# MapCVAE: Probabilistic Prediction of Diverse Pedestrian Behaviors on General Roads

Konosuke Kobayashi* and Satoru Fujita†
* Hosei University, Japan
E-mail: konosuke.kobayashi.8j@stu.hosei.ac.jp Tel/Fax: +81-42-387-4545
† Hosei University, Japan
E-mail: fujita_s@hosei.ac.jp Tel/Fax: +81-42-387-4545

***Abstract*—Accurate pedestrian trajectory prediction is crucial for autonomous vehicles, but capturing the diverse and uncertain nature of pedestrian movements remains a significant challenge. This paper introduces MapCVAE, a new model based on Conditional Variational Autoencoders (CVAE), which predicts multiple possible future paths. By integrating semantic map information, MapCVAE produces contextually appropriate trajectories and captures diverse intentions such as changing direction or stopping. Experiments show that MapCVAE significantly outperforms conventional deterministic models in predicting diverse pedestrian behaviors. This research makes self-driving cars safer because it helps them better understand where a person might walk next.**

## I. Introduction

In modern self-driving technology, it is crucial for cars to capture their surroundings and react to pedestrian behaviors for safer driving. On roads and near crosswalks, people often move in unpredictable ways, like stopping suddenly or changing direction. As these sudden actions increase accident risks, self-driving cars must accurately detect and predict these behaviors. Though research on pedestrian movement prediction has typically provided only a single possible path, people probabilistically move to unexpected directions in real life. On general roads, a single-path prediction is insufficient for estimating pedestrian's possible paths.

Our research aims to solve this issue by using deep learning to predict pedestrian paths, especially near crosswalks. We focus on predicting various and complex actions, such as quick changes in direction and sudden stops. We use a Conditional Variational Autoencoder (CVAE) because it works well for predicting likelihood of future paths based on past observations and understanding uncertain human behaviors. Using this model, we aim to accurately capture a diversity of pedestrian's complex behaviors. This will lead to building more reliable self-driving systems that can quickly and safely respond to unexpected situations.

The main contributions of this work are three-fold:

1) We propose MapCVAE, a novel CVAE-based architecture that effectively integrates map information to predict diverse pedestrian trajectories on general roads.
2) We introduce a discrete latent space using the Gumbel-Softmax technique to learn interpretable walking modes.
3) Experiments on the Argoverse 2 dataset demonstrate our model's superiority and validate its key components.

## II. Related Work

### A. Deterministic Pedestrian Trajectory Prediction

Researchers have actively studied predicting future pedestrian trajectories since self-driving cars first appeared. Initially, many studies focused on modeling pedestrian behaviors by considering social interactions among individuals [1]. These approaches predicted future paths by analyzing how pedestrians influence each other's movements in crowded environments. While such models predicted human's collective behavior, they still struggled to accurately forecast individual trajectories in complex scenarios.

With the development of deep learning, the field of pedestrian trajectory prediction achieved significant improvements. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) are often used to predict pedestrian paths from their past movements [2][3]. These models work deterministically to output only a single prediction. They learn common behavior patterns from training data to estimate future paths. However, pedestrian behavior on general roads does not always follow a single deterministic path. A major problem remains: these deterministic models cannot represent the diverse possibilities for uncertain behaviors like sudden stops or changes in direction.

### B. Diverse Pedestrian Trajectory Prediction using CVAE

To understand and predict uncertain and diverse human movements in deep, recent research has shifted from single predictions to probabilistic models that can forecast several possible future paths. Variational Autoencoders(VAE) and similar generative models are promising solutions. Conditional Variational Autoencoders (CVAE), in particular, can generate many possible future paths by using past data and a latent space [4][5].

Existing research that applies CVAE to pedestrian trajectory prediction has mainly focused on dense environments or specific scenarios [6]. While these studies demonstrate the CVAE's ability to model behavioral diversity, most ones do not focus on accurate capturing of complex and unpredictable human behaviors in general roads near crosswalks.

## III. Driving Dataset

We use the Argoverse 2 dataset [7] as real-world path data. This data set was collected using actual vehicles in six

979-8-3315-7206-8/25/$31.00 

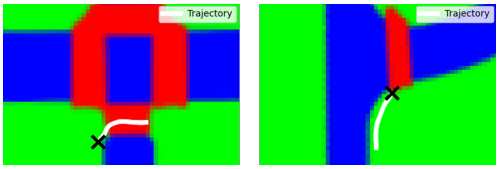


Fig. 1. Examples of pedestrian trajectories and surrounding map in the Argoverse2 dataset

US cities. The Argoverse 2 dataset contains about 250,000 scenarios. It provides detailed path information for various object types, including cars, pedestrians, motorcycles, bicycles, and buses. Each scenario includes data in eleven seconds, collected at 100 milliseconds per frame. This allows us to analyze fine-grained pedestrian movements. The dataset also provides map information, such as crosswalk locations and road–sidewalk boundaries, represented as vector maps (i.e., sequences of points). We use these vector maps to identify areas like crosswalks and roads, and convert them into color-coded environmental images.

Figure 1 shows an example of a pedestrian's path. The solid white line represents the pedestrian's path, and the 'X' mark indicates the path end. On the map, red refers to crosswalks, blue to roads, and green to other areas.

## IV. MapCVAE

We propose MapCVAE, a trajectory prediction model based on Conditional Variational Autoencoders (CVAE), to model pedestrian's probabilistic behaviors.

### A. Problem Formulation

This section defines the problem of predicting the future paths of pedestrians on general roads. Given a past trajectory $X_{1:T} = \{X_1, X_2, \ldots, X_T\}$ over $T$ time steps, the model predicts a future trajectory distribution $\hat{Y}_{T+1:T+U} = \{\hat{Y}_{T+1}, \hat{Y}_{T+2}, \ldots, \hat{Y}_{T+U}\}$ over $U$ time steps. Here, $X_t = (x_t, y_t)$ denotes the coordinates and $\hat{Y}_t = (\mu_{x,t}, \mu_{y,t}, \log \sigma^2_{x,t}, \log \sigma^2_{y,t}, \rho_t)$ represents the parameters of a bivariate Gaussian distribution. In addition to the past trajectory, segmentation images M of the surrounding map are also used as input. The prediction task is formulated as:

$$\hat{Y}_{T+1:T+U} = f(X_{1:T}, M) \tag{1}$$

where $f$ is a prediction model.

### B. Framework

Figure 2 illustrates the MapCVAE framework. MapCVAE is designed to learn the uncertainty in pedestrian behavior based on past pedestrian trajectories and the surrounding map, enabling the prediction of future trajectory distributions. Specifically, the Past Encoding Module encodes and integrates the path and map data. This allows the model to understand how pedestrians have moved in various environments in deep. The Probabilistic Multi-Path Prediction Module, implemented using a CVAE, then probabilistically predicts diverse pedestrian behaviors on general roads, generating multiple plausible future trajectories. The red arrows of the figure indicate the training-only paths.

### C. Past Encoding Module

*1) Path Encoder:* A Transformer Encoder model captures the temporal features of past pedestrian trajectories. Given the past trajectory $X_{1:T} = \{X_1, X_2, \ldots, X_T\} \in \mathbb{R}^{T\times 2}$ over $T$ time steps as input, the model outputs a feature representation $h_{path} \in \mathbb{R}^{1\times D}$, which represents the final step's feature, as follows:

$$h_{path} = TransformerEncoder(\phi(X_{1:T})) \tag{2}$$

Here, $\phi(\ldots)$ represents a Linear layer, D is the internal dimension.

*2) Map Encoder:* In pedestrian trajectory prediction on general roads, it is important to examine not only past trajectories but also the surrounding environment. This is because the environment influences human behaviors like stops or changes in direction. To address this issue, we incorporate a surrounding map $M \in \mathbb{R}^{R\times\theta\times 4}$ into our model.

Figure 3 shows a map example. We redraw this map at final time step $t$ based on the pedestrian's position, covering an area within a radius $R = 15[m]$ and an angle of $\pm\pi/2[rad]$ from the walking direction. Each pixel on the map corresponds to an area of 1 meter in radial length and $\pi/18[rad]$ in angular width. Each map consists of $R \times \theta$ pixels. The map is segmented into four regions: red pixels represent crosswalks, blue pixels represent roads, green pixels represent sidewalk areas, and black pixels represent other areas. Because the original data lacked a "sidewalk" label, we defined the "sidewalk" area as a 5-meter buffer around the "road" and "crosswalk" regions within the remaining ("other") areas. By using an ego-centric map representation that imitates a pedestrian's field of view, our model learns to associate the pedestrian's heading with surrounding features (e.g., crosswalks) to better understand actions like crossing the street.

Map Encoder consists of a CNN-based front-end module that captures local environmental features , followed by a Transformer Encoder module that aggregates radial features using the attention mechanism. The Map Encoder takes the map information $M$ as input and outputs an environmental feature representation $h_{map} \in \mathbb{R}^{R\times\theta\times D}$ as follows:

$$h_{map} = TransformerEncoder(\phi(CNN(M))) \tag{3}$$

*3) Cross Encoder:* A Cross Attention mechanism effectively integrates the features extracted by the Path Encoder and the Map Encoder. It is crucial to properly fuse these two different types of information for improving the accuracy of future trajectory prediction. In the Cross Attention module, the output from the Path Encoder serves as the Query (Q), while the output from the Map Encoder serves as both the Key (K) and Value (V). This combination helps the model focus on the most important map information based on the pedestrian's past movements. The fused feature $h_{past} \in \mathbb{R}^{1\times D}$ obtained from the Cross Encoder effectively captures rich interactions

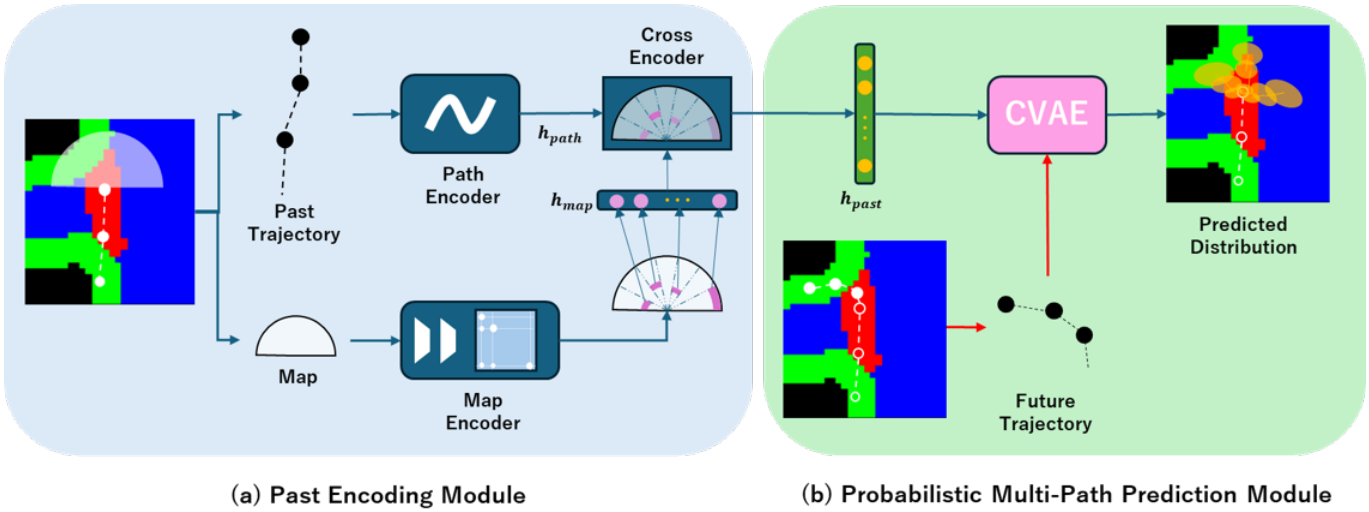


Fig. 2. The framework of MapCVAE. (a) Past Encoding Module: This module encodes the pedestrian's past steps. A Path Encoder encodes past pedestrian trajectory, and a Map Encoder encodes surrounding map. These two types of data are fused via a Cross Encoder. (b) Probabilistic Multi-Path Prediction Module: This module predicts future trajectory distributions using a CVAE model. The red arrows indicate the training-only paths.

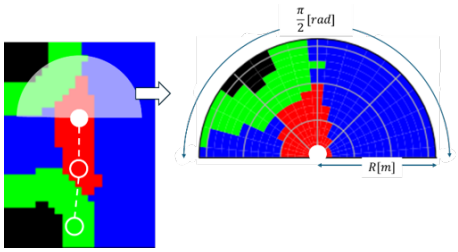


Fig. 3. An example of surrounding map extraction used in the model

between pedestrian behavior and the surrounding environment. The process is defined by:

$$F^1_{past} = MultiHead(h_{path}, h_{map}, h_{map}) \tag{4}$$

$$F^2_{past} = LayerNorm(F^1_{past} + h_{path}) \tag{5}$$

$$h_{past} = LayerNorm(FFN(F^2_{past}) + F^2_{past}) \tag{6}$$

$MultiHead(\dots)$ denotes a multi-head attention mechanism that helps the model jointly attend to different parts of the input information. $LayerNorm(\dots)$ is layer normalization, which stabilizes training by normalizing the features. $FFN(\dots)$ is a feed-forward network that enhances the features using fully connected neural network layers.

### D. Probabilistic Multi-Path Prediction Module

To capture the diversity of pedestrian behavior on general roads and perform probabilistic trajectory prediction, we incorporate a CVAE structure into the Probabilistic Multi-Path Prediction Module. Figure 4 shows the architecture of this model. The red arrows in the figure indicate the data flow used only during training. The model takes the fused feature $h_{past}$, output from the Past Encoding Module, as conditional input, and generates the future pedestrian trajectory over $U$ steps. Here, while the input $Y \in \mathbb{R}^{U\times 2}$ is a sequence of 2D coordinate $(x, y)$, the output $\hat{Y} \in \mathbb{R}^{U\times 5}$ is a sequence of a set of parameters $(\mu_x, \mu_y, \log\sigma_x^2, \log\sigma_y^2, \rho)$ that define a bivariate Gaussian distribution.

The continuous latent space in standard CVAEs often produces unrealistic trajectories by interpolating between distinct walking modes (e.g., 'straight' and 'turn'). To address this, we introduce the Gumbel-Softmax technique [8] to create a discrete latent space. This allows the model to learn K clearly separated modes, enabling the generation of more realistic and diverse paths.

During training, the Encoder first uses a Path Encoder to extract time-series features from the reconstruction target Y. A Cross Encoder then fuses these features with the conditional input $h_{past}$ to output the fused feature $F_{rec} \in \mathbb{R}^{U\times D}$. Then, $F_{rec}$ is averaged along the temporal dimension. This averaged feature and $h_{past}$ are both passed through the latent encoder $E_{latent}$ to compute the prior probabilities $prob_p$ and the posterior probabilities $prob_q$:

$$prob_p = E_{latent}(h_{past}) \tag{7}$$

$$prob_q = E_{latent}(F_{rec}) \tag{8}$$

Subsequently, $prob_q$ is passed through a Gumbel-Softmax layer to generate a one-hot vector representing the selected mode. The Gumbel-Softmax layer is a differentiable technique for generating a probabilistic one-hot vector. It works by adding Gumbel noise to an input $prob_q$ and then applying a softmax function controlled by a temperature parameter $\tau$. The temperature controls how sharp the output is and how close it is to a real one-hot vector. After that, the generated one-hot vector is expanded to $U$ future time steps. To help the model understand temporal order, the expanded one-hot vector is passed through positional encoding $E_{pos}$, resulting in $Z$. Finally, the Transformer Decoder–based Decoder takes

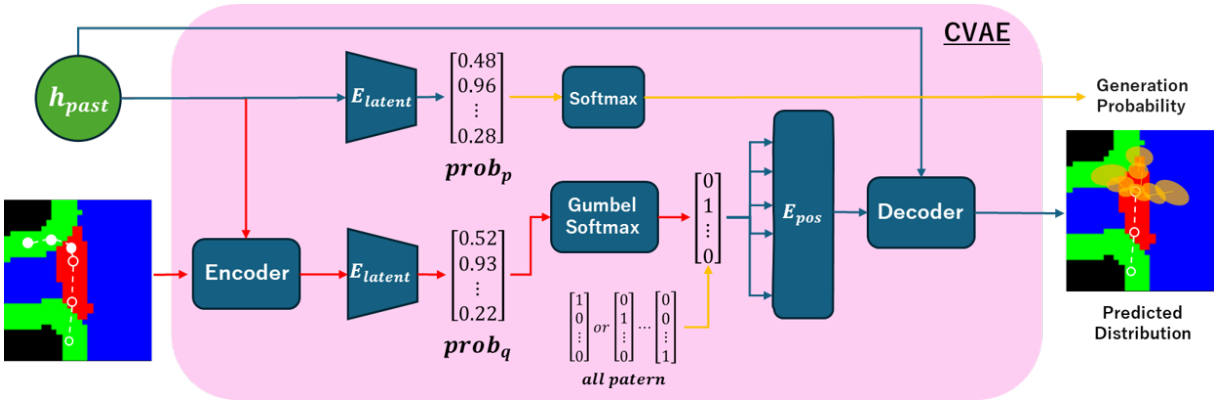


Fig. 4. Probabilistic Multi-Path Prediction Module : The red arrows indicate the training-only paths, while the yellow arrows indicate the inference-only paths.

both $h_{past}$ and $Z$ as input, generating the predicted future trajectory distribution $\hat{Y}$ over $U$ steps.

$$\hat{Y} = Decoder(Z, h_{past}) \tag{9}$$

During inference, the ground-truth trajectory is unavailable, so the posterior probabilities cannot be computed. Instead, our model systematically generates a future path for each of the K possible modes by inputting its corresponding one-hot vector. The probability of each generated path is then determined by applying a softmax function to the prior probabilities, as shown by the yellow arrows in Figure 4.

### E. Loss Function

Our model is trained end-by-end by minimizing a multi-task loss:

$$\begin{aligned}\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}\Big(&\sum_{t=T+1}^{T+U}(\lambda_1\|\hat{Y}^{n,t}_{(\mu_x,\mu_y)} - Y^{n,t}\|_2 \\ &+ \lambda_2 \mathrm{NLL}(\hat{Y}^{n,t}, Y^{n,t})) \\ &+ \lambda_3 D_{\mathrm{KL}}(prob_p \,\|\, prob_q)\Big)\end{aligned} \tag{10}$$

The first term is an L2 loss between the predicted mean position and the ground truth. The second term is the negative log-likelihood (NLL) loss of the predicted distribution. The third term is the Kullback–Leibler (KL) divergence between the prior probabilities and the posterior probabilities. $\lambda_1, \lambda_2, \lambda_3$ are weighting coefficients for each loss term.

## V. Experiments

### A. Experimental Setup

*1) Dataset:* In experiments, we reduce the data size by sampling frames every 200 milliseconds. The training data includes about 25,000 scenarios with around 25,000 pedestrians. The test data includes about 3,000 scenarios with around 3,000 pedestrians. We use 5 seconds (25 frames) of past observations and predict the next 6 seconds (30 frames). Additionally, for data normalization and consistent orientation, we transform the coordinates of each trajectory. The final observed position is set to the origin, and the direction of travel is aligned with the positive y-axis. We perform data augmentation to enhance the diversity of our training data. This involves two methods: first, we balance the dataset by the future direction of travel (e.g., straight, left, and right turns) , and second, we horizontally flip the entire dataset to increase its volume and variety.

*2) Evaluation Metrics:* We evaluate the predicted trajectories using two metrics: the k-Average Path Distribution Evaluation (k-APDE) and the k-Final Path Distribution Evaluation (k-FPDE). These metrics calculate the probability of the predicted distributions falling within a k-meter radius of the ground-truth points. Specifically, k-APDE assesses this probability for the distributions at each timestep along the entire trajectory , while k-FPDE assesses it only for the distribution at the final point. These metrics are formulated as follows:

$$k-APDE = \frac{1}{NU}\sum_{n=1}^{N}\sum_{u=1}^{U}\int\int_{A_{n,u}} \mathcal{N}(\mu_{n,u}, \Sigma_{n,u})dA \tag{11}$$

$$k-FPDE = \frac{1}{N}\sum_{n=1}^{N}\int_{A_n} \mathcal{N}(\mu_n, \Sigma_n)dA \tag{12}$$

Where $N$ is the number of pedestrians, $T$ is the prediction time, $A$ is the circular area with a radius of k[m] centered on the ground truth point, and $\mu$ and $\Sigma$ are the mean vector and the covariance matrix of the model output, respectively.

Since our model generates multiple future paths, we evaluate its performance using three sets of metrics. We use best-APDE and best-FPDE for the prediction closest to the ground truth; mode-APDE and mode-FPDE for the most probable prediction; and MDN-APDE and MDN-FPDE, which consider the entire distribution of predicted paths and their probabilities.

*3) Implementation details:* The input to the model consists of pedestrian coordinates $(x, y)$ and map images within a radius $R$ and an angle of $\pm\pi/2[rad]$ in the direction of movement as environmental information for the past 4 seconds. The output is a predicted trajectory over the next 6 seconds. Our model predicts the distribution $(\mu_x, \mu_y, \log\sigma_x^2, \log\sigma_y^2, \rho)$ of the future path. We set the internal dimension $D$ to 128, the number of walking modes $K$ to 3, and the dropout rate to 0.1.

For training, we set the number of epochs to 600, the learning rate to $5 \times 10^{-4}$, and used the AdamW optimizer.Furthermore, for the hyperparameters used in the loss

TABLE I
QUANTITATIVE RESULTS: PREDICTION PERFORMANCE

Higher is better; best results are in **bold**.

| | MLP | Transformer | Ours(mode) | Ours(best) | Ours(MDN) |
|---|---|---|---|---|---|
| 1-APDE | 0.465 | 0.468 | 0.590 | **0.683** | 0.560 |
| 3-APDE | 0.823 | 0.828 | 0.889 | **0.965** | 0.869 |
| 1-FPDE | 0.057 | 0.061 | 0.154 | **0.219** | 0.129 |
| 3-FPDE | 0.424 | 0.419 | 0.630 | **0.827** | 0.569 |

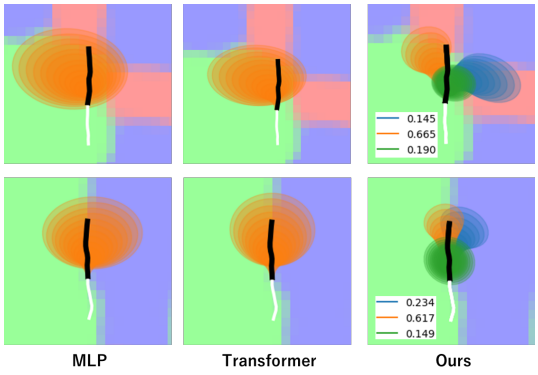


Fig. 5. Comparative visualizations: Past (white), ground-truth (black), and predicted distributions (colored), with path probabilities in the legend.

TABLE II
QUANTITATIVE RESULTS: ABLATION STUDY ON MODEL ARCHITECTURE

Higher is better; best results are in **bold**.

| | N-Ours | B-Ours | S-Ours | Ours |
|---|---|---|---|---|
| 1-APDE (MDN) | 0.460 | 0.536 | 0.461 | **0.560** |
| 3-APDE (MDN) | 0.790 | 0.848 | 0.816 | **0.869** |
| 1-FPDE (MDN) | 0.065 | 0.117 | 0.053 | **0.129** |
| 3-FPDE (MDN) | 0.362 | 0.513 | **0.780** | 0.569 |

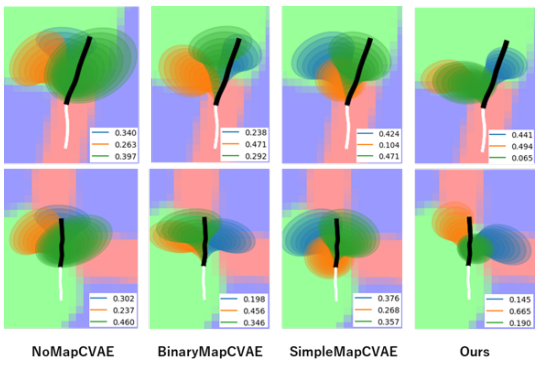


Fig. 6. Visualization of the Ablation Study: Past (white), ground-truth (black), and predicted distributions (colored), with path probabilities in the legend.

function, we set $\lambda_1 = 1$ and $\lambda_2 = 1$, while $\lambda_3$ is increased from 0.0 to 1.0 as the training epochs progress. Additionally, the temperature parameter $\tau$ for the Gumbel-Softmax is similarly annealed from 3.0 to 0.1.

*4) Comparison Model:* Our model predicts multiple possible future trajectories instead of just one. To evaluate its performance, we implemented the following models and compared them using the same evaluation metrics.

- MLP : This model uses simple MLP layers to output a single predicted trajectory distribution.
- Transformer : This model uses simple Transformer layers to output a single predicted trajectory distribution.

### B. Quantitative analysis

The performance of our proposed MapCVAE and the reference models was evaluated using APDE and FPDE metrics. The quantitative results are presented in Table I.

The results show that MapCVAE achieves the highest scores across all metrics. This consistent outperformance suggests that MapCVAE's ability to model multiple paths is more effective for this task than the deterministic approaches.

### C. Qualitative analysis

We can observe the diverse predictions of MapCVAE and their effectiveness in detail. Figure 5 shows the prediction results for each model. In the figure, the white line represents the pedestrian's past trajectory, the black line represents the true future trajectory, the orange, blue, and green lines represent the predicted distribution. The legend shows the generation probability of each path.

MapCVAE effectively predicts diverse pedestrian behaviors. At crosswalks, where path changes are probable, the model's predictions spread across multiple distinct paths, accurately reflecting uncertain choices. Conversely, on roadways with fewer behavioral options, predictions become more linear.

### D. Ablation Study

*1) Effect of the Model Architecture:* To validate the effectiveness of the surrounding map and our model architecture, we compared MapCVAE's performance against three ablated variants: 1) No-MapCVAE (N-Ours), which excludes the map input; 2) Binary-MapCVAE (B-Ours), which uses a binarized map (walkable vs. non-walkable); and 3) Simple-MapCVAE (S-Ours), in which all layers are replaced with MLP layers.

Table II presents the APDE and FPDE scores from our comparative experiment. MapCVAE clearly outperforms all ablated variants on both metrics. This suggests that using surrounding map is crucial (vs. No-MapCVAE). Additionally, the results support the importance of detailed map representations (vs. Binary-MapCVAE) and the efficacy of our Transformer and CNN-based feature extraction (vs. Simple-MapCVAE). We conclude that the combination of these components is crucial for achieving high accuracy in pedestrian prediction.

This effect is also visually confirmed in Figure 6. NoMapCVAE tends to generate similar predictions regardless of the pedestrian's location. In contrast, a comparison between BinaryMapCVAE and MapCVAE in the bottom row shows that MapCVAE is aware of the nearby crosswalk, enabling it to generate more realistic predictions, such as waiting before crossing.

*2) Effect of the Number of Modes K:* To verify the effect of the number of modes K, we conducted an ablation study by varying K from 3 to 10. Figure 7 shows the results. We use MDN-APDE and MDN-FPDE as our evaluation metrics.

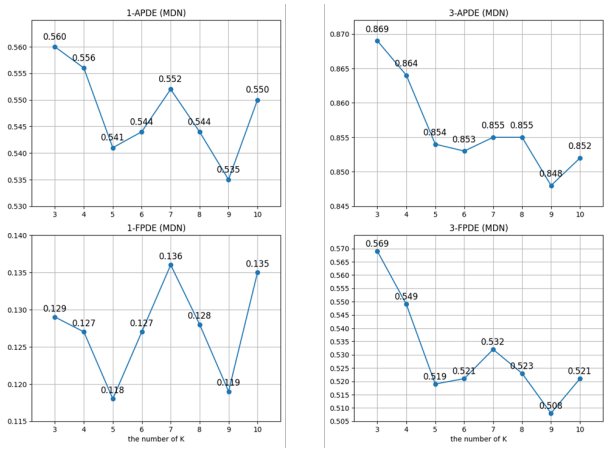


Fig. 7. Performance evaluation with a varying number of K. The x-axis represents the number of modes, while the y-axis shows the APDE and APDE.

The results show that performance evaluation, especially for 3-APDE and 3-FPDE, degrades as K increases. This indicates that most pedestrian behavior in our dataset can be explained by three main modes. When K is too large, probability is spread over unused modes, lowering the overall likelihood.

## VI. Discussion

In this research, we proposed MapCVAE, a model that integrates surrounding map with a CVAE to probabilistically predict the diverse behaviors of pedestrians on general roads. Experiments demonstrated the model's effectiveness, validated the importance of its components.

Performance experiments shows that MapCVAE outperforms deterministic baselines, revealing its ability to generate diverse and accurate predictions (Table I). This confirms that single-path prediction is insufficient, and probabilistic outputs are essential to capture the uncertainty of pedestrian behavior. Such predictions are especially valuable for autonomous vehicles, allowing safer decisions in tasks like collision avoidance.

The ablation study shows the importance of surrounding maps (Table II). MapCVAE with 4-class semantic maps achieves the best results, outperforming No-MapCVAE and Binary-MapCVAE. This suggests the model leverages semantic context, not just walkability, to generate more realistic predictions, such as crossing at intersections.

The analysis of prediction modes provides another key insight. Contrary to the common assumption that more modes are better for capturing diversity, our experiments showed that performance is best at K=3. This suggests that a pedestrian's short-term intentions can be simplified to a few main behaviors. Choosing the right K is crucial, as too large a value may generate unrealistic behaviors and lower the quality of the distribution.

While MapCVAE shows strong performance, it has several limitations. The current model considers only a single pedestrian and static maps, without explicitly modeling interactions with other agents or dynamic factors such as traffic signals. Furthermore, this research was evaluated on a specific dataset (Argoverse 2), and its generalization performance in other geographical environments remains unverified.

Future work will address these limitations by adding social attention to model interactions with other agents, incorporating dynamic environmental factors, and testing on diverse datasets. Another challenge is to develop a mechanism that adaptively determines the optimal number of modes K based on environmental complexity.

## VII. Conclusion

This paper proposed MapCVAE, a CVAE-based pedestrian trajectory prediction model for general road environments. Compared to deterministic prediction models, the introduction of CVAE enabled the probabilistic prediction of diverse pedestrian behaviors, leading to more accurate predictions.

The key contribution of this research is showing the effectiveness of a probabilistic approach for autonomous driving. By generating a diverse and possible futures, MapCVAE provides the rich information needed for a vehicle's planning module to make safer decisions. While future work will address challenges like multi-agent interactions, the principles validated here represent a key step toward more reliable autonomous systems.

## References

[1] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, Sep. 2000, ISSN: 1476-4687. DOI: 10.1038/35035023. [Online]. Available: http://dx.doi.org/10.1038/35035023.

[2] P. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990. DOI: 10.1109/5.58337.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997. DOI: 10.1162/neco.1997.9.8.1735.

[4] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, 2022. arXiv: 1312.6114 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1312.6114.

[5] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, *Semi-supervised learning with deep generative models*, 2014. arXiv: 1406.5298 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1406.5298.

[6] W. Xiang, H. Yin, H. Wang, and X. Jin, *Socialcvae: Predicting pedestrian trajectory via interaction conditioned latents*, 2024. arXiv: 2402.17339 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2402.17339.

[7] B. Wilson, W. Qi, T. Agarwal, *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[8] E. Jang, S. Gu, and B. Poole, *Categorical reparameterization with gumbel-softmax*, 2017. arXiv: 1611.01144 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1611.01144.