MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction

Patrick Dendorfer*   Sven Elflein*   Laura Leal-Taixé

Technical University Munich

{patrick.dendorfer,sven.elflein,leal.taixe}@tum.de

Abstract

Pedestrian trajectory prediction is challenging due to its uncertain and multimodal nature. While generative adversarial networks can learn a distribution over future trajectories, they tend to predict out-of-distribution samples when the distribution of future trajectories is a mixture of multiple, possibly disconnected modes. To address this issue, we propose a multi-generator model for pedestrian trajectory prediction. Each generator specializes in learning a distribution over trajectories routing towards one of the primary modes in the scene, while a second network learns a categorical distribution over these generators, conditioned on the dynamics and scene input. This architecture allows us to effectively sample from specialized generators and to significantly reduce the out-of-distribution samples compared to single generator methods.

1. Introduction

To safely navigate through crowded scenes, intelligent agents such as autonomous vehicles or social robots need to anticipate human motion. Predicting human trajectories is particularly difficult because future actions are multimodal: given a past trajectory, there exist several plausible future paths, depending on the scene layout and social interactions among pedestrians. Recent methods leverage conditional generative adversarial networks (GANs) [14, 16, 34, 22] to learn a distribution over trajectories. These methods present significant improvements over deterministic models [1, 18]. However, they suffer from limitations observed in the context of GANs [38, 20] that manifest in mode collapse or prediction of undesired out-of-distribution (OOD) samples, effectively yielding non-realistic trajectories. Mode collapse can be tackled with best-of-many sampling [6] or regularizations of the latent space [22, 2] but the problem of OOD samples remains unsolved. These OOD samples are particularly problematic in real-world applications where high

precision of predictions matters. Imagine an autonomous vehicle driving through crowded environments and interacting with pedestrians. To ensure the safety of pedestrians, the vehicle needs to anticipate their future motion and react accordingly, *e.g.*, brake or turn. As a consequence, unrealistic predictions may lead to sudden reactions that pose danger to other traffic participants.

To understand why OOD samples are produced by state-of-the-art GAN methods, we need to understand the underlying geometry of the problem. Consider a pedestrian reaching the junction in Figure 1a. There are three plausible main directions that the pedestrian can take, namely, going straight, left, or right. Furthermore, there exist several paths that route towards these directions. While all recent works agree that such trajectory distribution is inherently multimodal, we further observe that the distribution consists of several disconnected modes. Each mode is shown in Figure 1c in different colors, and as we can observe, the three modes are disconnected in space. Existing GAN models do not consider this property, and hence generate undesirable OOD samples in between modes, visualized as red trajectories in Figure 1b. This is an inherent problem of single-generator GANs, as they cannot learn a mapping from a continuous latent space to a disconnected, multimodal target distribution [38].

In this paper, we address this issue and explicitly focus on learning such disconnected multimodal distributions for pedestrian trajectory prediction. To this end, we propose a novel multi-generator GAN that treats the multimodal target distribution as a mixture of multiple continuous trajectory distributions by optimizing a continuous generator for each mode. Unlike previous multi-generator models [19, 7], our model needs to adapt to the selection of generators to different scenes, *e.g.*, two- and three-way junctions. For this, we employ a fixed number of generators and allow the model to learn the necessary number of modes directly from visual scene information. Towards this end, we train a second module estimating the categorical probability distribution over the individual generators, conditioned on the input observations. At test time, we first select a specific gener-

---

*Equal contribution.

(a) Target Distribution p_D (b) Single Generator distribution p_G (c) Multi-generator distribution
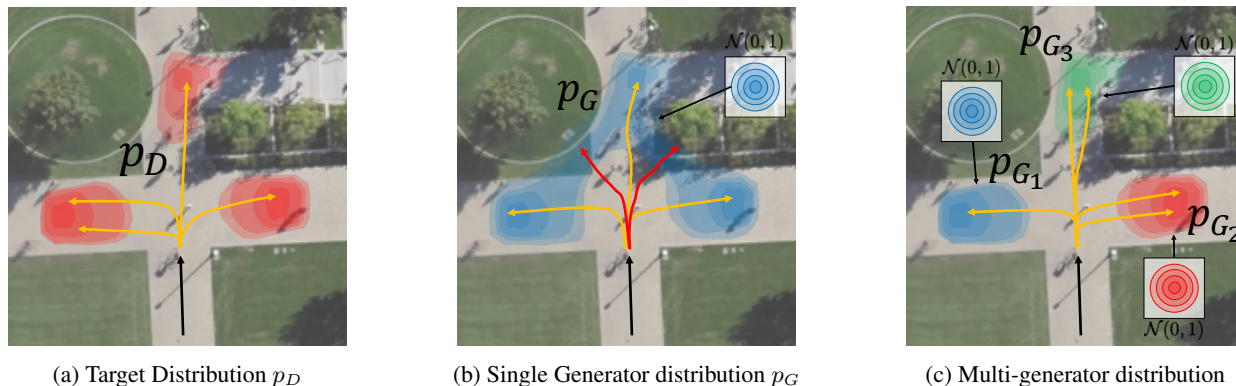
Figure 1: The figure illustrates a pedestrian reaching a junction (black) including (a) the multimodal target distribution of future paths, (b) learned future trajectory distribution by a single generator GAN predicting out-of-distribution samples (red), and (c) learned trajectory distribution of multi-generator mixture model.

ator based on its categorical probability and sample then trajectories specialized to that particular mode present in the scene. For measuring the quality of the predictions, we extend the concept of traditional L_2 error measures with a precision and recall metric [36, 23]. Our experimental evaluation shows that our proposed model overcomes state-of-the-art and single-generator methods when comparing the behavior of predicting OOD samples.

We summarize our **main contributions** as follows: (i) we discuss the limitations of single generator GANs and propose a novel multi-generator method that learns a multimodal distribution over future trajectories, conditioned on the visual input. To this end, we (ii) present a model that estimates a conditional distribution over the generators and elaborate a training scheme that allows us to jointly train our model end-to-end. Finally, (iii) we introduce recall and precision metrics for pedestrian trajectory prediction to measure the quality of the entire predictive distribution, and in particular OOD samples. We demonstrate our method's efficiency and robustness through extensive ablations. The source code of the model and experiments is available: https://github.com/selflein/MG-GAN.

2. Related Work

Trajectory Forecasting. Since its inception, the field of pedestrian trajectory prediction has moved from hand-crafted [18] to data-driven [1] methods. While the first learning methods used deterministic LSTM encoder-decoder architectures (S-LSTM [1]), deep generative models [16, 34, 22, 2, 12, 8] quickly emerged as state-of-the-art prediction methods. This development enabled the shift from predicting a single future trajectory to producing a distribution of possible future trajectories. S-GAN [16] establishes a conditional Generative Adversarial Networks [14] to learn the ground-truth trajectory distribution and S-GAN-P [16] and SoPhie [34] extend S-GAN with visual and social

interaction components. Further, S-BiGAT [22] increases the diversity of the samples by leveraging bicycle GAN training [42] that encourages the connection between the output and the latent code to be invertible. Goal-GAN [8] circumvents the problem of mode collapse by conditioning the decoder on a goal position estimated based on the topology of the scene.

GANs [14] have well-known issues with mode collapse, this is why many models [16, 34] use an L_2 variety loss [6] or modify the GAN objective [2] to encourage diversity of the samples. While producing highly diverse samples ensures coverage of all modes in the distribution, we also obtain many unrealistic out-of-distribution samples. The problem of OOD samples has been remained unnoticed partially due to the evaluation metrics used in the field which only measure the minimum L_2 distance between the set of predictions and the ground truth, namely the recall. Nonetheless, the realism of predicted trajectories, equivalent to a precision metric, is seldomly evaluated. We advocate that trajectory prediction methods should be evaluated concerning both of the aforementioned aspects.

Other work uses conditional variational autoencoders (VAE) [21] for multimodal pedestrian trajectory prediction [24, 35, 26, 5]. More recently, Trajectron++ [37] uses a VAE and represents agents' trajectories in a graph-structured recurrent neural network. PECNet [28] proposes goal-conditioned trajectory forecasting. Similar to GANs, VAEs are also continuous transformations and suffer from the limitations of generating distributions on disconnected manifolds [32].

Lastly, P2TIRL [9] learns a grid-based policy with maximum entropy inverse reinforcement learning. In summary, existing methods pay little attention to the resulting emergence of out-of-distribution samples and do not discuss the topological limitation in learning a distribution on disconnected supports.

Generation of Disconnected Manifolds. Understanding the underlying geometry of the problem is important when training deep generative models [11]. More precisely, learning disconnected manifolds requires disconnectedness within the model. A single generator preserves the topology of the continuous latent space and cannot exclusively predict samples on disconnected manifolds [38].

For image generation, the problem of multimodal learning is well-known and widely studied. Addressing this issue, [38] proposes a rejection sampling method based on the norm of the generator's Jacobian. InfoGAN [7] discretizes the latent space by introducing extra dimensions. Other works use mixtures of generators [40, 39, 41, 17, 13, 4] to construct a discontinuous function. However, these models assume either a uniform or unconditional probability for their discrete latent code or generators. As a result, these methods are unable to adapt to different scenes and thus unsuitable for the trajectory prediction task.

Our research is the first to address the problem of learning disconnected manifolds using multiple generators for the task of pedestrian trajectory prediction by modeling a conditional distribution over the generators.

3. Problem Definition

In this work, we tackle the problem of jointly predicting future trajectories for all pedestrians in the scene. For each pedestrian i, we generate a set of K future trajectories $\{\hat{Y}_i^k\}_{k=1,...,K}$ with $t \in [t_{obs}+1, t_{pred}]$ for a given input trajectory X_i with $t \in [t_1, t_{obs}]$. This implies learning the true distribution of trajectories conditioned on the input trajectories and scene layout.

In many real-world scenarios such as in Figure 1, the target distribution p_D is multimodal and composed of disconnected modes.

Why do Single Generator GANs produce OOD Samples? State-of-the-art methods use the standard conditional GAN architecture [14] and its modifications [29, 3] to learn a distribution over future trajectories. These models learn a continuous mapping $G : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$ from the latent space $\mathcal{Z}$ combined with the observations' space $\mathcal{X}$ to the space of future trajectories $\mathcal{Y}$. The probability prior $p(z)$ on $\mathcal{Z}$ is mainly a standard multivariate normal distribution with $z \sim \mathcal{N}(0, 1)$. When modeling G with a neural network, the mapping is continuous and preserves the topology of the space. Hence, the transformation $G(x, \mathcal{Z})$ of the support of the probability distribution $\mathcal{Z}$ is connected in the output space [38]. Therefore, theoretic work [38, 20] discusses that learning a distribution on disconnected manifolds is impossible; we also observe this phenomenon in our experiments.

Why are OOD Samples problematic? Real world-applications relying on trajectory predictions, *e.g.* au-

tonomous vehicles, have to treat every prediction as a possible future scenario and need to adjust their actions accordingly. Thus, not only missed but also unrealistic predictions may crucially hurt the performance of those applications. As OOD samples without support in the ground-truth distribution are likely to be unrealistic, we aim to keep their number small while still covering all modes.

How can we prevent OOD Samples? All single generator models will predict OOD if the target distribution lies on disconnected manifolds. Theoretically, there are only two ways to achieve disconnectedness in $\mathcal{Y}$: making $\mathcal{Z}$ disconnected or making the generator mapping $G : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$ discontinuous. We discuss both approaches in our paper but find the latter to be more effective.

How to measure OOD Samples? Best-of-many L_2 distance metrics focus on minimizing the error between a single sample out of a set of predictions without assessing the quality of the remaining trajectories. Therefore, we compare our model on both, recall and precision [36, 23], which are commonly used to assess the quality of generative models. While existing distance measures highly correlate with recall, we are equally interested in precision that correlates with the number of OOD samples.

4. Method

In this section, we present our multi-generator framework for pedestrian trajectory prediction. Our model learns a discontinuous function as a mixture of distributions modeled by multiple generators (Section 4.1).

To adapt to different scenes, we train a second network estimating the categorical distribution over generators (Section 4.2) for new unseen scenes.

4.1. MG-GAN

Visual and Trajectory Encoders. We outline the architecture of our model in Figure 2. First, the feature encoders extract visual and dynamic features d_i from the input sequences X_i and scene image patches I_i of each pedestrian i. The attention modules combine these encodings to compute the physical attention [34] features v_i and social attention [2] features s_i. After the encoding and attention, we concatenate the dynamic d_i, physical v_i, and social s_i features to $c_i = [d_i, v_i, s_i]$. In the following, we omit the index indicating individual pedestrians to avoid notation clutter. Note that we leverage established modules to model physical and social interactions [34, 2, 16], as our contribution is the multi-generator framework. We provide more details on these components in the supplementary.

Multi-generator Model. In our model, we leverage n_G different generators $\{G_g\}$, where each generator specializes in learning a different trajectory distribution conditioned on
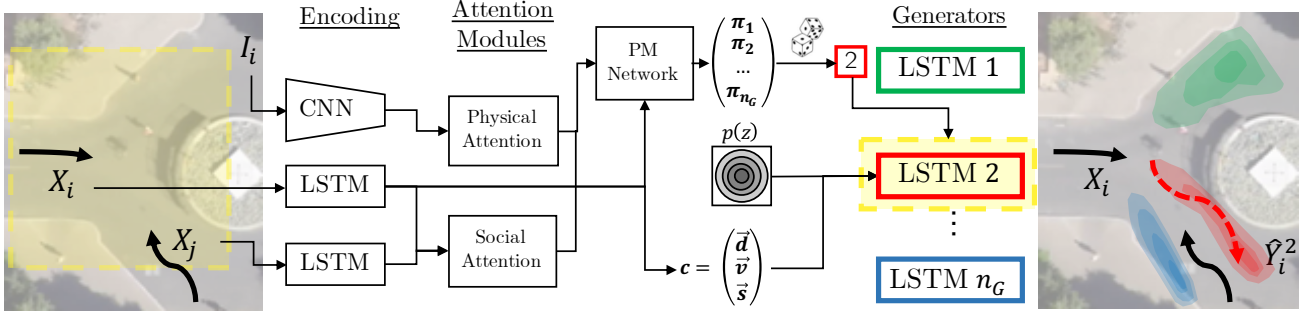
Figure 2: Architecture of MG-GAN. The scene image I_i and observed trajectories X are encoded and passed to the physical and social attention modules. The n_G generators can predict different conditional trajectory distributions for the given scene observation. The PM-Net estimates probabilities $\boldsymbol{\pi}$ for the generators. The model samples or selects a generator from $\boldsymbol{\pi}$ and predicts a trajectory $\hat{Y}$ conditioned on the features c and the noise vector z.

the input c. All generators share the same network architecture, however, they do not share weights. The generator architecture consists of a LSTM decoder, initialized with the features c and a random noise vector $z \sim \mathcal{N}(0, 1)$ as the initial hidden state h^0. The final trajectory $\hat{Y}$ is then predicted recurrently:

$$\Delta \hat{Y}^t = \text{LSTM}_g \left(\Delta X^{t-1}, h^{t-1} \right). \qquad (1)$$

Existing multi-generator modules proposed in the context of image generation assume the distribution over the generators to be constant [19, 17]. However, in the case of trajectory prediction, the number of modes is unknown a priori. Therefore, we train a module that adapts to the scene by activating specific generators, conditioned on the observations and interactions c.

4.2. Path Mode Network (PM-Net)

The Path Mode Network (PM-Net) parameterizes a distribution over the indices of the generators $p(g|c) = [\pi_1, \cdots, \pi_{n_G}]$ conditioned on the features c and is modelled with a multi-layer perceptron $\text{MLP}(c)$. The outputs $\{\pi_g\}$ assign probabilities to each of the n_G generators. During inference, we can sample different generators based on the predicted distribution. Note, that this provides a major advantage over existing methods [19, 20], where the distribution is fixed and cannot adjust to different scenes. In comparison, our PM-Net is capable of selecting the relevant generators for a given scene while deactivating unsuitable ones.

4.3. Model Training

We now present a training algorithm that jointly optimizes the distribution over generators parameterized by PM-Net and the multi-generator GAN model. For this, we propose an alternating training scheme, inspired by expectation-maximization [15, 20].

4.3.1 GAN Training

We train our model using a conditional generator, discriminator network D [14] that distinguishes between real and fake trajectories and a classifier C [19] learning to identify which generator predicted a given trajectory. More details on these networks' architectures can be found in the supplementary.

Adversarial Loss. We define each generator G_g as $\hat{Y}_{g,z} = G_g(c, z)$ inducing an implicit distribution $p_{G_g}(\hat{Y}|c)$. All n_G generators together describe a joint probability distribution $\sum_{g=1}^{n_G} \pi_g \, p_{G_g}(\hat{Y}|c)$, thus the established results [14] for GANs hold. We use the original adversarial loss $\mathcal{L}_{Adv}$ [14]. The discriminator D learns to distinguish between real samples Y and samples $\hat{Y}$ generated by the model encouraging realism of the predictions. However, D by itself does not prevent the generators from collapsing to the same mode.

Classification Loss. To incentivize the generators to cover different, possibly distinct modes occupying different regions of the output space, we follow [19] and introduce a classifier C which aims to identify the generator index g that generated a sample $\hat{Y}_{g,z}$. The cross-entropy loss $\mathcal{L}_{Cl}$ between the classifier output and the true generator label of the predicted trajectory encourages the generators to model non-overlapping distributions and drives the trajectories of different generators spatially apart. This behavior is regularized through the adversarial loss $\mathcal{L}_{Adv}$ that constrains the samples to be realistic and not diverge from the real distribution. Overall, the training object reads as follows

$$\min_G \max_D \mathcal{L}_{Adv} + \lambda_{Traj}\mathcal{L}_{Traj} + \lambda_{Cl}L_{Cl}, \qquad (2)$$

where we additionally apply a L_2 best-of-many loss [6, 16] $\mathcal{L}_{Traj}$ with q samples to increase the diversity of predicted trajectories. λ_{Traj} and λ_{Cl} are weighting hyperparameters.

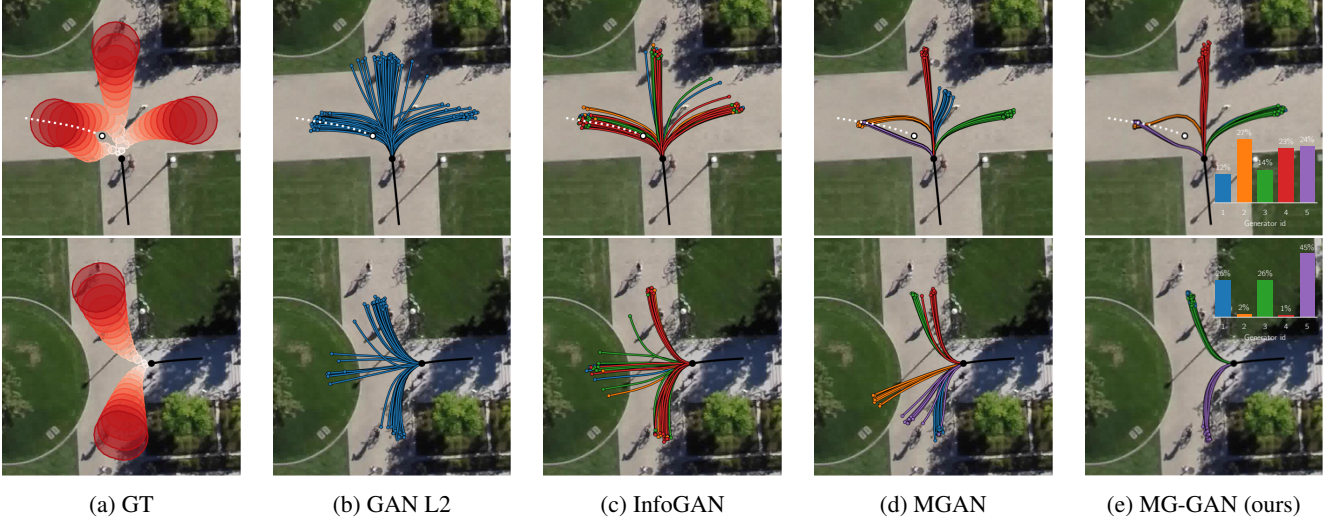|				
(a) GT	(b) GAN L2	(c) InfoGAN	(d) MGAN	(e) MG-GAN (ours)

Figure 3: Predicted trajectories for two scenarios in the synthetic dataset. The upper row contains scene on a junction with 3 modes and an interacting pedestrian (white). The lower row shows a scenario with two modes. Figures (a) represent the support of the conditional multimodal ground-truth distributions for these scenes. Figures (e) of MG-GAN also show the probabilities $\boldsymbol{\pi}$ of the PM Network. We visualize trajectories of one generator/discrete latent variable in the same color.

4.3.2 PM-Net Training

To train PM-Net, we approximate the likelihood of a particular generator distribution p_{G_g} supporting the trajectory Y by the generated trajectories $\hat{Y}_{g,c,z_i} = G_g(c, z_i)$ as:

$$p(Y|c, g) \propto \frac{1}{l} \sum_{i=1}^{l} \exp \left(\frac{-\left\| \hat{Y}_{g,c,z_i} - Y \right\|_2^2}{2\sigma} \right). \quad (3)$$

Here, we marginalize the GAN noise z and assume a normally distributed and additive error $\epsilon \sim N(0, \sigma I)$ between $\hat{Y}$ and Y as common for regression tasks [10]. We obtain the conditional probability over generators by applying Bayes' rule:

$$p(g|c, Y) = \frac{p(Y|c, g)}{\sum_{g'}^{n_G} p(Y|c, g')}. \quad (4)$$

Finally, we optimize the PM-Net with the approximated likelihood minimizing the cross entropy loss:

$$\mathcal{L}_\Pi = H(p(g|c, Y), \Pi(c)). \quad (5)$$

Intuitively, the network is trained to weigh the generator that generates trajectories closest to the ground-truth sample the highest. We provide the full derivation of the objective in the supplementary.

4.3.3 Alternating Training Scheme

Our training scheme consists of two alternating steps similar to an expectation-maximization algorithm [15]:

1. PM-Net Training Step: We sample l trajectories for each generator and optimize the parameters of PM-Net using Equation (5) while keeping the rest of the network's parameters fixed.

2. Generator Training Step: In the generator training step, we use PM-Net to generate probabilities $\boldsymbol{\pi}$ and sample q generators predicting trajectories. With these predictions, we update the model excluding PM-Net optimizing Equation (2). We provide pseudo-code detailing our training procedure in the supplementary.

4.4. Trajectory Sampling

We can use the estimated probabilities $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_{n_G}]$ generated by the PM-Net to establish different mechanisms to sample trajectories from the multiple generators. This helps us to cover all modes present in the scene with as-few-as-possible predictions. In single generator models [22, 34] the relation between regions in the Gaussian latent space and different modes in the output space is implicit and unknown. However, for MG-GAN we can use the estimated probabilities $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_{n_G}]$ from the PM-Net to control and to cover predictions for all modes present in the scene. Next to randomly sampling k trajectories (*Random*) from $\boldsymbol{\pi}$ we introduce an additional strategy (*Expectation*) where we compute the expected number of samples for each generator as $n_g = k \cdot \pi_g$. We round all n_g to the nearest integer and adjust the number of the generator with the highest score to ensure that all numbers sum up to k.
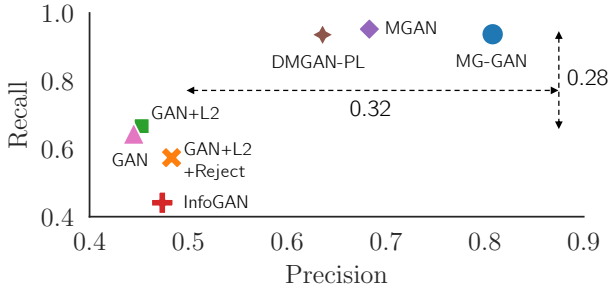
Figure 4: Precision vs. Recall on synthetic dataset.

Figure 5: Generated samples of our MG-GAN, Trajectron++, and PECNet.

5. Experimental Evaluation

We evaluate our model on four publicly available datasets [30, 25, 31, 27] for pedestrian trajectory prediction and compare our results with state-of-the-art methods. Furthermore, we conduct experiments on synthetic datasets. Compared to real data, synthetic data provides access to the ground-truth trajectory distribution which enables us to identify OOD samples by comparing ground-truth and generated trajectory distributions. Finally, we run an ablation on the individual components of MG-GAN and study the robustness of our model w.r.t. the number generators n_G.

5.1. Experimental Setup

We follow prior work [31, 1] and observe 8 past time steps (3.2 seconds) and predict the future 12 time steps (4.8 seconds) for every pedestrian in the scene.

Metrics. We evaluate results using the following metrics: *Average Displacement Error (ADE)* is defined as a mean L_2 distance between the prediction and ground-truth trajectory. *Final Displacement Error (FDE)* is defined as the distance between the prediction and ground-truth trajectory position at time t_{pred}.

For both metrics, ADE and FDE, we follow the *Minimum over k* procedure [16, 34, 22] with $k = 20$. Note that this approach only considers a single prediction with the lowest ADE and FDE, but not the entirety of the set of k generated output trajectories combined. Therefore, we include additional metrics commonly used in the GAN literature [36, 23], namely *recall* and *precision*. Recall measures the coverage of all ground-truth modes, while precision measures the ratio of generated samples in the support of the ground truth distribution. Hence, the precision is directly related to the number of OOD samples. We also compute the $F1$ score, combining recall and precision.

Datasets. We perform the evaluation using the following datasets. ETH [30] and UCY datasets [25] contain five sequences (ETH: ETH and HOTEL, UCY: UNIV, ZARA1, and ZARA2), recorded in four different scenarios. We follow the standard leave-one-out approach for training and
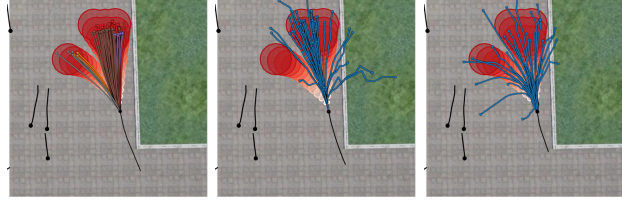
testing, where we train on four datasets and test on the remaining one. The Stanford Drone Dataset (SDD) [31] consists of 20 video sequences captured from a top view at the Stanford University campus. In our experiments, we follow the train-test-split of [33] and focus solely on pedestrians. The recently proposed Forking Paths Dataset (FPD) [27] is a realistic 3D simulated dataset providing multi-future trajectories for a single input trajectory. To study the ability of our model to predict multimodal trajectories while preventing OOD samples, we create a synthetic dataset where we simulate multiple possible future paths for the same observation emerging due to the scene layout and social interactions. Detailed information on the generated dataset is provided in the supplementary material.

Baselines. We compare our method with several single and multi-generator GAN baselines. We evaluate a (i) vanilla *GAN* baseline, (ii) *GAN L2* trained with variety loss [6], (iii) *GAN L2 Reject* [38] that filters OOD samples based on gradients in the latent space, and (iv) InfoGAN [7] with discrete random latent variable. Furthermore, we compare MG-GAN to multi-generator models MGAN [19] and DMGAN-PL [20], proposed in the context of image generation, that we adapt for the task of trajectory prediction. To ensure comparability, all models use the same base model following SoPhie [34] with attention modules as described in Section 4.1. For qualitative comparison, we evaluate our method against state-of-the-art prediction models presented in Section 2 on the standard benchmarks for trajectory forecasting.

5.2. Experiments on Synthetic data

We first study our model on a synthetic dataset in which we have access to the ground-truth distribution of the future trajectories. In this experiment, we show that MG-GAN achieves better performance in learning a multimodal trajectory distribution with disconnected support and is more efficient than the baselines.

Results. The results in Figure 4 show that MG-GAN outperforms the single-generator baselines and increases Recall by 0.28 and Precision by 0.32. To this end, we find that all multi-generator methods have a similar recall but MG-GAN achieves a 15% higher Precision corresponding to a lower number of OOD samples.
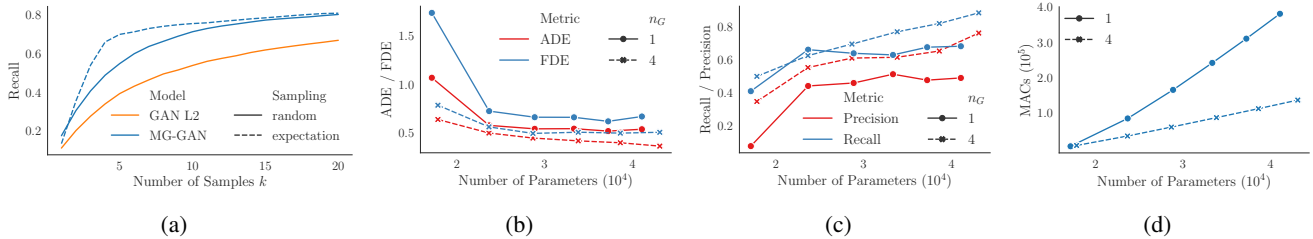
Figure 6: Comparison between single generator model *GAN+L2* and MG-GAN. (a) recall for different number of samples k and sampling methods. (b) - (c) compares ADE/FDE, recall/precision, and MACs (Multiply–accumulate operations) for varying total number of model parameters.

Visual Results. In Figure 3, we visualize predicted trajectories for two different scenarios where the white trajectory represents another interacting pedestrian. The support of the ground-truth distribution for each timestep is shown as a red circle in Figure 3a. A model achieves low precision in Figure 4 if many trajectory points lie outside the corresponding red circle for a particular timestep. Similarly, a model has high recall if its samples cover most of the area of red circles. Single generator models, GAN+L2 (Figure 3b), and InfoGAN (Figure 3c) produce many OOD samples leading to low precision. In particular, we find that InfoGAN is not able to learn the correspondence between the discrete latent space and the modes in the trajectory space. While theoretically plausible, these results indicate that a discretized latent space is not well-suited for learning distribution on disconnected support. Contrarily, MGAN can learn the distribution but is incapable to adjust generators resulting in OOD samples in Figure 3d when the number of modes does not match the number of generators. Finally, our MG-GAN is able to adjust to both scenarios in Figure 3e as the PM-Net deactivates generators which are unsuitable and prevents OOD samples explaining the high Precision in Figure 4.

Effective Mode Covering. Figure 6a shows the recall depending on the number of samples k. Our method covers more modes of the ground-truth distribution than the single generator model for the same number of samples as indicated by the higher recall. Additionally, we observe significant improvements compared to random sampling by using expectation sampling leveraging PM-Net as described in Section 4.4, especially for fewer samples.

Number of Parameters and Computational Cost. In this experiment, we show that our MG-GAN does not require more resources w.r.t. parameters or computations compared to a single generator baseline. For this, we compare MG-GAN using four generators with the single-generator baseline while keeping the total number of parameters of both models fixed by only using approx. $1/4$ of the parameters for each generator. As can be seen in Figures 6b and 6c, MG-GAN outperforms the single generator GAN w.r.t. to ADE/FDE (50%) and recall/precision (30%) using

Dataset	S-LSTM [1]	S-GAN [16]	SoPhie [34]	S-BiGAT [22]	CGNS [26]	GoalGAN [8]	PECNet [28]	Trajectron++ [37]	MG-GAN (Ours)
ETH	1.09/2.35	0.81/1.52	0.70/1.43	0.69/1.29	0.62/1.40	0.59/1.18	0.54/0.87	**0.39/0.83**	0.47/0.91
HOTEL	0.79/1.76	0.72/1.61	0.76/1.67	0.49/1.01	0.70/0.93	0.19/0.35	0.18/0.24	**0.12/0.21**	0.14/0.24
UNIV	0.67/1.40	0.60/1.26	0.54/1.24	0.55/1.32	0.48/1.22	0.60/1.19	0.35/0.60	**0.20/0.44**	0.54/1.07
ZARA1	0.47/1.00	0.34/0.69	0.30/0.63	0.30/0.62	0.32/0.59	0.43/0.87	0.22/0.39	**0.15/0.33**	0.36/0.73
ZARA2	0.56/1.17	0.42/0.84	0.38/0.78	0.36/0.75	0.35/0.71	0.32/0.65	0.17/0.30	**0.11/0.25**	0.29/0.60
AVG	0.72/1.54	0.58/1.18	0.54/1.15	0.48/1.00	0.49/0.97	0.43/0.85	0.29/0.48	**0.19/0.41**	0.36/0.71

Table 1: Quantitative results on ETH [30] and UCY [25]. We report ADE ($\downarrow$) /FDE ($\downarrow$) in meters. Underlined results denote the second best.

	S-LSTM [1]	S-GAN [16]	CAR-NET [35]	DESIRE [24]	SoPhie [34]	CGNS [26]	CF-VAE [5]	P2TIRL [9]	GoalGAN [8]	PECNet [28]	MG-GAN (4) (Ours)
ADE	57.0	27.3	25.7	19.3	16.3	15.6	12.6	12.6	12.2	**10.0**	13.6
FDE	31.2	41.4	51.8	34.1	29.4	28.2	22.3	22.1	22.1	**15.9**	25.8

Table 2: Quantitative results on Stanford Drone Dataset (SDD) [31]. We report ADE and FDE in pixels.

the same number of total parameters across various parameter budgets. In Figure 6a, the computational cost measured by MACs for the prediction of a trajectory is always lower for MG-GAN compared to the baseline. The model only runs one selected generator with $1/4$ amount of parameters during the forward pass while the cost of running PM-Net is negligible.

5.3. Benchmark Results

In this section, we compare our method to the state-of-the-art on the standard benchmarks ETH [30], UCY [25], and SDD [31], as well as the recently proposed Forking Path Dataset (FPD) [27]. We report the performance of the model with the lowest validation error as we train our method with different numbers of generators $n_G \in \{2, \ldots, 8\}$. We discuss the robustness w.r.t. the number of generators in Section 5.4.

ADE & FDE. Our MG-GAN achieves competitive results for the ADE and FDE on the ETH/UCY and Stanford Drone Dataset (SDD) shown in Table 1 and Table 2, respectively. Even though our method does not achieve SOTA performance on the ADE and FDE metrics on these benchmarks, we still argue that our method provides significant improvement to the task. That is since the distance-based L_2 measures can be drastically reduced by increasing the variance of the predictions for the price of producing

	ADE ↓	FDE ↓	Precision ↑	Recall ↑	F1 ↑
GAN+L2	28.81	58.37	0.55	0.87	0.67
PECNet	**13.14**	**24.55**	0.46	0.95	0.62
Trajectron++	13.15	32.00	0.38	**0.96**	0.54
MG-GAN (Ours)	22.09	46.38	**0.71**	0.89	**0.79**

Table 3: Results on FPD. We report ADE/FDE in pixels.

more OOD samples. A visual comparison of the trajectories produced by Trajectron++ and PECNet in Figure 5 shows that these methods produce high variance predictions without accounting for any constraints in the scene. Contrarily, MG-GAN only predicts trajectories inside the ground-truth manifold (red). While covering all modes, our predictions remain in the support of the ground-truth distribution. To quantify this observation, we compute the recall and precision metrics.

Precision & Recall. As ADE and FDE do not consider the quality of the entire generated distribution, we add results using precision/recall metrics [36, 23] on the FPD dataset [27]. This is possible on FPD as it contains multiple feasible, human-annotated ground-truth trajectories.

In Table 3, MG-GAN outperforms *GAN+L2* by 29%, PECNet by 54% and Trajectron++ by 86% in terms of Precision, while the difference in Recall with 0.02, 0.06, and 0.07 points is small. Single generator models predict overly diverse trajectories, thus increasing Recall slightly and reducing ADE/FDE, but produce OOD samples leading to low Precision. These results confirm that MG-GAN is significantly more reliably at predicting paths that align well with the human-annotated future trajectories (high precision), while also covering a similar amount of modes in the scene (high recall). Overall, we conclude that MG-GAN does not match SOTA performance on traditional evaluation metrics in Table 1 and Table 2. However, studying precision and recall reveals that our model can lower the number of OOD and achieves an overall better *F1* than current SOTA methods.

5.4. Ablation Studies

In this section, we ablate the key modules of MG-GAN. We emphasize that the goal of the paper is to demonstrate the need and effectiveness of a conditional multi-generator framework for pedestrian trajectory prediction. Hence, the study of attention modules used within our model described in Section 4.1, is not the goal of this work and has been extensively done in prior work [16, 2, 34, 22].

Effectiveness of Key Modules. We perform the ablation on our synthetic dataset by removing key components from our final model: multiple generators, the classifier C, and the PM-Net in Table 4. Reducing the number of generators to 1 results in a significant drop in performance of almost

M	C	PM	ADE ↓	FDE ↓	Precision ↑	Recall ↑
			0.94	1.58	0.46	0.48
✓			0.59	0.79	0.37	0.68
✓		✓	0.35	0.49	0.72	0.91
✓	✓		0.37	0.53	0.73	0.91
✓	✓	✓	**0.32**	**0.44**	**0.77**	**0.95**

Table 4: Ablation experiments: (M) Multi-generator, (C) Classifier, and (PM) Path Mode network.

	2	3	4	5	6	7	8	Best
ADE	0.37	0.38	0.38	0.39	0.37	0.36	0.37	0.36
FDE	0.72	0.74	0.75	0.76	0.71	0.71	0.72	0.70

Table 5: Results for $n_G \in \{2, \ldots, 8\}$ on ETH/UCY.

50% in recall and 31% in precision.

As described in Section 4.1, the classifier C encourages individual generators to specialize and increases precision from 37% to 73%. Similarly, with PM-Net learning a distribution over generators, the precision increases from 37% to 72%. Finally, leveraging PM-Net and classifier C, combining the advantages of both, further improves the performance on all considered metrics.

Robustness over the Number of Generators. The multimodality over future trajectories depends on social interactions and the scene layout, imposing a significant challenge when choosing the number of generators n_G at training time. To this end, we introduced the PM-Net that learns to activate generators depending on the observed scene features. As can be seen in Table 5, PM-Net successfully makes MG-GAN robust w.r.t. the choice of n_G as results only deviate 7% from the best reported values at maximum.

6. Conclusion

In this paper, we addressed the issue of single-generator GAN models for pedestrian trajectory prediction. While existing generative networks learn a distribution over future trajectories, they are fundamentally incapable of learning a distribution consisting of multiple disconnected modes. To overcome this problem, our proposed MG-GAN leverages multiple generators that specialize in different modes and learns to sample from these generators conditioned on the scene observation. We demonstrated the efficacy of MG-GAN at reducing out-of-distribution samples in comparison to the existing state-of-the-art. Finally, we emphasized the importance of precision next to recall metrics and hope to encourage a discussion on preventing OOD in future work.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2, 6, 7

[2] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social Ways: Learning multi-modal distributions of pedestrian trajectories with GANs. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 1, 2, 3, 8

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, 2017. 3

[4] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and Equilibrium in Generative Adversarial Nets GANs. In *International Conference on Machine Learning*, 2017. 3

[5] Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional Flow Variational Autoencoders for Structured Sequence Prediction. In *Neural Information Processing Systems*, 2019. 2, 7

[6] Apratim Bhattacharyya, Bernt Schiele, and Mario Fritz. Accurate and diverse sampling of sequences based on a "Best of Many" sample objective. In *Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 4, 6

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Neural Information Processing Systems*, 2016. 1, 3, 6

[8] Patrick Dendorfer, Aljoša Ošep, and Laura Leal-Taixé. Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation. In *Asian Conference on Computer Vision*, 2020. 2, 7

[9] Nachiket Deo and Mohan M. Trivedi. Trajectory Forecasts in Unknown Environments Conditioned on Grid-Based Plans. *arXiv e-prints*, page arXiv:2001.00735, 2020. 2, 7

[10] Norman R Draper and Harry Smith. *Applied regression analysis*, volume 326. John Wiley & Sons, 1998. 5

[11] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the Manifold Hypothesis. *Journal of the American Mathematical Society*, 2013. 3

[12] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. GD-GAN: Generative Adversarial Networks for Trajectory Prediction and Group Detection in Crowds. In *Asian Conference on Computer Vision*, 2018. 2

[13] A. Ghosh, V. Kulharia, V. Namboodiri, P. H. S. Torr, and P. K. Dokania. Multi-agent Diverse Generative Adversarial Networks. In *Conference on Computer Vision and Pattern Recognition*, 2018. 3

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Neural Information Processing Systems*, 2014. 1, 2, 3, 4

[15] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Neural Information Processing Systems*, 2017. 4, 5

[16] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 3, 4, 6, 7, 8

[17] Hao He, Hao Wang, Guang-He Lee, and Yonglong Tian. Bayesian Modelling and Monte Carlo Inference for GAN. In *International Conference on Learning Representations*, 2019. 3, 4

[18] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51, 1995. 1, 2

[19] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Q. Phung. MGAN: Training Generative Adversarial Nets with Multiple Generators. In *International Conference on Learning Representations*, 2018. 1, 4, 6

[20] Mahyar Khayatkhoei, Maneesh K. Singh, and Ahmed Elgammal. Disconnected Manifold Learning for Generative Adversarial Networks. In *Neural Information Processing Systems*, 2018. 1, 3, 4, 6

[21] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2013. 2

[22] 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. In *Neural Information Processing Systems*, 2019. 1, 2, 5, 6, 7, 8

[23] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved Precision and Recall Metric for Assessing Generative Models. In *Neural Information Processing Systems*, 2019. 2, 3, 6, 8

[24] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2, 7

[25] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by Example. *Comput. Graph. Forum*, 2007. 6, 7

[26] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional Generative Neural System for Probabilistic Trajectory Prediction. In *International Conference on Intelligent Robots and Systems*, 2019. 2, 7

[27] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2020. 6, 7, 8

[28] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European Conference on Computer Vision*, 2020. 2, 7

[29] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled Generative Adversarial Networks. In *International Conference on Learning Representations*, 2017. 3

[30] S. Pellegrini, Andreas Ess, and L. Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European Conference on Computer Vision*, 2010. 6, 7

[31] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision*, 2016. 6, 7

[32] Jason Tyler Rolfe. Discrete Variational Autoencoders. In *International Conference on Learning Representations*, Apr. 2017. 2

[33] Amir Sadeghian, Vineet Kosaraju, Agrim Gupta, Silvio Savarese, and Alexandre Alahi. TrajNet: Towards a Benchmark for Human Trajectory Prediction. *arXiv preprint*, 2018. 6

[34] 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 *Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 5, 6, 7, 8

[35] Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-Net: Clairvoyant attentive recurrent network. In *European Conference on Computer Vision*, 2018. 2, 7

[36] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing Generative Models via Precision and Recall. In *Neural Information Processing Systems*, 2018. 2, 3, 6, 8

[37] T. Salzmann, B. Ivanovic, Punarjay Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, 2020. 2, 7

[38] Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jeremie Mary. Learning disconnected manifolds: a no GANs land. In *Proceedings of Machine Learning and Systems*, pages 6767–6776, 2020. 1, 3, 6

[39] Ilya O Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. AdaGAN: Boosting generative models. In *Neural Information Processing Systems*, 2017. 3

[40] Yaxing Wang, Lichao Zhang, and Joost Van De Weijer. Ensembles of generative adversarial networks. *arXiv preprint arXiv:1612.00991*, 2016. 3

[41] Peilin Zhong, Yuchen Mo, Chang Xiao, Pengyu Chen, and Changxi Zheng. Rethinking Generative Mode Coverage: A Pointwise Guaranteed Approach. In *Neural Information Processing Systems*, 2019. 3

[42] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Neural Information Processing Systems*, 2018. 2