generated by our EWTAD sampling network on the proposed framework EWTAD-MDF.

6. Qualitative WTA variant comparison

Following [4], we analyze our EWTA in a simulation to see if our variant's hypotheses result in a Voronoi Tesselation. Results are shown in Figure 4. We see that WTA fails, since it leaves many hypotheses untouched. RWTA similarly leaves 8 hypotheses at the mean position. Our EWTA not only gives hypotheses as close to Voronoi Tesselation as possible, it also assigns equal number of hypotheses to each cluster, which is relevant for distribution fitting.

7. Failure Cases

In Figure 5 we depict several failure cases that we found. We show results for MDN (first row) and our EWTAD-MDF (second row). In the first column we see that for a scene that has never been seen during training, both models do not generalize well. Note that our predicted variance is still more reasonable. In the second column, we see another example of missing a mode. This failure is due to the unbalanced training data, where turning right in this scene happens very rarely. In the last column, the object of interest is a car, which is an under-sampled class in SDD. The probablity that there is a car in a scene is usually less than 1% and thus this is also a case rarely seen during training.

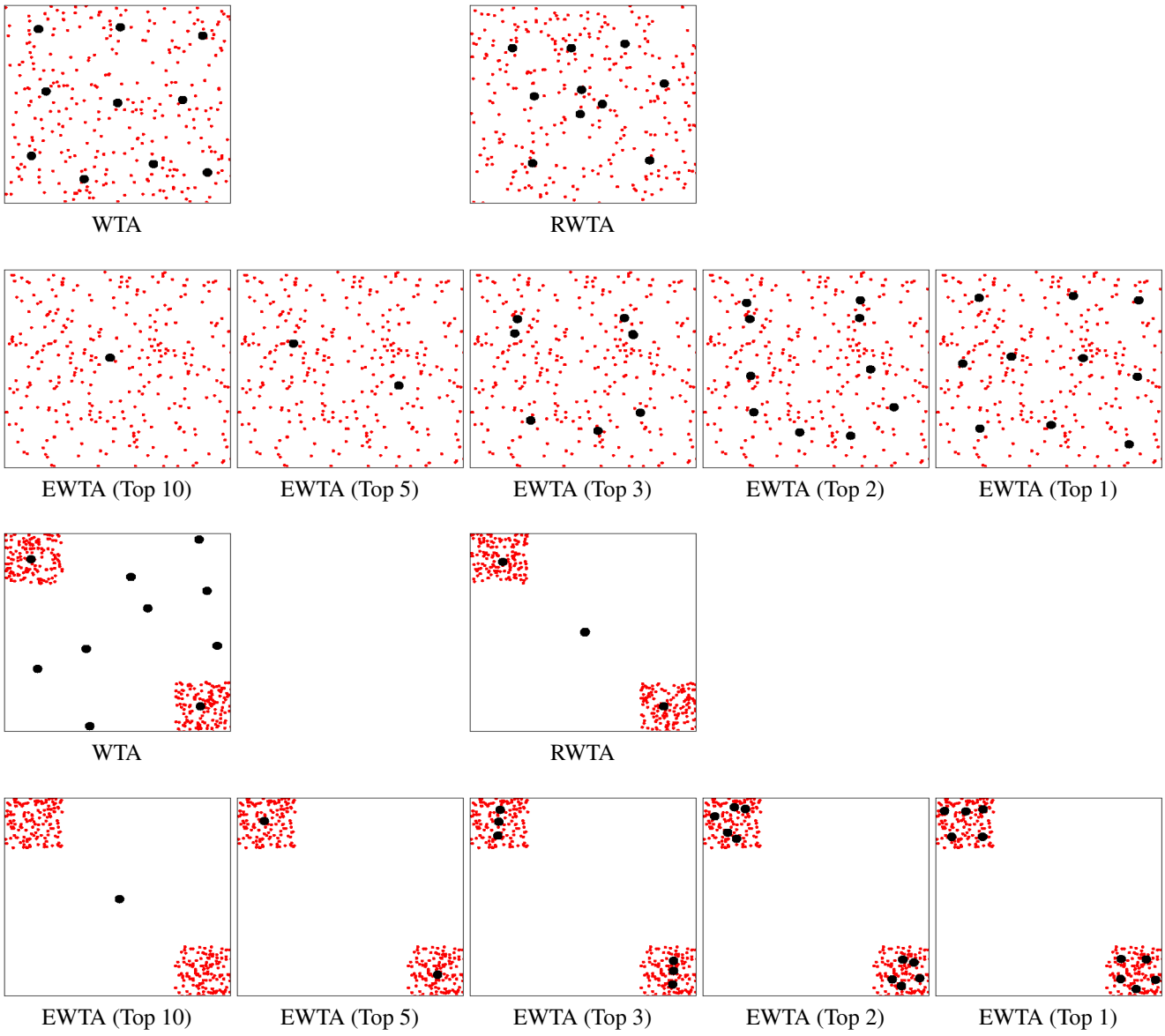


Figure 4: The simulation results from WTA, RWTA and EWTA. First 2 rows are for uniformly distributed samples over the whole space, while the last 2 rows are uniformly distributed samples centered in upper left and bottom right boxes. 300 ground truth samples are shown as red dots and 10 hypotheses as black dots. EWTA produces hypotheses closer to Voronoi Tessellation. Note that for the third row, 8 hypotheses are moved to the center and only 2 capture the ground-truth samples and RWTA fails to produce a Voronoi Tessellation.

Figure 5: Failure cases for MDN (first row) and our EWTAD-MDF (second row) on SDD. Three past locations of the target object are shown as red boxes, while the ground truth is shown as a magenta box. A heatmap overlay is used to show the predicted distribution over future locations. For interpretation see text.

References

[1] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[2] R. E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 1960.

[3] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. 03 2017.

[4] C. Rupprecht, I. Laina, R. DiPietro, M. Baust, F. Tombari, N. Navab, and G. D. Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *International Conference on Computer Vision (ICCV)*, 2017.