2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

DD-MDN: Human Trajectory Forecasting with Diffusion-Based Dual Mixture Density Networks and Uncertainty Self-Calibration

Manuel Hetzel[1] Kerim Turacan[1] Hannes Reichert[1] Konrad Doll[1] Bernhard Sick[2]

Abstract— **Human Trajectory Forecasting (HTF) predicts future human movements from past trajectories and environmental context, with applications in Autonomous Driving, Smart Surveillance, and Human-Robot Interaction. While prior work has focused on accuracy, social interaction modeling, and diversity, little attention has been paid to uncertainty modeling, calibration, and forecasts from short observation periods, which are crucial for downstream tasks such as path planning and collision avoidance. We propose DD-MDN, an end-to-end probabilistic HTF model that combines high positional accuracy, calibrated uncertainty, and robustness to short observations. Using a few-shot denoising diffusion backbone and a dual mixture density network, our method learns self-calibrated residence areas and probability-ranked anchor paths, from which diverse trajectory hypotheses are derived, without predefined anchors or endpoints. Experiments on the ETH/UCY, SDD, inD, and IMPTC datasets demonstrate state-of-the-art accuracy, robustness at short observation intervals, and reliable uncertainty modeling. The code is available at: https://github.com/kav-institute/ddmdn.**

I. Introduction

Human Trajectory Forecasting (HTF) is crucial for enabling intelligent autonomous systems such as autonomous driving [1], [2], robot guidance [3], and service robots to operate safely among humans [4]. It aims to estimate the most probable future states of human motion that exhibit intrinsic multimodality: identical histories can branch into diverse futures influenced by environmental and social constraints, as well as evolving intentions. Forecasting multiple diverse future trajectories is insufficient for downstream tasks, such as path planning and decision-making, which internally rely on probability estimates. Therefore, it is essential to model appropriate, calibrated uncertainty estimates for each discrete forecast. Numerous studies support the added value of probabilistic output modeling with uncertainty-related forecasts compared to discrete forecasts without any uncertainty-related output for path planning and situational assessment [5], [6], [7].

Over the past few years, HTF research has gained significance, with several dozen publications emerging each year, all employing stochastic trajectory prediction as the primary forecasting objective. It aims for the best positional accuracy. A model outputs a distribution from which K hypotheses are derived with at least one sample closely matching the Ground Truth (GT). However, these methods often lack evaluation of their underlying distributions, uncertainty quantification, and reliability calibration. Cognitive biases, such as the Dunning–Kruger effect, highlight misplaced confidence [8], which is mirrored by neural predictors that often over- or underestimate confidence [9], thereby undermining downstream decision-making processes. We demonstrate that accuracy-focused HTF methods are prone to incorrect confidence estimates due to their focus on K-shot accuracy. Moreover, the vast majority do not provide probability scores for the forecasted hypotheses. Both represent research gaps we aim to address.

We propose DD-MDN (Diffusion-based Dual Mixture Density Network): an end-to-end probabilistic HTF model unifying multimodal accuracy with self-calibrated uncertainty from the very first observations. A few-shot diffusion backbone and dual Mixture Density Network (MDN) generate uncertainty calibrated residence areas alongside confidence-related anchor-paths, ensuring aleatoric uncertainty modeling, probability-sorted K-shot hypotheses ranking, and robustness to short observation periods. Experiments on the ETH/UCY, SDD, inD, and IMPTC datasets demonstrate state-of-the-art (SOTA) accuracy, calibrated uncertainty estimates, and robustness, without requiring predefined waypoints or knowledge distillation.

II. Related Work

Most HTF methods internally use probabilistic approaches and can therefore be described as probabilistic models. However, their results are deterministic (i.e., discrete future trajectories without uncertainty estimates); therefore, with respect to the final output, such methods can also be described as deterministic. The designation depends on the respective perspective. In the following, we refer to a model's output as either deterministic or probabilistic classification.

A. Deterministic Methods

Deterministic forecasting methods aim to generate at least one discrete forecast that is as close as possible to the GT. [10], [11], [12] provide detailed surveys of deterministic HTF methods, including the latest developments and achievements. Social-**LSTM** [13] introduced social pooling for interactions, enhanced by Group-LSTM and SS-LSTM. **CNNs** capture spatial features and demonstrate ways to include map data (Y-net [14], Next [15]). **GNNs** model agent interactions using nodes and edges (Social-STGCNN [16], GroupNet [17]). **GANs** such as Social-GAN [18] and its adaptations Sophie [19] combine encoded features with Gaussian noise and learn discriminators for

[1]The authors are with the Faculty of Engineering, University of Applied Sciences Aschaffenburg, Germany. E-mail: {manuel.hetzel,kerim.turacan,hannes.reichert,konrad.doll}@th-ab.de.

[2]The author is with the Intelligent Embedded Systems Lab, University of Kassel, Germany. E-mail: {bsick}@uni-kassel.de.

trajectory validation. **CVAEs** explicitly learn conditional trajectory distributions (Trajectron++ [20], PECNet [21]). **Normalizing Flow** networks (STGlow [22], FlowChain [23]) explicitly model data distributions via invertible transformations. Finally, **Transformer**-based architectures have recently excelled by incorporating self-attention and denoising diffusion, representing the current SOTA in deterministic HTF (LED [24], SingularTrajectory [25], MoFlow [26]). All methods focus on social-interaction handling and positional-accuracy modeling, achieving continuous improvements in both.

B. Probabilistic Methods and Uncertainty Modeling

Uncertainty handling and the generation of accurate probabilistic forecasts are treated significantly less frequently in HTF than in deterministic approaches. [10] provides an extensive overview of probabilistic and deterministic HTF methods. In addition, the reliability of the estimates and the calibration evaluation are rarely considered, leaving a research gap. Social-STAGE [27] ranks multimodal Gaussian distributions using probabilities for each mode. TUTR [28] uses a Transformer architecture to forecast K trajectories with associated probabilities encoded from their internal data distribution. In *PSU-TF* [29], uncertainties from the perception level are included in the forecasting process, and in [5], Bayesian Deep Learning techniques are used to provide realistic uncertainty estimates on the computed forecasts. All listed methods aim to partially or fully integrate uncertainty estimation, but none address uncertainty calibration or reliability evaluation. Some works, such as [30], utilize MDNs to model continuous probability distributions with Confidence Levels (CL) and provide uncertainty evaluation, a methodology comparable to our contribution but without denoising diffusion, discrete hypothesis generation/evaluation, or context integration.

C. Contributions

With DD-MDN, we address the gaps outlined above. Our main contributions are:

- An end-to-end HTF framework coupling a few-shot denoising diffusion backbone with a dual MDN, producing uncertainty-calibrated residence areas and probability-ranked trajectory forecasts.
- Direct modeling of aleatoric uncertainty, yielding well-calibrated likelihood-ranked hypotheses without post-hoc recalibration.
- State-of-the-art accuracy, calibration, and short-observation robustness on various HTF datasets.

III. Methodology

We first define the HTF problem, then provide a high-level overview of the architecture, and describe our input encoding. Next, we explain how DD-MDN performs probabilistic modeling with self-calibration, and dynamic mode pruning is implemented. Finally, we detail the semi-unsupervised denoising diffusion process and the generation of K confidence-enriched discrete trajectory hypotheses.

A. Problem Statement

HTF aims to forecast the future 2D positions of multiple agents based on their past movements and environmental context. Let A be the number of agents, T_{hist} the history length, and T_{fut} the forecast horizon. We define:

$$\mathbf{X}_a^{\text{in}} = \left\{ \mathbf{x}_{tp}^a \in \mathbb{R}^2 \;\middle|\; tp = 1, \ldots, T_{\text{hist}} \right\} \tag{1}$$

$$\mathbf{Y}_a^{\text{gt}} = \left\{ \mathbf{x}_{tf}^a \in \mathbb{R}^2 \;\middle|\; tf = T_{\text{hist}}+1, \ldots, T_{\text{hist}}+T_{\text{fut}} \right\} \tag{2}$$

$$\mathcal{X}^{\text{in}} = \left\{ \mathbf{X}_a^{\text{in}} \right\}_{a=1}^{A} \quad , \quad \mathcal{Y}^{\text{gt}} = \left\{ \mathbf{Y}_a^{\text{gt}} \right\}_{a=1}^{A} \tag{3}$$

The model takes as input all past trajectories, denoted as $\mathcal{X}^{\text{in}}$, and an occupancy grid $\mathbf{G}$ that encodes scene context. To capture multimodality and aleatoric uncertainty, we predict for each agent a set of K hypotheses $\dot{\mathcal{Y}}_a$ with corresponding probabilities $\dot{\mathcal{P}}_a$, see Eq. (4), ensuring that at least one high-probability trajectory sample $\dot{\mathbf{Y}}_a^k$ per agent lies close to the GT.

$$\dot{\mathcal{Y}}_a = \left\{ \dot{\mathbf{Y}}_a^k \right\}_{k=1}^{K} \quad \text{with} \quad \dot{\mathcal{P}}_a = \left\{ \dot{P}_a^k \right\}_{k=1}^{K} \tag{4}$$

Explicitly, we learn a probabilistic model $F_\Theta\big(\dot{\mathcal{Y}}_a, \dot{\mathcal{P}}_a \mid \mathcal{X}^{\text{in}}, \mathbf{G}\big)$ with parameters Θ, which jointly models multimodality and uncertainty. Because future paths are inherently uncertain, we adopt the standard SOTA stochastic trajectory prediction procedure and extend it as described in Section I with calibrated aleatoric uncertainty estimates and CLs.

B. Architecture

DD-MDN is an end-to-end learning framework comprising three parts: Encoding, Probabilistic Modeling, and Deterministic Hypotheses Generation. An architectural overview is illustrated by Fig. 1. Classic encoder networks encode input data. A temporal encoder (LSTM) is used for temporal inputs (past agent motions), a spatial encoder (CNN) for spatial data (occupancy grid), and transformers (TF) for self- and social-attention matters, see Section III-C. Probabilistic modeling processes temporal and spatial input features by using a dual MDN that comprises a shared TF-based denoising diffusion backbone and three probabilistic heads, which generate two distinct distributional representations. The architecture can handle various continuous probability distributions, including the Gaussian, Laplace, and Cauchy distributions. For this work, we use Gaussian Mixtures (GM). We model future agent positions and their corresponding uncertainty estimates using two complementary GM representations, as illustrated in Fig. 2. The first is a per-future-timestep GM Θ^{step}, which provides calibrated one-step uncertainty at each future time step tf. For each timestep the conditional distribution of an arbitrary 2D position $\mathbf{x}_{tf} \in \mathbb{R}^2$ can be described as in Eq. (5):

$$\Theta^{step}(\mathbf{x}_{tf}) = \sum_{m=1}^{M} \alpha_{tf,m} \mathcal{N}\big(\mathbf{x}_{tf} \mid \boldsymbol{\mu}_{tf,m}, \Sigma_{tf,m}\big) \tag{5}$$

We use multivariate normal distributions $\mathcal{N}$ with $\{\boldsymbol{\mu}_{tf,m} \in \mathbb{R}^2, \Sigma_{tf,m} \in \mathbb{R}^{2\times2}, \alpha_{tf,m}\}$ being the mean, covariance, and weight of mode m at time tf. At each future time step, the model forecasts M multivariate normal distributions and their corresponding weights, thereby constructing the

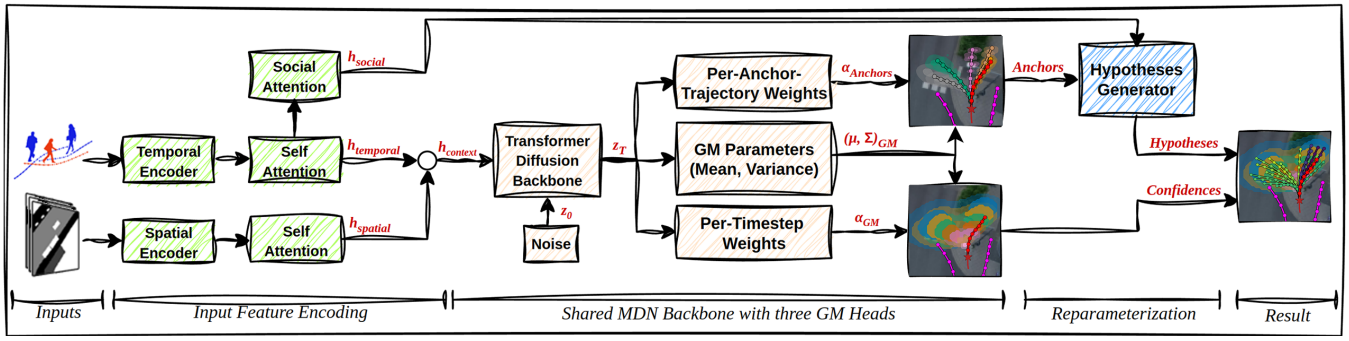

Fig. 1: Architectural overview of DD-MDN. Input encoding blocks are green, probabilistic blocks are yellow, and deterministic ones are blue.

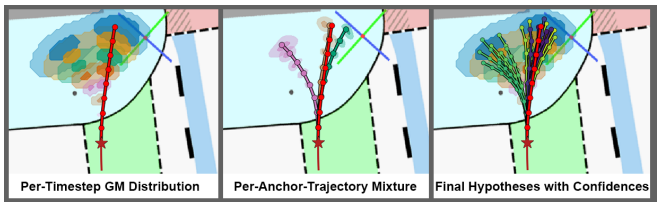

Fig. 2: Internal GM representations, f.l.t.r.: **First:** Per-timestep GM representation in $\mathbb{R}^2$ space as 95- and 68 % mixture CLs. **Second:** Per-anchor-trajectory GM representation in $\mathbb{R}^{2T_{fut}}$ trajectory space visualized by three anchor trajectories and their uncertainties. **Third:** Final discrete K generated hypotheses. GT path is red.

GM Θ^{step}. The second representation is called: per-anchor-trajectory GM Θ^{anchor}, and it is used to generate multiple coherent future anchor trajectories $\widehat{\mathbf{Y}}$ from the same shared mean and covariance parameters $(\boldsymbol{\mu}_{tf,m}, \Sigma_{tf,m})$. Therefore, we form a joint GM in a $2T_{fut}$-dimensional trajectory space and define a set of M future anchor trajectories as $\widehat{\mathcal{Y}}_a = \left[\widehat{\mathbf{Y}}_1, \widehat{\mathbf{Y}}_2, ..., \widehat{\mathbf{Y}}_M\right] \in \mathbb{R}^{2T_{fut}}$. For each mode m in Θ^{anchor}, we stack the shared means and covariances as described in Eq. (6) and Eq. (7).

$$\widehat{\boldsymbol{\mu}}_m = \begin{bmatrix} \boldsymbol{\mu}_{1,m} \\ \vdots \\ \boldsymbol{\mu}_{T_{fut},m} \end{bmatrix} \in \mathbb{R}^{2T_{fut}} \tag{6}$$

$$\widehat{\Sigma}_m = \text{blockdiag}\left(\Sigma_{1,m}, \ldots, \Sigma_{T_{fut},m}\right) \in \mathbb{R}^{2T_{fut} \times 2T_{fut}} \tag{7}$$

We deliberately omit off-diagonal cross-time covariances to keep $\widehat{\Sigma}_m$ tractable; multiple mixture modes instead capture temporal diversity. Furthermore, we assign anchor trajectory weights $\widehat{\alpha}_m$ via a softmax over m. The resulting joint mixture over a position in the joint trajectory space $\widehat{\mathbf{x}} \in \mathbb{R}^{2T_{fut}}$ is:

$$\Theta^{anchor}(\widehat{\mathbf{x}}) = \sum_{m=1}^{M} \widehat{\alpha}_m \, \mathcal{N}\left(\widehat{\mathbf{x}} \mid \widehat{\boldsymbol{\mu}}_m, \widehat{\Sigma}_m\right) \tag{8}$$

The per-anchor-trajectory distribution Θ^{anchor} is represented as a single multimodal GM in the trajectory space, where each mode m represents a discrete future trajectory along with its associated variance. Fig. 3 illustrates the principle. Θ^{step} and Θ^{anchor} share the same core Gaussian parameters $\{\boldsymbol{\mu}_{tf,m}, \Sigma_{tf,m}\}$ predicted by the shared backbone and core parameter head. Initially, the parameter pairs (mean and variance) are uncorrelated. This is established through the GM representations. The per-timestep GM and weights $\alpha_{tf,m}$ connect parameter pairs to form a related GM for each future timestep, a timestep-based clustering. However, there is no correlation between the individual time steps. Therefore, the means in this representation type cannot represent realistic future trajectories over time. That is where the per-anchor-trajectory GM representation and the $\mathbb{R}^{2T_{fut}}$ trajectory space come in. Here, the model is encouraged to make reasonable time-related connections among means across timesteps to generate realistic future trajectories. Those are used as anchor trajectories to derive the final K hypotheses. This dual-GM design yields both trustworthy per-timestep uncertainty estimates and natural, time-consistent future anchor trajectories. Section III-D and Section III-E provide additional details. Finally, the Hypotheses Generator takes Θ^{step} and Θ^{anchor} and processes temporal and social input features to generate K uncertainty-related discrete future trajectory hypotheses using affine reparameterization sampling, further described in Section III-F.

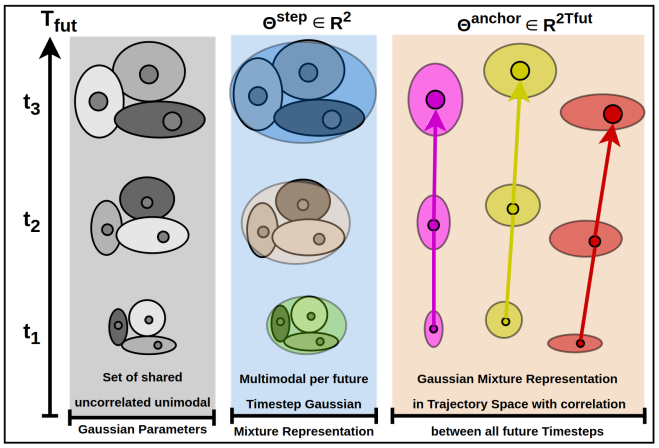

Fig. 3: Schematic illustration of the two GM types, f.l.t.r.: **First:** Uncorrelated Gaussian parameters per-timestep (gray). **Second:** Representation in $\mathbb{R}^2$ space. **Third:** per-anchor-trajectory GM representation in $\mathbb{R}^{2T_{fut}}$ trajectory space.

C. Input Encoding

The environmental context (map data, static objects, and vehicles) is represented as a single-layer occupancy grid centered on the target agent within a 10-meter radius. It distinguishes walkable areas (e.g., pedestrian walks) from inaccessible ones (e.g., buildings, walls). To align image

coordinates $[u, v]$ with world coordinates $[x, y]$, we apply two Coordinate Convolutions (CordConv) [31], one per dimension, linking spatial and temporal features. A small CNN with self-attention processes the resulting three-layer grid into spatial features $\mathbf{h}_{spatial}$. Past trajectories of all pedestrian agents are encoded by an LSTM followed by self-attention, producing temporal features $\mathbf{h}_{temporal}$. Finally, Social-Out-Way Attention [32], an augmented Transformer with exit gates, filters irrelevant or distant neighbors without fixed thresholds. Each head outputs relevance and exit probabilities, pruning uninformative links end-to-end and yielding more accurate, socially compliant forecasts. Passing $\mathbf{h}_{temporal}$ through this module yields the social feature vector $\mathbf{h}_{social}$.

D. Probabilistic Modeling

We aim for uncertainty self-calibration on a per-timestep basis to ensure that the probability density of Θ^{step} reflects the data's genuine aleatoric uncertainty so that we can derive trustworthy and reliable confidence areas, which describe how likely the target agent will reside within at each future timestep tf. Training with the Negative Log-Likelihood (NLL) corresponds to minimising the continuous logarithmic score, a strictly proper scoring rule [33]. Consequently, the network learns both the central tendency and the dispersion of the predictive distribution directly from the data, yielding an aleatorically calibrated forecast without the need for manual tuning of the covariance.

Unimodal Case: A neural network parameterizes mean and covariance as functions of input trajectory $\mathbf{X}^{in}$ and weights W. Minimizing the NLL aligns predicted distributions with GT data. For a single Gaussian, the NLL decomposes as

$$NLL_{tf} = \underbrace{\tfrac{1}{2}\, d_{tf}^2(\mathbf{x}_{tf}, W, \mathbf{X}^{in})}_{\text{error term}} + \underbrace{\tfrac{1}{2}\, \log\det \Sigma_{tf}(W, \mathbf{X}^{in})}_{\text{residual term}}. \tag{9}$$

Here d_{tf}^2 is the Mahalanobis distance of $\mathbf{x}_{tf}$ to the predicted mean. The error term penalizes inaccurate means, while the residual term penalizes excessive covariance, reflecting the calibration–sharpness trade-off [34].

Multimodal Case: Human motion is often multimodal, modeled with a Gaussian mixture:

$$GM(\mathbf{x}_{tf}) = \sum_{m=1}^{M} \alpha_{tf,m}\, \mathcal{N}_{tf,m}(\mathbf{x}_{tf}), \tag{10}$$

$$NLL_{tf} = -\log(GM(\mathbf{x}_{tf})). \tag{11}$$

Unlike the unimodal case, the NLL has no additive split. Components far from $\mathbf{x}_{tf}$ contribute little, resulting in gradients that pull the means $\boldsymbol{\mu}_{tf,m}$ toward the data. Covariances are implicitly regularized: inflating $\Sigma_{tf,m}$ lowers density at the GT point, so diffuse modes contribute negligibly and self-penalize. Mixture weights $\alpha_{tf,m}$ are normalized via softmax, concentrating on components that better match the data. During training, modes specialize, collectively capturing multimodality with calibrated probabilities.

Mode Pruning: Not all M modes are always necessary to represent reliable forecasts. We prune low-weight components via an epoch-dependent threshold $\delta(e)$, increasing from δ_0 to δ_f over E epochs:

$$\delta(e) = \delta_0 + (\delta_f - \delta_0)\, \tfrac{e}{E}, \quad e = 1, \ldots, E. \tag{12}$$

A sigmoid gate $G_m(e)$ filters each weight:

$$G_m(e) = \sigma\Big(\tfrac{\alpha_m - \delta(e)}{\eta(e)}\Big), \quad \dot{\alpha}_m(e) = \tfrac{G_m(e)\,\alpha_m}{\sum_{j=1}^{M} G_j(e)\,\alpha_j}. \tag{13}$$

Here $\eta(e)$ anneals from η_0 to η_f for smooth soft-to-hard gating. Active modes are those with $\dot{\alpha}_m(e) > \delta(e)$, yielding $M^\star$ effective components whose weights renormalize to 1. This allows the model to adapt the mixture complexity to each input, retaining only the essential modes.

E. Unsupervised Transformer Diffusion

Denoising diffusion models learn the gradient of the log-density by corrupting observable GT data with a known Markov noise process and then training a neural network to invert that corruption. We transplant this idea into the parameter space of continuous probability distributions themselves, here using the shared multivariate normal parameters $\mathcal{N}\big(\mathbf{x}_{tf} \mid \boldsymbol{\mu}_{tf,m}, \Sigma_{tf,m}\big)$ and the separated weights $\alpha_{tf,m}$ and $\widehat{\alpha}_m$. The theoretical motivation for this parameter-space approach is to provide a generative prior over the manifold of valid distributions, ensuring global temporal coherence that standard point-estimate MDNs often lack. These parameters lie on a highly structured manifold, specifically the mean space $\mathbb{R}^2$, the simplex for weights, and the Symmetric Positive Definite (SPD) covariance cone.

While direct Mean-Squared Error (MSE) regularization ignores this complex geometry, the iterative denoising process serves as a global regularizer that integrates local corrections with a context-aware score. Validity on the SPD manifold is maintained implicitly: the NLL objective serves as a geometric constraint, whereby degenerate or non-SPD covariance structures lead to a sharp drop in the probability density at the GT point. This triggers a self-penalizing gradient that steers the diffusion backbone back toward valid, stable regions of the parameter space. In our case, the observables are time-series data on future positions, whereas the continuous probability distributions are latent. Since no corresponding GT distribution for Θ^{step} and Θ^{anchor} exists, we cast them as the outputs of a conditional diffusion prior: starting from pure noise, the model is only guided by the NLL of the induced densities w.r.t. the GT trajectory $\mathbf{Y}^{gt}$ and a context feature $\mathbf{h}_{context}$ conditioned through Feature-wise Linear Modulation (FiLM) [35]. For the shared diffusion backbone f_ϕ , let $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be Gaussian noise in a learnable latent space. For $\tau = 1, \ldots, T_{Diff}$ diffusion steps, we iterate as described in Eq. (14).

$$\mathbf{z}_\tau = f_\phi\Big(\underbrace{\mathrm{FiLM}(\mathbf{h}_{context}, \tau)}_{\text{context \& diffusion step}}, \mathbf{z}_{\tau-1}\Big) \tag{14}$$

$\mathbf{h}_{context}$ is the concatenation of $\mathbf{h}_{temporal}$ and $\mathbf{h}_{spatial}$. FiLM modulation injects this context at each diffusion step,

so f_ϕ learns a context-guided generative prior over valid GM parameters. Afterwards, the final latent state $\mathbf{z}_{T_{Diff}}$ branches into three heads to derive the dual mixture representations. Because both distribution representations are latent, optimization relies solely on the per-timestep and per-anchor-trajectory NLLs and their respective weightings (Eq. (15)).

$$\mathcal{L}_{prob} = \lambda_{step} NLL_{step} + \lambda_{anchor} NLL_{anchor} \tag{15}$$

The right balance between λ_{step} and λ_{anchor} is crucial due to the shared core parameters ($\boldsymbol{\mu}_{tf,m}$, $\Sigma_{tf,m}$). λ_{step} must be the main gradient driver; otherwise, per-timestep uncertainty calibration is not guaranteed, and on the other side, λ_{anchor} must be in the right spot not to prevent per-timestep calibration while still having enough gradient influence to establish time-correlated and stable connections between means at different future timesteps to establish natural and realistic anchor-trajectories for the hypotheses generation. Across all experiments, $\lambda_{anchor} = 0.05$ and $\lambda_{step} = 1.0$ best meet the requirements, as evaluated through a parameter sweep.

Back-propagating through the exact log-likelihoods drives the diffusion before generating mixture parameters that place high probability mass on the observed trajectory, both on a per-timestep basis and across the full trajectory, due to the dual mixture approach. This unsupervised, likelihood-based calibration signal enables the usage of the denoising diffusion process for GM parameter prediction without having corresponding GT mixture parameters. The process optimizes itself solely on the basis of future GT positions and their correlations. The parameter manifold is curved and coupled across timestep positions and complete trajectories. Iterative denoising regularizes this geometry: each step blends local corrections with a global, context-aware score, yielding (i) smooth evolution across the forecast horizon, (ii) diverse modes, and (iii) stable covariances. Ablations in Section IV show that diffusion-augmented training improves the model's overall reliability compared to classic MDNs.

F. Deterministic Hypotheses Generation

Hypotheses Sampling: The generated anchor trajectories based on the correlated means of Θ^{anchor} are transformed into a finite set of $\dot{\mathcal{Y}} \in \mathbb{R}^{K \times 2T_{fut}}$ discrete trajectory hypotheses through an affine reparameterization sampling scheme. Each mode m in Θ^{anchor} represents a time-stable mean-trajectory $\widehat{\mathbf{Y}}_m \in \mathbb{R}^{2T_{fut}}$ and a positive-definite block diagonal covariance matrix $\widehat{\Sigma}_m \in \mathbb{R}^{2T_{fut} \times 2T_{fut}}$ together representing a future anchor trajectory and its variance. According to the anchor trajectory, the affine reparameterization is given by Eq. (16).

$$\dot{\mathcal{Y}}_m = \widehat{\mathbf{Y}}_m + \hat{\sigma}_m \mathcal{B}_m \tag{16}$$

Here $\hat{\sigma}_m$ represents the overall variance scale of $\widehat{\Sigma}_m$ derived by the root-mean-square of the diagonal covariance values. In combination with a set of normalized residual trajectories of unit variance $\mathcal{B}_m$, i.e., error patterns, we can derive a set of multiple future trajectory hypotheses $\dot{\mathcal{Y}}_m$ per anchor trajectory. The residual trajectories capture the typical ways a pedestrian might deviate from the central path. This two-step process cleanly separates the shape of likely deviations from their magnitude (mode-specific), yielding diverse, context-aware predictions while preserving full differentiability. A small MLP encoder takes the covariances as input and predicts a mode-specific scaling feature, $\mathbf{h}^m_{scale}$. This feature is concatenated with $\mathbf{h}_{temporal}$ and $\mathbf{h}_{social}$, passed through an MLP decoder, and simultaneously predicts the residuals $\mathcal{B}_m$. We take a mean field and a scalar variance scale that have already been learned from Θ^{anchor}_m and combine them with learning residual patterns guided by the input features. This affine reparameterization can be applied to each mode of Θ^{anchor}, allowing for the generation of any number of hypotheses.

Each active mode contributes at least its mean anchor trajectory. The remaining $K - M^\star$ hypotheses are apportioned in proportion to the pruned weights $\dot{\alpha}_m$ via a largest-remainder Hamilton rule quota q_m, Eq. (17). Thus, the model learns for itself the number of hypotheses each anchor contributes to the final K hypotheses.

$$q_m = 1 + \lfloor r_m \rfloor, \quad r_m = \frac{\hat{\alpha}_m}{\sum_{j=1}^{M^\star} \hat{\alpha}_j} (K - M^\star) \tag{17}$$

The leftover $L = K - \sum_m q_m$ slots are given one-by-one to the modes with the largest fractional remainders $r_m - \lfloor r_m \rfloor$. This guarantees $\sum_m q_m = K$ while maintaining a split as close as possible to the ideal proportional split. The number of contributed hypotheses depends on the model's learned confidence in specific modes, given the current situation.

Hypotheses Loss: To train the hypotheses generator, we combine three complementary objectives. Given K forecasted trajectory hypotheses and the future GT trajectory, the generator loss is composed as described in Eq. (18).

$$\mathcal{L}_{\text{hypo}} = \lambda_{MSE}\, \mathcal{L}_{MSE} + \lambda_{\text{WTA}}\, \mathcal{L}_{\text{WTA}} + \lambda_{\text{Conf}}\, \mathcal{L}_{\text{Conf}} \tag{18}$$

The $\mathcal{L}_{MSE}$ term tends to pull the hypotheses toward the truth. In contrast, the Winner-Takes-It-All (WTA) $\mathcal{L}_{\text{WTA}}$ term focuses on the most accurate candidate by softly emphasizing the best hypothesis with a temperature annealing schedule [36]. The $\mathcal{L}_{\text{Conf}}$ term keeps all hypotheses within a defined confidence region. For every future step tf we compute a log-probability threshold γ_{tf} as the $(1-\beta)$-quantile of Θ^{step}_{tf}. A hypothesis k is penalised at that step only if its log-probability falls below this threshold. We compute the worst deficit across the K hypotheses and average it over batches and time. A hinge ensures that hypotheses lying inside the confidence region incur zero cost. At the same time, the maximization selects the single most offending hypothesis at each batch-timestep pair. This ensures that all hypotheses fall within the desired confidence interval.

Confidence estimation: We assign each forecasted hypothesis a confidence score $C_k \in [0, 1]$ based on its likelihood under Θ^{step} using Monte Carlo (MC) percentiles. At each time tf, we draw I MC samples $\{\mathbf{d}_{tf,i}\}_{i=1}^{I} \sim \Theta^{step}_{tf}$, compute their densities $\rho_{tf,i}$ as well as the hypothesis density $\rho_{tf,k}$ as described in Eq. (19).

$$\rho_{tf,i} = p\big(\mathbf{d}_{tf,i} \mid \Theta^{step}_{tf}\big) \;\; ; \;\; \rho_{tf,k} = p\big(\mathbf{x}_{tf,k} \mid \Theta^{step}_{tf}\big) \tag{19}$$

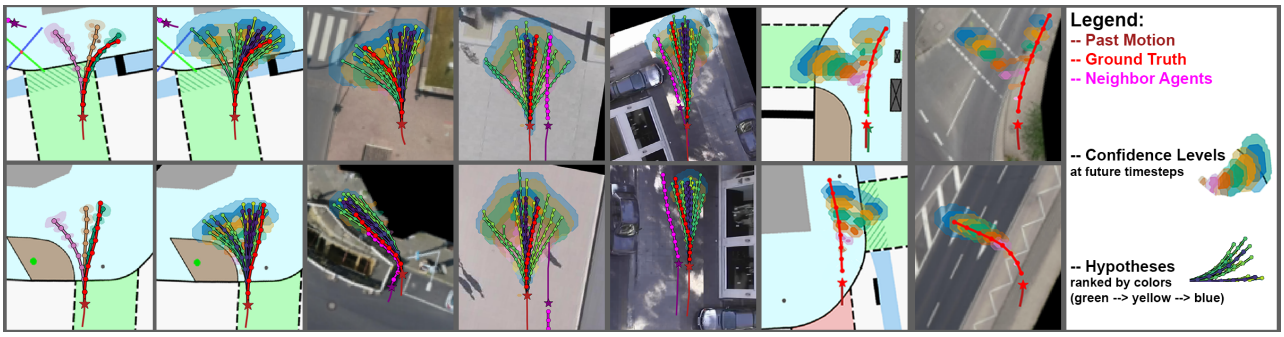

Fig. 4: DD-MDN results in various datasets, f.l.t.r.: (1) IMPTC per-anchor-trajectory GM representation and (2) resulting final K hypotheses with 68- and 95 % CLs, (3-5) inD, SDD, and ETH examples, (6) IMPTC- and (7) inD per-timestep 68- and 95 % isolated CL forecasts.

$$r_{tf,k} = \frac{1}{I}\sum_{i=1}^{I} \mathbf{1}\left[\rho_{tf,i} \geq \rho_{tf,k}\right] \in [0,1] \tag{20}$$

$r_{tf,k}$ is the fraction of MC samples whose density is at least as high as that of the hypothesis. We then estimate the upper-tail percentile of $\rho_{tf,k}$ among the I samples (see Eq. (20)). To convert that percentile into a confidence weight, we quantize it into one of J equal-width bins of size $L = 1/J$. The bin index, Eq. (21), identifies which percentile interval the hypothesis density falls into. We then assign a confidence weight equal to one minus the lower edge of that bin, so that hypotheses falling into lower percentiles (i.e. higher-density samples) receive scores closer to 1. In contrast, outliers in higher percentile bins receive proportionally lower confidence, as shown in Eq. (22). Finally, the overall confidence of hypothesis k is the average per-timestep confidence.

$$j_{tf,k} = \min\Big(\lfloor r_{tf,k}/L \rfloor,\ J-1\Big) \tag{21}$$

$$c_{tf,k} = 1 - j_{tf,k}L \quad \in [0,1] \tag{22}$$

G. Training Details

We train the model on an NVIDIA RTX 4090 and an AMD Ryzen 9950X using the AdamW optimizer, a cosine learning rate scheduler, and dynamic input-horizon scaling. The architecture is highly compact, comprising only 4.5 MB of raw weights, representing a significant reduction in footprint compared with models such as LED (60.0 MB) or MoFlow (60.2 MB). Empirical analysis shows an average inference latency of 15.5 ms at $B = 64$ and 21.9 ms at $B = 128$. Under FP32 precision, the model maintains a low memory footprint of 818.7 MB ($B = 64$), including the fixed CUDA context. Using mixed-precision (FP16) or quantized (FP8) arithmetic can further reduce memory requirements by 40 % to 60 %, thereby improving edge deployment suitability. All parameters and hyperparameters are available in the public repository. Because of the model's probabilistic nature, we report the average of three training runs.

IV. Experiments

A. Experimental Setup

We evaluate our framework on the ETH/UCY [37], SDD [38], and inD [39] benchmarks, providing pedestrian trajectories and scene context. We follow the ETH/UCY leave-one-out protocol [18] and use the SDD/inD splits from [14]. All datasets utilize a 2.5 Hz sampling rate, with 8 observed frames (3.2 s) and a 12-frame horizon (4.8 s). Due to the limited size of standard benchmarks, we also utilize the full SDD, inD, and IMPTC [40], [41] datasets (≥40 000 samples), denoted as SDD*, inD*, and IMPTC*, for robust reliability evaluation.

For accuracy, we report Best-of-K Average and Final Displacement Errors ($min_k ADE$, $min_k FDE$) in meters $min_k ADE$ measures the mean L_2 distance of the best forecast $\dot{\mathbf{Y}}_k$ to the ground truth $\mathbf{Y}^{gt}$, while $min_k FDE$ evaluates only the endpoints. Forecast reliability is assessed via R_{avg} and R_{min} [30], comparing predicted confidence levels $1-\beta(\mathbf{x})$ against observed frequencies $f_o(1-\beta)$. Scores are reported as percentages (100% = perfect calibration) and visualized with Q–Q plots.

B. Evaluation Results

Method	ETH	HOTEL	UNIV	ZARA1	ZARA2	**AVG ↓**
Trajectron	0.61/1.03	0.20/0.28	0.30/0.55	0.24/0.41	0.18/0.32	0.31/0.52
STAGE	0.44/0.77	0.28/0.50	0.40/0.77	0.30/0.56	0.20/0.37	0.32/0.59
TUTR	0.40/0.61	**0.11**/0.18	0.23/0.42	0.18/0.34	0.13/0.25	0.21/0.36
LED	0.39/0.58	**0.11/0.17**	0.26/0.43	0.18/ **0.26**	0.13/ 0.22	0.21/0.33
SingTraj	0.35/0.42	0.13/0.19	0.25/0.44	0.19/0.32	0.15/0.25	0.21/0.32
MNRF	**0.26/0.37**	**0.11/0.17**	0.28/0.49	0.17/0.30	0.14/0.25	**0.19**/0.32
MoFlow	0.40/0.57	**0.11/0.17**	0.23/0.39	**0.15/0.26**	**0.12**/0.22	0.20/0.32
Ours	0.30/0.46	0.13/**0.17**	0.23/0.41	0.16/0.28	**0.12/0.21**	**0.19/0.31**

Table. I: Stochastic evaluation on the ETH/UCY benchmark with $min_k ADE/FDE$ (K=20) in meters. **Bold** represents the best result, underline represents the second best result.

We compare our method with the SOTA approaches in the established ETH/UCY, SDD, and inD benchmarks. Regarding ETH/UCY, the overall progress has reached a plateau since 2023; nonetheless, our model slightly outperforms the best available methods with a $min_k ADE/FDE$ of 0.19/0.31 m, as shown in Tab. I. Concerning practical relevance, our method sets a new benchmark for momentary observation (shortened input horizon of two frames), outperforming the current SOTA SingTraj [25] by over 20 % (0.20/0.32 m vs. 0.25/0.40 m), see Tab. II. The performance of most other methods drops significantly in this scenario. DD-MDN, on the other hand, is capable of generating highly accurate hypotheses with an input length of just two frames (position and velocity), which is very important for critical traffic situations with short observation and reaction times.

Dataset	MID	EigenTraj	SingTraj	**Ours**
ETH	0.63/1.05	0.46/0.76	0.45/0.67	**0.32/0.51**
HOTEL	0.29/0.49	0.17/0.28	0.18/0.29	**0.13/0.18**
UNIV	0.30/0.56	0.25/0.44	**0.24**/0.43	**0.24/0.42**
ZARA1	0.30/0.56	0.19/0.35	0.19/0.33	**0.17/0.28**
ZARA2	0.22/0.40	0.15/0.27	0.17/0.28	**0.12/0.21**
AVG ↓	0.35/0.61	0.25/0.42	0.25/0.40	**0.20/0.32**

Table. II: Momentary observation evaluation on ETH/UCY benchmark with $min_k ADE/FDE$ (K=20) in meters.

On the SDD benchmark in Tab. III, using the full input horizon length of 8 past timesteps, marked as (@8), DD-MDN matches the performance of the best available methods regarding $min_k ADE$, but misses a top spot for $min_k FDE$. Regarding momentary observation, marked as (@2), no other method has reported results for SDD yet; nonetheless, our method confirms the strong performance already presented in the ETH/UCY momentary evaluation, still reporting the second-best $min_k ADE$ after just two input frames.

The inD benchmark is less frequently used for evaluation. Here, as listed on the right side in Tab. III, DD-MDN significantly outperforms the SOTA, delivering a 46% and 38% improvement in $min_k ADE/FDE$ and confirming the excellent momentary observation performance.

The robustness of DD-MDN with respect to variations in input length is based on our probabilistic modeling approach, combined with ego-coordinates and the use of a dynamic input trajectory length during training. This effectively quadruples the input training pattern, thereby enhancing the fine-tuning of the mixture parameters and improving the distinction between input data clusters.

SDD (px)		**inD (m)**	
Method	ADE/FDE ↓	Method	ADE/FDE ↓
SocialVAE	8.88/14.81	Y-Net	0.55/0.93
HighGraph	7.81/11.09	AC-VRNN	0.42/0.80
LMTraj	7.80/**10.10**	Agentformer	0.57/0.87
MNRF	**7.20**/11.29	Goar-SAR	0.44/0.70
MoFlow	7.50/11.96	Di-Long	0.37/0.59
Ours (@8)	**7.19**/11.82	**Ours (@8)**	**0.20/0.36**
Ours (@2)	7.42/12.43	**Ours (@2)**	0.21/0.38

Table. III: Stochastic evaluation with $min_k ADE/FDE$ (K=20). **Left:** SDD benchmark in pixels. **Right:** inD benchmark in meters.

Despite delivering excellent positional accuracy, DD-MDN is capable of handling uncertainty by providing a probability score for each discrete forecast, representing the model's confidence. Due to the probabilistic modeling and the use of the NLL as part of the training loss, the model aims to achieve the best possible calibration of aleatoric uncertainty. In Tab. IV (A - Comparison), we evaluate DD-MDNs' uncertainty calibration performance regarding the *Reliability* metric compared to other SOTA methods. For methods lacking native density estimation, we utilized Kernel Density Estimation (KDE) with 1×10^4 samples as an approximation. Our method's approach is beneficial, resulting in the highest *Reliability* scores across all benchmarks; specifically, the *Reliability* scores of DD-MDNs are the most accurate and trustworthy. In the upper row of Fig. 5, the Q-Q plots for the Zara01 subtest are visualized. The results for the other ETH/UCY subtests are similar. Accuracy-focused models tend toward overconfidence with overly tight distributions. While not reaching perfect calibration, DD-MDN outperforms current SOTA methods by a significant margin.

A - Comparison				**B - Ablation**			
Set	Method	$\mathbf{R}_{avg}$	$\mathbf{R}_{min}$	Set	Method	$\mathbf{R}_{avg}$	$\mathbf{R}_{min}$
ETH	LED	84.4	71.1	**IMPTC***	Linear	97.9	92.9
	SingTraj	85.9	61.9		MLP	97.6	90.8
	Ours	**91.8**	**76.4**		Diffusion	**98.3**	**95.1**
SDD	MID	83.4	72.8	**inD***	Linear	95.7	91.7
	MoFlow	87.6	77.9		MLP	98.3	94.4
	Ours	**94.3**	**84.6**		Diffusion	**99.2**	**98.0**
inD	Agentformer	85.8	77.7	**SDD***	Linear	92.4	80.2
	Di-Long	89.2	84.3		MLP	92.1	83.3
	Ours	**98.3**	**94.8**		Diffusion	**93.6**	**83.8**

Table. IV: Uncertainty calibration evaluation with *Reliability* (R_{avg}, R_{min}) in %. **Left (A)**: Benchmark comparison with SOTA methods. **Right (B):** Ablations on full datasets.

C. Ablation Studies

To better demonstrate our methods' uncertainty handling and self-calibration capabilities, we took the full SDD*, inD*, and IMPTC* datasets and extracted train-eval-test splits, each with (>= 40 K samples). The right side of Tab. IV (B - Ablation) represents the results obtained using identical training parameters for all three complete datasets. Provided that sufficient training and test data are available, DD-MDN achieves near-perfect average- and strong minimum *Reliability* scores for the inD* and IMPTC* datasets. The bottom row of Fig. 5 provides the corresponding Q-Q calibration plots. The results for SDD* show a slight decline, suggesting that training may have ended prematurely.

Moreover, in Tab. IV (B - Abalation), we also demonstrate the positive influence of the diffusion-based backbone. Compared to classic MDN modeling backbones (Linear Layers or MLPs), the diffusion backbone slightly improves the average and significantly boosts the minimum *Reliability* scores across all three complete datasets, demonstrating the architecture's strong ability to calibrate uncertainty. Fig. 4 illustrates multiple forecasting examples with varying representation levels from all the datasets and benchmarks used.

V. Conclusion

This work introduced a diffusion-based dual-MDN model for HTF. It predicts calibrated residence areas and diverse trajectories, offering trustworthy forecasts and uncertainty estimates for downstream tasks. DD-MDN achieves superior accuracy compared to SOTA models, with significant gains at short observation periods, and sets a benchmark for uncertainty calibration in HTF. We further showed how denoising diffusion enhances MDNs without mixture ground-truth data by leveraging an unsupervised likelihood-based calibration

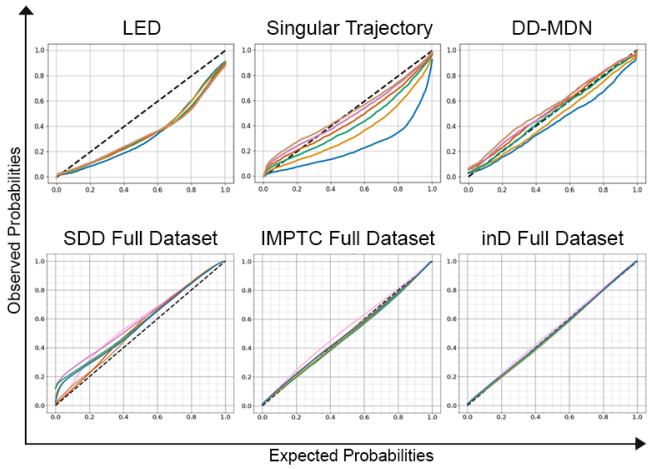

Fig. 5: **Upper Row:** Calibration plots of SOTA methods (LED, SingularTraj) and DD-MDN for ETH/UCY Zara01 benchmark. **Lower Row:** Calibration plots for the SDD*, IMPTC* and inD* full datasets. Colors represent future timesteps (0.8, 1.6, ..., 4.8 s), the black dotted diagonal represents perfect calibration.

signal. Remaining limitations are evident in crowded scenes, where forecasts can be underconfident and broad. Future work will incorporate richer environmental contexts and agent forecasts to improve interaction modeling.

REFERENCES

[1] C. Jiang and et al., "Motiondiffuser: Controllable multi-agent motion prediction using diffusion," *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[2] H. Girase and C. Choi, "Loki: Long term and key intentions for trajectory prediction," *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[3] A. Rudenko and L. Palmieri, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, 2020.

[4] B.-J. Lee and B.-T. Zhang, "Robust human following by deep bayesian trajectory prediction for home service robots," *Conference on Robotics and Automation*, 2018.

[5] A. Nayak and A. Eskandarian, "Uncertainty estimation of pedestrian future trajectory using bayesian approximation," *IEEE Open Journal of Intelligent Transportation Systems*, 2022.

[6] A. Eskandarian and C. Wu, "Research advances and challenges of autonomous and connected ground vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[7] J. Eilbrecht and O. Stursberg, "Model-predictive planning for autonomous vehicles anticipating intentions of vrus by artificial neural networks," *IEEE Symposium on Computational Intelligence*, 2017.

[8] J. Kruger and D. Dunning, "Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments," *Journal of Personality and Social Psychology*, 1999.

[9] C. Guo and G. Pleiss, "On calibration of modern neural networks," *IEEE International Conference on Machine Learning*, 2017.

[10] M. Hetzel and B. Sick, "Are we pursuing the right objective? A survey in human trajectory prediction," *Zenodo*, vol. https://doi.org/10.5281/zenodo.18605645, 2026.

[11] H. Renhao and P. Maurice, "Multimodal trajectory prediction: A survey," *ArXiv*, vol. abs/2302.10463, 2023.

[12] F. Zheng and K. Kun, "Summary and reflections on pedestrian trajectory prediction in the field of autonomous driving," *IEEE Transactions on Intelligent Vehicles*, 2024.

[13] A. Alahi and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[14] K. Mangalam and J. Malik, "From goals, waypoints and paths to long term human trajectory forecasting," *IEEE International Conference on Computer Vision*, 2021.

[15] J. Liang and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[16] A. Mohamed and C. Claudel, "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[17] C. Xu and S. Chen, "Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning," *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[18] A. Gupta and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[19] A. Sadeghian and A. Sadeghian, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[20] T. Salzmann and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," *IEEE European Conference on Computer Vision*, 2020.

[21] M. Karttikeya and A. Shreya, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," *IEEE European Conference on Computer Vision*, 2020.

[22] L. Rongqin and L. Xia, "Stglow: A flow-based generative framework with dual graphormer for pedestrian trajectory prediction," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[23] T. Maeda and N. Ukita, "Fast inference and update of probabilistic density estimation on trajectory prediction," *IEEE International Conference on Computer Vision*, 2023.

[24] W. Mao and Y. Wang, "Leapfrog diffusion model for stochastic trajectory prediction," *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[25] I. Bae and H. Jeon, "Singulartrajectory: Universal trajectory predictor using diffusion model," *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

[26] F. Yuxiang and L. Renjie, "Moflow: One-step flow matching for human trajectory forecasting via implicit maximum likelihood estimation based distillation," *Computer Vision and Pattern Recognition*, 2025.

[27] S. Malla and C. Choi, "Social-stage: Spatio-temporal multi-modal future trajectory forecast," *IEEE International Conference on Robotics and Automation*, 2021.

[28] S. Liushuai and W. Le, "Trajectory unified transformer for pedestrian trajectory prediction," *IEEE International Conference on Computer Vision*, 2023.

[29] B. Ivanovic and P. Marco, "Propagating state uncertainty through trajectory forecasting," *International Conference on Robotics and Automation*, 2022.

[30] M. Hetzel and B. Sick, "Reliable probabilistic human trajectory prediction for autonomous applications," *IEEE European Conference on Computer Vision*, 2024.

[31] R. Liu and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution," *ArXiv*, vol. abs/1807.03247, 2018.

[32] X. Wei and X. Lei, "Trajectory prediction via proposal guided transformer with out way attention," *Nature - Scientific Reports*, 2025.

[33] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *American Statistical Association*, 2007.

[34] T. Gneiting, "Probabilistic forecasts, calibration and sharpness," *Royal Statistical Society: Series B*, 2007.

[35] P. Ethan and C. Aaron Courville, "Film: Visual reasoning with a general conditioning layer," *AAAI Conference on Artificial Intelligence*, 2017.

[36] X. Yihong and M. Cord, "Annealed winner-takes-all for motion forecasting," *ArXiv*, vol. abs/2409.11172, 2024.

[37] S. Pellegrini and L. V. Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," *IEEE International Conference on Computer Vision*, 2009.

[38] A. Robicquet and A. Sadeghian, "Learning social etiquette: Human trajectory understanding in crowded scenes," *IEEE European Conference on Computer Vision*, 2016.

[39] J. Bock and R. Krajewski, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," *IEEE Intelligent Vehicles Symposium*, 2020.

[40] M. Hetzel and B. Sick, "The imptc dataset: An infrastructural multi-person trajectory and context dataset," *IEEE Intelligent Vehicles Symposium*, 2023.

[41] M. Hetzel and H. Reichert, "Smart infrastructure: A research junction," *IEEE International Smart Cities Conference*, 2021.