A Inconsistent Practice of the Evaluation for Trajectory Prediction

Previous research in the area of trajectory forecasting, i.e. trajectory prediction, has focused on multiple datasets for quantitative evaluation. However, we notice that the evaluation settings of previous works are inconsistent thus making noisy and unfair comparisons in the benchmarks we usually refer to.

A.1 Inconsistent evaluation convention on ETH/UCY

The ETH/UCY [20, 33] dataset, comprising five subsets of data, serves as the primary benchmark for human trajectory prediction. It is not proposed as a single dataset in the first place but merges many different resources. Therefore, there are no official guidelines for data splitting and model evaluation metrics. Consequently, previous studies employ different evaluation conventions and falsely confuse results from different conventions together for benchmarking.

The benchmark widely adopted by the research community was initially proposed by Social-GAN [11]. It adheres to the following principles:

1. Utilizing data with a sampling rate of 10 FPS in all subsets.
2. Employing a leave-one-out approach for splitting, where the model is trained on four subsets and tested on the remaining one subset.
3. Dividing the raw trajectory samples into specific train, eval, and test sets.

Subsequent works like Trajectron++ [39], AgentFormer [52], and EqMotion [50] have widely embraced this benchmarking and evaluation setting.

However, not all studies adhere to this benchmark. To identify examples, we conducted a review of recent open-access conference papers, which reveals the following divergence.

1. Sampling Rate on ETH dataset. There are two widely used sampling rates for the evaluation of the ETH dataset. While Social-GAN [11] utilized the data with a sampling rate of 10FPS (SR=10), other works, such as SR-LSTM [53], V2Net [46], SocialCircle [47], STAR [51], PCCSNet [42], Stimulus Verification [41], and MG-GAN [5], used the version with a sampling rate of 6FPS (SR=6). The SR=6 version contains more data, with a total of 8,908 frames, whereas the SR=10 version consists of only 5,492 frames. Based on our experience, the same model tends to yield higher evaluation scores (ADE/FDE) on the SR=6 version compared to the SR=10 version.

2. Data Splitting. Social-GAN follows a specific scheme and ratio for splitting the data into train/val/test sets, while some works have adopted different conventions. For instance, Sophie [38] selected fewer training scenes, MG-GAN [5] used the complete training scene data for training while separating a portion of the test set for evaluation. Also, works that choose the SR=6 version data in the ETH dataset adopt different splitting conventions, because they do not share the same raw trajectory data samples with Social-GAN, which uses the SR=10 version data.

3. Inconsistent Data Pre-processing. Some other studies, such as Y-Net [25], Introvert [40], and Next [23], provide processed data without the raw data and the processing scripts. The provided processed data for val/test sets does not align with the Social-GAN benchmark.

A.2 Inconsistent evaluation convention on SDD

Compared to ETH/UCY, SDD [36] is a more recent dataset consisting of 20 scenes captured in bird's eye view. SDD contains various moving agents such as pedestrians, bicycles, and cars.

Most works follow the setting of TrajNet [37] which comes from a public challenge. However, some works adopt different evaluation way compared to TrajNet. SimAug [22] reprocesses the raw videos and gets a set of data files different from TrajNet's. Besides, it uses a different data splitting convention. Subsequent works such as V2Net [46] and SocialCircle [47] follow the same setting that SimAug starts. DAG-Net [30] shared the same data file with TrajNet, but used a different data splitting approach. Social-Implicit [29] followed its setting.

Therefore, there are multiple different evaluation protocol conventions on the ETH/UCY and SDD datasets. Because the data splitting for training/test and evaluation details are different, putting the evaluation numbers from them together provides misleading quantitative observations, which the community has been using for a while. We point it out and make a complete summary of these misalignments, wishing future research aware of this to avoid potential continuity of the mis-practice and provide a fair comparison.

Table 5: minADE/minFDE of worst-N predictions on UNIV dataset. By adding augmented data along with their corresponding cluster centers, our method significantly improves the performance on the corner cases.

N	FlowChain		Augment-MGF	
	ADE	**FDE**	**ADE**	**FDE**
10	3.13	6.54	0.75	1.30
50	2.40	4.90	0.90	1.57
100	2.06	4.29	1.07	1.86

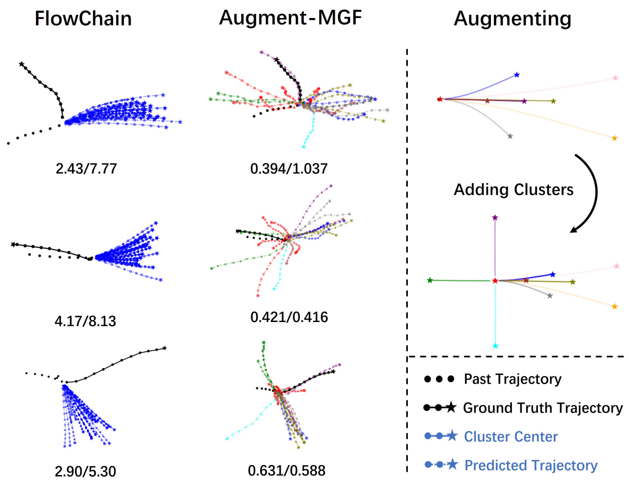

Figure 6: By adding augmented data along with their corresponding clusters into the construction of the mixed Gaussian prior, we could manipulate the generation patterns as we desire. For example, we could inject some under-represented trajectory patterns. Then the model can generate corner cases that existing models fail to generate in a reasonable probability. We selected three examples from the UNIV dataset, namely sharp left/right turns and U-turns. The left column shows the predictions from FlowChain, while the middle column shows the predictions of MGF with augmented priors

A.3 Summary and Our Practice

When comparing baseline results, many previous studies fail to meticulously verify whether they adhere to the same convention, thereby leading to unfair comparisons. To compare the performance of various models fairly, we follow the practice that the community adheres to the most: Social-GAN's convention for ETH/UCY and TrajNet's convention for SDD.

More specifically, for ETH/UCY dataset, we recommend to use the preprocessed data and dataloader from SocialGAN [11]/Trajectron++ [39]/AgentFormer [52]. For the SDD dataset, we recommend using the preprocessed data and dataloader from Y-Net [25]. Although their data processing methods may differ, they share the same data source and data splitting approach, facilitating fair comparisons.

We also note that, in the early version of Trajectron++, a misuse of the *np.gradient* function during computation resulted in the model accessing future information. Rectifying this bug typically leads to a significant decrease in scores. Consequently, several Trajectron++-based studies have achieved improved scores.

B Controllable generation by data augmentation

Besides the fashion of manipulating generation results given the good property of our constructed mixed Gaussian prior in the main paper, we also use data augmentation to alter the data patterns in our training set, thereby obtaining different priors. This enables our model to fix corner cases that are difficult to handle with traditional flow-based models like FlowChain. Taking Figure 6 as an example, to generate the clusters representing green "U-turn", purple "left-turn", and cyan "right-turn" clusters in the right-middle figure, we duplicate and rotate the original future trajectory data by 180°, 90°, and -90°, respectively. Subsequently, we mix these rotated data with the original data in a fixed ratio to produce the augmented dataset (in this particular case, a 2:2:1:1 ratio is employed for the original:180°:90°:-90°). Then we apply k-means to the augmented dataset, thereby obtaining the new augmented prior distribution (depicted in the right-middle figure). Finally, we train the model using this augmented dataset. Compared to the generated results from FlowChain, after using the augmented data to construct the mixed Gaussian prior, our method can generate the under-explored trajectory patterns with a higher chance. After this manipulation, we could change the mixed Gaussian prior as we desire, such as amplifying the chance of generating corner cases in this example.

On the other hand, we quantitatively evaluate the ability to generate under-represented trajectory patterns. Table 5 compares the ADE/FDE scores of their worst-N samples on the UNIV dataset. Typically, the samples from the test set with the worst ADE/FDE relate to the under-represented corner cases of future trajectories. The results demonstrate quantitatively that MGF can better generate the under-represented motion patterns after injecting the desired corner cases on purpose by manipulating the mixed Gaussian prior as mentioned above. We note that all the provided examples of manipulating the mixed Gaussian prior to controlling the generation statistics do not require fine-tuning or any operation to the normalizing flow itself. As manipulating the mixed Gaussian prior is purely a parameter updating processing without any training and gradient backpropagation, all the manipulation is very fast in practice. This suggests the good efficiency and flexibility of our proposed method to achieve controllable generations.

6 Limitations

Limited by computing resources, we did not utilize the map information in our model. Some generated trajectories may overlap with obstacles, thus decreasing the upper bound of MGF's ability. Also, we found that agents can occasionally collide with each other due to the limited ability of the history encoder. Future works may take more consideration to the collision among agents or between agents and the environment.