Analyzing the Variety Loss in the Context of Probabilistic Trajectory Prediction

Luca Anthony Thiede [*]
Georg-August-Universität Göttingen
luca.thiede@yahoo.com

Pratik Prabhanjan Brahma
Innovation and Engineering Center California
Volkswagen Group of America
pratik.brahma@vw.com

arXiv:1907.10178v1 [cs.LG] 23 Jul 2019

Abstract

Trajectory or behavior prediction of traffic agents is an important component of autonomous driving and robot planning in general. It can be framed as a probabilistic future sequence generation problem and recent literature has studied the applicability of generative models in this context. The variety or Minimum over N (MoN) loss, which tries to minimize the error between the ground truth and the closest of N output predictions, has been used in these recent learning models to improve the diversity of predictions. In this work, we present a proof to show that the MoN loss does not lead to the ground truth probability density function, but approximately to its square root instead. We validate this finding with extensive experiments on both simulated toy as well as real world datasets. We also propose multiple solutions to compensate for the dilation to show improvement of log likelihood of the ground truth samples in the corrected probability density function.

1. Introduction

Trajectory prediction is an important problem with many applications. It can be used for tracking [31], anomaly detection [39], video games [20] or safety simulation [33]. Arguably, the most safety critical application is to use trajectory prediction to help robots navigate environments that they share with other people, for example in the case of self driving cars. While driving, humans have an intuitive anticipation of what other traffic participants are likely to do and react accordingly. This is remarkable since future trajectories are non-deterministic and multimodal (See Figure 1). For this reason, a recent line of research takes the approach to model the natural probability distribution of recorded data, for example with mixture density networks [4, 10], occupancy grids [23] or generative models [40, 18, 13, 26]. One of the recent works, Social–GAN [18], trained their generative model with a combination of the adversarial loss [16] and the variety loss (hereafter referred to as Minimum over N or MoN). They, and many other recently published works [38, 37, 26, 29, 41], used the same MoN loss as a metric to benchmark their model against others, arguing that it is better suited to measure performance on multimodal data as compared to the widely used average displacement error [35]. Additionally we noticed that researchers in other fields apart from trajectory prediction used variations of the MoN loss/metric as well [12, 15, 17, 45, 9, 5, 30]. However, we could not find any theoretical analysis of the MoN loss/metric.

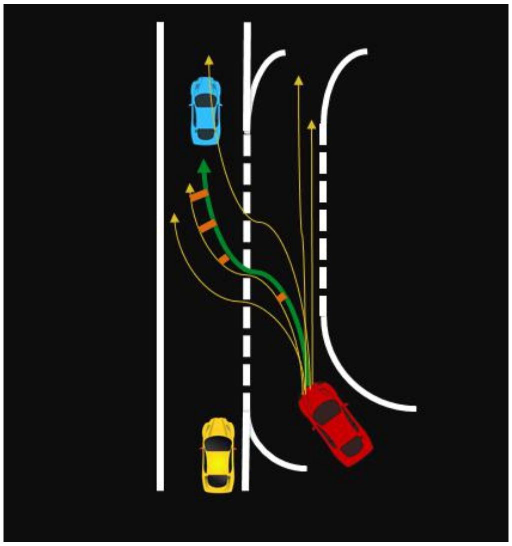


Figure 1: Trajectory prediction is a multimodal problem. In order to learn this distribution, the variety loss can be used. It is computed as the distance between the groundtruth (green) trajectory and its closest prediction.

In this work, we present a proof in section 3 that the optimal solution of the MoN loss is not the ground truth PDF but its square root instead. In section 4, we discuss if MoN can be used as a viable metric nonetheless. Then, in section 5, we propose various algorithms to recover the

[*] Work done during internship at Volkswagen Group of America Innovation and Engineering Center located in Belmont, California.

true PDF from the learned one. We verify these results experimentally on simulated low–dimensional datasets in section 6. We also validate our hypotheses on a highway vehicles and a pedestrian trajectory prediction datasets by applying a compensating transformation on the distribution learned by MoN loss based generative model and show an improvement of ground truth samples in the sense of average marginalized log likelihood.

2. Related work

Trajectory prediction of traffic participants is a difficult problem. The model has to capture the many possible outcomes as well as the interactions of the person/vehicle to be predicted with other traffic participants and the environment. Early attempts of predicting the trajectories of humans under consideration of social interactions used a model of attractive and repulsive social forces [19, 32, 8, 27, 35] with promising results. Other approaches include using Gaussian processes [43] and continuum dynamics [44].
Newer works are more data driven. Some [4, 10] use data to teach a network to predict the parameters of base distributions (Mixture Density Networks) [6, 10]. Others discretize the prediction space into a grid and predict the probability that one of these grid cells is occupied [23, 34]. While these models show promising results, it is difficult to sample trajectories with a longer time horizon. This limitation is overcome by modeling longer trajectories directly using generative models. Generally these models learn to transform samples from a latent space into samples from a data distribution. The best known representatives for generative models are variational autoencoders (VAE) [24, 13, 17] and generative adversarial networks (GANs) [16, 18, 40]. VAEs are trained by auto-encoding samples and optimizing a variational lower bound on the data distribution. GANs, on the other side, learn a discriminator jointly with the generator. The discriminator has the task to separate real data samples from generated ones while the generator has to produce samples that fool the discriminator. It was shown that this training procedure reaches the optimum if and only if the generator has learned the true data distribution [16]. Both models have seen successful applications on a wide array of tasks like texture synthesis [28], super resolution [25], text to image synthesis [36] or image synthesis from a mask [46]. MoN was originally introduced by [12] in the context of 3d point cloud generation and adopted by [18, 11, 9] for trajectory prediction. Other works used MoN loss/metric or similar concepts for depth map prediction [30], 3d reconstruction [15], activity prediction [17], to improve the optimization of the variational lower bound in VAE [5] or for pixel flow prediction [45].

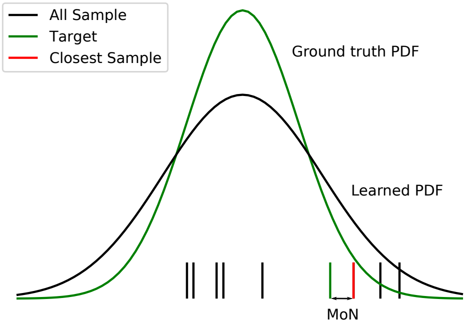


Figure 2: An illustration of the MoN loss in one dimension. Only the error of the sample with the smallest distance to the target sample is considered. This leads a model to learn the square root of the true PDF.

3. The Minimum over N loss

Given is a generative model $\boldsymbol{P(X|}\mathrm{I}\boldsymbol{)}$, where $\boldsymbol{X} \in \mathbb{R}^n$ for some $n \in \mathbb{N}$ (for example $n = 2T$ for 2 dimensional trajectories of length T) is the output to be generated and I is a set of inputs. Then the MoN loss is defined as

$$\mathrm{MoN}_P(\boldsymbol{x^*}) = \min_{\boldsymbol{x_1},...,\boldsymbol{x_N} \overset{\text{iid}}{\sim} P} \left(d(\boldsymbol{x^*}, \boldsymbol{x_1}), d(\boldsymbol{x^*}, \boldsymbol{x_2}), ..., d(\boldsymbol{x^*}, \boldsymbol{x_N})\right) \quad (1)$$

where x^* is a ground truth sample and $\boldsymbol{x_1} \dots \boldsymbol{x_N} \sim \boldsymbol{P(X|}\mathrm{I}\boldsymbol{)}$ are samples generated from the model. The function $d(\cdot, \cdot)$ is some distance metric. One natural choice is the l2 distance $d(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||_2$. An illustration in the one dimensional case is shown in Figure 2.

In this paper we consider the question: given some ground truth probability distribution $P_T(x^*)$, does a model that was learned with the MoN loss converge towards this $P_T(x^*)$? To get a better theoretical grasp, we consider the expectation value of the MoN loss:

Definition 1. *EMoN: Given a probability density* $P(\boldsymbol{x}) : \mathbb{R}^n \to [0, 1]$ *and some point* $\boldsymbol{x^*} \in \mathbb{R}^n$. *Then we define the Expected-Minimum-over-N function*

$$\mathit{EMoN}_P(\boldsymbol{x^*}) = \int \min\left(||\boldsymbol{x^*} - \boldsymbol{x_1}||_2, ||\boldsymbol{x^*} - \boldsymbol{x_2}||_2, ..., ||\boldsymbol{x^*} - \boldsymbol{x_N}||_2\right) P(\boldsymbol{x_1})P(\boldsymbol{x_2}) \dots P(\boldsymbol{x_N})\, \mathrm{d}\boldsymbol{x_1}\, \mathrm{d}\boldsymbol{x_2} \dots \mathrm{d}\boldsymbol{x_N} \quad (2)$$

We can estimate $\text{EMoN}_P(\boldsymbol{x^*})$ with

$$\widehat{\text{EMoN}}_P(\boldsymbol{x^*}) =$$

$$\frac{1}{R}\sum^{R} \min_{\boldsymbol{x_1},...,\boldsymbol{x_N} \overset{\text{iid}}{\sim} P} (||\boldsymbol{x^*}-\boldsymbol{x_1}||_2, ||\boldsymbol{x^*}-\boldsymbol{x_2}||_2, ..., ||\boldsymbol{x^*}-\boldsymbol{x_N}||_2) \tag{3}$$

where R is the sample size for the expectation value. In the referenced literature R in equation (3) is set to $R = 1$. Since the variance of 3 is $\mathcal{O}(\frac{1}{R})$ one could question, if our theoretical results that are based on equation 2 still hold. The experiments show though, that this is indeed the case.

Next, we can consider the expected MoN loss (MoN loss for short):

Definition 2. *MoN loss: Given some target probability $P_T(x) : \mathbb{R}^n \to [0,1]$ we define the Minimum-over-N loss as*

$$L_N(P_T, P) = \int P_T(\boldsymbol{x^*}) EMoN_P(\boldsymbol{x^*})\,\mathrm{d}x^* \tag{4}$$

In a practical context, we would estimate this with samples from our dataset $\mathbf{D}$:

$$\hat{L}_N(P_T, P) = \frac{1}{|\mathbf{D}|}\sum_{x_T \in \mathbf{D}} \widehat{\text{EMoN}}_P(\boldsymbol{x^*}) \tag{5}$$

The following theorem answers the question, whether a model trained with the MoN loss converges towards the true data distribution P_T:

Theorem 1. *For N big enough and P_T differentiable with finite support, the differentiable PDF that minimizes the MoN loss is*

$$\arg\min_P \hat{L}_N(P_T, P) \approx \frac{\sqrt{P_T}}{C} \tag{6}$$

with some normalization constant C.

A proof is presented in the supplementary material. It is remarkable that this means that the MoN loss is a likelihood free loss (similar to the adversarial loss), that is, it does not assume any parametric form of the target distribution and can therefore be used to train a generative model.

If $N = 1$, there is a high chance that a model learns the mean of the known PDF. However with a large number of tries, it has the tendency to put more probability mass into regions with low ground truth probability. This is because putting more samples in areas with high probability will decrease the expected error only a little bit if there are already many samples, while, even if not likely, the prospect of a high error in the low probability area out weighs this decrease. This leads us to the following proposition:

Proposition 1. *Given $N_1 < N_2$, a ground truth PDF $P(\boldsymbol{x})$ and the family of PDFs*

$$P_k(\boldsymbol{x}) := \frac{1}{C_k} P(\boldsymbol{x})^k \tag{7}$$

Let $P_{k_i}(\boldsymbol{x})$ be the PDF out of this family, that minimizes MoN_i. Then

$$k_2 \leq k_1 \tag{8}$$

We verify this proposition in section 6 experimentally. Proposition 1 means, that only considering the family $P_k(x)$, the exponent $k(N)$ that minimizes MoN is monotonically falling with N. Because of Theorem 1, $k = 0.5$ is a strict lower bound. Note, that this does not guarantee that a learner actually converges towards $P_k(\boldsymbol{x})$ (in fact it is easily seen that this is not the case for a multimodal distribution and $N = 1$). We assume that the transformation that recovers the ground truth PDF from the PDF that minimizes MoN (we call this the compensation transformation from now on) belongs to the following family of transformations:

$$T_{\bar{k}}(P(\boldsymbol{x})) = \frac{1}{\int P(\boldsymbol{x})^{\bar{k}}\,\mathrm{d}x} P(\boldsymbol{x})^{\bar{k}} \tag{9}$$

For some practical N (where $\bar{k} = \frac{1}{k}$), proposition 1 gives us the intuition that $\bar{k}$ is going to be less than 2.

4. MoN as a metric

The ideal metric to compare probabilistic models would be a statistical divergence like the Kullback–Leibler divergence or the Jensen-Shannon divergence between the learned and the ground truth distribution. Comparing the KL divergence is equivalent to comparing the log likelihood of the ground truth samples under the models. Since two dimensional trajectories with T time steps in the future is $2T$ dimensional, estimating this log likelihood with generative models, where we do not have direct access to the likelihood of samples, is unfeasible for anything but very small T (for small T the learned PDF can be estimated by sampling from the model).

Recent work [18] in trajectory prediction used MoN as a metric to compare their results against previous ones (*e.g.* [4]). This can be problematic though (particularly if one of the models was trained with the MoN loss while the others were not), as it rewards a model that learned a less sharp distribution. Therefore, following [10] we advocate to use additionally to the MoN a second metric: The average log likelihood of the ground truth from the test set $\mathbf{D}_{\text{Test}}$

under the marginalized learned distribution for every time step t:

$$\mathcal{L}^t_{\mathbf{D}_{\text{Test}}}(P) = \frac{1}{|\mathbf{D}_{\text{Test}}|} \sum_{(x^*_{i,t}, y^*_{i,t}) \in \mathbf{D}_{\text{Test}}} \log \int P(x_1, y_1, ..., x^*_{i,t}, y^*_{i,t}, ..., x_T, y_T) \, \mathrm{d}x_1 \, \mathrm{d}y_1, ..., \mathrm{d}x_{t-1}, \mathrm{d}y_{t-1}, \mathrm{d}x_{t+1}, \mathrm{d}y_{t+1}, ..., \mathrm{d}x_T, \mathrm{d}y_T \tag{10}$$

Since the marginalized distribution is only two dimensional, it can easily be estimated by sampling from the learned generative model and subsequently using some simple density estimation technique like kernel density estimation (KDE) [42].

Using this metric has two advantages. Firstly, when combined with MoN, it gives a better estimation of how well the model really learned the underlying PDF as the marginalized log likelihood favours a model with sharper probability but ignores inter-time step dependencies. The MoN metric, on the other hand, can give a decent estimate of the joint probability of prediction for all time steps even for large T. Secondly, it is useful to have the per time step probability distribution to generate the grid based cost map in order to do ego path planning [14] in autonomous driving. We will elaborate further in section 5.2.

5. Recovering the ground truth PDF

5.1. Sample from squared distribution

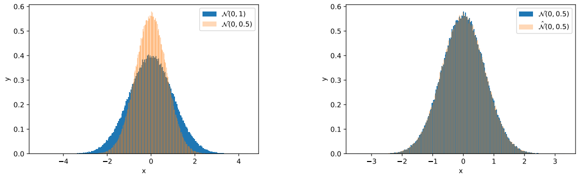
Assuming a learner $P(\boldsymbol{x})$ converged towards $\sqrt{P_T(\boldsymbol{x})}$, we now want to recover the ground truth PDF $P_T(\boldsymbol{x})$. For low–dimensional tasks, we show here a simple way to sample from $P(\boldsymbol{x})^2$ thereby cancelling the square root. For this consider a PDF $P(\boldsymbol{x})$ and bin it in bins of width ϵ. Then the probability that two iid samples fall in the same bin b_i is

$$\begin{aligned}\int_{b_i}\int_{b_i} P(\boldsymbol{x_1}, \boldsymbol{x_2}) \, \mathrm{d}\boldsymbol{x_1} \, \mathrm{d}\boldsymbol{x_2} &= \int_{b_i}\int_{b_i} P(\boldsymbol{x_1}) \cdot P(\boldsymbol{x_2}) \, \mathrm{d}\boldsymbol{x_1} \, \mathrm{d}\boldsymbol{x_2} \\ &= \int_{b_i} P(\boldsymbol{x_1}) \, \mathrm{d}\boldsymbol{x_1} \cdot \int_{b_i} P(\boldsymbol{x_2}) \, \mathrm{d}\boldsymbol{x_2} \\ &= P(b_i) \cdot P(b_i) = P(b_i)^2 \end{aligned} \tag{11}$$

This can be realized by two different algorithms:

- Bin the sample space in bins of width ϵ. Sample from $P(\boldsymbol{x})$, and count in which bin the sample falls. Repeat this until there are two samples in one bin, and choose one of those samples.

- Sample two times from $P(\boldsymbol{x})$. If the samples have a distance of less then $|\boldsymbol{x_1} - \boldsymbol{x_2}| < \epsilon$, choose one of those samples. Otherwise repeat.



(a) Histogram of the Normal Gaussian (blue) and the analytically squared Gaussian (orange).

(b) Histogram of the analytically squared Gaussian (blue) and the estimated squared Gaussian (orange).

Figure 3: (a) Histogram of samples from the Normal Gaussian $\mathcal{N}(0, 1)$ and the analytically squared Gaussian $\mathcal{N}(0, \sqrt{0.5})$. (b) Histogram of samples from $\mathcal{N}(0, \sqrt{0.5})$ and samples from $\hat{\mathcal{N}}(0, \sqrt{0.5})$ that was obtained by applying the squaring compensation on $\mathcal{N}(0, 1)$. It is clearly visible, that $\hat{\mathcal{N}}(0, \sqrt{0.5})$ matches $\mathcal{N}(0, \sqrt{0.5})$ very closely.

The two variations are a trade-off of speed vs memory, with the first one being faster but more memory intensive. Primarily, these two can be used during inference time, to generate samples that come from the target distribution. Without testing this, for unconditional problems one could also think of using it during training with stochastic gradient descent [22] to sample the data point shown to the network. However, due to the curse of dimensionality, this is only possible in relatively low dimensions (or in higher dimensions, if the distribution has a very small support), since otherwise it is too unlikely to generate two samples with $|\boldsymbol{x_1} - \boldsymbol{x_2}| < \epsilon$ for small enough ϵ.

To validate the algorithm, we sampled from $\mathcal{N}(0, 1)$ and from $\frac{1}{C}\mathcal{N}^2(0, 1) = \mathcal{N}(0, \sqrt{0.5})$ and with the proposed algorithm from $\mathcal{N}(0, 1)$ which gives us an estimate of $\frac{1}{C}\mathcal{N}^2(0, 1)$ that we denote as $\hat{\mathcal{N}}(0, \sqrt{0.5})$. The results are depicted in Figure 3.

5.2. Maximum likelihood based recovery

In the previous section, we assumed that the learner converged to $\sqrt{P_T(\boldsymbol{x})}$. However, there are several reasons, why this might not be exactly the case: the model is not expressive enough, the training got stuck in a local minima, N was too small or MoN was used in addition to other losses, like the adversarial loss, that does converge to $P_T(\boldsymbol{x})$. In those cases, squaring the learned distribution could actually move it farther away from the ground truth distribution. At least for some very specific applications, there is still a possibility to compensate for the dilating effect of MoN. One of these applications is trajectory prediction for path planning in autonomous cars. A possible approach to path planning is to create a cost map [14], and then run a path finding algorithm that minimizes these costs under the physical constrains of the vehicle dynamics. In this framework, one can imagine that the probability of a traffic participant being at a certain point in time can simply be framed as costs on the cost map. Finding a path that minimizes these costs then is equivalent to minimizing the probability of crashing with another traffic participant. Since these algorithms only care about whether there will likely be a traffic participant at a certain point in time at a certain point in space, and not how it got there, we only care about the marginalized probability distribution per time step. Since this distribution is only two dimensional, we can easily estimate it by sampling from the trained model and using a kernel density estimator [42] to recover the PDF. The bandwidth of the KDE can be selected via cross validation on a left out set of generated samples [7]. Subsequently, the KDE can be evaluated on a grid and a transformation as defined in (9) can be applied for various $\bar{k}$. At the end the $\bar{k}$ is selected, that maximizes the log likelihood of the ground truth sample for each of these samples. Algorithm 1 makes these steps precise. The parameter $\bar{k}_{\text{opt}}$ can then be found to improve the estimated PDF during inference time by doing the steps of the innermost loop in algorithm 1 with the found $\bar{k}$. Note that this compensation is very lightweight, once $\bar{k}_{\text{opt}}$ is found, for tasks where the KDE reconstruction has to be done anyway.

Algorithm 1 Algorithm to find the best compensation parameter $\bar{k}$ for transformation $T_{\bar{k}}(P)$ of model $P(\boldsymbol{X}|\boldsymbol{I})$ under the inputs $\{\boldsymbol{I}_i\}_{i=0,\dots,K}$ so that the marginalized log likelihood of the ground truth sample $\{(x_i^*, y_i^*)\}_{i=0,\dots,K}$ is maximized. Here, $\{\bar{k}\}_{\text{search}}$ is the search space and $(\boldsymbol{x}, \boldsymbol{y})$ are grid points. n_{sample} is an sufficiently big integer and $\alpha_{\text{split}} \in (0, 1)$.

```
procedure FindBestCompensationParameter($P(\boldsymbol{X}|\boldsymbol{I})$, $\{\boldsymbol{I}_i\}_{i=0,\dots,K}$, $\{(x_i^*, y_i^*)\}_{i=0,\dots,K}$, $n_{\text{sample}}$, $\alpha_{\text{split}}$, $\{\bar{k}\}_{\text{search}}$, $(\boldsymbol{x}, \boldsymbol{y})$)
    $L_{\text{max}} \leftarrow -\infty$
    $\bar{k}_{\text{best}} \leftarrow 0$
    for $\bar{k}$ in $\{\bar{k}\}_{\text{search}}$ do
        $L_{\text{run}} \leftarrow 0$
        for $\boldsymbol{I}_i$ in $\{\boldsymbol{I}_i\}_{i=0,\dots,K}$ do
            $\{s_i^j\}_{j=0,\dots,n_{\text{sample}}} \overset{\text{iid}}{\sim} P(\boldsymbol{X_t}|\boldsymbol{I}_i)$
            Use $\{s_i^j\}_{j=0,\dots,\alpha_{\text{split}} n_{\text{sample}}}$ to fit a KDE and $\{s_i^j\}_{j=\alpha_{\text{split}} n_{\text{sample}},\dots,n_{\text{sample}}}$ to find best bandwidth for the KDE [7]
            $L(\boldsymbol{x}, \boldsymbol{y}) \leftarrow$ evaluate KDE on grid $(\boldsymbol{x}, \boldsymbol{y})$
            $L(\boldsymbol{x}, \boldsymbol{y}) \leftarrow \frac{L(\boldsymbol{x}, \boldsymbol{y})^{\bar{k}}}{\sum_{\boldsymbol{x}, \boldsymbol{y}} L(\boldsymbol{x}, \boldsymbol{y})^{\bar{k}}}$
            $L_{\text{run}} \leftarrow L_{\text{run}} + \log L(x_i^*, y_i^*)$
        end for
        if $L_{\text{run}} > L_{\text{max}}$ then
            $L_{\text{max}} \leftarrow L_{\text{run}}$
            $\bar{k}_{\text{best}} \leftarrow \bar{k}$
        end if
    end for
    return $\bar{k}_{\text{best}}$
end procedure
```

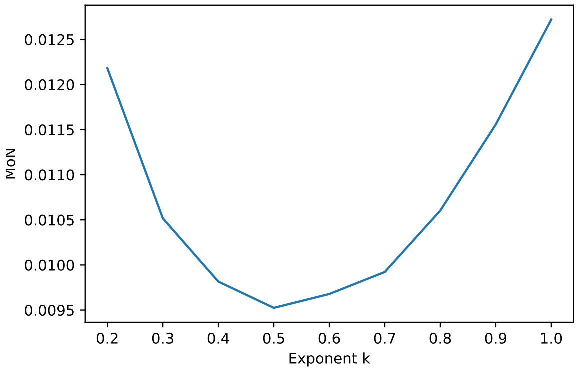


Figure 4: MoN values for different k by using samples from f_1 as the groundtruth distribution and f_1^k as the test distribution with $N = 256$

6. Experiments

6.1. MoN minimum of Mixture of Gaussians

We verify our result on two toy experiments:
For the first experiment we sample $M = 50000$ times from

$$\{x_{1,i}\}_{i=1,\dots,M} \overset{\text{iid}}{\sim} f_1 := \mathcal{N}(0, 1) \tag{12}$$

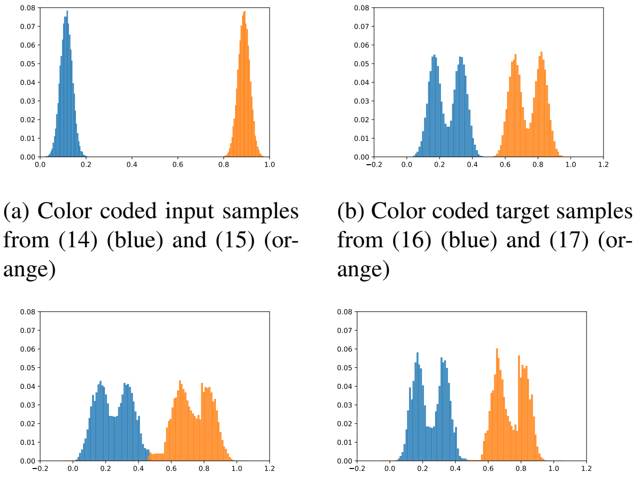
We then consider the family of PDFs

$$\{x_{1,i}^k\}_{i=1,\dots,256} \overset{\text{iid}}{\sim} f_1^k := \frac{1}{C_{f_1,k}} \mathcal{N}(0, 1)^k \tag{13}$$

and sample 256 data points. Subsequently we calculate for each of the M sample from the original distribution the minimum distance from the 256 samples. All of this is averaged over $R = 100$ tries. Note, that we do not learn a model here, but merely search for the k, that minimizes the MoN loss for the respective PDFs. The results are reported in Figure 4. As expected, the minimum value is within our search resolution exactly $k = 0.5$, which means the PDF that minimizes the MoN loss is the square root of the ground truth PDF. This validates Theorem 1.

6.2. Learn Mixture of Gaussians

Theorem 1 and the previous experiment show that the PDF that minimizes the MoN loss for big N is actually the square root of the ground truth PDF. It is not clear though, if a generative model, trained with the MoN loss actually converges towards this solution or if it gets stuck in local minima. We test this with another toy dataset and a very



(a) Color coded input samples from (14) (blue) and (15) (orange)

(b) Color coded target samples from (16) (blue) and (17) (orange)

(c) The learned distribution P_{learned}

(d) The learned distribution squared P^2_{learned}

Figure 5: The input distribution and corresponding targets distribution are shown in (a) and (b) respectively. In (c), samples from the learned distribution P_{learned} are plotted and (d) shows P^2_{learned} which is square of the learned distribution. Obviously P^2_{learned} matches the ground truth distribution better.

simplistic generative model: the dataset consist of inputs, which are randomly sampled either from

$$f_2 = \mathcal{N}(-1.5, 0.1) \tag{14}$$

or

$$f_3 = \mathcal{N}(+1.5, 0.1) \tag{15}$$

and of targets, which are randomly sampled from

$$f_{2,\text{target}} = \frac{1}{C_{f_2,\text{target}}} \left(\mathcal{N}(-2, 1) + \mathcal{N}(-4, 1)\right) \tag{16}$$

or

$$f_{3,\text{target}} = \frac{1}{C_{f_3,\text{target}}} \left(\mathcal{N}(2, 1) + \mathcal{N}(4, 1)\right) \tag{17}$$

respectively. The inputs and targets are illustrated and color coded in Figure 5a and 5b.

For the generative model, we used a very simple neural network consisting of an encoder that predicts the mean and variance of a Gaussian by encoding the samples from distribution 14 or 15 to predict parameter of a Gaussian, and a decoder that takes N samples from the Gaussian and transforms them to minimize the MoN loss (for architecture details we refer to the supplementary details). Note that the model would not be able to learn the correct distribution with a simple mean squared error loss, as it would only learn to generate the mean of the distribution.

Metric	P_{learned}	P^2_{learned}
Jensen–Shannon divergence	0.1282	0.0191

Table 1: The JS divergence between the ground truth PDF and the learned PDF P_{learned} is worse than that between the ground truth and the compensated version P^2_{learned}.

We train the model with the MoN loss with $N = 128$. However we noticed that this consistently led to poor local minima, where the modes that are farther away from the center were poorly predicted. We found it vastly helpful to start with a low N, and then slowly increase it during training till the final N is reached. The resulting learned PDF is depicted in Figure 5c. As one can see, the learned PDF looks dilated. However, since theorem 1 tells us that this should be approximately the square root of the ground truth PDF, we can simply square and normalize over the bins, to recover the ground truth. This is shown in 5d. The qualitative superiority of the compensated PDF is obvious. Also the numerically estimated Jensen–Shannon divergence becomes almost an order of magnitude smaller (See Table 1).

6.3. Dependence of minimizing exponent on N

Next we want to verify proposition 1 experimentally, by repeating the experiment from section 6.1. This time however we search for the MoN minimizing exponent k for different N. The results are plotted in Figure 6. It is obvious that Proposition 1 holds at least for this particular PDF. The same experiment is repeated with a 10 dimensional version of the PDF (See Figure 6). Surprisingly, the results imply that for higher dimensional PDF, the MON loss prefers k close to 0.5 even for small N. This is especially important in the context of using MoN as a metric.

7. Application to trajectory prediction in autonomous vehicles

The problem considered here is to find a model $\boldsymbol{P}(\boldsymbol{Y}|\text{I})$, where $\boldsymbol{Y}$ is a trajectory with $\boldsymbol{Y} = \{(x_1, y_1), (x_2, y_2), ..., (x_T, y_T)\}$ of length T and I is the input. We experiment with the prediction of highway vehicles and pedestrian trajectories.

7.1. NGSIM Dataset

In this section, we will train a generative model using MoN on the Next Generation Simulation (NGSIM) dataset and show that compensating the learned probability distribution using Algorithm 1 will improve the average log likelihood of ground truth samples.

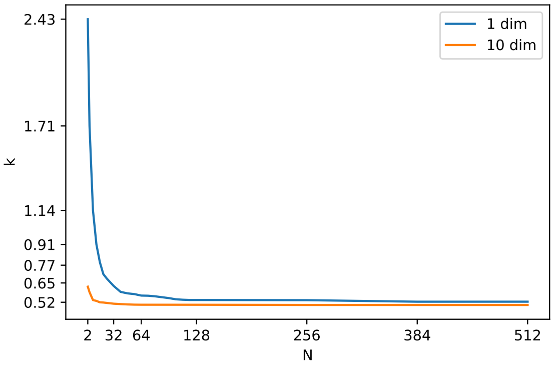
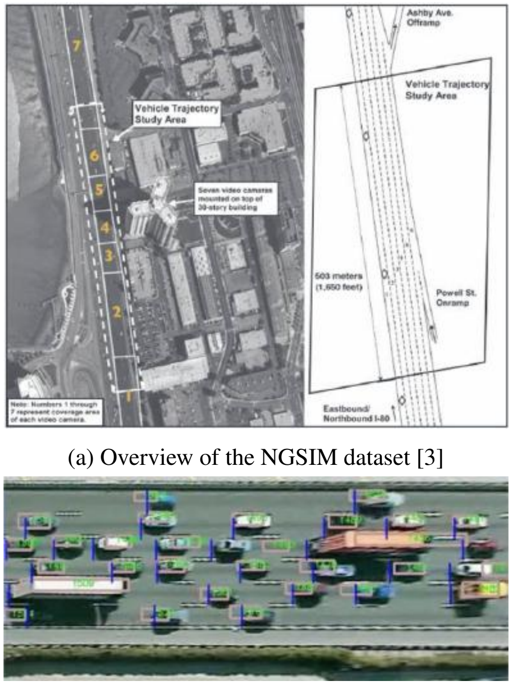


Figure 6: The variation of the k that minimizes the MoN loss is plotted with respect to N for PDFs with dimensionality 1 and 10. Note that the 10 dimensional one converges much faster. This implies that a widespread PDF is preferred by MoN in higher dimensions even for small N.


Ashby Ave. Offramp
Vehicle Trajectory Study Area
Vehicle Trajectory Study Area
Seven video cameras mounted on top of 30-story building
503 meters (1,650 feet)
Powell St. Onramp
Eastbound/ Northbound I-80
Note: Numbers 1 through 7 represent coverage area of each video camera.
7
6
5
4
3
2
1

(a) Overview of the NGSIM dataset [3]

(b) Close up of the highway and tracking of the vehicle [2].

Figure 7: (a) An overview of the highway section the NGSIM dataset was recorded on [3]. (b) Close up of the I-80 with a visualization of the tracking the vehicle.

The NGSIM dataset consists of 45 minutes of vehicles tracked along a section of the I-80 highway which is approximately 0.5 kilometer long (see Fig 7).

Our generative model consists of an LSTM with 128 units that encodes the trajectory of a vehicle and predicts the parameters of a 12 dimensional Gaussian distribution. Then we sample $N = 100$ times from this distribution. These samples are encoded by 2 dense layers, each with 128 units and ReLu activations. Finally, a decoder LSTM with 128 units predicts the Δx_t and Δy_t, so that $x_{t-1} + \Delta x_t = x_t$ and $y_{t-1} + \Delta y_t = y_t$ respectively. We downsample the data by a factor of 16 and consider 3 time steps, which amounts to a time horizon of 4.8 seconds. Since the vehicle moves much faster in x direction than in y direction, which means the errors in x direction are much higher, we weight the error in y direction with a factor of 20 during training (not during test time).

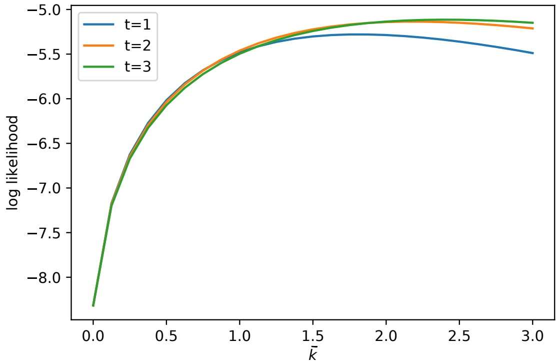


Figure 8: The average log likelihood in dependency of the $\bar{k}$ used for the transformation in (9).

As described in 5.2, for the problem of trajectory prediction in the context of path planning with a cost map, it is enough to only consider the marginalized distribution (See the supplementary materials for plots of the uncompensated marginalized PDF, reconstructed with a KDE as described in 5.2).

Since we are using the MoN loss, the learned PDF has to be compensated for the dilation effect. We apply algorithm 1 for this purpose. We set n_{sample} to 1000 and α_{split} to 0.7. As the set of possible compensation parameter $\{\bar{k}_{\text{search}}\}$, we use 25 values between 0.001 and 3. Our experiments showed $\bar{k}_{\text{opt,t=1}} = 1.88$, $\bar{k}_{\text{opt,t=2}} = 2.12$ and $\bar{k}_{\text{opt,t=3}} = 2.12$ which are close to the expected value of 2. A plot of the average log likelihood dependency from the chosen $\bar{k}$ is shown in Figure 8. The supplementary material shows the compensated reconstructed PDFs. Furthermore, if we use Algorithm 1 to find the $\bar{k}_{\text{opt}}$ that is optimal for all 3 time steps simultaneously, the algorithm yields $\bar{k}_{\text{opt}} = 2.00$ withing the search resolution, which is exactly the theoretically expected value. After obtaining the $\bar{k}_{\text{opt}}$, we applied the compensation on a left out test dataset and observed an improvement in the average log likelihood of ground truth trajecto-

PDF	$\mathcal{L}^1_{\mathbf{D}_{\text{Test}}}(P)$	$\mathcal{L}^2_{\mathbf{D}_{\text{Test}}}(P)$	$\mathcal{L}^3_{\mathbf{D}_{\text{Test}}}(P)$
Original PDF	-5.48	-5.46	-5.50
Compensated PDF	**-5.28**	**-5.13**	**-5.12**

Table 2: The results for the marginalized log likelihood as defined in (10) on the NGSIM dataset for the first 3 time steps (4.8 seconds). The compensated PDF consistently outperforms the uncompensated one.

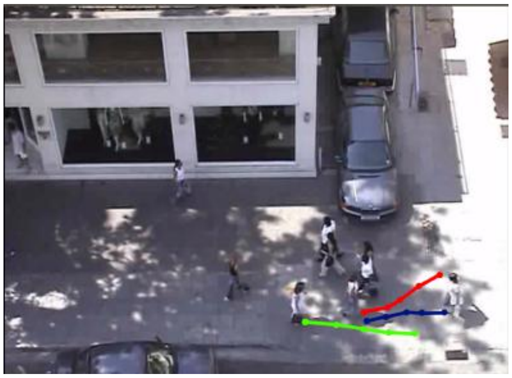


Figure 9: Illustration of the Zara dataset with ground truth trajectories.

ries. The results (see table 2) show that our compensated PDF clearly outperforms the uncompensated one.

7.2. Social–GAN

Next, we experiment on pedestrian trajectory data with Social–GAN [18] to show that even a state–of–the–art model can be improved by using our proposed compensation. Here, the authors use a combination of the MoN loss and the adversarial loss. They also designed a social pooling mechanism for efficient modelling of the social interactions of the pedestrians. We consider the Zara 1 dataset [1] and use the best performing model provided by the authors of [18] (https://github.com/agrimgupta92/sgan). The Zara dataset consists of 489 trajectories extracted from 13 minutes of videos on the corner of a sidewalk in a city (See Figure 9). In the supplementary materials, a few plots of the uncompensated PDFs are shown. We use algorithm 1 with the same settings as in 7.1. The resulting optimal compensation parameters are $\bar{k}_{\text{opt,t = 1}} = 2.50$ and $\bar{k}_{\text{opt,2}} = 1.63$. The variation of the average log likelihood with respect to the chosen $\bar{k}$ is shown in Figure 10. The supplementary materials show the compensated reconstructed PDFs. The final results of the marginalized log likelihoods are presented in Table 3 where we clearly see an advantage over the uncompensated version. For $t \geq 3$ and more difficult pedestrian datasets, this compensation however does not work. This is probably because too many samples fall in the low probability regions of the PDF that was learned by Social–GAN. Therefore, sharpening the learned distribution moves it even farther away from the ground truth distribution.

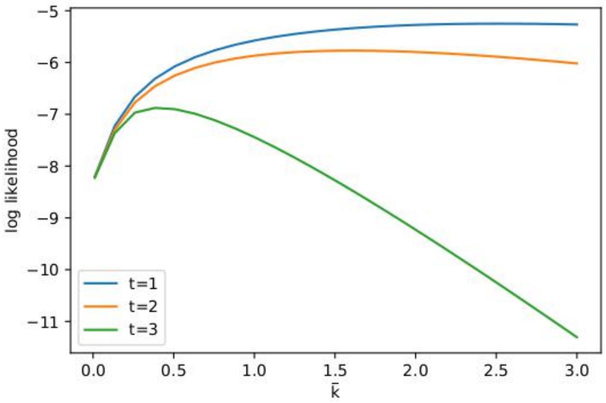


Figure 10: The average log likelihood in dependence of the $\bar{k}$ used for the transformation in (9).

PDF	$\mathcal{L}^1_{\mathbf{D}_{\text{Test}}}(P)$	$\mathcal{L}^2_{\mathbf{D}_{\text{Test}}}(P)$
Original PDF	-5.57	-5.87
Compensated PDF	**-5.24**	**-5.77**

Table 3: The results for the marginalized log likelihood as defined in (10) on the Zara1 dataset for the first 2 time steps. The compensated PDF outperforms the uncompensated one. For $t = 3$ the compensation parameter $\bar{k}_{\text{opt}}$ is however smaller than 1, which means that it has not learned the PDF well enough and our compensation does not make sense.

8. Conclusion

In this paper, we proved that the minimum of the MoN loss is not the ground truth PDF, but instead its square root. We validated this result using different experiments on toy and real world datasets. This means that the PDF that minimizes the MoN is a dilated version of the true one. Restricted to a certain class of PDFs, we also showed empirically that the MoN minimizing PDF becomes monotonically further stretched out with bigger N. This leads us to the conclusion that MoN should not be trusted as the only metric to compare models. For trajectory prediction, we instead advocate to also use the log likelihood of the marginalized PDF. Furthermore, we verify empirically that a learner trained with MoN loss can indeed converge to the square root of the PDF. Finally, we show that for certain low–dimensional applications, it is possible to compensate for the dilating effect of MoN and show that the ground truth dataset is more likely in the compensated distribution.

References

[1] ”crowds-by-example” data set (zara1 dataset). https://graphics.cs.ucy.ac.cy/research/downloads/crowd-data. Accessed: 2018-11-02.

[2] I-80 ngsim validation. http://www2.ece.ohio-state.edu/ coifman/documents/I80-NGSIM/. Accessed: 2018-11-02.

[3] Next generation simulation dataset. https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.html. Accessed: 2018-11-02.

[4] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[5] A. Bhattacharyya, B. Schiele, and M. Fritz. Accurate and diverse sampling of sequences based on a best of many sample objective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8485–8493, 2018.

[6] C. M. Bishop. Mixture density networks. Technical report, 1994.

[7] S.-T. Chiu. Bandwidth selection for kernel density estimation. *Ann. Statist.*, 19(4):1883–1905, 12 1991.

[8] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, pages 215–230, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[9] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1809.10732*, 2018.

[10] N. Deo and M. M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. *CoRR*, abs/1805.06771, 2018.

[11] P. K. et. al. Human Trajectory Prediction using Adversarial Loss.

[12] H. Fan, H. Su, and L. Guibas. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. *arXiv e-prints*, page arXiv:1612.00603, Dec 2016.

[13] P. Felsen, P. Lucey, and S. Ganguly. Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[14] D. Ferguson. Efficiently using cost maps for planning complex maneuvers. 2008.

[15] M. Gadelha, A. Rai, S. Maji, and R. Wang. Inferring 3D Shapes from Image Collections using Adversarial Networks. *arXiv e-prints*, page arXiv:1906.04910, Jun 2019.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[17] J. Guan, Y. Yuan, K. M. Kitani, and N. Rhinehart. Generative Hybrid Representations for Activity Forecasting with No-Regret Learning. *arXiv e-prints*, page arXiv:1904.06250, Apr 2019.

[18] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. *arXiv e-prints*, page arXiv:1803.10892, Mar. 2018.

[19] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. 51:4282–4286, May 1995.

[20] S. Hladky and V. Bulitko. An evaluation of models for predicting opponent positions in first-person shooter video games. In *2008 IEEE Symposium On Computational Intelligence and Games*, pages 39–46, Dec 2008.

[21] B. (https://math.stackexchange.com/users/297308/bgm). Expected minimum absolute difference to a given point correctly computed? Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/3000933 (version: 2018-11-16).

[22] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.*, 23(3):462–466, 09 1952.

[23] B. Kim, C. M. Kang, S. Lee, H. Chae, J. Kim, C. C. Chung, and J. W. Choi. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. *CoRR*, abs/1704.07049, 2017.

[24] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *arXiv e-prints*, page arXiv:1312.6114, Dec. 2013.

[25] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv e-prints*, page arXiv:1609.04802, Sep 2016.

[26] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chand raker. DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents. *arXiv e-prints*, page arXiv:1704.04394, Apr 2017.

[27] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

[28] C. Li and M. Wand. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. *arXiv e-prints*, page arXiv:1604.04382, Apr 2016.

[29] J. Liang, L. Jiang, J. C. Niebles, A. Hauptmann, and L. Fei-Fei. Peeking into the Future: Predicting Future Person Activities and Locations in Videos. *arXiv e-prints*, page arXiv:1902.03748, Feb 2019.

[30] C. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. Furukawa. Planenet: Piece-wise planar reconstruction from a single RGB image. *CoRR*, abs/1804.06278, 2018.

[31] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–942, June 2009.

[33] M. O Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi. Scalable end-to-end autonomous vehicle testing via rare-event simulation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9827–9838. Curran Associates, Inc., 2018.

[34] S. H. Park, B. Kim, C. Mook Kang, C. Choo Chung, and J. W. Choi. Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture. *arXiv e-prints*, page arXiv:1802.06338, Feb 2018.

[35] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268, Sep. 2009.

[36] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative Adversarial Text to Image Synthesis. *arXiv e-prints*, page arXiv:1605.05396, May 2016.

[37] N. Rhinehart, K. Kitani, and P. Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *European Conference on Computer Vision*. Springer, 2018.

[38] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine. PRECOG: PREdiction Conditioned On Goals in Visual Multi-Agent Settings. *arXiv e-prints*, page arXiv:1905.01296, May 2019.

[39] B. Ristic, B. La Scala, M. Morelande, and N. Gordon. Statistical analysis of motion patterns in ais data: Anomaly detection and motion prediction. In *2008 11th International Conference on Information Fusion*, pages 1–7, June 2008.

[40] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, S. H. Rezatofighi, and S. Savarese. SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints. *arXiv e-prints*, page arXiv:1806.01482, June 2018.

[41] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll. The Simpler the Better: Constant Velocity for Pedestrian Motion Prediction. *arXiv e-prints*, page arXiv:1903.07933, Mar 2019.

[42] D. W. Scott. Multivariate density estimation and visualization. 2012.

[43] M. K. C. Tay and C. Laugier. *Modelling Smooth Paths Using Gaussian Processes*, pages 381–390. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[44] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. *ACM Trans. Graph.*, 25(3):1160–1168, July 2006.

[45] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An Uncertain Future: Forecasting from Static Images using Variational Autoencoders. *arXiv e-prints*, page arXiv:1606.07873, Jun 2016.

[46] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

9. Supplementary Material

Proof of Theorem 1

Proof. For the sake of simplicity we only consider the one dimensional case.
First we bin the support of P_T in M equally sized bins $b_1, b_2, ..., b_M$ of width 2ϵ. Then we can write the MoN Loss as

$$L_N(P_T, P) \approx \sum_{i=1}^{M} P_T(b_i) \int_{b_i} \mathrm{EMoN}_{P,b_i}(x^*)\, \mathrm{d}x^* \tag{18}$$

with

$$\begin{aligned}\mathrm{EMoN}_{P,b_i}(x^*) = \int_{b_i} &\min\left(|x^* - x_1|, |x^* - x_2|, ..., |x^* - x_N|\right) \\ &P(x_1)P(x_2)\dots P(x_N)\,\mathrm{d}x_1\,\mathrm{d}x_2\dots\mathrm{d}x_N\end{aligned} \tag{19}$$

In expectation there are $NP(b_i)$ samples in bin b_i. Assume, that $z_i = NP(b_i)$ is an integer. Denote the sample that fall in b_i as $H_i := \{x_1^i, x_2^i, ..., x_n^i\}$. Then, for the calculation of $\mathrm{EMoN}_{P,b_i}(x^*)$ we can ignore all samples that are not in H_i. Then we can write

$$\int_{b_i} \mathrm{EMoN}_{P,b_i}(x^*)\, \mathrm{d}x^* = \tag{20}$$

$$\begin{aligned}&\int_{b_i}\int_{b_i} \min\left(|x^* - x_1^i|, |x^* - x_2^i|, ..., |x^* - x_{z_i}^i|\right) \\ &P(x_1^i)P(x_2^i)...P(x_{z_i}^i)\,\mathrm{d}x_1^i\,\mathrm{d}x_2^i, ..., \mathrm{d}x_{z_i}^i\,\mathrm{d}x^*\end{aligned} \tag{21}$$

For small ϵ and because we only consider differentiable functions, we can approximate $P(x_1^i)P(x_2^i)...P(x_{z_i}^i)$ withing the bin as uniform. Next we have to calculate the inner integral in (21). For that, center the bins around the origin (which is possible because we have uniform probability distributions) and consider the survival function of individual random samples $|x_i - x^*|$ [21]:

$$\begin{aligned}S(x) &= \Pr\{|x_i - x^*| > x\} && (22)\\ &= \Pr\{x_i > x^* + x\} + \Pr\{x_i < x^* - x\} && (23)\\ &= \begin{cases} 1 & \text{if } x \leq 0 \\ 1 - \frac{x}{\epsilon} & \text{if } 0 < x \leq \epsilon - |x^*| \\ \frac{\epsilon + |x^*| - x}{2\epsilon} & \text{if } \epsilon - |x^*| < x < \epsilon + |x^*| \\ 0 & \text{if } x \geq \epsilon + |x^*| \end{cases} && (24)\end{aligned}$$

Then the survival function $S_{MoN,z_i}(x)$ of $\min|x_i - x^*|$ is the probability, that all z_i sample will independently be bigger than x. Thus the survival function is $S_{MoN,z_i}(x) = S(x)^{z_i}$ and therefore

$$\begin{aligned}&E[\min|x_i - x^*|] = && (25)\\ &= \int_0^{+\infty} S(x)^{z_i} dx && (26)\\ &= \int_0^{\epsilon - |x^*|} \left(1 - \frac{x}{\epsilon}\right)^{z_i} dx + \int_{\epsilon - |x^*|}^{\epsilon + |x^*|} \left(\frac{\epsilon + |x^*| - x}{2\epsilon}\right)^{z_i} dx && (27)\\ &= \frac{-\epsilon}{z_i + 1}\left(1 - \frac{x}{\epsilon}\right)^{z_i + 1}\Bigg|_0^{\epsilon - |x^*|} + \frac{-2\epsilon}{z_i + 1}\left(\frac{\epsilon + |x^*| - x}{2\epsilon}\right)^{z_i + 1}\Bigg|_{\epsilon - |x^*|}^{\epsilon + |x^*|} && (28)\\ &= \frac{\epsilon}{z_i + 1}\left(1 - \frac{|x^*|^{z_i+1}}{\epsilon^{z_i+1}} + 2\frac{|x^*|^{z_i+1}}{\epsilon^{z_i+1}}\right) && (29)\\ &= \frac{\epsilon}{z_i + 1}\left(1 + \frac{|x^*|^{z_i+1}}{\epsilon^{z_i+1}}\right) && (30)\end{aligned}$$

Substituting (30) in (18) therefore yields

$$\begin{aligned}L_N(P_T, P) &\approx \sum_{i=1}^{M} P_T(b_i) \int_{-\epsilon}^{\epsilon} \frac{\epsilon\left(1 + \frac{|x^*|^{z_i+1}}{\epsilon^{z_i+1}}\right)}{z_i + 1}\,\mathrm{d}x^* && (31)\\ &= \epsilon \sum_{i=1}^{M} P_T(b_i) \int_{-\epsilon}^{\epsilon} \frac{\left(1 + \frac{|x^*|^{z_i+1}}{\epsilon^{z_i+1}}\right)}{z_i + 1}\,\mathrm{d}x^* && (32)\\ &= \epsilon \sum_{i=1}^{M} P_T(b_i) \frac{2\epsilon + 2\int_0^{\epsilon} \frac{x^{*\,z_i+1}}{\epsilon^{z_i+1}}\,\mathrm{d}x^*}{z_i + 1} && (33)\\ &= \underbrace{2\epsilon^2}_{:=a} \sum_{i=1}^{M} P_T(b_i) \frac{1 + \frac{1}{NP(b_i)+2}}{NP(b_i) + 1} && (34)\\ &\approx a \sum_{i=1}^{M} P_T(b_i) \frac{1 + \frac{1}{NP(b_i)}}{NP(b_i)} && (35)\\ &= a \sum_{i=1}^{M} P_T(b_i) \left(\frac{1}{NP(b_i)} + \frac{1}{(NP(b_i))^2}\right) && (36)\\ &\approx a \sum_{i=1}^{M} P_T(b_i) \frac{1}{NP(b_i)} && (37)\end{aligned}$$

where we used in line (35) and (37) that $N \to \infty$. We can find the minimum of $L_N(P_T, P)$ under the constraint that $\sum_{i=1}^{M} P(b_i) = 1$ by using a Lagrange multiplier. The objective is therefore:

$$f(P(b_i), \lambda) := a \sum_{i=1}^{M} \frac{P_T(b_i)}{NP(b_i)} - \lambda\left(1 - \sum_{i=1}^{M} P(b_i)\right) \tag{38}$$

Optimization for $P(b_i)$ yields:

$$\nabla_{P(b_i)} f(P(b_i), \lambda) = \lambda - a \frac{P_T(b_i)}{NP(b_i)^2} \overset{!}{=} 0 \tag{39}$$

$$\Leftrightarrow P(b_i) = \sqrt{\frac{aP_T(b_i)}{\lambda N}} \tag{40}$$

$$\nabla_\lambda f(P(b_i), \lambda) = 1 - \sum_{i=1}^{M} P(b_i) \overset{!}{=} 0 \tag{41}$$

$$\Leftrightarrow \sum_{i=1}^{M} P(b_i) = 1 \tag{42}$$

where we can omit the ± in (40), because probabilites can not be negative. Putting (40) and (42) together results in:

$$\Rightarrow \sum_{i=1}^{M} P_T(b_i) \left(\sqrt{\frac{aP_T(b_i)}{\lambda N}} \right) = 1 \tag{43}$$

$$\Rightarrow \lambda = \left(\frac{\sum_{i=1}^{M} \sqrt{aP_T(b_i)}}{\sqrt{N}} \right)^2 \tag{44}$$

$$\Rightarrow P(b_i) = \frac{\sqrt{P_T(b_i)}}{\sum_{i=1}^{M} \sqrt{P_T(b_i)}} \tag{45}$$

Note, that this result is true for any distance metric in any dimensions, as long as the expected minimum distance of N sample to a target sample goes locally with $\mathcal{O}(\frac{1}{N})$. □

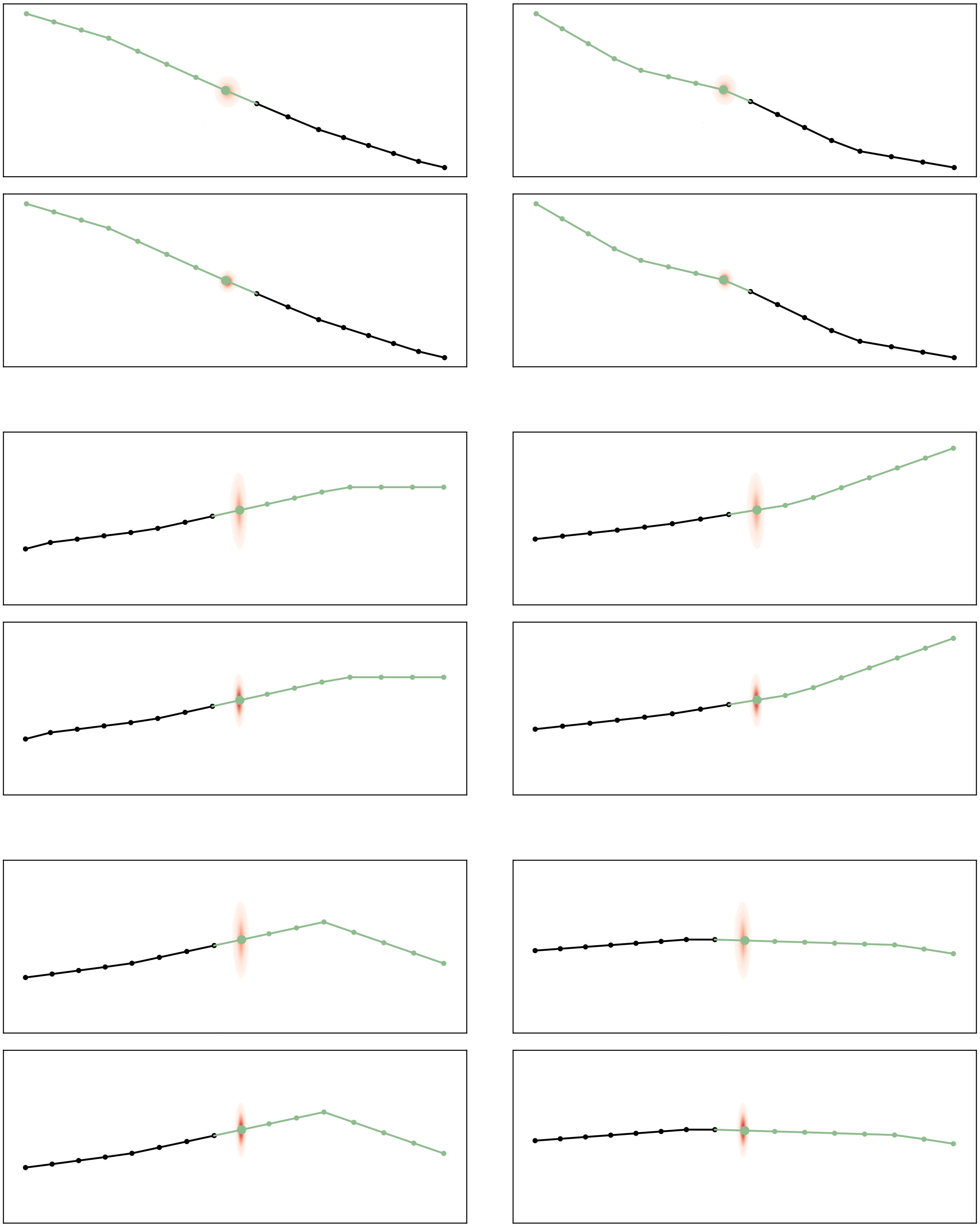


Figure 11: Marginalized probabilities of Social–GAN on the Zara dataset for $t = 1$. Upper panel is the uncompensated PDF and the lower panel is the compensated one. Black is the observed trajectory and green the ground truth future.

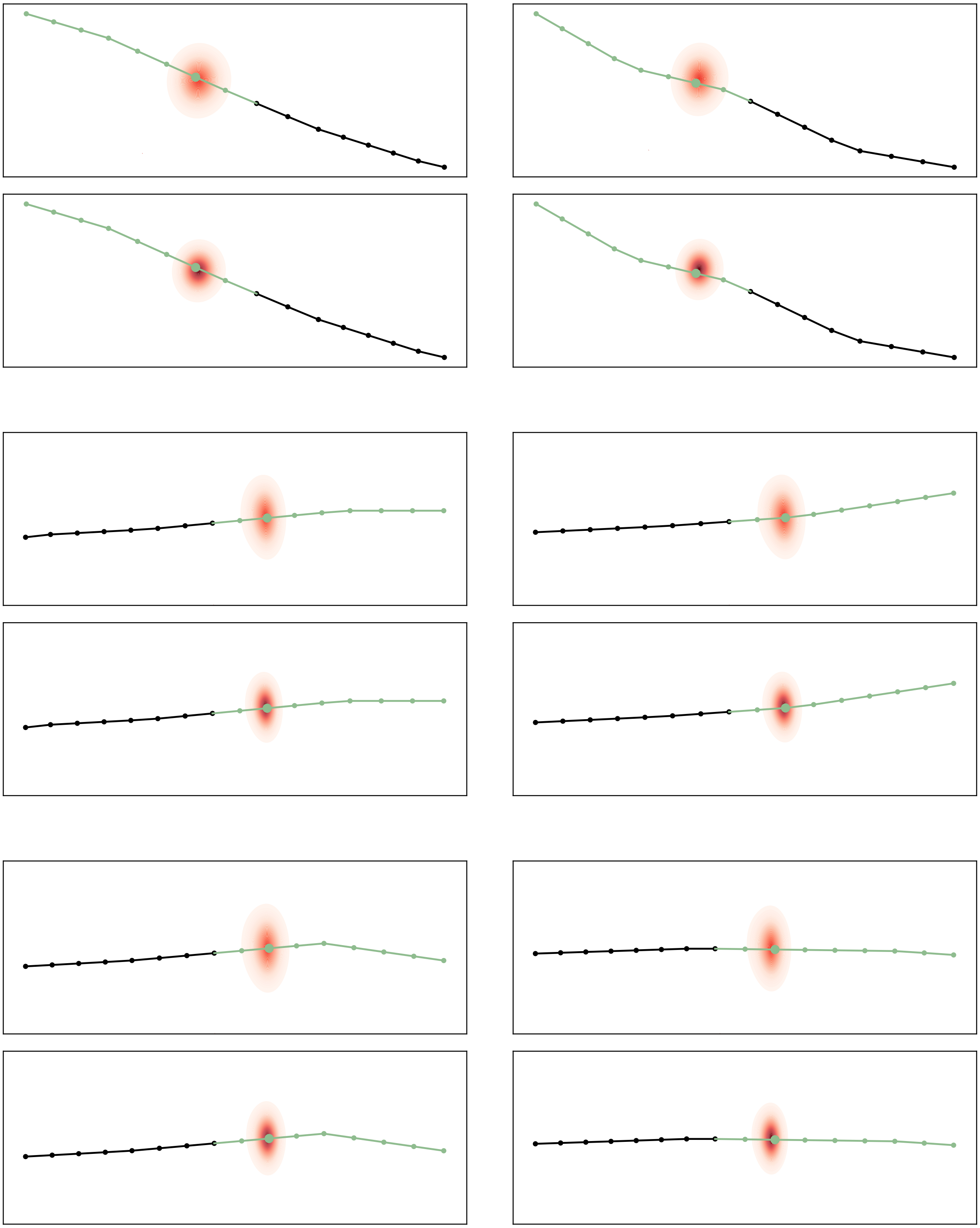


Figure 12: Marginalized probabilities of Social–GAN on the Zara dataset for $t = 2$. Upper panel is the uncompensated PDF and the lower panel is the compensated one. Black is the observed trajectory and green the ground truth future.

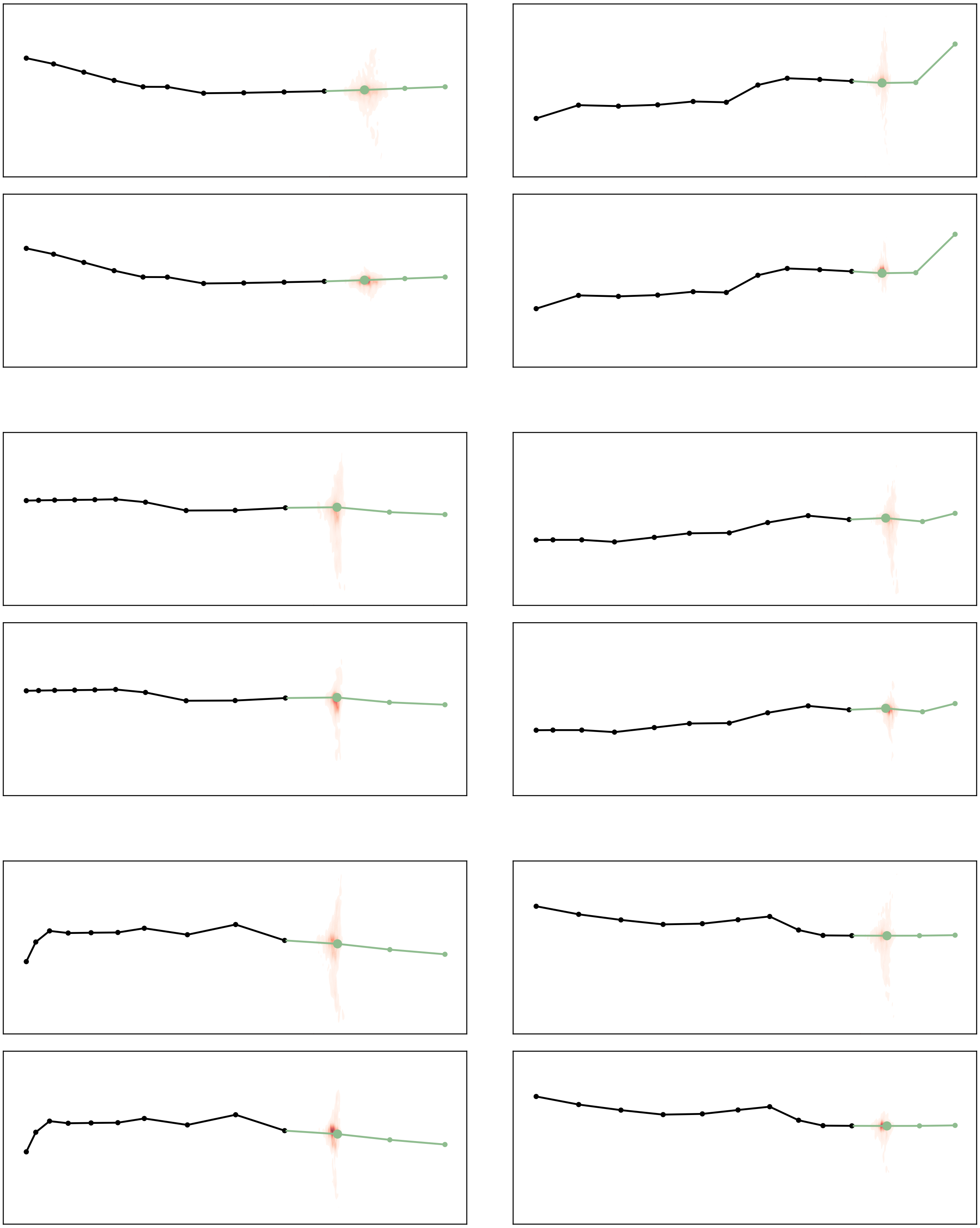


Figure 13: Marginalized probabilities of our own model on the NGSIM dataset for $t = 1$. Upper panel is the uncompensated PDF and the lower panel is the compensated one. Black is the observed trajectory and green the ground truth future.

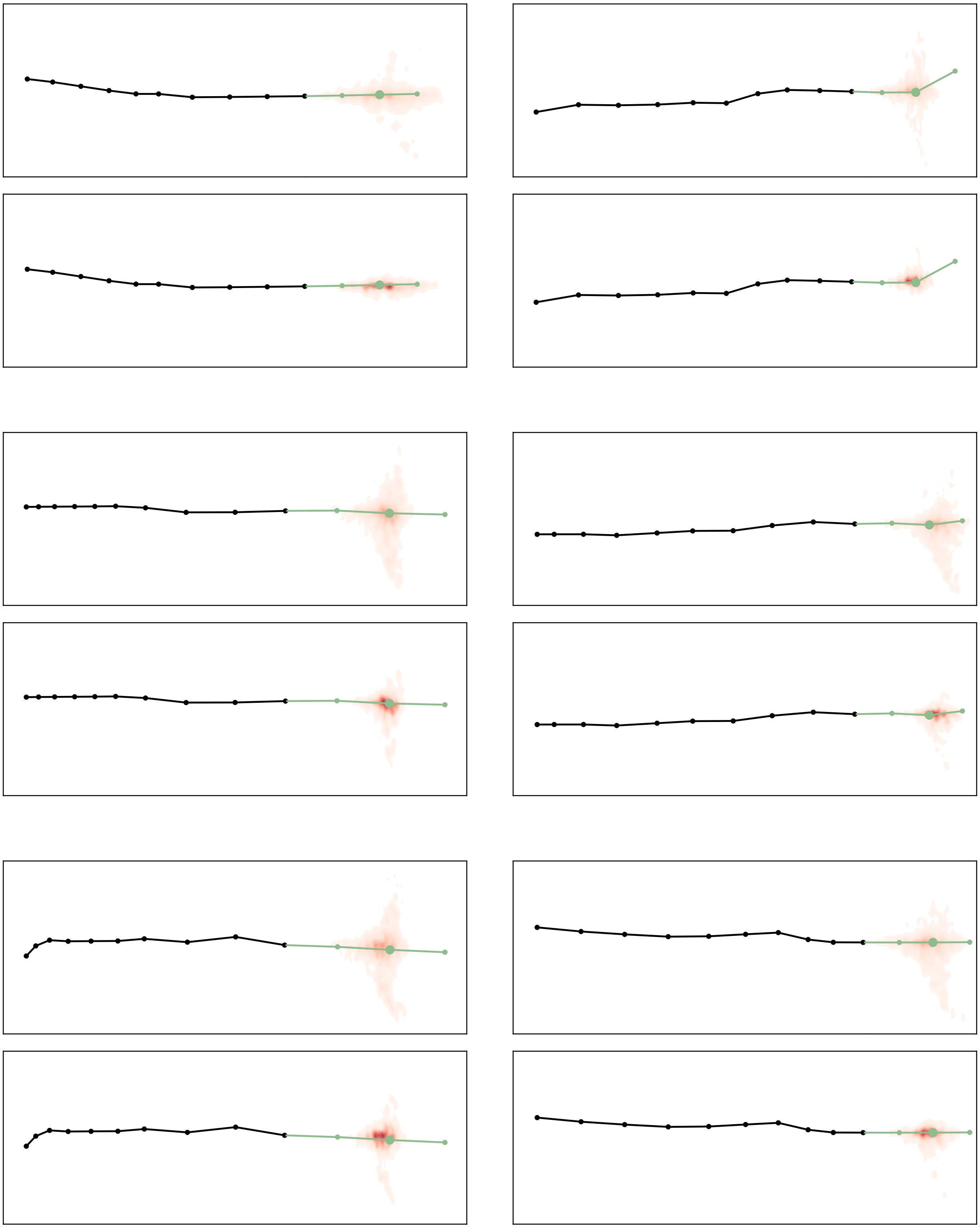


Figure 14: Marginalized probabilities of our own model on the NGSIM dataset for $t = 2$. Upper panel is the uncompensated PDF and the lower panel is the compensated one. Black is the observed trajectory and green the ground truth future.

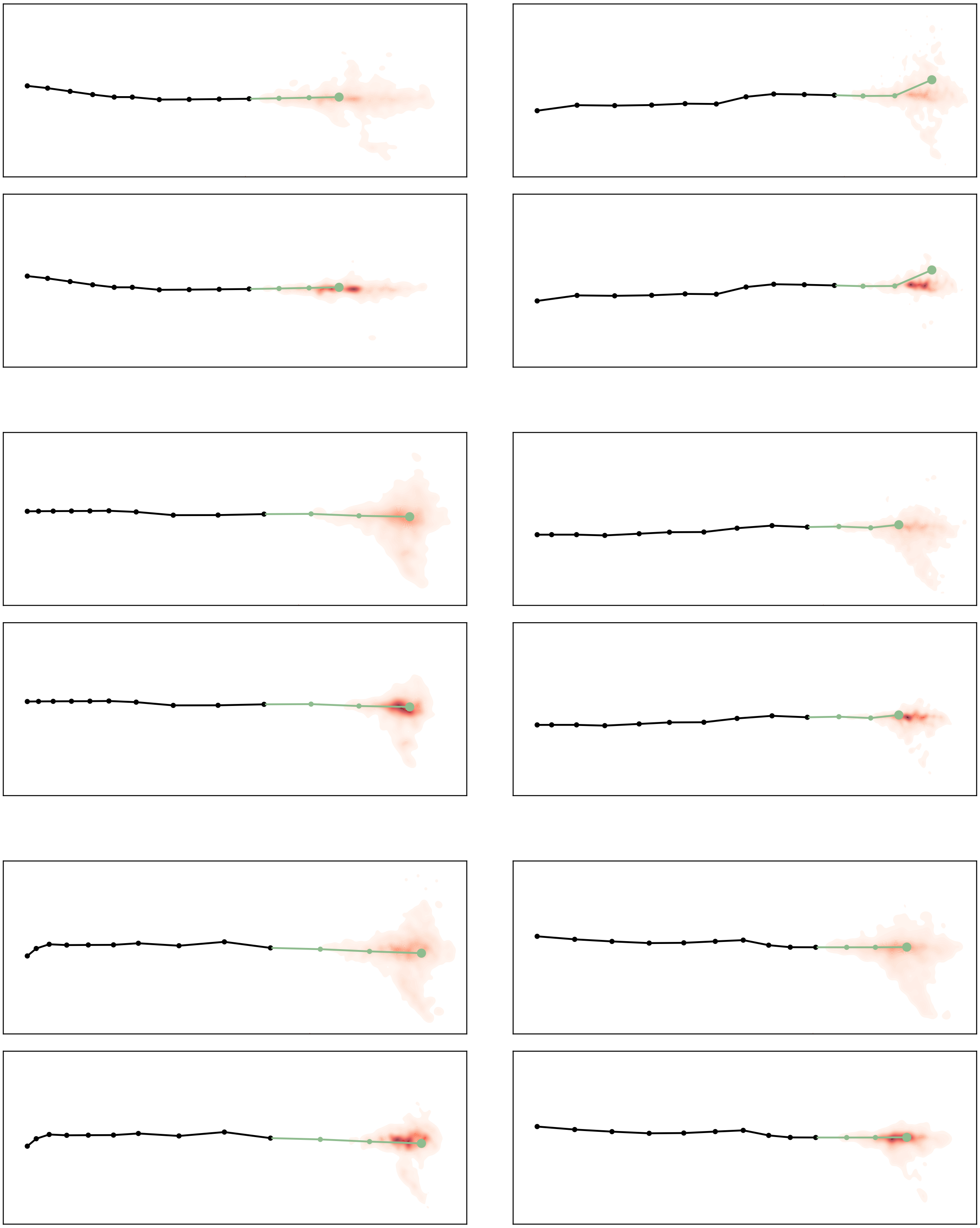


Figure 15: Marginalized probabilities of our own model on the NGSIM dataset for $t = 3$. Upper panel is the uncompensated PDF and the lower panel is the compensated one. Black is the observed trajectory and green the ground truth future.