Spatio-Temporal Graph-Based Pedestrian Trajectory Prediction with Environmental Context Integration and Constraint Learning*

Yuka Takahara and Yuka Kato[1]

***Abstract*— Accurate pedestrian trajectory prediction is fundamental for safe navigation of autonomous robots and vehicles in crowded environments. Although graph-based methods such as Social-STGCNN efficiently model social interactions among pedestrians, they largely ignore static environmental constraints such as walls and obstacles, which can lead to physically infeasible predictions. In this paper, we propose an extended trajectory prediction method that integrates local and global environmental information into Social-STGCNN via Multi-Head Cross-Attention, and incorporates environmental constraint learning through contrastive MapNCE and collision avoidance EnvCol losses. To generate diverse prediction while maintaining scene consistency, we further introduce a low-dimensional trajectory representation based on Singular Value Decomposition and Adaptive Anchors derived from K-means clustering. We evaluate the proposed method on the ETH/UCY benchmark across five scenes using Average Displacement Error (ADE), Final Displacement Error (FDE), obstacle collision rate, and inference time. The results show that the proposed method consistently improves the FDE over Social-STGCNN, while ADE increases. The obstacle collision rate also increases, revealing a trade-off between endpoint accuracy and full-trajectory environmental compliance.**

I. INTRODUCTION

Pedestrian trajectory prediction in crowded environments is a fundamental technology for safe and smooth path planning and velocity control in autonomous mobile robots and self-driving systems. In spaces such as shopping centers and university campuses, estimating future pedestrian positions in advance and making decisions based on these predictions is directly linked to collision avoidance and improved mobility efficiency. However, pedestrian motion depends not only on interactions with other pedestrians but also on environmental structures such as walls, obstacles, and navigable regions, making accurate prediction a challenging problem.

To address this challenge, rule-based models such as the Social Force Model [1] and collision avoidance models such as ORCA [2] have been widely used. While these approaches offer high interpretability and can explicitly handle kinematic and geometric constraints, they are limited in representing complex interactions and diverse future behaviors. Consequently, deep learning (DL)-based trajectory prediction methods have become prevalent in recent years. Existing DL approaches have advanced along two main directions: modeling social interactions among pedestrians, and incorporating environmental information. For the former, LSTM-based methods [3] [4] and graph-based methods have been proposed to represent pedestrian interactions in a temporal and structural manner. We have also previously proposed a data construction method for LSTM-based trajectory prediction [5]. Notably, Social Spatio-Temporal Graph Convolutional Neural Network (Social-STGCNN) [6] efficiently captures pedestrian interactions via spatio-temporal graph convolution, achieving both high prediction accuracy and fast inference. For the latter, methods such as SoPhie [7] integrate environmental features from scene images and map information into prediction models, enabling trajectory prediction that accounts for both social and physical constraints.

However, existing methods still face notable limitations. Methods that excel at modeling social interactions often fail to explicitly account for environmental information, leading to physically infeasible trajectories that conflict with obstacles or walls. On the other hand, methods that incorporate environmental information may improve endpoint prediction but do not consistently enhance full-trajectory accuracy or environmental compliance throughout the entire prediction sequence. Furthermore, integrating environmental information and constraint learning increases inference time, making the balance between environmental awareness and computational efficiency an important consideration.

To solve these problems, this paper proposes a pedestrian trajectory prediction method that builds upon Social-STGCNN and combines local and global environmental information integration with environmental constraint learning. The proposed method efficiently represents social interactions among pedestrians via a spatio-temporal graph, while dynamically integrating local and global environmental features extracted from scene maps, and further incorporates environment-constraint-based training losses to promote scene-consistent predictions. In addition, we introduce diverse prediction candidate generation based on a low-dimensional trajectory representation.

The main contributions of this paper are: (i) the development of a trajectory prediction method that jointly incorporates local and global environmental information integration and environmental constraint learning into a graph-based social interaction model; and (ii) a comprehensive evaluation covering not only prediction accuracy but also obstacle collision rate and inference time, revealing trade-offs between endpoint accuracy, environmental compliance, and computational efficiency

*This work was partially supported by JSPS KAKENHI (23K11087, 23K11073), the Telecommunications Advancement Foundation, the Cooperative Research Project Program of RIEC, Tohoku University, and Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures (JHPCN) and High Performance Computing Infrastructure (HPCI) in Japan (Project ID: jh250015).

[1]Yuka Takahara and Yuka Kato are with Division of Information and Mathematical Sciences, Tokyo Woman's Christian University, Tokyo, Japan. yuka@cis.twcu.ac.jp

II. RELATED WORK

Research on pedestrian trajectory prediction encompasses methods based on social interactions among pedestrians, methods that integrate environmental information, and more recently, methods that explicitly learn to avoid environmental collisions. This section provides an overview of them.

A. Social Interaction-Based Methods

An early DL-based approach is Social LSTM [3], which models each pedestrian's trajectory with an LSTM and aggregates the states of neighboring pedestrians via social pooling, enabling trajectory prediction that accounts for crowd interactions. Subsequently, Social GAN [4] was proposed to handle multiple plausible future trajectories through adversarial training between a generator and discriminator combined with a variety loss, achieving diverse and socially acceptable trajectory generation. Social-STGCNN [6] was further proposed as a method that achieves both computational efficiency and expressive power. By representing a group of pedestrians as a spatio-temporal graph and directly modeling their interactions through Spatial-Temporal Graph Convolutional Neural Network (ST-GCNN), it delivers improved prediction accuracy while also reducing the number of parameters and inference time. AgentFormer [8] is a Transformer-based multi-agent trajectory prediction method that jointly handles the temporal dimension and inter-agent interactions without separation. Although it enables more flexible interaction modeling, its high computational cost and data requirements make it not always the optimal choice in small-scale datasets or latency-critical applications.

These social interaction-based methods excel at capturing relationships among pedestrians, yet may fail to adequately handle static environmental constraints, posing a challenge for deployment in real environments.

B. Environment Integration-Based Methods

Pedestrians move not only in response to interactions with others but also in accordance with static environmental constraints. Models that take only trajectory coordinates as input may therefore generate physically infeasible trajectories. To address this, prediction methods that incorporate scene images or map-based environmental constraints have been proposed. A representative example is SoPhie [7], which employs both social attention and physical attention to simultaneously handle pedestrian interactions and environmental context derived from scene images, generating environment-aware paths. Another example is Trajectron++ [9], a graph-structured recurrent model that integrates heterogeneous data such as dynamic constraints and semantic maps, with emphasis on integration with planning and control systems.

Although these methods demonstrate that incorporating environmental information contributes to improved prediction performance and physical plausibility, they lack structured interaction representations and face challenges in simultaneously achieving high inference speed and prediction accuracy.

C. Methods with Explicit Environmental Collision Avoidance

Recent work has gone beyond using environmental information merely as input features and has instead explicitly formulated obstacle collision avoidance as a learning objective. Environmental Collision Avoidance Module (ECAM) [10] is a module that can be incorporated into existing trajectory prediction models. Its key characteristics are a focus on improving collision-free predictions and the introduction of contrastive learning-based Map Noise-Contrastive Estimation (MapNCE) and an environmental collision loss (EnvCol).

However, while such modules are effective when added to existing models, their performance can vary significantly depending on how they are combined with the social interaction representation of the underlying model. In particular, how to coherently integrate environmental constraint learning with high-efficiency graph-based models has not been explored.

This work builds upon the efficient spatio-temporal graph representation of Social-STGCNN and aims to achieve both endpoint prediction accuracy and environmental compliance by combining local and global environmental information integration with environmental constraint learning.

III. METHOD

A. Problem Formulation

The problem addressed in this work is to predict future pedestrian trajectories given observed trajectories and environmental information of the scene. Let $\boldsymbol{x}_t^i \in \mathbb{R}^2$ denote the position of pedestrian i at time t, N the number of pedestrians in the scene, T_{obs} the number of observed frames, and T_{pred} the number of predicted frames. The observed trajectory sequence $\boldsymbol{X}$ is defined as a matrix whose columns are the sequences $\{\boldsymbol{x}_0^i, ..., \boldsymbol{x}_{T_{\text{obs}}-1}^i\}$ for all N pedestrians. Similarly, the predicted trajectory sequence $\boldsymbol{Y}$ is a matrix whose columns are the sequences $\{\boldsymbol{x}_{T_{\text{obs}}}^i, ..., \boldsymbol{x}_{T_{\text{obs}}+T_{\text{pred}}-1}^i\}$ for all N pedestrians. The task is to estimate the conditional probability distribution of future trajectories $\boldsymbol{Y}$ given the observed trajectories $\boldsymbol{X}$ and environmental information.

The static environment of the scene is represented as a binary map $\boldsymbol{M} \in \{0,1\}^{H \times W}$, where H and W denote the height and width of the map, with values indicating walkable regions and obstacle regions respectively. Since pedestrian trajectories are given in a metric world coordinate system while the environment map is expressed in image coordinates, a homography matrix $\boldsymbol{H} \in \mathbb{R}^{3\times 3}$ is used to establish the correspondence for each scene. In addition, a vector field $\boldsymbol{F}$ is used as auxiliary information representing the dominant walking tendencies in each scene. The vector field is not a fixed external input but is computed from historical pedestrian trajectory sequences in the training data: velocities are computed from first-order differences and accelerations from second-order differences, and they are combined into a matrix representation.

The prediction problem is thus formulated as estimating the conditional distribution $f(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{M}, \boldsymbol{H}, \boldsymbol{F})$ of future

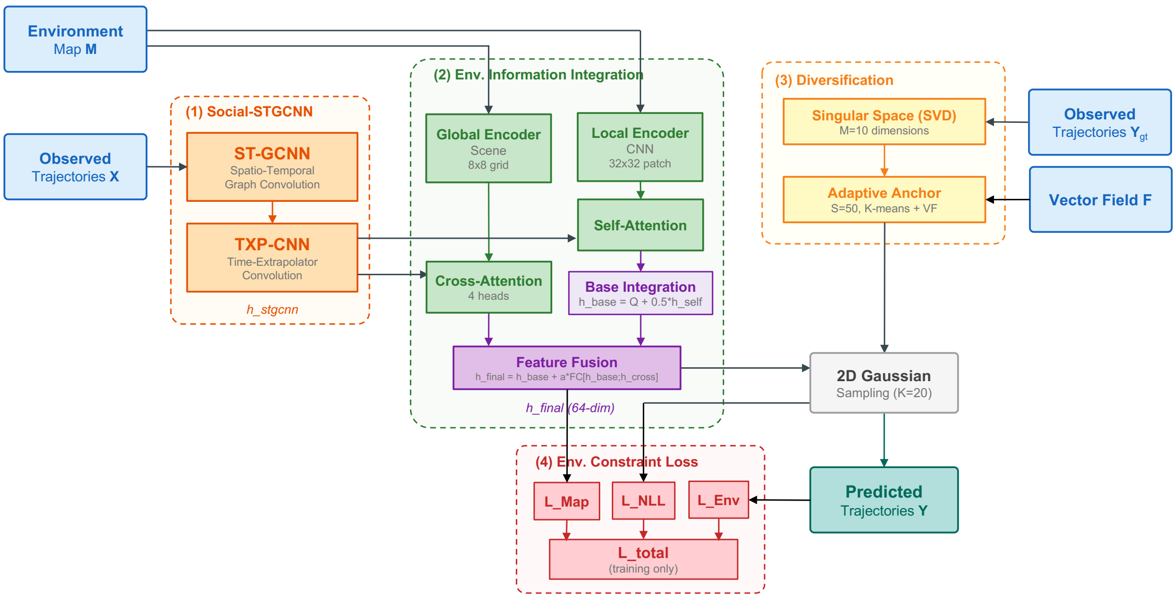

Fig. 1. System architecture of the proposed method.

trajectories $\boldsymbol{Y}$ given the observed trajectories $\boldsymbol{X}$, the environment map $\boldsymbol{M}$, the homography matrix $\boldsymbol{H}$, and the vector field $\boldsymbol{F}$ computed from historical trajectories.

Observed trajectories are normalized based on the maximum displacement during the observation period to improve training stability. Specifically, the current position (last observed position) of pedestrian i is set as the origin $\boldsymbol{o}^i$, and normalization is performed using the maximum displacement $s_i = \max_t \|\boldsymbol{x}_t^i - \boldsymbol{o}^i\|_2$ as

$$\bar{\boldsymbol{x}}_t^i = \frac{\boldsymbol{x}_t^i - \boldsymbol{o}^i}{\max(s_i, 1.0)} \tag{1}$$

where $\|\cdot\|_2$ denotes the L2 norm. Velocities and accelerations used to construct the vector field are also normalized using $\bar{\boldsymbol{x}}$. Although the number of pedestrians varies across scenes, it is fixed to N for batch processing; when the actual number of pedestrians is less than N, the remaining entries are filled with zeros to maintain consistent input/output dimensions.

B. Method Overview

The system architecture of the proposed method is shown in Fig. 1. The method consists of the following four modules:

- **Social Interaction Modeling**: A spatio-temporal graph captures interactions among pedestrians.
- **Environmental Information Integration**: Local environmental features and the global scene structure are dynamically integrated via Cross-Attention.
- **Low-Dimensional Representation and Prediction Diversification**: A trajectory representation in Singular Space improves computational efficiency, while Adaptive Anchors generate diverse predictions.
- **Explicit Environmental Compliance Learning**: Map-NCE and EnvCol losses explicitly train the model for obstacle avoidance, improving the environmental compliance of all generated samples.

C. Method Details

1) Social Interaction Modeling: Social-STGCNN is used as the base model to represent social interactions among pedestrians. At each time step t, the positions of pedestrians form a vertex set $\mathcal{X}_t = \{\boldsymbol{x}_t^i \mid i \in \{0, ..., N-1\}\}$. Inter-pedestrian interactions are represented by an edge set $\mathcal{E}_t = \{e_t^{ij} \mid i, j \in \{0, ..., N-1\}\}$, forming a spatial graph $\mathcal{G}_t = (\mathcal{X}_t, \mathcal{E}_t)$. The edge weight e_t^{ij} is defined as

$$e_t^{ij} = \begin{cases} 1/\|\boldsymbol{x}_t^i - \boldsymbol{x}_t^j\|_2 & (\|\boldsymbol{x}_t^i - \boldsymbol{x}_t^j\|_2 \neq 0) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

so that pedestrians in closer proximity are assigned larger weights. Spatial graphs over the entire observation period T_{obs} are unified to form a spatio-temporal graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$.

This $\mathcal{G}$ is fed into the ST-GCNN layers to extract interaction patterns and temporal dependencies among pedestrians. The output is then extended along the time dimension by the Time-Extrapolator CNN (TXP-CNN) layers, converting the representation from the observation period T_{obs} to the prediction period T_{pred}, yielding the hidden feature $\boldsymbol{h}_{\text{stgcnn}}^i$.

2) Environmental Information Integration: An environmental patch centered on the current position of each pedestrian is cropped and fed into a CNN encoder to extract a local environmental feature vector $\boldsymbol{m}^i$ specific to each pedestrian. Simultaneously, a Global Scene Encoder divides the entire environment map into multiple grids and generates feature vectors, constructing global environmental tokens $\boldsymbol{T}_{\text{global}}$.

To selectively incorporate only the environmentally relevant information for each pedestrian from these local and global features, Multi-Head Cross-Attention is employed. Here, the linear combination $\boldsymbol{Q}^i = \boldsymbol{h}_{\text{stgcnn}}^i + 0.5 \times \boldsymbol{m}^i$ serves as the Query, and $\boldsymbol{T}_{\text{global}}$ serves as both Key and Value, computing Cross-Attention as

$$\boldsymbol{h}_{\text{cross}}^i = ATT(\boldsymbol{Q}^i, \boldsymbol{T}_{\text{global}}, \boldsymbol{T}_{\text{global}}). \tag{3}$$

This enables each pedestrian to dynamically and selectively incorporate the relevant portions of the global scene, rather than adding environmental information in a fixed manner.

Next, a base integration is performed by incorporating inter-pedestrian interaction features $\boldsymbol{h}^i_{\text{self}}$ obtained via Self-Attention (attention applied across all pedestrians on $\boldsymbol{h}^i_{\text{stgcnn}}$), yielding $\boldsymbol{h}^i_{\text{base}} = \boldsymbol{Q}^i + 0.5 \times \boldsymbol{h}^i_{\text{self}}$. This is then concatenated with $\boldsymbol{h}^i_{\text{cross}}$ and passed through a fully connected layer FC to align dimensions, fusing social and environmental features and reshaping them into residual-compatible dimensions. The final integrated feature is constructed as

$$\boldsymbol{h}^i_{\text{final}} = \boldsymbol{h}^i_{\text{base}} + \alpha \ \text{FC}[\boldsymbol{h}^i_{\text{base}}; \boldsymbol{h}^i_{\text{cross}}] \tag{4}$$

where $[\cdot;\cdot]$ denotes concatenation and α is a coefficient controlling the contribution of environmental features.

3) Low-Dimensional Representation and Prediction Diversification: Although future trajectories have a high-dimensional representation of $T_{\text{pred}} \times 2$, the dominant motion modes are assumed to be expressible in a low-dimensional subspace. During training, each ground-truth trajectory sample is converted to a vector of length $T_{\text{pred}} \times 2$ and arranged as columns in a matrix $\boldsymbol{Y}_{\text{gt}} \in \mathbb{R}^{(T_{\text{pred}} \times 2) \times N}$, to which Singular Value Decomposition (SVD) $\boldsymbol{Y}_{\text{gt}} = \boldsymbol{U\Sigma V}^T$ is applied for dimensionality reduction. Using the matrix $\boldsymbol{U}_{\text{trunc}} \in \mathbb{R}^{(T_{\text{pred}} \times 2) \times M}$ composed of the top M left singular vectors, the low-dimensional coefficients of each sample are obtained as

$$\boldsymbol{C} = \boldsymbol{U}^T_{\text{trunc}} \boldsymbol{Y}_{\text{gt}} \in \mathbb{R}^{M \times N} \tag{5}$$

where $\boldsymbol{C}$ represents the coordinates of each training trajectory (ground truth) in the singular space, with each column corresponding to a single trajectory. To ensure prediction diversity, K-means clustering is applied to $\boldsymbol{C}$ to obtain S cluster centers. Denoting each cluster center as $a_s \in \mathbb{R}^M$, the anchor set is given by $\boldsymbol{C}_{\text{anchor}} = \{a_1, ..., a_S\} \in \mathbb{R}^{M \times S}$, where each a_s represents a representative trajectory pattern in the singular space.

At inference phase, each anchor a_s is reconstructed from the singular space back to the original trajectory space and used as a reference trajectory. Each reference trajectory is then adaptively deformed per pedestrian using the scene's vector field $\boldsymbol{F}$. Specifically, near the endpoint of the anchor trajectory, the vector field is referenced according to the final observed position and heading direction of each pedestrian, and the relative scale and heading are corrected to obtain the s-th adaptive anchor trajectory for pedestrian i. This allows generating diverse prediction candidates that preserve representative trajectory patterns while being consistent with each pedestrian's past motion and scene-specific movement tendencies.

4) Explicit Environmental Compliance Learning: In addition to the negative log-likelihood loss, the model is trained using the MapNCE and EnvCol losses proposed in ECAM [10] for environmental compliance.

For the base loss, the negative log-likelihood of a bivariate Gaussian distribution is used. The Gaussian parameters are obtained per pedestrian per time step by applying 1×1 convolution to $\boldsymbol{h}^i_{\text{stgcnn}}$, specifically the mean $\boldsymbol{\mu}^i_t$, variance $\boldsymbol{\sigma}^i_t$, and correlation ρ^i_t. The base loss is defined as

$$\mathcal{L}_{\text{NLL}} = -\frac{1}{N} \sum_{i=0}^{N-1} \sum_{t=0}^{T_{\text{pred}}-1} \log p\big(\boldsymbol{x}^i_t \mid \boldsymbol{\mu}^i_t, \boldsymbol{\sigma}^i_t, \rho^i_t\big) \,. \tag{6}$$

For MapNCE, the integrated feature $\boldsymbol{h}^i_{\text{final}}$ is mapped to a query vector, and contrastive learning is performed with the ground-truth future position as a positive sample and points near obstacle contours as negative samples. Positive samples are obtained by adding Gaussian noise to the ground truth: $\boldsymbol{p}^i_{\text{pos}} = \boldsymbol{x}^i_{T_{\text{pred}}-1} + \mathcal{N}(0,\ \sigma^2 \boldsymbol{I})$. Negative samples are generated using contour points $\boldsymbol{c}_f$ and direction angles $\theta_d = 2\pi d/8$ as $\boldsymbol{p}^{f,d}_{\text{neg}} = \boldsymbol{c}_f + \rho \cdot (\cos\theta_d,\ \sin\theta_d)^T + \mathcal{N}(0,\ \sigma^2 \boldsymbol{I})$, where ρ is a predefined distance from contour points. Letting j be a sequential index over (f, d) pairs ($j = 1, \ldots, J = F \times 8$), the MapNCE loss is defined as

$$\mathcal{L}_{\text{Map}} = -\frac{1}{N} \sum_{i=0}^{N-1} \log \frac{P_i}{P_i + \sum_{j=1}^{J} N^i_j} \tag{7}$$

where J is the total number of negative samples. Denoting the query encoder that maps $\boldsymbol{h}^i_{\text{final}}$ to a query vector as ψ and the key encoder that maps positive/negative samples to key vectors as ϕ,

$$P_i = \exp(\psi(\boldsymbol{h}^i_{\text{final}}) \cdot \phi(\boldsymbol{p}^i_{\text{pos}})/\tau) \tag{8}$$

$$N^i_j = \exp(\psi(\boldsymbol{h}^i_{\text{final}}) \cdot \phi(\boldsymbol{p}^j_{\text{neg}})/\tau) \tag{9}$$

where τ is a temperature parameter.

EnvCol is a loss that penalizes only the set of predicted samples $\mathcal{S}_i$ that collide with obstacles, pulling them toward the ground-truth trajectory:

$$\mathcal{L}_{\text{Env}} = \frac{1}{N} \sum_{i=0}^{N-1} \sqrt{\frac{1}{|\mathcal{S}_i| \cdot T_{\text{pred}}} \sum_{k \in \mathcal{S}_i} \sum_{t=0}^{T_{\text{pred}}-1} \left\| \hat{\boldsymbol{x}}^{i,k}_t - \boldsymbol{x}^i_t \right\|^2_2}. \tag{10}$$

When $|\mathcal{S}_i| = 0$, the contribution is set to 0.

The total loss is given as the sum of these losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NLL}} + \lambda_{\text{map}} \mathcal{L}_{\text{Map}} + \lambda_{\text{env}} \mathcal{L}_{\text{Env}}. \tag{11}$$

D. Inference

At inference phase, K sample trajectories (in this paper, $K = 20$) are generated by sampling from the bivariate Gaussian distribution. Inverse normalization is then applied to the samples to convert them back to the metric coordinate system. Note that MapNCE and EnvCol are computed only during training and are not evaluated at inference phase. This allows the model to learn environmental compliance during training while keeping inference overhead low.

IV. EXPERIMENTAL SETUP

This paper evaluates whether the proposed method can improve prediction accuracy and environmental compliance compared to existing methods. In particular, both Final Displacement Error (FDE) and Average Displacement Error (ADE) are assessed, and Collision Rate (COL) and inference time are also compared to provide a comprehensive evaluation from the perspectives of accuracy, safety, and efficiency.

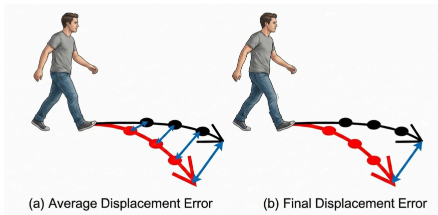

Fig. 2. Illustration of ADE and FDE.

A. Dataset

We evaluate on the ETH/UCY dataset [11] [12], a widely used benchmark for trajectory prediction research. The dataset comprises five scenes—ETH, HOTEL, UNIV, ZARA01, and ZARA02—covering diverse environments including open spaces, corridors, and crossing flows. The original data is sampled at 2.5 fps; in this work, the observation sequence is set to 8 frames (3.2 sec.) and the prediction one to 12 frames (4.8 sec.). Coordinates are given in a metric world coordinate system, and for each scene we make use of an environment map, a homography matrix, and a vector field representing walking tendencies. The characteristics of each scene are summarized below:

- **ETH**: The plaza in front of the main building of ETH Zurich. An open outdoor space with few occlusions.
- **HOTEL**: A sidewalk in front of a hotel in Zurich. An outdoor space where pedestrians walk along a building.
- **UNIV**: The campus of the University of Cyprus. A mix of narrow corridors and open plazas where pedestrian flows cross in complex patterns.
- **ZARA01**: A well-known shopping street in Madrid. Predominantly straight pedestrian flows.
- **ZARA02**: A well-known shopping street in Madrid. Characterized by crossing flows and turning corners.

Data splits follow the predefined train/validation/test partitions for each scene. Training is performed on the full training set, and the validation set is used for hyperparameter tuning. Final evaluation is conducted on the test set.

B. Evaluation Metrics

ADE and FDE are used as prediction accuracy metrics. ADE measures the mean positional error over the entire prediction sequence, reflecting overall accuracy from short-term to long-term predictions. FDE measures the positional error at the final time step, evaluating the accuracy of destination prediction. Both metrics are computed in the metric coordinate system, and Best-of-N evaluation is applied by selecting the sample closest to the ground truth among the multiple predicted samples generated by the proposed method. In this paper, 20 samples are generated per input. An illustration of these metrics is shown in Fig. 2.

In addition, COL is used as a metric for environmental compliance. COL is defined as the proportion of predicted samples for which the distance to an obstacle falls below a threshold of $\theta_{\mathrm{col}} = 0.2$ m. Collision detection targets the first 3 seconds of predictions, with temporal resolution enhanced through interpolation.

C. Baselines

To evaluate the effectiveness of the proposed method, Social-LSTM and Social-STGCNN are used as baselines. Social-LSTM is a classical method that represents inter-pedestrian interactions using LSTM and Social Pooling. Social-STGCNN is a method that efficiently models social interactions via spatio-temporal graph convolution. Since the proposed method extends Social-STGCNN by adding local and global environmental integration and ECAM, the difference between the two directly reflects the effect of environmental information integration.

D. Implementation Details

AdamW is used for optimization with an initial learning rate of 0.001 and weight decay of 10^{-5}. The effective batch size is 8 (mini-batch of 2 with 4 gradient accumulation steps), and the maximum gradient norm for clipping is set to 1.0. CosineAnnealingWarmRestarts from PyTorch is used as the learning rate scheduler, and the maximum number of epochs is 250. Loss weights are set to $\lambda_{\mathrm{NLL}} = 1.0$ (fixed); λ_{map} and λ_{env} are initialized to $\lambda_{\mathrm{map}} = 0.3$ and $\lambda_{\mathrm{env}} = 0.05$, and dynamically adjusted based on validation performance.

Key hyperparameters for the model architecture are as follows: the Social-STGCNN component uses 1 ST-GCNN layer (hidden dimension 64), 5 TXP-CNN layers (with residual connections), and dropout of 0.1. For environmental information integration, the local patch size is 32×32 pixels, the global environment representation uses an 8×8 grid, Cross-Attention uses 4 heads (embedding dimension 64), and the coefficient α is set to 0.3. For ECAM-related settings, the Singular Space dimension is 10, the number of Adaptive Anchors is 50, the MapNCE temperature is 0.05, and the number of negative samples is 80. At inference phase, 20 samples are generated and Best-of-20 evaluation is applied.

V. EXPERIMENTAL RESULTS

A. Overall Performance Comparison

Table I presents the comparison results averaged over all scenes. The proposed method improves FDE over the baseline Social-STGCNN from 0.81 to 0.50, demonstrating superior performance in destination prediction accuracy. On the other hand, ADE increases from 0.48 to 0.65, indicating degraded average error over the entire prediction sequence. The inference time per sample is 26.64 ms, which is higher than Social-STGCNN (1.02 ms) but lower than Social-LSTM (38.69 ms). These results indicate that the proposed method involves a trade-off between improved endpoint accuracy and the stability of intermediate trajectory predictions as well as computational cost.

TABLE I

COMPARISON RESULTS AVERAGED OVER ALL SCENES.

Method	ADE	FDE	Time (ms)
Social-LSTM	1.38	2.28	38.69
Social-STGCNN	**0.48**	0.81	**1.02**
Our method	0.65	**0.50**	26.64

B. Per-Scene Performance Comparison

Table II presents the per-scene comparison results. The proposed method achieves the best FDE in all five scenes, confirming consistent improvement in destination prediction. In particular, the ETH scene shows improvement in both ADE (from 0.74 to 0.71) and FDE (from 1.23 to 0.47). In contrast, for the others, FDE improves while ADE deteriorates, suggesting that it introduces increased trajectory fluctuation or detour behavior at intermediate time steps. This trend indicates that while integrating environmental information enhances the plausibility of long-term destinations, additional design considerations are needed to suppress them.

Per-scene inference times (ms) are relatively short for ETH (15.18 ms) and HOTEL (15.26 ms), while UNIV exhibits the highest inference time at 59.80 ms. This may reflect that the complex spatial structure increase the processing load for environmental feature integration and candidate generation.

C. Environmental Compliance Evaluation

Table III presents the comparison results of COL. The average rate of the proposed method is 17.34%, which is higher than that of Social-STGCNN at 4.97%. On a per-scene basis, the ETH scene yields a relatively low value of 2.01%, whereas UNIV exhibits a notably high value of 50.98%. In the ETH scene, the low collision rate is consistent with the simpler environment structure. In contrast, UNIV contains complex corridors and crossing structures, suggesting that environmental information integration alone is insufficient.

These results indicate that unresolved challenges remain from the perspective of environmental compliance. In particular, the loss design, negative sample generation strategy, and resolution of the map representation may be insufficient for scenes with high structural complexity.

VI. CONCLUSIONS

In this paper, we proposed a pedestrian trajectory prediction method that builds upon Social-STGCNN and combines local and global environmental information integration with environmental constraint learning. Evaluation on the ETH/UCY dataset demonstrated that the proposed method consistently improves FDE, confirming its effectiveness in enhancing destination prediction accuracy. On the other hand, ADE and COL both deteriorated, and it was found that environmental compliance is not sufficiently ensured particularly in scenes with complex environmental structures. Inference time increased compared to Social-STGCNN but remained shorter than Social-LSTM, confirming that a certain level of practical computational efficiency is maintained.

Future work includes adaptive tuning of environmental constraint weights according to scene characteristics, higher-resolution environment representations, and introduction of losses that directly control intermediate trajectory shapes.

TABLE II

PER-SCENE COMPARISON RESULTS.

Scene	Method	ADE	FDE	Time (ms)
ETH	Social-LSTM	1.37	2.23	37.27
	Social-STGCNN	0.74	1.23	**0.93**
	Our method	**0.71**	**0.47**	15.18
HOTEL	Social-LSTM	1.39	2.32	37.20
	Social-STGCNN	**0.41**	0.67	**1.00**
	Our method	0.61	**0.43**	15.26
UNIV	Social-LSTM	1.42	2.29	42.18
	Social-STGCNN	**0.49**	0.91	**1.00**
	Our method	0.67	**0.75**	59.80
ZARA01	Social-LSTM	1.38	2.29	38.84
	Social-STGCNN	**0.33**	0.52	**1.09**
	Our method	0.63	**0.40**	17.52
ZARA02	Social-LSTM	1.36	2.27	37.95
	Social-STGCNN	**0.30**	0.48	**1.07**
	Our method	0.62	**0.43**	25.43

REFERENCES

[1] D. Helbing and P. Molnar, “Social Force Model for Pedestrian Dynamics,” *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[2] J. von den Berg *et al.*, “Reciprocal n-body Collision Avoidance,” in *Robotics Research*. Springer, 2011, pp. 3–19.

[3] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human Trajectory Prediction in Crowded Spaces,” in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, 2016, pp. 961–971.

[4] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social GAN: Socially Acceptable Trajectories With Generative Adversarial Networks,” in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, 2018, pp. 2255–2264.

[5] R. Akabane and Y. Kato, “Pedestrian Trajectory Prediction Based on Transfer Learning for Human-Following Mobile Robots,” *IEEE Access*, vol. 9, pp. 126 172–126 185, 2021.

[6] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, “Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction,” in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*, 2020, pp. 14 424–14 432.

[7] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, “SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints,” in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, 2019, pp. 1349–1358.

[8] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, “AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting,” in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, 2021, pp. 9813–9823.

[9] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, “Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data,” in *Computer Vision – ECCV 2020*, 2020, pp. 683–700.

[10] G. Rosin, M. R. Ur Rahman, and S. Vascon, “ECAM: A Contrastive Learning Approach to Avoid Environmental Collision in Trajectory Forecasting,” in *2025 International Joint Conference on Neural Networks (IJCNN)*, 2025, pp. 1–8.

[11] S. Pellegrini *et al.*, “You’ll never walk alone: Modeling social behavior for multi-target tracking,” in *IEEE 12th International Conference on Computer Vision*, 2009, pp. 261–268.

[12] A. Lerner, Y. Chrysanthou, and D. Lischinski, “Crowds by Example,” *Computer Graphics Forum*, vol. 26, no. 3, pp. 655–664, 2007.

TABLE III

COMPARISON RESULTS OF OBSTACLE COLLISION RATES (%).

Scene	Social-STGCNN	Our method
ETH	1.38	2.01
HOTEL	4.03	8.07
UNIV	9.69	50.98
ZARA01	2.49	5.81
ZARA02	7.25	19.85
Average	4.97	17.34