Improving Transferability for Cross-Domain Trajectory Prediction via Neural Stochastic Differential Equation

Daehee Park, Jaewoo Jeong, Kuk-Jin Yoon

Visual Intelligence Lab., KAIST, Korea
{bag2824, jeong207, kjyoon}@kaist.ac.kr

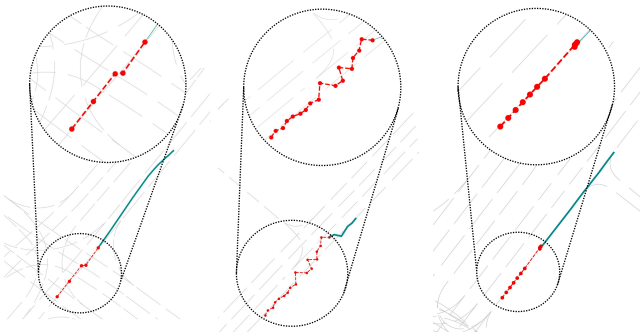
Abstract

Multi-agent trajectory prediction is crucial for various practical applications, spurring the construction of many large-scale trajectory datasets, including vehicles and pedestrians. However, discrepancies exist among datasets due to external factors and data acquisition strategies. External factors include geographical differences and driving styles, while data acquisition strategies include data acquisition rate, history/prediction length, and detector/tracker error. Consequently, the proficient performance of models trained on large-scale datasets has limited transferability on other small-size datasets, bounding the utilization of existing large-scale datasets. To address this limitation, we propose a method based on continuous and stochastic representations of Neural Stochastic Differential Equations (NSDE) for alleviating discrepancies due to data acquisition strategy. We utilize the benefits of continuous representation for handling arbitrary time steps and the use of stochastic representation for handling detector/tracker errors. Additionally, we propose a dataset-specific diffusion network and its training framework to handle dataset-specific detection/tracking errors. The effectiveness of our method is validated against state-of-the-art trajectory prediction models on the popular benchmark datasets: nuScenes, Argoverse, Lyft, INTERACTION, and Waymo Open Motion Dataset (WOMD). Improvement in performance gain on various source and target dataset configurations shows the generalized competence of our approach in addressing cross-dataset discrepancies.

Introduction

Trajectory prediction stands as one of the most crucial challenges to corroborate the safety of autonomous driving systems. Its objective of predicting future trajectories allows autonomous agents to respond optimally to actively changing environments. As a response, a number of large-scale trajectory datasets such as nuScenes, Argoverse, WOMD, Lyft, INTERACTION, and TrajNet++ have been established (Caesar et al. 2020; Chang et al. 2019; Zhan et al. 2019; Houston et al. 2021; Ettinger et al. 2021; Kothari, Kreiss, and Alahi 2021) to pursue a data-driven approach towards constructing a reliable motion forecasting system (Li et al. 2021; Tang et al. 2021; Bae and Jeon 2023; Wu et al. 2023; Ge, Song, and Huang 2023; Shi et al. 2022; Liang et al. 2021).

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



(a) nuScenes (b) Argoverse (c) WOMD

Figure 1: Unique uncertainty manifested across each trajectory prediction dataset due to discrepancy in data acquisition strategy. The red dotted line and darkgreen solid line represent past and future trajectories. The two main sources of discrepancy are time step configuration difference and tracklet noise. Tracklet errors are uniquely shown as lateral position error in nuScenes, ID switch in Argoverse, and longitudinal position error in WOMD, all of which our framework handles in a dataset-wise exclusive manner.

	History (s)	Prediction horizon (s)	Frequency (Hz)	Training size
nuScenes	2	6	2	30k
Argoverse	2	3	10	200k
WOMD	1	8	10	500k
TrajNet++	3.2	4.8	2.5	250k

Table 1: Various temporal configurations during data acquisition across trajectory datasets. These discrepancies - 1) past/future time length and 2) data acquisition rate - severely limit cross-dataset transferability.

One well-known issue with data-driven models is their limited performance when discrepancies in data distributions are manifested between training and test data. Therefore, to construct a trajectory prediction system on a specific environment, the optimal way is to collect data from that environment. However, recent models require abundant data for optimal performance, which require a cumbersome pro-

cess of acquiring such an amount of data. In that sense, adequate utilization of existing large-scale datasets grants an advantage in circumventing this hurdle. Recent approaches have attempted to overcome such challenge by proposing domain adaptation (Xu et al. 2022; Wang et al. 2022b) or increasing model generalizability via multi-source dataset training (Wang et al. 2022a). Compared to these efforts in handling domain gaps, the dataset-specific discrepancies caused by disparity between each data acquisition strategy have been excluded from being considered as a domain gap and have been less visited. Our work shows that adequate handling of these dataset-specific discrepancies unlocks a collective potential from cross-dataset motion patterns.

In doing so, we focus on two representative distinctions across datasets. First is the time step configuration difference, including observed/predicted time lengths and sampling frequencies as shown in Tab. 1. This results in the discrepancy of feature manifold of input/output trajectory in the feature space. For instance, a model trained on the WOMD dataset, which is to predict 8 seconds of future from 1 second of past with 10Hz, learns a mapping function between the past 1-second motion feature and the future 8-second motion feature. However, when evaluating this model on the nuScenes dataset which involves predicting 6 seconds into the future from 2 seconds of observed data in 2Hz, the model struggles to map past trajectory features to future ones accurately.

Secondly, trajectory datasets are obtained by detecting and tracking surrounding agents from the sensor data taken from the ego-agent. As a result, the tracked results (tracklets) are prone to both sensor noise and also inaccurate detection and tracking results (Saleh et al. 2021; Park and Park 2020), and it adversely affects prediction performance (Weng, Ivanovic, and Pavone 2022). Moreover, each dataset manifests unique tendencies of tracklet errors. It is because they use different types of sensors and detector/tracker configurations in the acquisition process. Their unique tendencies of tracklet errors are shown in Fig. 1. The tracklet noise is also influenced by the time step configuration, for different sampling rates exhibit unique noise patterns. Namely, tracklet noise tends to be more severe with smaller Δt, as shown in Fig. 1, where Argoverse has more severe tracklet noise than nuScenes with the same past length.

To address these disparities holistically, we adapt the continuous and stochastic representation of Neural Stochastic Differential Equation (NSDE). Rather than dealing with time series data discretely as conventional approaches, we leverage NSDE to handle time series data in a continuous space. Additionally, we show the capability of stochastic representation in handling the tracklet errors. Specifically, we propose a dataset-specific diffusion network of NSDE and its training method to enhance robustness against dataset-specific tracklet errors. Our contributions are summarized as follows:

- We utilize a continuous representation of NSDE for trajectory prediction to diminish internal discrepancies across datasets collected in arbitrary temporal configurations.
- We propose a framework of dataset-specific diffusion network and its training method to handle unique tracklet errors across datasets.
- The proposed methods are validated against state-of-the-art prediction methods including regression-based and goal-conditioned method, the two mainstreams of trajectory prediction methodology.
- We validate our methods across five datasets: nuScenes, Argoverse, WOMD, Lyft, and INTERACTION, and show consistent improvement in prediction accuracy with state-of-the-art prediction models.

Related Works

Trajectory Prediction

Trajectory prediction involves predicting the future paths of road agents based on observed past trajectories and environmental information, such as HD maps (Wang et al. 2023; Park et al. 2022).With its increasing interest, a number of large-scale trajectory datasets have been established (Kothari, Kreiss, and Alahi 2021; Malinin et al. 2021). These datasets acquire trajectories by detecting and tracking surrounding agents using sensor input installed on ego-agent. HD map information can be obtained from pre-built HD maps or derived from sensor data (Hu et al. 2023). The introduction of large-scale datasets has resulted in improved performance of data-driven trajectory prediction models. Various methods have been proposed to capture agent interactions or better relationship between HD maps (Meng et al. 2022; Salzmann et al. 2020). The methodology for motion forecasting based on these datasets' patterns could be broadly classified into two categories: regression-based and goal prediction-based models. Regression-based models predict the entire trajectory at once, while goal prediction-based models initially predict the endpoints, followed by conditional generation of motion path for each end points. However, despite the rapid advancements in the past few years, prediction in cross-domain scenarios remains relatively underexplored, as all these methods have been individually trained and evaluated on each pre-existing large-scale dataset.

Cross-Domain Trajectory Prediction

Recent research has highlighted the presence of domain discrepancies among various trajectory datasets (Gilles et al. 2022). Analyzing datasets such as nuScenes, Argoverse, Interaction, and Shift, it has been confirmed that transferability between datasets is limited. From a general domain adaptation perspective, approaches have been proposed to address such discrepancies (Xu et al. 2022; Wang et al. 2022a,b). Besides, trajectory datasets exhibit discrepancies due to various factors. For instance, geographical (external) factors can lead to variations in driving environments and agent density, resulting in different driving patterns. To tackle such discrepancies related to road structure curvature, one method (Ye, Zhou, and Wang 2023) proposed a domain normalization technique using Frenet coordinates. However, the methods mentioned above do not consider discrepancies arising from differenct data acquisition strategies including varing time step configuration and tracklet errors. While these domain adaptation papers all set up cross-domain experiments, they restricted the time steps to an overlap of all dataset time steps. For example, in (Gilles et al. 2022), they used a common time

step configuration of 1 second past and 3 seconds future, which is shared by the datasets. Additionally, although they acknowledged that detection/tracking errors during dataset collection could affect transferability (Ivanovic et al. 2022; Weng et al. 2022), the different tendencies of error in cross-domain environments are yet to be addressed.

Neural Differential Equation (NDE)

The proposal of Neural Ordinary Differential Equations (NODE) (Chen et al. 2018) has made significant strides in applying continuous representation to time series data, making it intuitive for various tasks like predicting continuous functions (Anumasa and Srijith 2022; Norcliffe et al. 2020) or other applications that need continuous time series representation (Cao et al. 2023; Park et al. 2021). Neural ODEs have been employed in encoder-decoder structures and have been used to represent the latent space of entire time series data in continuous form (Qian, Kacprzyk, and van der Schaar 2022). Therefore, motion forecasting has been deemed as an epitome of a pattern recognition task solvable via NODEs for its temporally coordinated time series structure. The first work to utilize NODEs for motion forecasting was social ODE (Wen, Wang, and Metaxas 2022) which applied Neural ODEs to pedestrian trajectory prediction to enable interaction modeling. Moreover, the MTP-GO (Westny et al. 2023) constructed a graph-based NODE model for trajectory prediction. Nevertheless, these prior works have not fully leveraged the continuous characteristics of neural ODEs and lack the incorporation of stochastic nature of NSDE modeling, thereby our method substantially differs from previous frameworks based on NODEs.

Method

Preliminaries

Problem Definition Given N agents, a position of a road agent $n \in \{1, ..., N\}$ at a specific time t can be denoted as $\mathbf{x}_t^n$ for the past, and $\mathbf{y}_t^n$ for the future. In general, trajectory prediction aims to predict future trajectory $\mathrm{Y} = \{ \mathbf{y}_{\Delta t}^n, ..., \mathbf{y}_{T_f}^n \}$ from map information M (optional) and observed history trajectory $\mathrm{X} = \{ \mathbf{x}_{-T_p}^n, ..., \mathbf{x}_{-\Delta t}^n, \mathbf{x}_0^n \}$ with fixed time step Δt, history length T_p, and prediction horizon T_f. In our problem definition, we assume that we have large-scale source dataset $\{\mathrm{X}^{sr}, \mathrm{Y}^{sr}\}$ and small-scale target dataset $\{\mathrm{X}^{tg}, \mathrm{Y}^{tg}\}$. Because each dataset has own time step configuration (T_p, T_f, and Δt), we design a model that can handle arbitrary time-step sampled trajectory $\mathrm{Y} = \{\mathbf{y}_t^n\}_{t \in (0, T_F]}$ and $\mathrm{X} = \{\mathbf{x}_t^n\}_{t \in [-T_P, 0]}$ where T_F and T_P are maximum values of prediction horizon and history length across datasets. From now, we omit the superscript n for simplicity.

Neural Stochastic Differential Equation Neural Ordinary Differential Equation (NODE) is an approach that models the derivative of hidden state $\mathbf{h}_t$ employing neural networks to model the transition of features over time. Neural Stochastic Differential Equation (NSDE) introduces stochasticity by incorporating a term resembling Brownian motion into the transition of the hidden state (Li et al. 2020; Tzen and Raginsky 2019). This can be represented as follows:

$$d\mathbf{h}_t = f(\mathbf{h}_t, t)dt + g(\mathbf{h}_t, t)dW_t$$

Here, W represents the standard Brownian motion, while f and g respectively denote the drift and diffusion functions and are parametrized by neural networks. The stochastic noise term acts as a regularizer, mitigating perturbations present in the data. With the above derivatives, we can get a hidden state at a specific time t with initial value problem (IVP) solvers.

Thanks to its continuous nature across time, NDE is known as effective for handling irregularly sampled time series data (Anumasa and Srijith 2022; Kidger et al. 2020). Therefore, we use NSDE to encode and decode temporal trajectory, which is originally performed with discrete networks like transformer (Vaswani et al. 2017), or LSTM (Hochreiter and Schmidhuber 1997) in previous methods.

Proposed Framework

Modeling Time-Wise Continuous Latent Following conventions in both NDE and trajectory prediction, our model follows an encoder-decoder (sequence-to-sequence) structure as shown in Fig. 2. At first, we encode past trajectories of agents with SDE-GRU. We adopt ODE-RNN structure to handle incoming irregularly sampled data, where ODE is replaced with SDE. When the input positions is not observed at a time stamp, the latent is continuously translated via NSDE. If the agent position at time t ($\mathbf{x}_t$) is observed, the latent vector ($\mathbf{h}_t$) is updated using encoded incoming data ($\mathbf{h}_{\mathrm{x}_t}$) via GRU following (De Brouwer et al. 2019; Rubanova, Chen, and Duvenaud 2019). How to obtain next step latent from current input and latent is as:

$$\begin{aligned} d\mathbf{h}_t &= f(\mathbf{h}_t, t)dt + g(\mathbf{h}_t, t)dW_t \\ \mathbf{h}'_{t+\Delta t} &= \mathbf{h}_t + f(\mathbf{h}_t, t)\Delta t + g(\mathbf{h}_t, t)\sqrt{\Delta t}W_t \\ \mathbf{h}_{t+\Delta t} &= \mathrm{GRU}(\mathbf{h}'_{t+\Delta t}, \mathbf{h}_{\mathrm{x}_t}) \end{aligned} \tag{1}$$

Here, GRU cell is represented as:

$$\begin{aligned} \mathbf{r}_t &= \sigma(W_r(\mathbf{h}'_{t+\Delta t} \oplus \mathbf{h}_{\mathrm{x}_t})) + \mathbf{b}_r) \\ \mathbf{z}_t &= \sigma(W_z(\mathbf{h}'_{t+\Delta t} \oplus \mathbf{h}_{\mathrm{x}_t})) + \mathbf{b}_z) \\ \mathbf{g}_t &= \tanh(W_g((\mathbf{r}_t \odot \mathbf{h}'_{t+\Delta t}) \oplus \mathbf{h}_{\mathrm{x}_t})) + \mathbf{b}_g) \\ \mathbf{h}_{t+\Delta t} &= \mathbf{z}_t \odot \mathbf{h}_t + (1 - \mathbf{z}_t) \odot \mathbf{g}_t \end{aligned} \tag{2}$$

where $\mathbf{r}_t, \mathbf{z}_t, \mathbf{g}_t$ correspond to reset gate, update gate, update vector and $\oplus, \odot$ correspond to concatenation, element-wise product. Here, we omit the superscript *past* for Eqs. 1, 2 for simplicity. Then, integrating latent feature from $-T_p$ to 0, we get a latent feature of each agent at current time step ($t = 0$):

$$\begin{aligned} \mathbf{h}_0^{past} &= \int_{-T_P}^{0} \mathrm{GRU}(\mathrm{SDEsolve}(\mathbf{h}_t^{past}, t), \mathbf{h}_{\mathrm{x}_t})dt \\ \mathbf{h}_0^{fut} &= \mathbb{E}(\mathbf{h}_0^{past}, \mathcal{M}) \end{aligned} \tag{3}$$

After encoding past trajectory as a single feature per agent, remaining part of encoder $\mathbb{E}$ is performed such as encoding with map information $\mathcal{M}$.

In case of decoder, it has different network design depending on whether the base model is regression-based model or

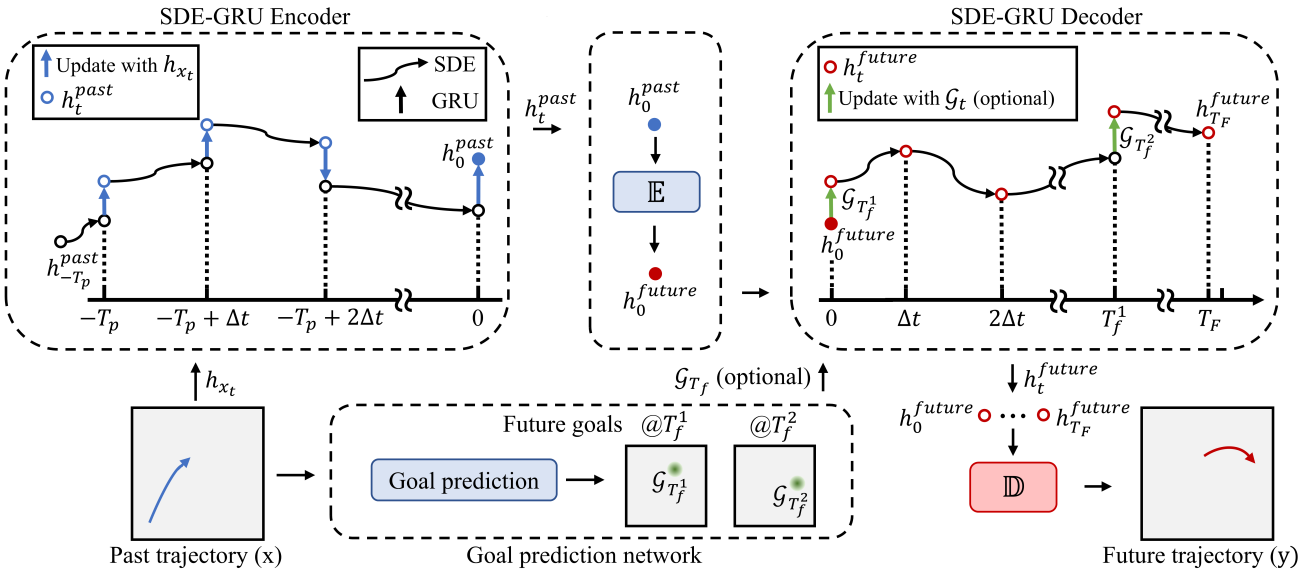


Figure 2: Overall network architecture of the proposed SDE-GRU-based encoder-decoder model. Agent features are extracted from the input trajectory for all observed time steps $\mathbf{h}_{x_t}$. SDE-GRU encoder then integrates the learnable parameter $h_{-T_p}^{past}$ from $-T_p$ to 0 along with feature updates via NSDE at unobserved time steps and GRU gating at observed past trajectory time steps. Followed by an additional encoding operation with encoder $\mathbb{E}$, the SDE decoder similarly integrates the encoded feature from time 0 to T_F. MLP decoder $\mathbb{D}$ then predicts the corresponding future motion.

goal-conditioned model. Unlike the past feature which needs to be updated as data coming as t passes, there is no incoming data for future decoding. Therefore, in regression-based model, hidden state at future time step can be obtained by vanilla SDE solver without GRU update. However, in case of goal-conditioned method, we propose a multi-scale-goal updating method as depicted in the right part of Fig 2. The goal-conditioned decoder predicts trajectory both from $\mathbf{h}_0$ and goal feature. Goal is predicted at the last time step for each dataset configuration: $\boldsymbol{\mathcal{G}}_{T_f^1}, \boldsymbol{\mathcal{G}}_{T_f^2}$. Here, T_f^1 is a time step which is smaller one between $\{ T_f^{sr}, T_f^{tg} \}$, and T_f^2 is the larger one which is identical with T_F. To adopt this multi-scale goal-conditioned, we additionally utilize SDE-RNN which can be represented as:

$$\mathbf{h}_{t\in(0,T_F]} = \begin{cases} \text{SDEsolve}(\text{GRU}(\mathbf{h}_0, \boldsymbol{\mathcal{G}}_{T_f^1})) & 0 < t \leq T_f^1 \\ \text{SDEsolve}(\text{GRU}(\mathbf{h}_{T_f^1}, \boldsymbol{\mathcal{G}}_{T_f^2})) & T_f^1 < t \leq T_F \end{cases} \tag{4}$$

Finally, a mlp decoder $\mathbb{D}$ is utilized to decode future position $\hat{\mathbf{Y}} = \{\mathbf{x}_t\}_{t\in(0,T_F]}$ from hidden state $\{\mathbf{h}_t\}_{t\in(0,T_F]}$.

Handling Tracklet Uncertainty As elaborated in previous sections, the tendencies of tracklet error are unique across datasets. Although the NSDE is known to be robust to data perturbation, it is troublesome to account for each and every uncertainty tendency at once. With this motivation, we adopt the concept of SDE-Net (Kong, Sun, and Zhang 2020). The SDE-Net utilizes both in-distribution data (*ind*) and out-of-distribution data (*ood*) to train the diffusion network. They accomplish this by training the diffusion net to assign 0 to *ind* and 1 to *ood* in the SDE derivative computation. With this approach, the model accomplishes two things: 1. Uncertainty measurement of previously unseen *ood* data, 2. Larger weighting of brownian motion term dW_t for uncertain samples when solving SDE. Its objective is the following:

$$\min_{\boldsymbol{\theta}_g} \mathrm{E}_{\mathbf{h}_t \sim P_{ind}} g\left(\mathbf{h}_t, t\right) + \max_{\boldsymbol{\theta}_g} \mathrm{E}_{\tilde{\mathbf{h}}_t \sim P_{ood}} g\left(\tilde{\mathbf{h}}_t, t\right) \tag{5}$$

Compared to the SDE-Net, our method differs in two aspects. Firstly, we focus on detecting and utilizing trajectory samples with tracklet error within a dataset, rather than addressing cross-dataset distribution. To this end, we define *ind* as a clean trajectory and *ood* as trajectory containing tracklet error. During training, the ego-agent's trajectory is used for clean data since it is obtained from GPS and localization information, thus free from tracklet errors originating from occlusion or ID switch. For *ood* data, Gaussian noise is added to the ego-agent's trajectory. It is based on the fact that most multi-object trackers are based on recursive Bayesian filters, so they produce Gaussian state uncertainty (Ivanovic et al. 2022). This way, the encoder NSDE is trained to assign larger weight on Brownian motion term dW_t for noisy trajectory data, and smaller weight for clean data. Since the Browninan motion term of NSDE is known to act as a regularizer (Liu et al. 2019), our method fosters robustness by intensifying regularization effect on noisy trajectory samples through diffusion network-driven Brownian motion weighting.

Secondly, we address the tracklet error variation across datasets by using separate diffusion networks per dataset. In the NSDE formulation, the drift net aims to achieve high

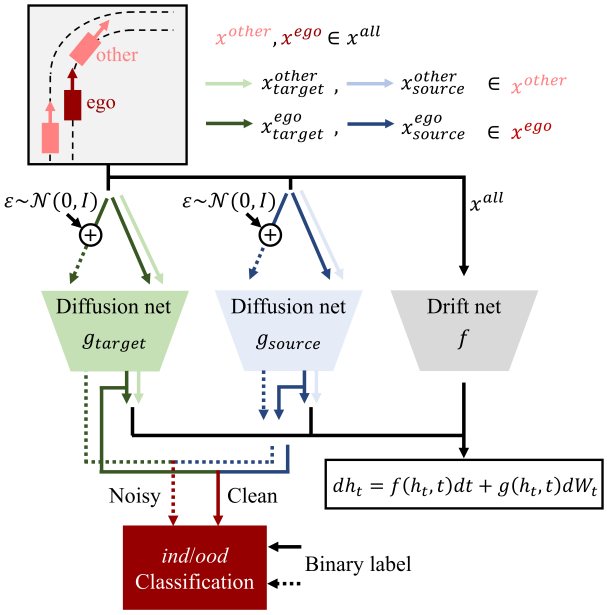


Figure 3: Visual representation of the proposed uncertainty training framework. We use a shared drift net and separated diffusion nets for source and target datasets. Unlike non-ego agents, ego-agents' trajectories with noise perturbation are additionally fed to the diffusion nets. The outputs from both drift networks and diffusion networks are utilized to solve the stochastic differential equation. In addition, diffusion networks are separately trained to perform in-distribution (*ind*) and out-of-distribution (*ood*) classification of ego-agent trajectories with and without perturbation. As a result, brownian motion dW_t takes larger weight on the subsequent SDE integration for uncertain samples.

prediction accuracy via system control, while the diffusion net captures aleatoric uncertainty. To leverage multi-source learning for prediction accuracy, we share the drift net across datasets and allocate distinct diffusion networks for each dataset. Ultimately, the derivative of NSDE encoder and its training objective is formulated as follows:

$$d\mathbf{h}_t = f(\mathbf{h}_t, t)dt + g(\mathbf{h}_t, t)dW_t, \quad g = \begin{cases} g_{sr} \text{ if } \mathrm{X}^{sr} \\ g_{tg} \text{ if } \mathrm{X}^{tg} \end{cases} \tag{6}$$

$$\min_{\boldsymbol{\theta}_{gsr}} \mathrm{E}_{\mathbf{h}_t \sim P^{sr}_{ego}} g\left(\mathbf{h}_t, t\right) + \max_{\boldsymbol{\theta}_{gsr}} \mathrm{E}_{\tilde{\mathbf{h}}_t \sim P^{sr}_{ego}} g\left(\tilde{\mathbf{h}}_t, t\right) + \min_{\boldsymbol{\theta}_{gtg}} \mathrm{E}_{\mathbf{h}_t \sim P^{tg}_{ego}} g\left(\mathbf{h}_t, t\right) + \max_{\boldsymbol{\theta}_{gtg}} \mathrm{E}_{\tilde{\mathbf{h}}_t \sim P^{tg}_{ego}} g\left(\tilde{\mathbf{h}}_t, t\right) \tag{7}$$

Here, g_{sr} and g_{tg} are diffusion networks for source and target dataset, respectively. $\mathbf{h}_t$ and $\tilde{\mathbf{h}}_t$ corresponds to hidden states encoded from clean ego-agent trajectories and noise-injected ego-agent trajectories. Figure. 3 illustrates the proposed uncertainty training framework.

Training Details In addition to the objective of Eq. 7, prediction loss between $\hat{\mathrm{Y}}$ and GT Y is also used in order to train the overall encoder-decoder structured model. In detail, the prediction loss trains the decoder SDE to learn a continuous transition function between the encoded feature $\mathbf{h}_0$ and future timestep hidden states $\{\mathbf{h}_t\}_{t\in(0,T_F]}$ and encoder SDE to produce a continuous representation of the observed trajectory X, thus enabling the model to handle arbitrary time lengths and steps. Additionally, if the baseline model is a goal-conditioned model, goal prediction loss between predicted goal position $\hat{\mathcal{G}}$ and gt goal $\mathcal{G}$ is used. For the implementation of SDE methods, *torchsde* (google-research 2021) is used. For the initial latent value ($\mathbf{h}^{past}_{-T_p}$), we assign a learnable parameter. Please refer to the supplementary materials for additional details on the training procedure and model implementation.

Experiment

Experiment Setup

Our goal is enhancing performance in target dataset via additional training on a large-scale source dataset. To assess whether the prediction model alleviates limited transferability across datasets, we jointly train with training set of both dataset and evaluate on validataion set of target dataset. In detail, the regression-based prediction model is trained only with nuScenes *train* set (**N**), jointly with nuScenes + Argoverse *train* sets (**N+A**) or nuScenes + WOMD *train* sets (**N+W**), then validated on nuScenes *val* set, with a widely used metric, mADE_{10}. To evaluate the model's generalizability on different target datasets, we additionally report training on WOMD. As for the goal-conditioned model, its improvement on nuScenes and Lyft validation sets are evaluated. Specifically, for nuScenes validation, we compare training only on nuScenes *train* set (**N**) and jointly with nuScenes + INTERACTION *train* sets (**N+I**). We additionally report its improvement on Lyft validation set by comparing only training on Lyft *train* set (**L**) and jointly with Lyft + INTERACTION *train* sets (**L+I**).

To enable discrete baseline models to be applied to two different time-step configured data, we have re-arranged both trajectory datasets. We create empty time series data that can contain both dataset time steps, then scatter each data to each time step. For example, with nuScenes (2/6s, 2Hz) as target and Argoverse (2/3s, 10Hz) as source, we create 81 bins (2/6s, 10Hz) of empty data. Then, nuScenes data is scattered with 5 time step intervals for overall lengths, while only the first 50 time steps are filled for Argoverse data.

Datasets and Baseline Models

We use nuScenes (30k) and Lyft (160k) as small-scale target dataset for their relatively smaller sizes. We utilize INTERACTION, Argoverse (200k) and WOMD (500k) datasets as largse-scale datasets for additional training. To utilize common information among datasets, we use past/future trajectories and lane centerline information only. Additionally, while these datasets have both vehicle and pedestrian trajectory data, we only train and evaluate vehicle trajectories for simplicity. To show the effectiveness of our framework, we select HiVT (Zhou et al. 2022) and MUSE-VAE (Lee et al. 2022) as the latest regression and goal prediction-based trajectory prediction method and show that even the state-of-the-art

Train Set	**N**	**N+A**	gain	**N+W**	gain
vanilla HiVT	1.045	0.966	7.56%	0.950	9.09%
HiVT + ODE-RNN	1.058	0.935	11.62%	0.913	13.71%
HiVT + latentSDE	1.044	0.943	9.67%	0.912	12.64%
HiVT + ours	1.044	**0.913**	**12.55%**	**0.893**	**14.46%**

Table 2: Effectiveness on regression-based method. All digits represent $mADE_{10}$ on nuScenes *val* set trained on each dataset and the corresponding gain. **N**, **A**, and **W** respectively denote nuScenes, Argoverse, and WOMD. Lower is better.

Valid set	Train set	museVAE	museVAE + Ours
N	**N**	2.304	2.333
	N+I	2.178	**1.953**
	gain	5.47%	**16.29%**
L	**L**	1.179	1.191
	L+I	1.073	**0.827**
	gain	8.90%	**30.56%**

Table 3: Effectiveness on goal-conditioned method on two different target datasets. All digits represent $mADE_{10}$ on nuScenes and Lyft *val* set trained on each dataset and the corresponding gain. **N**, **L**, and **I** respectively denote nuScenes, Lyft, and INTERACTION. The lower is better.

Valid set	Train set	HiVT	HiVT + Ours
W	**W** (5%)	0.9454	0.9445
	W (5%) + **A**	0.9286	**0.8496**
	gain	1.78%	**10.05%**

Table 4: Additional experiment of regression-based method on WOMD validation set as target dataset, again showing effectiveness on different target dataset. All digits represent $mADE_{10}$ on nuScenes *val* set trained on each dataset and the corresponding gain. **W** and **A** respectively denote Waymo and Argoverse. The lower is better.

method has room for improvement with the fusion of our proposed SDE framework.

Results

Effectiveness in Multi-Source Training

Table. 2 shows the improvements due to multi-source training on the regression-based model. We compare our method with original discrete models, as well as ODE-RNN (Rubanova, Chen, and Duvenaud 2019) and LatentSDE (Li et al. 2020) adaptations. Compared to training with **N**, the baseline model's mADE has improved 7.56% for **N+A**, and 9.09% for **N+W**. This improvement signifies an underfitted result when the model is only trained with nuScenes, a relatively smaller dataset that is comprised of only 30k training data. However, further performance gain has been limited since the discrete temporal encoding of vanilla HiVT is incapable of efficiently handling the cross-dataset discrepancy. By adopting ODE-RNN as the temporal encoder/decoder, the use of additional training data has brought about much more performance improvement (11.62% for **N+A**, 13.71% for **N+W**) thanks to its continuous modeling of latent transition across time. Although adopting latentSDE is known to be robust against data perturbation, its performance gain slightly decreases compared to ODE-RNN (9.67% for **N+A** and 12.64% for **N+W**). It is because the single diffusion network of latentSDE failed to address different type of tracklet error across datasets. Finally, with our proposed method, $mADE_{10}$ improves to 0.913 for **N+A**, and 0.893 for **N+W**. These improvements correspond to 12.55%/14.46% compared to nuScenes *only* training (**N**), and 5.49%/6% compared to vanilla HiVT (0.966 $\rightarrow$ 0.913 and 0.950 $\rightarrow$ 0.893) which empirically show the effectiveness of the proposed methods.

Table. 3 shows the effectiveness of our method on the goal-conditioned model. We conduct experiments on two different target validation datasets: **N** and **L**. For each case, we compare the performance gain when using **I** set as additional training data. The use of our method resulted in significant improvement in performance gain compared to the vanilla MUSE-VAE model. Specifically, its performance gains on both validation sets are threefold compared to the vanilla MUSE-VAE method, demonstrating the importance of improved transferability of our method across different types of backbone prediction models. In addition, similar to the goal-conditioned model's generalized improvement on two different target sets, the regression-based method also shows improvement in performance on a different target validation set. Table 4 reports the regresssion-based method's performance on **W** set as target set. Use of only 5% of **W** set is compared to additional use of **A** set to assume a situation where only a small amount of training data within target set distribution is available. The use of our method over the baseline HiVT again shows significant improvement in performance gain.

Effect of Target Dataset Size

Previous experiments have been conducted with the size of target dataset as 30k, the size of nuScenes *train* set. However, 30k is still a considerably large-scale dataset, and collecting a labeled dataset of an equivalent size could still be considered a cumbersome work. Therefore, we have also conducted the same experiments with smaller sizes of target dataset to show the effectiveness of our model even when the available target dataset is smaller. By randomly dropping a ratio of nuScenes *train* set, we construct the target datasets size of 20k, 15k, 10k, 3k, and 0 (no target dataset is used). Argoverse dataset is used as the source dataset. In the case of 0 target dataset setting, we share diffusion net and maintain the uncertainty training objective in Eq. 7. In Fig. 4, $mADE_{10}$

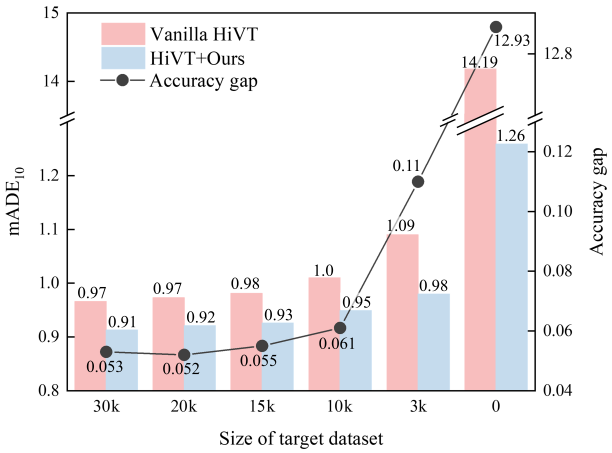


Figure 4: Prediction accuracy according to the size of the target dataset. The target dataset is nuScenes, and the source is Argoverse. X-axis denotes the amount of nuScenes training set used during training. The Y-axis bars indicate $mADE_{10}$ of both models, and the grey line indicates the accuracy gap between baseline HiVT and the proposed method.

of the baseline and our method are plotted with bar graph, and their difference is plotted as line graph. The effectiveness of the proposed method gradually increases in the range of 30k to 15k, and exponentially increases with the smaller target dataset sizes. Such larger improvements on smaller target datasets show that our proposed method effectively promotes transferability across datasets. Indeed, at an extreme with no target dataset for training, mADE of the baseline diverges over 14 while ours remain reasonable at 1.26. These results show that our proposed NSDE temporal networks's advantage of effective cross-dataset discrepancy handling is even more valued for smaller target datasets.

Effectiveness of Continuous Representation

We compare our NSDE with the baseline HiVT model equipping with other methods for handling dataset-wise unique time step configuration as reported in Tab. 5. First method is random dropping (RD) where some portion of time steps is randomly dropped during training. We expect RD to be equivalent to stochastic noise injection, thus improving generalizability. However, RD shows minimal improvement of only 0.006 since dropping time steps does not provide any extra time step data to a discrete temporal network. In that sense, we experiment with manipulating source data (1/8s, 10Hz) to target data's time step configuration (2/6s, 2Hz) (**S** $\rightarrow$ **T**) or vise-versa (**T** $\rightarrow$ **S**) through interpolation and extrapolation. Converting target dataset to source dataset configuration severely downgrads the prediction performance due to the source data's inaccurate information obtained from extreme extrapolation to unseen future time steps. While converting source dataset to target dataset slightly increases accuracy, its improvement remains minimal due to inaccurate extrapolation of past trajectory. Lastly, we apply domain-adpation method which is feature align loss $\mathcal{L}_{align}$ between source and target dataset following (Xu et al. 2022), where MMD loss with RBF kernel is used for the distance function. While the original paper tackled unsupervised domain adaptation problem, we provide labels for the target dataset for a fair comparison with other methods. However, applying $\mathcal{L}_{align}$ hinder prediction loss from target dataset supervision and had adverse effects on the prediction performance.

Baseline	RD	S $\rightarrow$ T	T $\rightarrow$ S	$\mathcal{L}_{align}$	SDE
0.950	0.944	0.987	0.947	0.971	0.912

Table 5: The effectiveness of time-wise continuous representation of NSDE compared to other methods. The digits represent $mADE_{10}$ on nuScenes *val* set for models trained on N+W. Lower is better.

Uncertainty Handling Ability

Our NSDE intensifies SDE's regularization effects by recognizing uncertain samples and assigning them large brownian motion weighting. Our method relies on the recognition of uncertain samples, therefore we quantify the recognized uncertainty to assess our method's adequate operation. The details of uncertainty quantification process are explained in the supplementary materials. For a qualitative review, Fig. 5 plots uncertain samples in red lines and others in yellow, thresholded by average standard deviation value of 0.06. In the nuScenes samples (1st row), it shows that our model can properly recognize uncertain samples due to tracking error with sudden position change (left) and meandering motion (right). Other dataset samples also show competent classification of samples with their dataset-specific uncertainties. More samples can be found in supplementary material.

Here, we analyze whether the diffusion net-based brownian-motion weighting indeed improve the model's robustness against tracklet error. In doing so, we compare the prediction results between our method (green) and the baseline (blue) on *ood* samples as in Fig. 5. Among 10 predictions for both models, only the most accurate predictions to GT (magenta) are plotted. Predictions of our method are consistently more accurate compared to the baseline's predictions. Such improvement is also quantitatively compared in Tab. 6 where $mADE_{10}$ is compared between predictions of baseline and ours on normal samples (*ind*) and uncertain samples (*ood*). Notably, our method exhibits larger accuracy gains in *ood*, demonstrating superior robustness against uncertain samples due to tracklet noise.

Ablation Studies

Ablation on model architecture appears in the upper part of Tab. 7, with HiVT as baseline. First, we model the past feature at time 0 (h_0^{past}) as Gaussian latent following general NDE methods. After obtaining mean and variance from h_0^{past} as in VAE, we sample past feature F times. (F is the number of prediction sample, here, set as 10). This approach severly worsen the performance since non-probability sampling is much better for multi-modal trajectory prediction (Bae, Park,

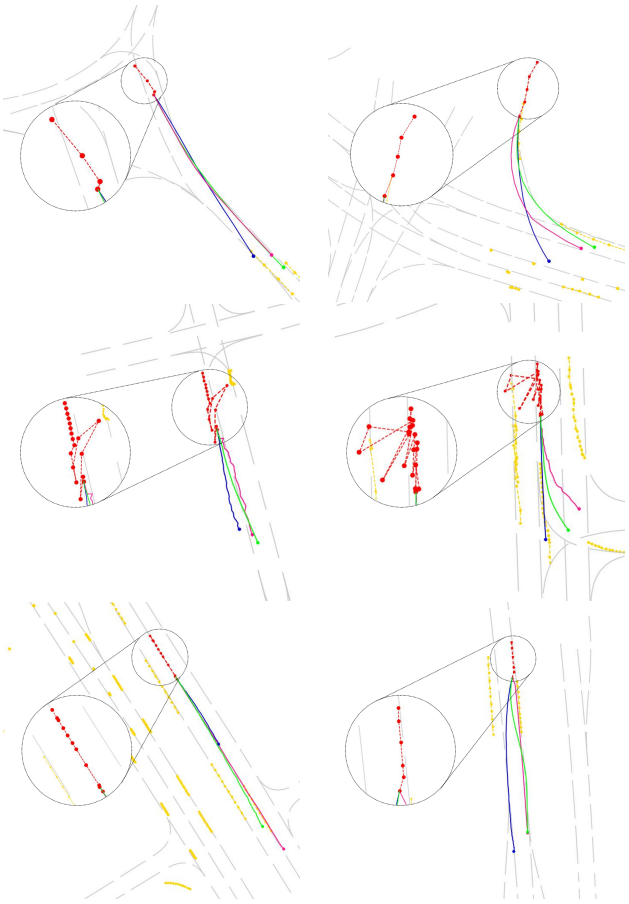


Figure 5: Prediction results against uncertain trajectory inputs. The dotted line is past trajectories of road actors. Among the yellow normal trajectories, the red ones are samples recognized as uncertain by the SDE-Net framework. Their GT future paths are plotted in magenta, prediction from baseline (HiVT) and ours are plotted in blue and green, respectively. Among 10 predictions of both models, most closest ones to the GT are plotted. Each row represents nuScenes (1st), Argoverse (2nd), and WOMD (3rd).

and Jeon 2022). Second, we adjust the number of layers of drift and diffusion network of encoder NSDE and decoder NSDE. Their number of layer is originally set as 4, and we reduce them as two. Comparing the results between encoder and decoder, model capacity decline resulted in larger performance drop for the encoder. We believe such discrepancy comes from higher complexity of encoder's task, as the encoder needs to translate past features while also considering incoming data on certain timesteps via GRU.

The lower part of Tab. 7 shows ablation on uncertainty training. Our model is comprised of shared drift network along with separate diffusion networks and we ablate each component. Although we lose multi-source training for temporal encoding when separating drift network, performance drop is relatively small since other components of the model are still shared. In case of sharing diffusion network, however, the performance drop is larger since a single diffusion network is insufficient to handle disparate types of noises. Indeed, noise in argoverse is more severe compared to nuScenes as shown in the second row in Fig. 5, so the diffusion network is dominated by argoverse data's distribution and results in wrong Brownian noise injection. In addition, we remove the uncertainty training objective in Eq. 7, which has also resulted in a considerable performance drop. The above results consistently reveal that uncertainty handling plays a crucial role during cross-domain trajectory prediction.

thres: 0.01/0.06	*ind*	*ood*
Baseline	0.580	1.414
Ours	0.551	1.333
gain	0.029	**0.081**

Table 6: Prediction accuracy (mADE_{10}) against normal samples (*ind*) and uncertain samples (*ood*) in nuScenes *val* set. Baseline HiVT and the proposed methods trained on N+A are compared. Lower is better.

	Experiments	mADE_{10}
Model architecture	Gaussian latent	1.225
	Encoder f/g layers=2	0.927
	Decoder f/g layers=2	0.918
Uncertainty training	w/o share Drift net	0.934
	w/o separate Diff net	0.944
	w/o $\mathcal{L}_{\text{uncertain}}$	0.940
	Full model	0.913

Table 7: Ablation studies on N+A training.

Conclusion

In this paper, we introduce a novel approach to addressing the challenges posed by discrepancies in trajectory datasets. By leveraging continuous and stochastic representations within NSDE, the proposed method tackles two key issues: varying time step configurations and different patterns of detection/tracking noise across datasets. The continuous representation effectively handles diverse time intervals, enabling seamless adaptation to different dataset structures, while the stochastic aspect accommodates the inherent uncertainties arising from tracklet errors. Through experimentation on nuScenes, Argoverse, Waymo, INTERACTION, and WOMD, our NSDE consistently improved upon both regression and goal prediction-based the state-of-the-art methods.

We not only highlight the importance of dataset-specific considerations in trajectory prediction but also introduce a practical solution that bridges the gap between diverse data sources. These contributions underscore the methodology's potential for advancing the reliability and safety of autonomous mobility systems, offering a promising avenue for further research and development in the field.

Acknowledgments

This work was partially supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2014-3-00123, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis), and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF2022R1A2B5B03002636).

References

Anumasa, S.; and Srijith, P. 2022. Latent time neural ordinary differential equations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6010–6018.

Bae, I.; and Jeon, H.-G. 2023. A Set of Control Points Conditioned Pedestrian Trajectory Prediction. In *AAAI Conference on Artificial Intelligence*.

Bae, I.; Park, J.-H.; and Jeon, H.-G. 2022. Non-probability sampling network for stochastic human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6477–6487.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Cao, D.; Enouen, J.; Wang, Y.; Song, X.; Meng, C.; Niu, H.; and Liu, Y. 2023. Estimating Treatment Effects from Irregular Time Series Observations with Hidden Confounders. *arXiv preprint arXiv:2303.02320*.

Chang, M.-F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. 2019. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8748–8757.

Chen, R. T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural Ordinary Differential Equations. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

De Brouwer, E.; Simm, J.; Arany, A.; and Moreau, Y. 2019. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. *Advances in neural information processing systems*, 32.

Ettinger, S.; Cheng, S.; Caine, B.; Liu, C.; Zhao, H.; Pradhan, S.; Chai, Y.; Sapp, B.; Qi, C. R.; Zhou, Y.; et al. 2021. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9710–9719.

Ge, C.; Song, S.; and Huang, G. 2023. Causal Intervention for Human Trajectory Prediction with Cross Attention Mechanism. In *AAAI Conference on Artificial Intelligence*.

Gilles, T.; Sabatini, S.; Tsishkou, D. V.; Stanciulescu, B.; and Moutarde, F. 2022. Uncertainty estimation for Cross-dataset performance in Trajectory prediction. *ArXiv*, abs/2205.07310.

google-research. 2021. torchsde. https://github.com/google-research/torchsde.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Houston, J.; Zuidhof, G.; Bergamini, L.; Ye, Y.; Chen, L.; Jain, A.; Omari, S.; Iglovikov, V.; and Ondruska, P. 2021. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, 409–418. PMLR.

Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; Lu, L.; Jia, X.; Liu, Q.; Dai, J.; Qiao, Y.; and Li, H. 2023. Planning-oriented Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ivanovic, B.; Lin, Y.; Shrivastava, S.; Chakravarty, P.; and Pavone, M. 2022. Propagating State Uncertainty Through Trajectory Forecasting. In *2022 International Conference on Robotics and Automation (ICRA)*, 2351–2358.

Kidger, P.; Morrill, J.; Foster, J.; and Lyons, T. 2020. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33: 6696–6707.

Kong, L.; Sun, J.; and Zhang, C. 2020. SDE-Net: Equipping Deep Neural Networks with Uncertainty Estimates. In *International Conference on Machine Learning*, 5405–5415. PMLR.

Kothari, P.; Kreiss, S.; and Alahi, A. 2021. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, 23(7): 7386–7400.

Lee, M.; Sohn, S. S.; Moon, S.; Yoon, S.; Kapadia, M.; and Pavlovic, V. 2022. MUSE-VAE: Multi-Scale VAE for Environment-Aware Long Term Trajectory Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2221–2230.

Li, L.; Yao, J.; Wenliang, L. K.; He, T.; Xiao, T.; Yan, J.; Wipf, D.; and Zhang, Z. 2021. GRIN: Generative Relation and Intention Network for Multi-agent Trajectory Prediction. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Li, X.; Wong, T.-K. L.; Chen, R. T.; and Duvenaud, D. 2020. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, 3870–3882. PMLR.

Liang, R.; Li, Y.; Li, X.; Tang, Y.; Zhou, J.; and Zou, W. 2021. Temporal pyramid network for pedestrian trajectory prediction with multi-supervision. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 2029–2037.

Liu, X.; Xiao, T.; Si, S.; Cao, Q.; Kumar, S.; and Hsieh, C.-J. 2019. Neural sde: Stabilizing neural ode networks with stochastic noise. *arXiv preprint arXiv:1906.02355*.

Malinin, A.; Band, N.; Gal, Y.; Gales, M.; Ganshin, A.; Chesnokov, G.; Noskov, A.; Ploskonosov, A.; Prokhorenkova, L.; Provilkov, I.; Raina, V.; Raina, V.; Roginskiy, D.; Shmatova,

M.; Tigas, P.; and Yangel, B. 2021. Shifts: A Dataset of Real Distributional Shift Across Multiple Large-Scale Tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Meng, M.; Wu, Z.; Chen, T.; Cai, X.; Zhou, X. S.; Yang, F.; and Shen, D. 2022. Forecasting Human Trajectory from Scene History. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Norcliffe, A.; Bodnar, C.; Day, B.; Moss, J.; and Liò, P. 2020. Neural ODE Processes. In *International Conference on Learning Representations*.

Park, D.; and Park, Y.-H. 2020. Identifying Reflected Images From Object Detector in Indoor Environment Utilizing Depth Information. *IEEE Robotics and Automation Letters*, 6(2): 635–642.

Park, D.; Ryu, H.; Yang, Y.; Cho, J.; Kim, J.; and Yoon, K.-J. 2022. Leveraging Future Relationship Reasoning for Vehicle Trajectory Prediction. In *The Eleventh International Conference on Learning Representations*.

Park, S.; Kim, K.; Lee, J.; Choo, J.; Lee, J.; Kim, S.; and Choi, E. 2021. Vid-ode: Continuous-time video generation with neural ordinary differential equation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2412–2422.

Qian, Z.; Kacprzyk, K.; and van der Schaar, M. 2022. D-CODE: Discovering Closed-form ODEs from Observed Trajectories. In *International Conference on Learning Representations*.

Rubanova, Y.; Chen, R. T.; and Duvenaud, D. K. 2019. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32.

Saleh, F.; Aliakbarian, S.; Rezatofighi, H.; Salzmann, M.; and Gould, S. 2021. Probabilistic tracklet scoring and inpainting for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14329–14339.

Salzmann, T.; Ivanovic, B.; Chakravarty, P.; and Pavone, M. 2020. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, 683–700. Springer.

Shi, L.; Wang, L.; Long, C.; Zhou, S.; Zheng, F.; Zheng, N.; and Hua, G. 2022. Social interpretable tree for pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2235–2243.

Tang, B.; Zhong, Y.; Neumann, U.; Wang, G.; Chen, S.; and Zhang, Y. 2021. Collaborative Uncertainty in Multi-Agent Trajectory Forecasting. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Tzen, B.; and Raginsky, M. 2019. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, C.; Chen, X.; Wang, J.; and Wang, H. 2022a. ATPFL: Automatic Trajectory Prediction Model Design Under Federated Learning Framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6563–6572.

Wang, L.; Hu, Y.; Sun, L.; Zhan, W.; Tomizuka, M.; and Liu, C. 2022b. Transferable and adaptable driving behavior prediction. *arXiv preprint arXiv:2202.05140*.

Wang, R.; Wang, S.; Yan, H.; and Wang, X. 2023. WSiP: Wave Superposition Inspired Pooling for Dynamic Interactions-Aware Trajectory Prediction. In *AAAI Conference on Artificial Intelligence*.

Wen, S.; Wang, H.; and Metaxas, D. 2022. Social ODE: Multi-agent Trajectory Forecasting withÂ Neural Ordinary Differential Equations. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, 217–233. Cham: Springer Nature Switzerland. ISBN 978-3-031-20047-2.

Weng, X.; Ivanovic, B.; Kitani, K.; and Pavone, M. 2022. Whose Track Is It Anyway? Improving Robustness to Tracking Errors with Affinity-based Trajectory Prediction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6563–6572.

Weng, X.; Ivanovic, B.; and Pavone, M. 2022. MTP: Multi-hypothesis Tracking and Prediction for Reduced Error Propagation. *IV*.

Westny, T.; Oskarsson, J.; Olofsson, B.; and Frisk, E. 2023. MTP-GO: Graph-Based Probabilistic Multi-Agent Trajectory Prediction with Neural ODEs. *IEEE Transactions on Intelligent Vehicles*, 1–14.

Wu, Y.; Wang, L.; Zhou, S.; Duan, J.; Hua, G.; and Tang, W. 2023. Multi-Stream Representation Learning for Pedestrian Trajectory Prediction. In *AAAI Conference on Artificial Intelligence*.

Xu, Y.; Wang, L.; Wang, Y.; and Fu, Y. 2022. Adaptive Trajectory Prediction via Transferable GNN. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6510–6521.

Ye, L.; Zhou, Z.; and Wang, J. 2023. Improving the Generalizability of Trajectory Prediction Models with Frenét-Based Domain Normalization. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 11562–11568.

Zhan, W.; Sun, L.; Wang, D.; Shi, H.; Clausse, A.; Naumann, M.; Kümmerle, J.; Königshof, H.; Stiller, C.; de La Fortelle, A.; and Tomizuka, M. 2019. INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps. *arXiv:1910.03088 [cs, eess]*.

Zhou, Z.; Ye, L.; Wang, J.; Wu, K.; and Lu, K. 2022. HiVT: Hierarchical Vector Transformer for Multi-Agent Motion Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8823–8833.