Towards Predicting Any Human Trajectory In Context

Ryo Fujii
Keio University
ryo.fujii0112@keio.jp

Hideo Saito
Keio University
hs@keio.jp

Ryo Hachiuma
NVIDIA
rhachiuma@nvidia.com

Abstract

Predicting accurate future trajectories of pedestrians is essential for autonomous systems but remains a challenging task due to the need for adaptability in different environments and domains. A common approach involves collecting scenario-specific data and performing fine-tuning via backpropagation. However, this process is often impractical on edge devices due to constrained computational resources. To address this challenge, we introduce TrajICL, an In-Context Learning (ICL) framework for pedestrian trajectory prediction that enables rapid adaptation without fine-tuning on the scenario-specific data. We propose a spatio-temporal similarity-based example selection (STES) method that selects relevant examples from previously observed trajectories within the same scene by identifying similar motion patterns at corresponding locations. To further refine this selection, we introduce prediction-guided example selection (PG-ES), which selects examples based on both the past trajectory and the predicted future trajectory, rather than relying solely on the past trajectory. This approach allows the model to account for long-term dynamics when selecting examples. Finally, instead of relying on small real-world datasets with limited scenario diversity, we train our model on a large-scale synthetic dataset to enhance its prediction ability by leveraging in-context examples. Extensive experiments demonstrate that TrajICL achieves remarkable adaptation across both in-domain and cross-domain scenarios, outperforming even fine-tuned approaches across multiple public benchmarks. The code will be released at project page.

1 Introduction

Predicting pedestrian trajectories is crucial for applications such as autonomous driving [9], robot navigation [22, 14], and surveillance systems [10]. Recent learning-based methods have shown strong performance in trajectory prediction [1, 19, 48, 64, 24, 36, 37, 18, 38, 50, 16, 15, 53]. However, most existing approaches are trained and evaluated within specific environments or domains, limiting their generalizability. For real-world deployment, autonomous systems must handle diverse scenarios, requiring trajectory prediction models to be robust across varying environments and domains (*e.g.*, map layouts, camera positions, and sensor types). The lack of adaptability to these factors significantly hinders their practical applicability. A common solution involves collecting scenario-specific data and fine-tune models [54, 30, 15]. However, this approach is often impractical on edge devices due to limited computational resources. Furthermore, managing multiple models tailored to different scenarios increases system complexity. Therefore, developing a single, adaptable model capable of generalizing across diverse real-world environments without fine-tuning remains an open challenge.

To address this challenge, we explore the use of In-Context Learning (ICL) [5, 62, 47] for trajectory prediction (Figure 1). ICL enables models to adapt to new tasks using only a few demonstration examples provided as part of the input, without requiring updates to the model weights. Unlike fine-tuning, which modifies model weights through backpropagation, ICL operates solely via forward passes, keeping the model weights fixed. Leveraging this capability, we utilize a small set of examples

Preprint. Under review.

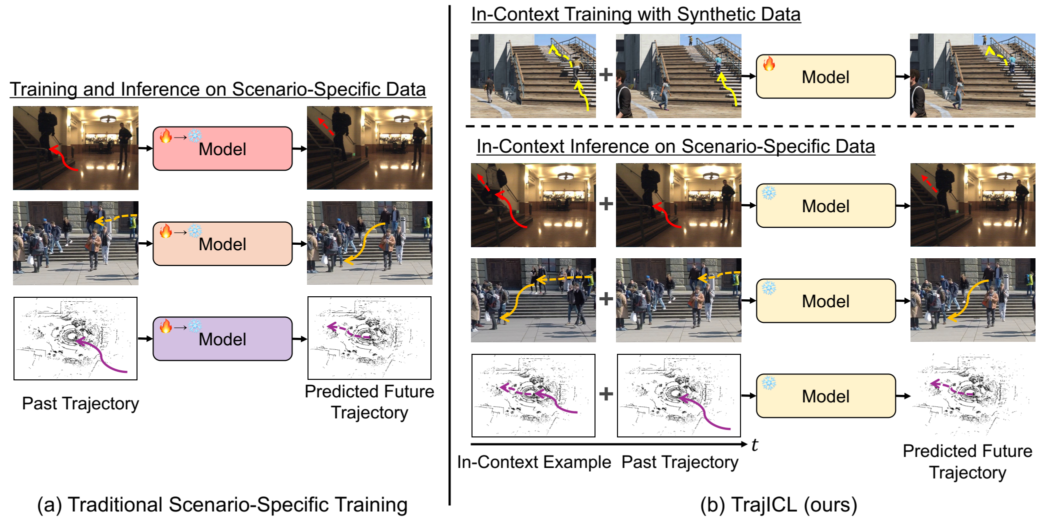


Figure 1: Illustration of real-world trajectory prediction scenarios and the adaptation pipeline. (a) The adaptation pipeline of traditional methods, where models are trained on scenario-specific data. (b) The adaptation pipeline of our proposed TrajICL, which automatically selects examples and adapts to novel scenarios by leveraging the scenario-specific examples without requiring training on scenario-specific data.

to enable a single model to quickly adapt to diverse scenarios. This approach facilitates the reuse of trajectory models across a wide range of scenarios, eliminating the need for costly fine-tuning.

However, incorporating ICL into trajectory prediction presents three major challenges: (1) Emerging ICL capability for trajectory prediction: Randomly selected examples provide minimal ICL capability to the model. Even when examples are selected from the same scene, if they represent different spatial locations or exhibit divergent movements, the effectiveness in improving the model's adaptation is limited. Such variations fail to provide the model with sufficiently relevant context to generalize effectively to new scenarios. (2) Suboptimal selection of examples based on past trajectory input: Existing methods select examples based on similarity to the input query (*i.e.*, past trajectory). However, short past trajectory segments often fail to capture long-term intentions. Moreover, pedestrian motion is inherently multi-modal, where similar past trajectories can lead to divergent future trajectories due to subtle influencing factors. Consequently, relying solely on past trajectories for example selection, without accounting for long-term dynamics, can result in misleading examples and hinder effective in-context adaptation. (3) Adaptability to diverse scenarios: Existing real-world datasets [71, 43, 26, 46, 4, 66, 40] have limited scenario diversity, often focusing on specific environments and interaction patterns. Training on these small-scale datasets restricts ICL's ability to generalize to unseen scenarios, reducing the model's adaptability to a broader range of real-world situations.

To address these challenges, we propose **TrajICL**, an ICL framework for trajectory prediction. First, we introduce a spatio-temporal similarity-based example selection (STES) that identifies relevant examples exhibiting similar motions at comparable locations in the past. By training the model using these spatially and temporally similar examples selected through STES, we enable the development of ICL capability, allowing the model to effectively adapt to new scenarios with minimal examples and without the need for additional training. Second, to address the suboptimal selection based on past trajectory, we propose a prediction-guided example selection (PG-ES) that refines the example selection process using prediction results. PG-ES consists of two phases. First, the model predicts the future trajectory based on its own past trajectory and examples selected according to their similarity with the query past trajectory using the proposed STES. In the second phase, the predicted future trajectory, in addition to the past trajectory, is utilized for example selection. This refinement process allows the model to select more relevant examples by incorporating long-term dynamics. Finally, instead of relying on small real-world datasets with limited scenario diversity, we train the model on a large-scale synthetic dataset [11] to learn predictive capabilities using in-context examples. This allows the trained model to be directly applicable to edge devices in various environments, enhancing its adaptability to real-world conditions.

Our main contributions are as follows: 1) We propose an ICL framework for trajectory prediction, enabling rapid adaptation to diverse scenarios without training on the scenario-specific data. 2) We introduce a spatio-temporal similarity-based example selection (STES) that allows the model to acquire ICL capability. 3) We present prediction-guided example selection (PG-ES), which utilizes both past and predicted future trajectories to select more relevant examples, rather than relying solely on past trajectory input. 4) We leverage large-scale synthetic trajectory datasets to enhance generalization to real-world conditions, addressing the limitations of small-scale real-world datasets to learn predictive capabilities using in-context examples.

2 Related Work

Adapting Trajectory Prediction. Pedestrian trajectory prediction aims to predict future positions based on past trajectories. Deep learning methods demonstrate strong performance due to their representational capabilities [1, 19, 48, 64, 24, 36, 37, 18, 38, 50, 16, 15]. Despite the significant advancements, previous trajectory prediction models are often tailored to specific training domains, limiting their generalizability to new scenarios. To address this, recent research has explored lightweight adaptation techniques for pretrained models. Some approaches focus on cross-domain transfer [65, 21, 30], while others emphasize online adaptation [42, 29, 8], continual learning [39, 63, 52], or prompt tuning-based strategies [54]. While these methods enhance forecasting performance, they often incur high computational costs due to backpropagation on the scenario-specific samples and require multiple models for different scenarios. In contrast, we leverage a small number of examples to seamlessly adapt to diverse scenarios, eliminating the need for costly fine-tuning.

In-Context Learning. In-Context Learning (ICL) [5, 62, 47] enables large language models (LLMs) to perform new tasks by providing a few input-output examples, or demonstrations, alongside the task input. ICL offers several advantages over traditional model adaptation methods, which typically involve pre-training followed by fine-tuning. A key benefit of ICL is that it circumvents the need for fine-tuning, which can be limited by computational resource constraints. Compared to parameter-efficient fine-tuning (PEFT) methods [20, 3], ICL is computationally cheaper and preserves the model's generality by leaving the model parameters unchanged. Following its success in Natural Language Processing (NLP), ICL has been extended to various domains within Computer Vision (CV), including images [3, 58, 59, 70, 2, 44, 67, 57, 27], video [23], point clouds [13, 51], and skeleton sequences [60], showcasing its ability to generalize across diverse, unseen tasks. Recent studies [25, 33] have explored the use of LLMs in vehicle trajectory prediction, highlighting their effectiveness in in-context learning [33]. In contrast, our approach prioritizes equipping significantly lighter trajectory prediction models with in-context learning capability, offering greater efficiency and practicality for real-world applications.

Prompt Selection. The effectiveness of In-Context Learning (ICL) is highly influenced by the selection of relevant examples [32, 41, 28]. Previous studies have shown that selecting in-context examples that are closer to the query improves performance [32, 61, 68]. To address this challenge, several methods have been proposed to select examples that are more similar to the query for ICL [32, 45, 69]. A simple approach is to retrieve the nearest neighbors of the input query based on their similarities [32, 45]. To enhance the robustness of ICL, some studies have employed iterative methods [45, 23, 27, 31]. However, in trajectory prediction tasks, a typical example selection based solely on past trajectory input can be suboptimal. In this work, we introduce a prediction-guided prompt selection approach that selects in-context examples based on their similarity to both the input past trajectory and the predicted future trajectory, aiming to choose more effective examples considering long-term dynamics.

3 TrajICL

3.1 Problem Formulation

Trajectory Prediction with In-Context Learning. Trajectory prediction aims to forecast the future positions of a target agent based on its own past trajectory and the trajectories of surrounding agents. Formally, let $X = (X_1, X_2, \dots, X_N) \in \mathbb{R}^{N \times T_{\text{obs}} \times 2}$ denote the past trajectories of N pedestrians

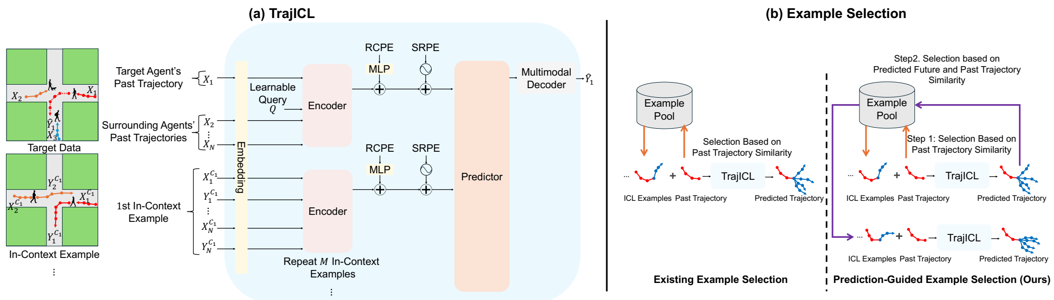


Figure 2: An illustration of our TrajICL framework. (a) The overall architecture includes an embedding layer, a trajectory encoder, an in-context-aware trajectory predictor, and a multi-modal decoder. (b) Rather than relying solely on past trajectories for example selection, we introduce prediction-guided example selection, which leverages both past and predicted future trajectories to identify more relevant examples.

over T_{obs} time steps. The trajectory of the n-th pedestrian is defined as $X_n = (x_1^n, x_2^n, \ldots, x_{T_{\text{obs}}}^n) \in \mathbb{R}^{T_{\text{obs}} \times 2}$, where each position $x_t^n = (u_t^n, v_t^n) \in \mathbb{R}^2$ indicates the location at time t. We consider the primary X_1 as the target pedestrian. Let $Y_1 = (y_1^1, y_2^1, \ldots, y_{T_{\text{pred}}}^1) \in \mathbb{R}^{T_{\text{pred}} \times 2}$ denote the future trajectory of the target pedestrian over T_{pred} time steps, where $y_t^1 = (u_t^1, v_t^1) \in \mathbb{R}^2$. Both X and Y are preprocessed such that the last observed position of the primary agent at time step T_{obs} is shifted to the origin. The goal is to learn a mapping function $\mathcal{F}$ from the past trajectories X to the future trajectory Y_1, such that $Y_1 = \mathcal{F}(X)$.

In this study, we propose leveraging in-context learning to enable trajectory prediction models to adapt to diverse scenarios without the need for updating model parameters through backpropagation. To predict Y_1, we use the past trajectories X in conjunction with an in-context set $\mathcal{C}$: $Y_1 = \mathcal{F}(X, \mathcal{C})$, where $\mathcal{C} = (\tilde{X}^i, \tilde{Y}_1^i)_{i=1}^M$ contains M pairs of past trajectories and the primary agent's ground-truth future trajectory. These example pairs are selected from the example pool, $\mathcal{D}^s$, which consists of pairs of past trajectories and their corresponding ground-truth future trajectories, all coming from the same scene s as the one to which X belongs.

3.2 Spatio-temporal Similarity-based Example Selection (STES)

While random selection of in-context learning examples represents the most straightforward approach, our empirical experiments consistently demonstrate that this method leads to suboptimal performance for trajectory prediction tasks. To address this limitation, we introduce a novel approach called spatio-temporal similarity-based example selection (STES) that automatically identifies the most relevant examples for enhanced trajectory prediction.

The core insight driving our proposed STES method is that trajectories with similar past patterns are likely to exhibit similar future behaviors. Building on this intuition, we propose retrieving the top-M past trajectories along with their corresponding ground-truth future trajectories based on a carefully designed similarity metric. Formally, given a past trajectory query X_1 and an example pool D^s, we identify the most relevant in-context examples as:

$$\mathcal{C} = \text{Top-}M_{\tilde{X}_1^i \in \mathcal{D}^s} \left[S(X_1, \tilde{X}_1^i) \right], \quad |\mathcal{C}| = M, \tag{1}$$

where $\mathcal{C}$ represents the set of top-M relevant trajectories selected from the example pool $\mathcal{D}^s$ according to the similarity metric $S(X_1, \tilde{X}_1^i)$. This metric quantifies the similarity between the target agent's past trajectory and each candidate primary agent's past trajectory in the example pool.

In our introduced STES approach, we define spatio-temporal similarity as $S(X_1, \tilde{X}_1^i) = \sigma(S_p(X_1, \tilde{X}_1^i)) + \sigma(S_v(X_1, \tilde{X}_1^i))$. S_p and S_v are the spatial and temporal similarities definied as follows:

$$S_p(X_1, \tilde{X}_1^i) = \frac{1}{1 + d_p(X_1, \tilde{X}_1^i)}, \quad S_v(X_1, \tilde{X}_1^i) = \frac{1}{1 + d_v(X_1, \tilde{X}_1^i)}, \tag{2}$$

where $d_p(\cdot,\cdot)$ and $d_v(\cdot,\cdot)$ represent the mean squared errors (MSE) of position and velocity, respectively, capturing both spatial proximity and motion similarity between trajectories. $\sigma(\cdot)$ is the normalization function that scales the similarity value within the range $[-1, 1]$ via min-max normalization across the entire set of similarities between the target agent's past trajectory and all candidate trajectories in the example pool.

3.3 Prediction-Guided Example Selection (PG-ES)

To address the suboptimal selection based solely on past trajectory, we propose prediction-guided example selection (PG-ES), which utilizes the trajectory prediction results to refine the example selection process. PG-ES consists of two steps, as shown in Figure 2 (b). In the first step, we predict multiple future trajectories based on the past trajectory and examples selected using past trajectory similarity, as outlined in Equation (1), yielding the prediction result $\hat{Y}_1 = \mathcal{F}(X, \mathcal{C})$. In the second step, these predicted future trajectories, $\hat{Y}_1$, are used to further refine the context selection, as shown in the following equation:

$$\mathcal{C}' = \text{Top-M}_{[\tilde{X}_1^i, \tilde{Y}_1^i] \in \mathcal{D}'^s} \left[\min_{k \in \{1,\dots,K\}} S\left([X_1, \hat{Y}_1^k], [\tilde{X}_1^i, \tilde{Y}_1^i] \right) \right], \quad |\mathcal{C}'| = M. \tag{3}$$

The similarity is computed between the concatenated past trajectory and each of the multiple predicted future trajectories of the target agent, as well as the concatenated past and ground-truth future trajectory of the example. The minimum similarity value across all K predictions is then used as the selection metric. By employing this two-step selection process, we can identify more relevant examples for in-context learning, taking into account both past and future trajectories.

3.4 Model Architecture

Our model consists of an embedding layer, a trajectory encoder, an in-context-aware trajectory predictor, and a multi-modal decoder, as presented in Figure 2 (a). Initially, the embedding layer $\mathcal{G}$ is applied to the agent's past trajectory to obtain d-dimensional features. We then concatenate $\mathcal{G}(X)$ with learnable query tokens $Q \in \mathbb{R}^{N \times T_{\text{pred}} \times d}$ and pass them through the trajectory encoders, Encoder to obtain the trajectory feature (*e.g.*, spatio-temporal features are extracted using a Social-Transmotion encoder [49]) as follows: $[H', Q'] = \text{Encoder}(\mathcal{G}(X), Q)$. Similarly, we use the embedding layer and trajectory encoder to obtain the trajectory features of M in-context examples. Here, instead of learnable queries, we apply $\mathcal{G}$ to both the in-context past trajectories X^i and future trajectories Y^i for all M examples: $[H^{i'}, Z^{i'}] = \text{Encoder}(\mathcal{G}(X^i), \mathcal{G}(Y^i)), \quad i = 1, \dots, M$.

Since the coordinates of the agents in the context examples are normalized to the position of each primary agent, it is crucial to integrate the relative position information of the target agent into the features of the in-context examples. We refer to this as the relative context position encoding (RCPE). RCPE is implemented using a simple one-layer MLP as follows: $\text{RCPE} = \text{MLP}(x_{rel}, y_{rel})$, where (x_{rel}, y_{rel}) represent the relative position of the primary agent of a context example with respect to the target agent. To incorporate information about which context example's primary agent is more similar to the target agent, we introduce similarity rank position encoding (SRPE), which encodes the similarity ranking of each context example's primary agent relative to the target agent. This is implemented using the original sinusoidal positional encoding from [55]. Both RCPE and SRPE are applied to the features of the context examples. These features are then fed into the in-context-aware trajectory predictor (Predictor), which consists of a three-layer Transformer encoder [56], allowing it to leverage the context examples for prediction as follows:

$$\begin{aligned}
[\hat{H}_1, \hat{Q}_1, \hat{H}_1^1, \hat{Z}_1^1, \dots, \hat{H}_1^M, \hat{Z}_1^M] = \text{Predictor}([&H_1', Q_1', \\
&H_1^{1'} + \text{RCPE}(1) + \text{SRPE}(1), Z_1^{1'} + \text{RCPE}(1) + \text{SRPE}(1), \dots, \\
&H_1^{M'} + \text{RCPE}(M) + \text{SRPE}(M), Z_1^{M'} + \text{RCPE}(M) + \text{SRPE}(M)]).
\end{aligned} \tag{4}$$

Finally, we implement the multimodal decoder, PredictionHead, which consists of K simple one-layer MLPs for multimodal prediction. The decoding process can be formulated as follows:

$$\hat{Y}_1 = \text{PredictionHead}(\hat{Q}'_1), \quad \hat{Y}_1 \in \mathbb{R}^{K \times T_{\text{pred}} \times 2}. \tag{5}$$

3.5 Training and Inference

Our framework comprises two sequential training phases—vanilla trajectory prediction (VTP) training and in-context training—followed by an inference phase.

Training. The VTP training phase equips TrajICL with foundational trajectory prediction capabilities. Analogous to conventional trajectory prediction models, TrajICL learns to forecast the future trajectory of a target agent based on its historical path and the trajectories of surrounding agents. The subsequent in-context training phase enables TrajICL to perform effective in-context learning. During this phase, the model trains on examples where similar instances are selected using the proposed metric as detailed in Section 3.2. Both training phases utilize MSE loss, implementing a winner-take-all strategy that optimizes only the most accurate prediction: $\mathcal{L} = \min_k \|\hat{Y}_1^{(k)} - Y_1\|^2$. Note that we only employ a large-scale synthetic dataset to train the model.

Inference. During inference, we adopt the prediction-guided example selection from the scenario-specific samples introduced in Section 3.3, allowing the model to adapt to environmental changes and domain shifts without any additional parameter updates.

4 Experiments

4.1 Experimental Settings

Datasets. We train our model using the MOTSynth [11] dataset, a synthetic pedestrian detection and tracking dataset, consisting of over 700 90-second videos captured from various camera viewpoints in diverse outdoor environments. For our experiments, we use a subset of 424 scenes for training and 107 scenes for evaluation, all captured with a static camera. In addition to in-domain evaluation on MOTSynth, we assess our method on five widely adopted datasets for cross-domain evaluation: JRDB [40], WildTrack [6], SDD [46], and JTA [12]. It is important to note that the model is not trained on these datasets for cross-domain evaluation; only a few examples are used for inference. Following standard practice, we predict 12 future timesteps based on the previous 9 timesteps. For consistency with the human seconds protocol adopted by [54], we sort the N identities chronologically based on their initial appearance in the scene. The earliest 80% of identities are used to construct the example pool, while the remaining 20% are reserved for evaluation.

Evaluation Metrics. We evaluate the performance of different trajectory prediction methods using two standard metrics: minADE$_K$ and minFDE$_K$. minADE$_K$ calculates the minimum average displacement error over time among the $K = 20$ predicted trajectories and the ground-truth future trajectory following the standard protocol [36, 18, 38],. minFDE$_K$ measures the minimum final displacement error, computing the distance between the closest predicted endpoint among the $K = 20$ predictions and the ground-truth endpoint.

Baselines. We combine TrajICL with the transformer-based model Social-Transmotion [49]. Although Social-Transmotion is designed to process multiple modalities, we use only trajectory data as input in all experiments to ensure fair comparisons. In addition, following recent works [7, 17, 15], we incorporate multiple forecasting heads to generate K possible future predictions. To verify the effectiveness of our method, we compare it with versions of Social-Transmotion that have been fine-tuned using various methods, including full fine-tuning (Full FT), PEFT methods such as LoRA [20], VPT [3], and head tuning, which adjusts the prediction heads. The data from the example pool is used for fine-tuning methods to ensure that all models have access to the same information.

Implementation Details. Our training process is divided into two stages: VTP training and in-context training. In the first stage, we train the model using the AdamW optimizer [35] with a base learning rate of 1×10^{-3} for 100 epochs. We perform a 3-epoch warmup and decay the learning rate to 0 throughout training using the cosine annealing scheduler [34]. In the second stage, we train the model for 400 epochs, with a 12-epoch warmup and the cosine annealing scheduler, following the same setup as in the first stage. The hyperparameters were determined through a standard coarse-to-fine grid search or step-by-step tuning. We set the batch size to 16 and train the model using one NVIDIA RTX A6000 GPU. The model configuration for Predictor consists of three layers and four attention

Table 1: Comparison with baseline methods on the MOTSynth, JRDB, WildTrack, SDD, and JTA datasets. $\text{minADE}_K/\text{minFDE}_K$ are reported. The unit for MOTSynth, WildTrack, and SDD is pixels, while the unit for JRDB-World and JTA is meters. **Bold** and <u>underlined</u> fonts represent the best and second-best results, respectively. The difference (Δ) represents the percentage improvement achieved by TrajICL over the vanilla Social-Transmotion.

Method	Training-free	In-Domain	Cross-Domain				
		MOTSynth	JRDB-Image	WildTrack	SDD	JRDB-World	JTA
Social-Transmotion [49]	✓	17.6/23.0	2.88/3.32	24.7/36.3	10.2/18.9	0.15/0.26	1.18/1.97
+Head Tuning	✗	17.1/22.6	2.70/3.13	23.9/34.7	9.81/18.3	0.11/0.16	0.68/1.07
+VPT Shallow [3]	✗	17.8/23.5	2.80/3.33	24.3/37.4	8.73/<u>14.8</u>	0.11/0.16	0.61/0.92
+VPT Deep [3]	✗	17.7/23.9	2.81/3.24	24.4/37.9	8.84/15.6	<u>0.10</u>/<u>0.15</u>	0.60/0.90
+LoRA ($r = 16$) [20]	✗	16.9/22.2	2.64/3.02	23.8/34.5	9.02/16.6	<u>0.10</u>/0.16	0.61/0.93
+LoRA ($r = 64$) [20]	✗	16.8/22.2	2.65/2.98	23.6/35.6	9.11/16.8	<u>0.10</u>/0.16	0.60/0.93
+Full FT	✗	<u>16.0</u>/<u>20.9</u>	**2.56**/<u>2.87</u>	<u>22.9</u>/<u>34.5</u>	**7.96/13.6**	**0.09/0.14**	**0.52/0.76**
+TrajICL (Ours)	✓	**15.3/17.5**	<u>2.61</u>/**2.68**	**21.1/28.3**	<u>8.40</u>/<u>14.8</u>	0.13/0.21	<u>0.59</u>/<u>0.85</u>
Δ		-14.2%/-23.9%	-7.6%/-19.2%	-14.6%/-22.0%	-17.6%/-21.7%	-3.3%/-19.2%	-41.5%/-56.9%

Table 2: Comparisons on JRDB-World and JTA under a few-shot evaluation setting. $\text{minADE}_K/\text{minFDE}_K$ are reported for different percentages of labeled real data available for the example pool for TrajICL and fine-tuning for the fine-tuning methods.

Method	Training-free	MOTSynth		JRDB-Image		WildTrack		SDD		JTA	
		10%	20%	10%	20%	10%	20%	10%	20%	10%	20%
Social-Transmotion [49]	✓	17.6/23.0		2.88/3.32		24.7/36.3		10.2/18.9		1.18/1.97	
+Head Tuning	✗	17.5/23.0	17.4/22.9	3.00/<u>3.50</u>	<u>2.84</u>/3.25	24.5/35.9	24.5/35.2	10.2/19.1	10.2/19.1	0.78/1.32	0.76/1.23
+VPT Shallow [3]	✗	21.3/30.2	20.5/28.7	3.51/4.23	2.98/3.75	25.7/41.5	24.9/39.5	11.2/20.4	10.5/18.1	0.77/1.17	0.70/1.03
+VPT Deep [3]	✗	21.2/29.2	19.8/26.8	3.48/4.16	3.16/3.60	24.4/35.6	24.5/37.2	10.2/<u>18.3</u>	10.3/18.1	0.73/<u>1.09</u>	0.68/1.00
+LoRA ($r = 16$) [20]	✗	17.4/23.0	17.4/22.9	<u>2.94</u>/3.54	2.75/3.24	<u>24.0</u>/**35.0**	23.9/<u>34.0</u>	<u>10.1</u>/19.0	10.1/18.8	0.75/1.22	0.70/1.14
+LoRA ($r = 64$) [20]	✗	17.4/22.9	17.3/22.8	3.00/3.63	2.76/3.20	<u>24.3</u>/34.8	23.8/34.1	<u>10.1</u>/18.8	10.2/19.1	0.74/1.22	0.70/1.13
+Full FT	✗	<u>17.0</u>/<u>22.2</u>	<u>16.8</u>/<u>21.8</u>	3.30/4.56	2.78/<u>3.16</u>	24.5/39.4	23.4/34.3	10.6/18.6	<u>9.83</u>/<u>17.2</u>	<u>0.68</u>/<u>1.09</u>	<u>0.64</u>/<u>0.99</u>
+ TrajICL (Ours)	✓	**16.7/20.4**	**16.4/19.9**	**2.85/3.24**	**2.68/2.94**	**23.4/35.0**	**22.6/33.2**	**9.76/17.4**	**9.60/16.8**	**0.62/0.96**	**0.61/0.90**

heads, with a model dimension of $d = 128$. We employ Leaky ReLU functions as the activation function. Data augmentation techniques, including rotation, noise addition, and masking, as adapted from [15], are applied.

4.2 Comparison with Baseline Methods

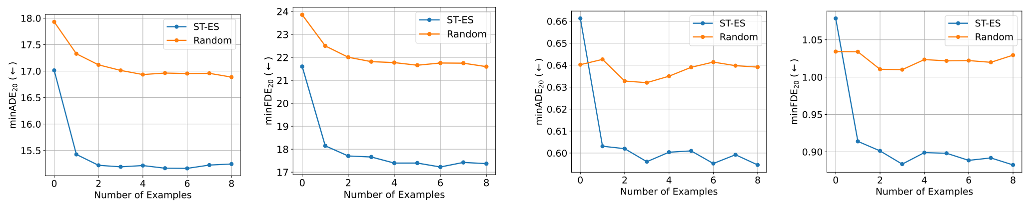
Table 1 shows that TrajICL consistently outperforms Social-Transmotion by a significant margin across all datasets, highlighting its adaptability to a wide range of scenarios. On the MOTSynth, JRDB-Image, WildTrack, and SDD datasets, TrajICL exceeds the performance of the best fine-tuned methods, including full fine-tuning, in terms of minFDE_K. Furthermore, on the JRDB-World and JTA datasets—where pedestrian positions are provided in real-world coordinates—TrajICL remains competitive, demonstrating its generalizability across various sensor configurations. Despite being trained solely on synthetic data, and without requiring any additional fine-tuning, these results showcase TrajICL's effectiveness in diverse scenarios, overcoming the third challenge of adaptability to new environments.

4.3 Comparison with Limited Pool Size

In Section 4.2, we validate the effectiveness of TrajICL using a sufficiently large example pool. However, in real-world scenarios, acquiring annotations manually incurs collection costs, and even when using detection and tracking algorithms, gathering a sufficient example pool takes time. Therefore, we evaluate the effectiveness of TrajICL with a limited pool size, which is a realistic and challenging scenario in real-world applications. We selected subsets of 5% and 10% from the example pool, using each subset as the example pool in TrajICL and as the fine-tuning data for the fine-tuning methods. The models were then evaluated on the same test set. As shown in Table 2, while PEFT methods demonstrate their effectiveness over full fine-tuning in the 100% example setting (Section 4.2), our method consistently outperforms both Social-Transmotion and fine-tuning methods. In the most challenging scenario, with only 10% of annotations, TrajICL achieves improvements of 8.8%, 7.0%, 4.9%, and 11.9% on MOTSynth, JRDB-Image, SDD, and JTA, respectively, compared to the best-performing fine-tuning model in terms of minFDE_K.

Table 3: Ablation study of the prediction-guided STES for inference. minFDE$_K$ is reported. The subscript percentage denotes relative minFDE$_K$ reduction over random selection.

Spatial	temporal	Prediction-guided	MOTSynth	WildTrack	JTA	SDD
			21.8	35.5	1.08	16.4
	✓	✓	18.0$_{-17.4\%}$	<u>28.9</u>$_{-18.6\%}$	**0.85**$_{-21.3\%}$	<u>15.1</u>$_{-7.9\%}$
✓		✓	**16.7**$_{-23.4\%}$	29.4$_{-17.2\%}$	<u>0.89</u>$_{-17.6\%}$	15.9$_{-3.0\%}$
✓	✓		19.2$_{-11.9\%}$	32.5$_{-8.50\%}$	0.91$_{-15.7\%}$	16.2$_{-1.2\%}$
✓	✓	✓	<u>17.5</u>$_{-19.7\%}$	**28.3**$_{-20.3\%}$	**0.85**$_{-21.3\%}$	**14.8**$_{-9.8\%}$



(a) MOTSynth (b) JTA

Figure 3: Performance of random example selection and the proposed STES at varying numbers of in-context examples.

4.4 Ablation Studies

Effectiveness of STES. We first evaluate the impact of incorporating our STES into in-context learning. As shown in Figure 3, training with our STES consistently results in improvements as the number of examples increases across different datasets. In contrast, training with randomly selected examples yields minimal performance gains, even as the number of examples increases in the MOTSynth dataset. In JTA, no improvement is observed despite the increase in examples. These results highlight the effectiveness of training with STES, which efficiently equips the model with ICL capabilities, overcoming the first limitation.

Effectiveness of the PG-STES. We conduct ablation experiments to evaluate the impact of our proposed PG-STES on inference. As shown in Table 3, PG-STES consistently outperforms random selection in terms of minFDE$_K$ across all datasets. By incorporating both the past trajectory and the predicted future trajectory for example selection, PG-STES improves over STES, which only uses past trajectory for selection, by 8.1%, 7.4%, 6.6%, and 8.6% on the MOTSynth, WildTrack, JTA, and SDD datasets in terms of minFDE$_K$, respectively. These results demonstrate the effectiveness of combining predicted future trajectories with past trajectories to enhance performance, addressing the second limitation. We also investigate the impact of spatial and temporal components in the STES example selection process. The results indicate that temporal similarity yields greater improvements than spatial similarity on the JTA and SDD datasets. However, on the MOTSynth dataset, spatial similarity provides more substantial gains. This suggests that the optimal selection dimension for demonstration examples is dataset-dependent. When both spatial and temporal similarities are utilized together, our method achieves the best performance across most datasets, particularly when prediction-guided retrieval is applied.

Effectiveness of VTP Training. Our experiments demonstrate that VTP training consistently delivers performance improvements across multiple datasets, as shown in Table 4.

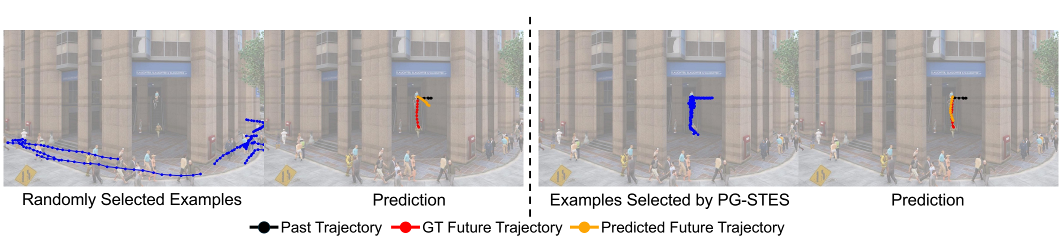


Figure 4: Qualitative comparison between random example selection and our proposed PG-STES.

Table 4: Ablation study of first-stage VTP training. minFDE$_K$ is reporeted.

VT	MOTSynth	WildTrack	JTA
	15.4/17.6	21.9/30.4	0.60/0.89
✓	**15.3/17.5**	**21.1/28.3**	**0.59/0.85**

Table 5: Ablation study of RCPE and SRPE.

RCPE	SRPE	MOTSynth	WildTrack	JRDB-World	JTA
		15.5/18.1	21.9/31.0	0.14/0.23	0.61/0.91
✓		15.4/17.6	21.8/29.9	0.14/0.24	0.59/0.88
	✓	**15.3**/17.6	21.8/29.6	0.14/0.23	0.59/0.89
✓	✓	**15.3/17.5**	**21.1 /28.3**	**0.13/0.21**	**0.59/0.85**

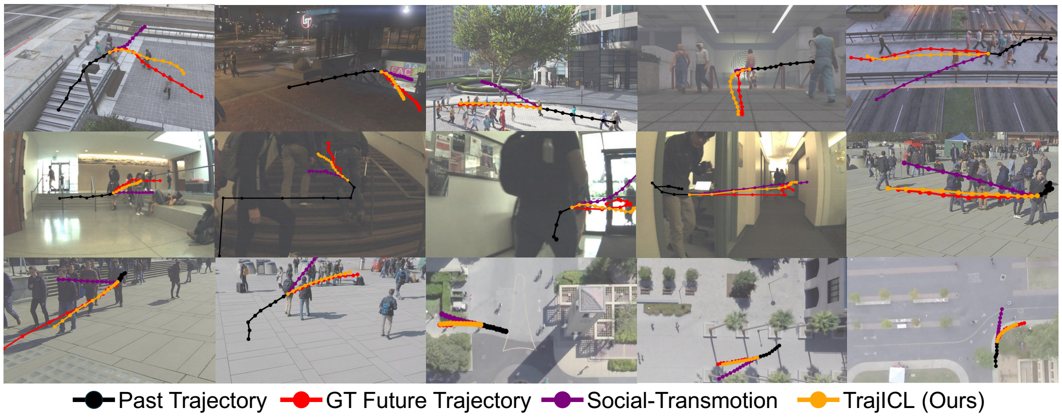


Figure 5: Qualitative results on MotSynth, JRDB, WildTrack, and SDD. These examples demonstrate scenarios where our TrajICL outperforms the Social-Transmotion baseline. TrajICL effectively learns the plausible motion patterns from examples.

Effectiveness of RCPE and SRPE. We investigate the impact of RCPE, which encodes the relative position with respect to the target agent, and SRPE, which encodes the similarity rank of the context examples' primary agents in relation to the target agent. As shown in Table 5, our ablation study demonstrates that both RCPE and SRPE contribute to improved performance across various datasets. On the in-domain MOTSynth dataset, RCPE alone achieves a slight improvement in performance. However, on the cross-domain datasets, WildTrack and JRDB-World, the combination of both RCPE and SRPE results in more significant improvements.

4.5 Qualitative Results

We compare randomly selected examples with those chosen by our PG-STES in Figure 4, along with their prediction results. PG-STES effectively selects spatially and temporally similar examples, enabling our model to generate more plausible predictions, such as a pedestrian riding down an escalator, with a better understanding of 3D structures compared to random selection. Figure 5 highlights the qualitative results of TrajICL and Social-Transmotion across various datasets, showcasing our method's adaptability in predicting future trajectories across domains. Unlike the Social-Transmotion baseline, which often predicts pedestrians floating in the air, our model aligns closely with the ground truth, even on non-planar surfaces like stairs. Furthermore, our approach incorporates finer-grained map awareness, avoiding obstacles like trees and respecting constraints (*e.g.*, not crossing fences), while capturing behavioral trends such as walking on sidewalks instead of roads.

5 Conclusion

In this paper, we introduce TrajICL, a novel in-context learning (ICL) framework for pedestrian trajectory prediction that enables rapid adaptation without the need for fine-tuning on the domain-specific data. We address the challenges of incorporating ICL into trajectory prediction by employing spatio-temporal similarity-based example selection, prediction-guided example selection, and leveraging a large-scale synthetic trajectory dataset. In our experiments, we thoroughly validate that our approach effectively adapts to environmental variations and domain shifts. Despite these promising results, there remains work to be done. While increasing the number of in-context examples improves accuracy, it also raises computational costs. We plan to explore this further in our future work.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *CVPR*, 2016.

[2] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L. Yuille, Trevor Darrell, Jitendra Malik, and Alexei A. Efros. Sequential Modeling Enables Scalable Learning for Large Vision Models. In *CVPR*, 2024.

[3] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *NeurIPS*, 2022.

[4] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections. In *IV*, 2020.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[6] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. WILDTRACK: A Multi-Camera HD Dataset for Dense Unscripted Pedestrian Detection. In *CVPR*, 2018.

[7] Jie Cheng, Xiaodong Mei, and Ming Liu. Forecast-MAE: Self-supervised pre-training for motion forecasting with masked autoencoders. *CVPR*, 2023.

[8] Yujiao Cheng, Weiye Zhao, Changliu Liu, and Masayoshi Tomizuka. Human motion prediction using semi-adaptable neural networks. In *ACC*, 2019.

[9] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks. In *ICRA*, 2019.

[10] Omar Elharrouss, Noor Almaadeed, and Somaya Al-Maadeed. A review of video surveillance systems. *JVCIR*, 2021.

[11] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljoša Ošep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking? In *ICCV*, 2021.

[12] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to Detect and Track Visible and Occluded Body Joints in a Virtual World. In *ECCV*, 2018.

[13] Zhongbin Fang, Xiangtai Li, Xia Li, Joachim M Buhmann, Chen Change Loy, and Mengyuan Liu. Explore in-context learning for 3d point cloud understanding. *NeurIPS*, 2024.

[14] Amalia Foka and Panos Trahanias. Probabilistic Autonomous Robot Navigation in Dynamic Environments with Human Motion Prediction. *IJSR*, 2010.

[15] Ryo Fujii, Hideo Saito, and Ryo Hachiuma. RealTraj: Towards Real-World Pedestrian Trajectory Forecasting. *arXiv preprint arXiv:2411.17376*, 2024.

[16] Ryo Fujii, Jayakorn Vongkulbhisal, Ryo Hachiuma, and Hideo Saito. A Two-Block RNN-Based Trajectory Prediction From Incomplete Trajectory. *IEEE Access*, 2021.

[17] Yang Gao, Po-Chien Luan, and Alexandre Alahi. Multi-Transmotion: Pre-trained Model for Human Motion Prediction. In *CoRL*, 2024.

[18] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic Trajectory Prediction via Motion Indeterminacy Diffusion. In *CVPR*, 2022.

[19] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. In *CVPR*, 2018.

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022.

[21] Boris Ivanovic, James Harrison, and Marco Pavone. Expanding the deployment envelope of behavior prediction via adaptive meta-learning. In *ICRA*, 2023.

[22] Nikolay Jetchev and Marc Toussaint. Trajectory prediction: learning to map situations to robot trajectories. In *ICML*, 2009.

[23] Kangsan Kim, Geon Park, Youngwan Lee, Woongyeong Yeo, and Sung Ju Hwang. VideoICL: Confidence-based Iterative In-context Learning for Out-of-Distribution Video Understanding. In *CVPR*, 2025.

[24] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, S. Hamid Rezatofighi, and Silvio Savarese. Social-BiGAT: Multimodal Trajectory Forecasting Using Bicycle-GAN and Graph Attention Networks. In *NeurIPS*, 2019.

[25] Zhengxing Lan, Lingshan Liu, Bo Fan, Yisheng Lv, Yilong Ren, and Zhiyong Cui. Traj-LLM: A New Exploration for Empowering Trajectory Prediction With Pre-Trained Large Language Models. *T-IV*, 2024.

[26] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an Image-Based Motion Context for Multiple People Tracking. In *CVPR*, 2014.

[27] Seung Hyun Lee, Jijun Jiang, Yiran Xu, Zhuofang Li, Junjie Ke, Yinxiao Li, Junfeng He, Steven Hickson, Katie Datsenko, Sangpil Kim, Ming-Hsuan Yang, Irfan Essa, and Feng Yang. Cropper: Vision-Language Model for Image Cropping through In-Context Learning. In *CVPR*, 2025.

[28] Junlong Li, Jinyuan Wang, Zhuosheng Zhang, and Hai Zhao. Self-prompting large language models for zero-shot open-domain QA. In *NAACL*, 2024.

[29] Maosen Li, Siheng Chen, Yanning Shen, Genjia Liu, Ivor W Tsang, and Ya Zhang. Online multi-agent forecasting with interpretable collaborative graph neural networks. *TNNLS*, 2022.

[30] Yiheng Li, Seth Z. Zhao, Chenfeng Xu, Chen Tang, Chenran Li, Mingyu Ding, Masayoshi Tomizuka, and Wei Zhan. Pre-training on Synthetic Driving Data for Trajectory Prediction. In *IROS*, 2024.

[31] Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. se^2: Sequential example selection for in-context learning. In *ACL*, 2024.

[32] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What Makes Good In-Context Examples for GPT-3? In *DeeLIO*, 2022.

[33] Jiawei Liu, Yanjiao Liu, Xun Gong, Tingting Wang, Hong Chen, and Yunfeng Hu. Harnessing and Evaluating the Intrinsic Extrapolation Ability of Large Language Models for Vehicle Trajectory Prediction. In *NAACL*, 2025.

[34] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *ICLR*, 2017.

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[36] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From Goals, Waypoints & Paths to Long Term Human Trajectory Forecasting. In *ICCV*, 2021.

[37] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It Is Not the Journey But the Destination: Endpoint Conditioned Trajectory Prediction. In *ECCV*, 2020.

[38] Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog Diffusion Model for Stochastic Trajectory Prediction. In *CVPR*, 2023.

[39] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Mantra: Memory augmented networks for multiple trajectory prediction. In *CVPR*, 2020.

[40] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments. *TPAMI*, 2021.

[41] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *EMNLP*, 2022.

[42] Daehee Park, Jaeseok Jeong, Sung-Hoon Yoon, Jaewoo Jeong, and Kuk-Jin Yoon. T4P: Test-Time Training of Trajectory Prediction via Masked Autoencoder and Actor-specific Token Memory. In *CVPR*, 2024.

[43] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings. In *ECCV*, 2010.

[44] Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, et al. UniControl: A Unified Diffusion Model for Controllable Visual Generation In the Wild. *NeurIPS*, 2023.

[45] Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. In-context learning with iterative demonstration selection. In *EMNLP*, 2024.

[46] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes. In *ECCV*, 2016.

[47] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning To Retrieve Prompts for In-Context Learning. In *NAACL*, 2022.

[48] Saeed Saadatnejad, Yang Gao, Kaouther Messaoud, and Alexandre Alahi. Social-Transmotion: Promptable Human Trajectory Prediction. In *ICLR*, 2024.

[49] Saeed Saadatnejad, Yang Gao, Kaouther Messaoud, and Alexandre Alahi. Social-Transmotion: Promptable Human Trajectory Prediction. In *ICLR*, 2024.

[50] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data. In *ECCV*, 2020.

[51] Feifei Shao, Ping Liu, Zhao Wang, Yawei Luo, Hongwei Wang, and Jun Xiao. MICAS: Multi-grained In-Context Adaptive Sampling for 3D Point Cloud Processing. In *CVPR*, 2025.

[52] Jianhua Sun, Yuxuan Li, Hao-Shu Fang, and Cewu Lu. Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In *ICCV*, 2021.

[53] Hiromu Taketsugu, Takeru Oba, Takahiro Maeda, Shohei Nobuhara, and Norimichi Ukita. Physical plausibility-aware trajectory prediction via locomotion embodiment. In *CVPR*, 2025.

[54] Neerja Thakkar, Karttikeya Mangalam, Andrea Bajcsy, and Jitendra Malik. Adaptive Human Trajectory Prediction via Latent Corridors. In *ECCV*, 2024.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*, 2017.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*, 2017.

[57] Chaoyang Wang, Xiangtai Li, Henghui Ding, Lu Qi, Jiangning Zhang, Yunhai Tong, Chen Change Loy, and Shuicheng Yan. Explore In-Context Segmentation via Latent Diffusion Models. In *AAAI*, 2024.

[58] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, 2023.

[59] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. SegGPT: Towards Segmenting Everything In Context. In *ICCV*, 2023.

[60] Xinshun Wang, Zhongbin Fang, Xia Li, Xiangtai Li, Chen Chen, and Mengyuan Liu. Skeleton-in-Context: Unified Skeleton Sequence Modeling with In-Context Learning. In *CVPR*, 2024.

[61] Zhenyu Wu, Yaoxiang Wang, Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Jingjing Xu, and Yu Qiao. OpenICL: An Open-Source Framework for In-context Learning. In *ACL*, 2023.

[62] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-context Learning as Implicit Bayesian Inference. In *ICLR*, 2022.

[63] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember Intentions: Retrospective-Memory-based Trajectory Prediction. In *CVPR*, 2022.

[64] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. EqMotion: Equivariant Multi-agent Motion Prediction with Invariant Interaction Reasoning. In *CVPR*, 2023.

[65] Yi Xu, Lichen Wang, Yizhou Wang, and Yun Fu. Adaptive trajectory prediction via transferable gnn. In *CVPR*, 2022.

[66] Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating Multi-Agent Trajectories using Programmatic Weak Supervision. In *ICLR*, 2019.

[67] Jiahao Zhang, Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Instruct Me More! Random Prompting for Visual In-Context Learning. In *WACV*, 2024.

[68] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. What makes good examples for visual in-context learning? In *NeurIPS*, 2023.

[69] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic Chain of Thought Prompting in Large Language Models. In *ICLR*, 2023.

[70] Zhen Zhao, Jingqun Tang, Chunhui Lin, Binghong Wu, Can Huang, Hao Liu, Xin Tan, Zhizhong Zhang, and Yuan Xie. Multi-modal In-Context Learning Makes an Ego-evolving Scene Text Recognizer. In *CVPR*, 2024.

[71] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors: Learning a Mixture model of Dynamic pedestrian-Agents. In *CVPR*, 2012.

Table 6: Comparison with baseline methods on the ETH-UCY [43, 26]. $\text{minADE}_K/\text{minFDE}_K$ are reported. The unit is meters. **Bold** and underlined fonts represent the best and second-best results, respectively. The difference (Δ) represents the percentage improvement achieved by TrajICL over the vanilla Social-Transmotion.

Method	Training-free	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Social-Trans [49]	✓	0.42/0.79	0.11/0.19	0.33/0.59	0.30/0.58	0.26/0.46	0.28/0.52
+Head Tuning	✗	0.46/0.85	0.17/0.31	0.30/0.53	0.25/0.47	0.23/0.42	0.28/0.52
+VPT Shallow [3]	✗	0.43/0.82	0.11/0.17	0.27/0.46	0.21/0.40	0.19/0.31	0.24/0.43
+VPT Deep [3]	✗	0.46/0.78	0.09/**0.13**	0.25/0.44	**0.19/0.35**	**0.17/0.29**	0.23/0.40
+LoRA ($r = 16$) [20]	✗	0.38/0.78	0.09/**0.13**	0.24/0.40	**0.19**/0.36	**0.17**/0.30	**0.21**/0.39
+LoRA ($r = 64$) [20]	✗	0.44/0.78	0.09/**0.13**	0.25/0.43	0.20/0.37	**0.17**/0.30	0.23/0.40
+Full FT	✗	0.36/**0.64**	**0.09/0.13**	**0.22/0.39**	**0.19**/0.37	0.18/0.32	**0.21/0.37**
+TrajICL (Ours)	✓	**0.34/0.64**	0.10/**0.13**	0.28/0.48	0.21/0.40	0.24/0.40	0.23/0.41
Δ		-19.0%/-9.1%	-7.6%/-31.6%	-15.1%/-18.6%	-36.7%/-31.0%	-7.7%/-13.0%	-17.9%/-21.2%

A Supplementary Material

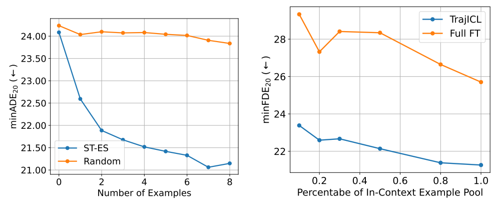
A.1 More Experiments

Comparison with Baseline Methods on ETH-UCY. Table 6 presents a performance comparison on the ETH-UCY. Across all subsets, TrajICL consistently surpasses Social-Transmotion by a notable margin, demonstrating its strong adaptability and robustness in diverse scenarios.

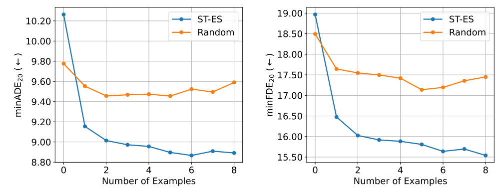
Effectiveness of the STES. Figure 6 demonstrates the impact of integrating our STES into in-context learning on the WildTrack SDD, JRDB, and JRDB-Image datasets. The inclusion of in-context examples consistently improves accuracy across all datasets, underscoring the effectiveness of training with STES in efficiently enhancing the model's ICL capability. Furthermore, as shown in Figure 6, we validate the effectiveness of STES with a larger number of examples. While training with STES continues to yield improvements as the number of examples increases across different datasets, the accuracy gains tend to saturate beyond a certain point.

Effect of Pool Size. We next investigate the impact of varying the size of the in-context pool. As shown in Figure 8, experiments on MotSynth reveal that increasing the number of examples in the in-context pool leads to improved performance. TrajICL consistently outperforms all fine-tuning methods on MotSynth. On the other hand, for JTA, when the in-context pool is small, TrajICL surpasses the fine-tuning methods by effectively preventing overfitting, as it does not require additional parameter updates. However, as the pool size increases, full fine-tuning begins to achieve better results on JTA.

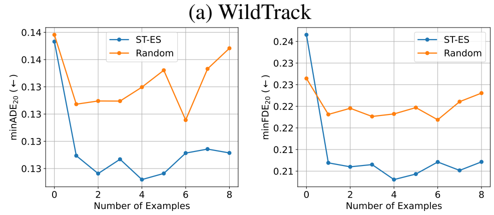
More Qualitative Results. We present further qualitative comparisons of TrajICL and Social-Transmotion on the MOTSynth, JRDB-Image, WildTrack, and SDD datasets in Figure 9. Compared to the Social-Transmotion baseline, our model demonstrates closer alignment with the ground truth by incorporating finer-grained map awareness and effectively avoiding obstacles.



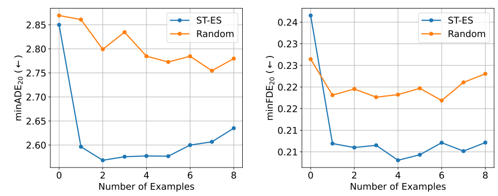
(a) WildTrack



(b) SDD

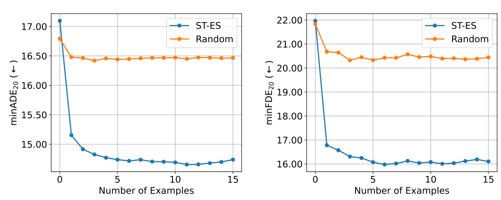


(a) JRDB

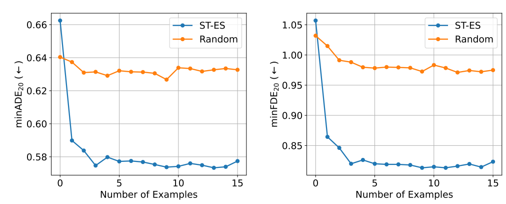


(b) JRDB-Image

Figure 6: Performance of random example selection and the proposed STES at varying numbers of in-context examples.

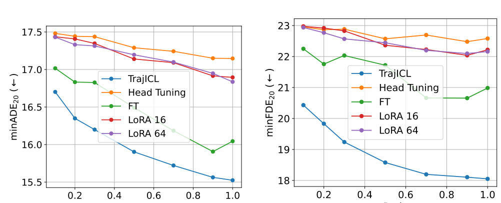


(a) MOTSynth

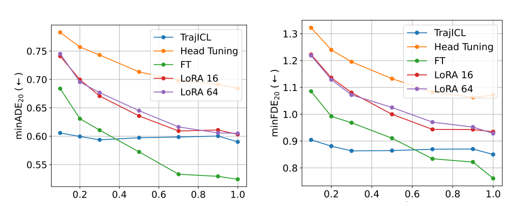


(b) JTA

Figure 7: Performance of random example selection and the proposed STES at varying numbers of in-context examples with a larger number of examples



(a) MOTSynth



(b) JTA

Figure 8: Performance change brought by different sizes of the in-context pool.

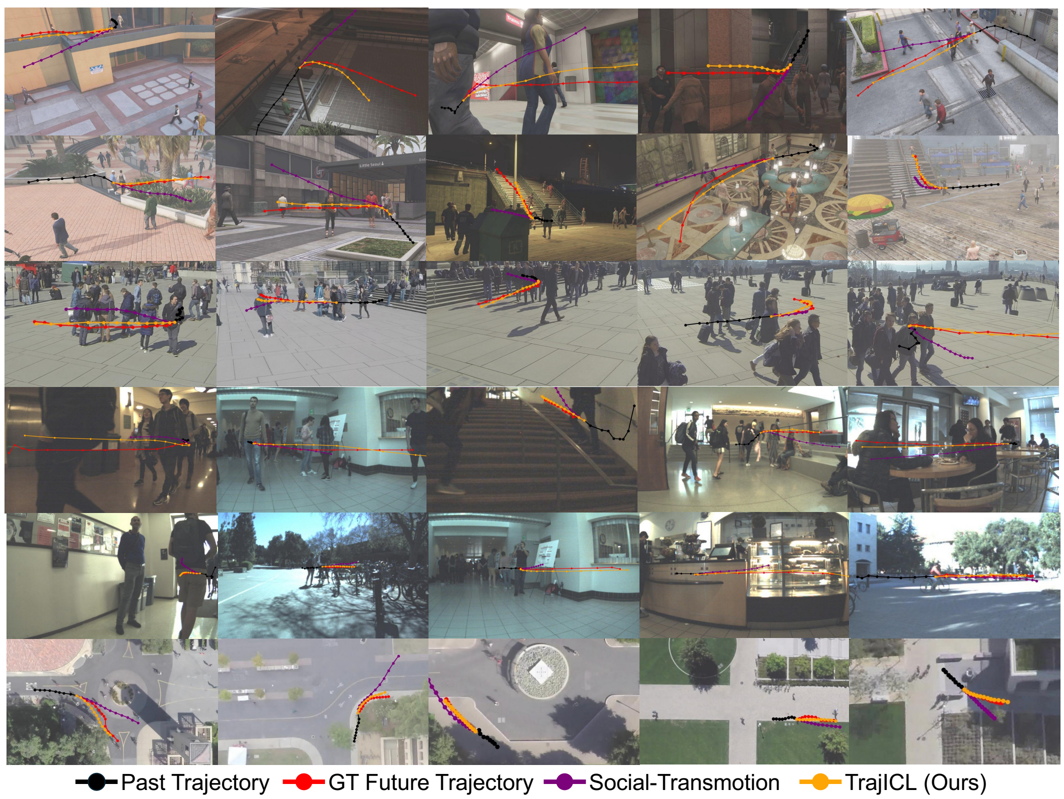


Figure 9: Additional qualitative results on MotSynth, JRDB, WildTrack, and SDD.