# COMMON SENSE INITIALIZATION OF MIXTURE DENSITY NETWORKS FOR MOTION PLANNING WITH OVERESTIMATED NUMBER OF COMPONENTS

**Thomas Kreutz, Max Mühlhäuser, and Alejandro Sanchez Guinea**
Technical University of Darmstadt, Telecooperation Lab
{<kreutz, sanchez>@tk, <max>@informatik}.tu-darmstadt.de

## ABSTRACT

Mixture density networks (MDNs) are a natural choice to model multi-modal predictions for trajectory prediction or motion planning. However, MDNs are often difficult to train due to mode collapse and a need for careful initialization, which becomes even more problematic when the number of mixture components are strongly overestimated. To address this issue in motion planning problems, we propose a pre-training scheme for MDNs called common sense initialization (CSI). Pre-training with CSI allows variety-encouraging optimization such as Winner-Takes-All (WTA) to exploit the initialized weights during training, so that the MDN can converge when the number of components are overestimated. This paper presents empirical evidence for the effectiveness of CSI when applied to motion planning of pedestrian agents in urban environments.

## 1 INTRODUCTION

Mixture density networks (MDNs) (Bishop, 1994) are often difficult to train due to mode collapse or a need for careful initialization (Makansi et al., 2019; Zhou et al., 2020). These problems may become even more severe when the true number of modes $K^*$ is unknown and the mixture components are strongly overestimated, i.e., when $K^* \ll K$. In the context of pedestrian motion planning, we propose a simple but effective pre-training scheme as common sense prior for MDNs. More specifically, we propose Common Sense Initialization (CSI), which pre-trains an MDN to output a probability distribution over the next common sense states at a realistic distance, as depicted in Figure 1.

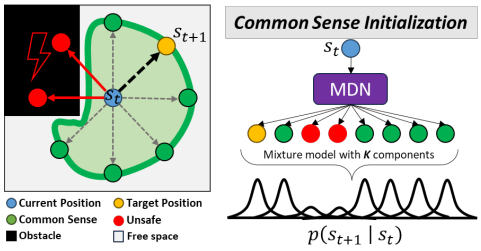


Figure 1: Given a current position $s_t$, we pre-train an MDN to output a "common sense" probability distribution $P(s_{t+1}|s_t)$, where common sense states are equally likely to be explored, while unsafe states are highly unlikely.

CSI is motivated by recent advances in pre-training methods for reinforcement learning such as Parrot (Singh et al., 2020), which allows a policy to be adapted quickly to new tasks either from a learned behavioral prior or from interpretable priors for trajectory prediction such as interpretable trajectory trees (Shi et al., 2022), where a neural network refines rule-based "common sense" future paths for multi-modal predictions. Complementing these works, we specifically address training difficulties when using MDNs for multi-modal motion planning tasks, where the true number of modes is unknown and interpretable common sense states exist. An advantage of our proposed pre-training scheme is that in a behavior cloning setup fine-tuning an MDN with variety encouraging losses such as Winner-Takes-All (WTA) (Guzman-Rivera et al., 2012; Cheng et al., 2023) or a variety loss (Gupta et al., 2018; Zhou et al., 2022) can exploit the initialized model to converge despite using a large number of components and reduce the risk of mode collapse.

We show the effectiveness of combining CSI with WTA for pedestrian motion planning, where a mixture density network with a large overestimation of mixture components is trained with behavior cloning in the grand central (GC) station dataset (Zhou et al., 2012).

## 2 APPROACH

Common sense is a skill that helps humans navigate safely towards a goal in arbitrary environments. Figure 1 shows how to model common sense navigation using an MDN. Common sense states do not necessarily satisfy desired behavior in a navigation scenario, such as moving towards a goal, but they encompass states that exclude dangerous situations. Given a current state $s_t$, an observation $o_t$, and a goal state $s_g$ (Figure 1 simplified to only include $s_t$), common sense induces a probability distribution $P(s_{t+1}|s_t, s_g, o_t)$ over possible next states $s_{t+1}$ that are safe. Let $w_k, \mu_k, \sigma_k$ , $1 \leq k \leq K$ be $K \in \mathbb{N}$ mixture weights and components, $x = (s_t, s_g, o_t)$ the current state, goal, and observation. An MDN $f$ approximates a probability distribution for the next state $s_{t+1}$ with:

$$P(s_{t+1}|x) = \sum_k w_k(x) \cdot \mathcal{N}(s_{t+1}|\mu_k(x), \sigma_k(x)) \tag{1}$$

For navigation scenarios, we propose common sense initialization (CSI), which pre-trains $f$, so that the mixture weights and components follow common sense. More specifically, with the interpretation that an agent can move in $K$ possible directions (modes) and the mixture weight constraint $\sum_k w_k = 1$, all modes from the current state $s_t$ are equally likely except for unsafe modes (for instance, modes that would collide with the environment). For each $\mu_k$, the target is one of $k$ points on a common sense circle around the current position $s_t$ with a radius corresponding to the ground truth distance to $s_{t+1}$, and each $\sigma_k$ to a small number $\epsilon$. Given a function $safe(\mu_k, o_t)$ that returns 1 if the action $\mu_k$ is safe and 0 otherwise, we compute the mixture weight targets $w_k$ as follows:

$$W_{init} = \sum_k^K safe(\mu_k, o_t) \tag{2}$$

$$w_k = \begin{cases} \frac{1}{W_{init}} & \text{if } safe(\mu_k, o_t) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

We pre-train $f$ with mean squared error on the common sense targets $\mu_k, \sigma_k, w_k$, initializing it with common sense (CSI). Next, $f$ is fine-tuned using WTA, which optimizes for the component nearest to the target state that satisfies $argmin_k||\mu_k - s_{t+1}||$. Intuitively, this approach leverages the pre-trained close-to-target mixture components from CSI, allowing WTA to focus on only updating the weights $w_k$, pushing the true target states to be more probable in $f$'s induced distribution.

## 3 EXPERIMENTS AND CONCLUSION

We evaluate our approach on the grand central station dataset (GC) (Zhou et al., 2012), where we train an MDN for motion planning with behavior cloning using WTA (details in Appendix B) on the first 90% of all trajectories, and test on the remaining 10%. We use the well-known average and final displacement error (ADE/FDE) as evaluation metrics. Our baseline is an MDN with random initialization and $K^* = 3$ components.

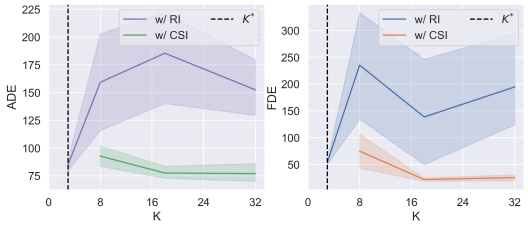


Figure 2: Comparing CSI when using different number of components against random initialization.

Figure 2 compares CSI against random initialization (RI), while using a growing number of components $K^{OE} \in [8, 18, 32]$. We trained each network configuration five times to give an estimate of the overall performance. Our results show that using CSI consistently improves performance that gets close to the baseline $K^* = 3$, despite using an unnecessarily large overestimation of the number of components. Our results indicate that WTA can exploit initialized CSI modes to consistently improve ADE, while giving less significant improvements to FDE over all configurations.

**Conclusion** We show that CSI combined with variety-encouraging losses such as WTA improves the performance of MDNs when the number of mixture components is overestimated in a behavior cloning setup for pedestrian motion planning. We argue that our results have practical relevance when the number of true modes $K^*$ of a motion planning problem cannot be determined and may be strongly overestimated, i.e., when $K^* \ll K$.

### URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of the ICLR 2024 Tiny Papers Track.

### ACKNOWLEDGEMENTS

This work has been funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY centre.

## REFERENCES

Christopher M Bishop. Mixture density networks. 1994.

Hao Cheng, Mengmeng Liu, Lin Chen, Hellward Broszio, Monika Sester, and Michael Ying Yang. Gatraj: A graph-and attention-based multi-agent trajectory prediction model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 205:163–175, 2023.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2255–2264, 2018.

Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. *Advances in neural information processing systems*, 25, 2012.

Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7144–7153, 2019.

Ahmed H Qureshi, Anthony Simeonov, Mayur J Bency, and Michael C Yip. Motion planning networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2118–2124. IEEE, 2019.

Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1349–1358, 2019.

Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Nanning Zheng, and Gang Hua. Social interpretable tree for pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2235–2243, 2022.

Avi Singh, Huihan Liu, Gaoyue Zhou, Albert Yu, Nicholas Rhinehart, and Sergey Levine. Parrot: Data-driven behavioral priors for reinforcement learning. *arXiv preprint arXiv:2011.10024*, 2020.

Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2871–2878. IEEE, 2012.

You Zhou, Jianfeng Gao, and Tamim Asfour. Movement primitive learning and generalization: Using mixture density networks. *IEEE Robotics & Automation Magazine*, 27(2):22–32, 2020.

Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8823–8833, 2022.

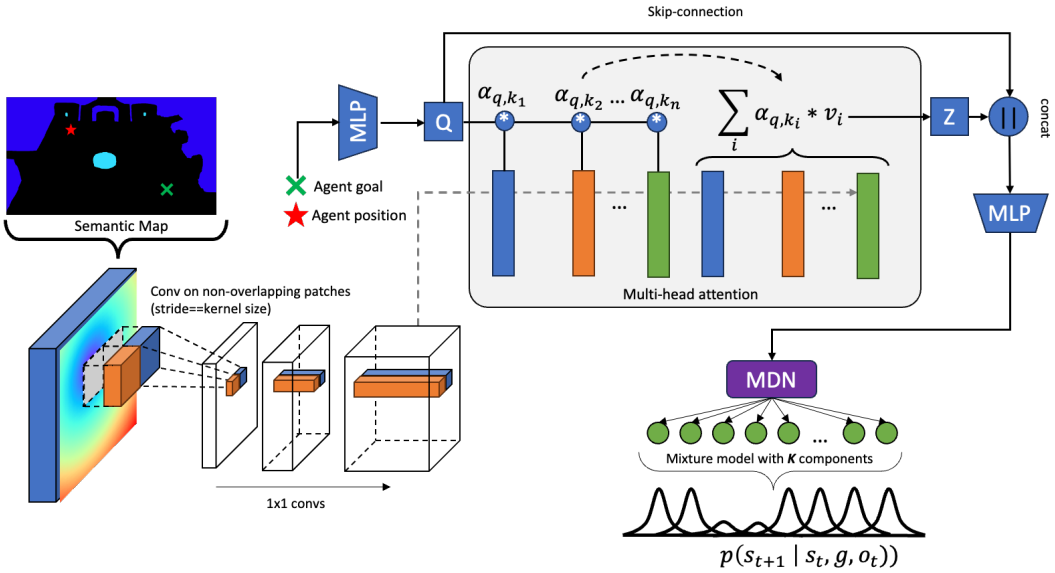


Figure 3: Our motion planner takes a semantic map, the current position, and the target position as an input. The map is processed with a NiN-like architecture, where non-overlapping patches are extracted first, which are afterward processed with a stack of 1x1 convolutions. Each patch receives a positional embedding that corresponds to the distance and angle w.r.t. to the agent in world coordinates. Using multi-head attention, the model can attend to the environment and the outputs the next position in dependency to the current position, goal, and environment.

## A Problem Statement

Given an agent at a starting position $s_0 = (x, y)$ and a goal position $g = (x_g, y_g)$, we train a variant of motion planning networks Qureshi et al. (2019) ,$f(\cdot)$, to compute a sequence of positions that lead the agent from $s_0$ to $g$. At each timestep, the motion planner takes the current position $s_t$, the goal position $g$, and a map observation $o_t$ as an input and predicts the next position $s_{t+1}$:

$$s_{t+1} = f(s_t, g, o_t) \tag{4}$$

A motion plan is produced by recursively quering the motion planning network with the predicted position:

$$s_* = f^*(s_t, g, o_t) \tag{5}$$

where $f^*$ denotes a recursive application that stops after $\star$ steps. To not run for infinity, we specify a maximum number of possible steps. When the agent reaches the goal before this number of steps, i.e., if the distance of the agent to the goal is smaller than a threshold ($||s_{t+1} - g|| < \epsilon$), the planner stops. If a ground truth plan for the given configuration is available, we set the maximum number of motion planning steps to the same number of steps as the ground truth plan with a small margin of extra steps. Otherwise, we set it to the largest number of steps recorded in the underlying dataset.

## B Model Details

Our architecture is shown in Figure 3. The agent's current position and goal is encoded first with an MLP to an embedding $Q$. To observe the environment, we implement an attention mechanism related to the physical attention proposed by previous works Sadeghian et al. (2019); Kosaraju et al. (2019). We implement a multi-head attention mechanism between agent and map patches with relative positional encodings based on polar coordinates. More specifically, given a semantic image of shape $M \in \mathbb{R}^{C \times H \times W}$, we encode $n = (H//16) * (W//16)$ non-overlapping patches with a

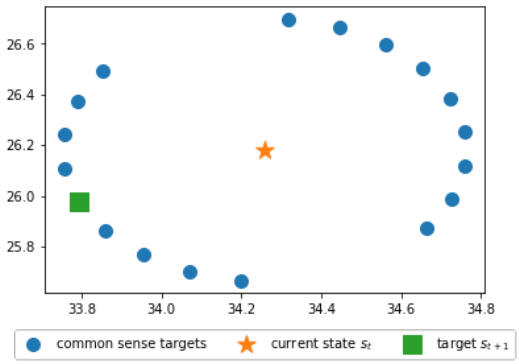


Figure 4: Common sense targets based on equidistant points in a specified fov towards front and back

Network-in-Network (NiN) like architecture Lin et al. (2013). We obtain $n$ patches with embedding dimension $e$, i.e., $p \in \mathbb{R}^{n \times e}$. Afterward, we compute relative positional embeddings w.r.t. the agents current position $s_t$ in world coordinates as follows:

1. Transform the center of the pixel coordinates of each patch into world coordinates $p_{world} \in \mathbb{R}^2$ using the homography matrix for the respective scene.
2. Transform the patch world coordinates to the agent's frame of reference $p_{rel} = p_{world} - s_t$
3. Transform $p_{rel}$ to polar coordinates to obtain distance $d$ and angle $\varphi$ from each patch to the agent's position, which we use as position encodings.

Afterward, the position encodings are added to the embeddings. Finally, cross attention is applied between the embeddings and the agent embedding $Q$ to obtain a map observation embedding $Z$. $Q$ is then concatenated with $Z$ and passed to a final MLP. For multi-modal predictions, a mixture density network is added after the final MLP layer which we use at the inference step to sample the final prediction. Recursively querying this model with the current position, goal, and map results in a motion plan.

**Training Details.** For multi-modal predictions, we use a mixture density network (Bishop, 1994) as a decoder after the last MLP layer in our architecture shown in Figure B that has $K$ mixture components and train the whole model with behavior cloning and adopt the Winner-Takes-All loss (Guzman-Rivera et al., 2012; Cheng et al., 2023) with a soft-displacement error as target probabilities. The model takes the map observation $o_t$, the current position $s_t$, and the goal $g$ as an input, which resembles the architecture of motion planning networks (Qureshi et al., 2019).However, our model differs from (Qureshi et al., 2019) as follows. First, we use a different map observation model (see Figure 3) to attend to the environment. Second, we use an MDN as a decoder to account for multi-modal predictions and instead of mean squared error, we train our model with WTA Guzman-Rivera et al. (2012) that encourages multi-modal predictions.

When using CSI, we pre-train the motion planning network for 250 epochs with a learning rate of 1e-4 and a batch size of 2048. After CSI, we finetune the network for 200 epochs with a learning rate of 1e-5 and an aggressive exponential decay scheduler. When not using CSI, we use the same training parameters of a 1e-5 learning rate, 200 epochs and an aggressive exponential decay scheduler to compare the effect of using CSI against random initialization.

As parameters for our model, we use an embedding dimension of 512 for all layers, downsampling factor of k=16 for the map, and a patch size of 16 for the observation model. The map is padded to be divisible by 16. Furthermore, we annotated the map with three semantic classes: Free space, obstacle, wall.

## C COMMON SENSE INITIALIZATION

The target $s_{t+1}$ in our motion planning problem is an $x, y$ coordinate position. Given the true target $s_{t+1}$ the CSI targets for $\mu_k$ are calculated as follows and visualized in Figure 4. We first compute the distance $d$ and angle $\varphi$ from $s_t = (x_t, y_t)$ to $s_{t+1} = (x_{t+1}, y_{t+1})$ using a polar coordinate transformation:

$$d = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \tag{6}$$

$$\varphi = atan2(y_{t+1} - y_t, x_{t+1} - x_t) \tag{7}$$

We compute $\frac{k}{2}$ equidistant common sense targets for the front and back in a field of view of $\frac{\pi}{3}$, i..e, $60^\circ$ because we assume that a pedestrian does not move sideways.