

TRAJEVO: Trajectory Prediction Heuristics Design via LLM-driven Evolution

Zhikai Zhao^{1*}, Chuanbo Hua^{1*}, Federico Berto^{1,2*}, Kanghoon Lee¹, Zihan Ma¹,
Jiachen Li³, Jinkyoo Park^{1,4}

¹KAIST, Daejeon, South Korea

²Radical Numerics, Palo Alto, CA, USA

³University of California, Riverside, CA, USA

⁴OMELET, Daejeon, South Korea

{zkk020202, cbhua, fberto, leehoon, zihanma}@kaist.ac.kr; jiachen.li@ucr.edu; jinkyoo.park@kaist.ac.kr

Abstract

Trajectory prediction is a critical task in modeling human behavior, especially in safety-critical domains such as social robotics and autonomous vehicle navigation. Traditional heuristics based on handcrafted rules often lack accuracy and generalizability. Although deep learning approaches offer improved performance, they typically suffer from high computational cost, limited explainability, and, importantly, poor generalization to out-of-distribution (OOD) scenarios. In this paper, we introduce TRAJEVO, a framework that leverages Large Language Models (LLMs) to automatically design trajectory prediction heuristics. TRAJEVO employs an evolutionary algorithm to generate and refine prediction heuristics from past trajectory data. We propose two key innovations: Cross-Generation Elite Sampling to encourage population diversity, and a Statistics Feedback Loop that enables the LLM to analyze and improve alternative predictions. Our evaluations demonstrate that TRAJEVO outperforms existing heuristic methods across multiple real-world datasets, and notably surpasses both heuristic and deep learning methods in generalizing to an unseen OOD real-world dataset. TRAJEVO marks a promising step toward the automated design of fast, explainable, and generalizable trajectory prediction heuristics. We release our source code to facilitate future research.

Code — <https://github.com/ai4co/trajevo>

Extended version — <https://arxiv.org/abs/2508.05616>

Introduction

Trajectory prediction is a cornerstone of intelligent autonomous systems (Madjid et al. 2025), with numerous real-world applications, including autonomous driving (Wang et al. 2025a,c; Li et al. 2024b; Lange, Li, and Kochenderfer 2024; Wang et al. 2025b), industrial safety (Robla-Gómez et al. 2017), robotic navigation (Li et al. 2024a; Yao et al. 2024), and planning (Li et al. 2022b, 2023b). The inherent stochasticity of these environments demands systems that can accurately process data with both temporal and spatial precision (Li et al. 2021). For instance, in autonomous vehicle navigation, accurately predicting pedestrian trajectories

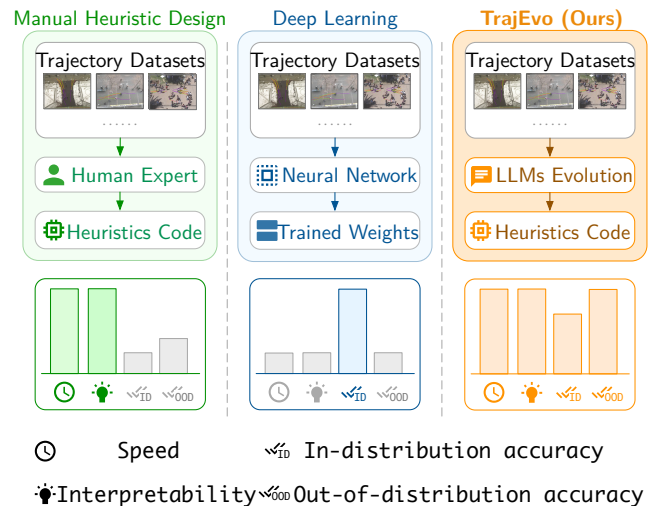


Figure 1: Motivation for our TRAJEVO framework. Traditional manual heuristic design (left) relies on human expertise and trial-and-error processes. Deep learning approaches (center) produce more accurate predictions but demand substantial computational resources, yield black-box models, and often struggle with generalization. TRAJEVO (right) automates the heuristic design process using evolutionary algorithms, enabling the generation of novel and effective trajectory prediction heuristics.

is essential for avoiding collisions in complex urban settings (Cao et al. 2021; Li et al. 2023a; Choi et al. 2021). Similarly, in industrial and indoor environments, robots must generate precise trajectory predictions to collaborate safely with humans in shared workspaces without causing harm (Ma et al. 2022).

Accurately predicting the movement patterns of multiple agents, such as pedestrians or vehicles, remains fundamentally challenging. Human motion is inherently complex, characterized by nonlinear behaviors, sudden directional changes, and spontaneous decisions influenced by individual factors such as habits or urgency (Li et al. 2020; Xu et al. 2024). Moreover, agents interact dynamically, adjusting their paths to avoid collisions, responding to traf-

*These authors contributed equally.

fic, or moving cohesively in groups, which results in high stochasticity (Zhou et al. 2022; Li et al. 2022a). Early efforts to model these behaviors relied on heuristic methods, including the Social Force model (Helbing and Molnar 1995), which captures interactions through physics-inspired forces (Farina et al. 2017; Chen et al. 2018); constraint-based approaches that define collision-free velocities (Ma, Manocha, and Wang 2018); and agent-based simulations of decision making processes (Yamaguchi et al. 2011). While these handcrafted heuristics offer interpretability, they are often difficult to tune for dynamic scenarios and tend to suffer from limited accuracy and generalization.

Deep learning methods have emerged as a powerful alternative to traditional heuristics (Salzmann et al. 2020). Social-LSTM (Alahi et al. 2016) was among the earliest influential models, leveraging LSTMs for trajectory prediction. Since then, a wide range of approaches, such as graph neural networks (GNNs) (Rainbow, Men, and Shum 2021; Ma et al. 2021) and generative adversarial networks (GANs) (Gupta et al. 2018; Li, Ma, and Tomizuka 2019), have been proposed to further improve prediction accuracy. More recently, Transformer-based frameworks (Kim et al. 2025) and diffusion models (Fu et al. 2025) have shown promise in capturing complex dependencies in human motion patterns.

However, deep learning methods suffer from several practical limitations. *a) Running Speed:* They often require significant computational resources and exhibit high latency, making them unsuitable for deployment on resource-constrained robots or vehicles (Itkina and Kochenderfer 2023). *b) Interpretability:* As black-box models, they lack transparency, which hinders verification and reduces trust, especially in safety-critical applications (Dax et al. 2023; Liu et al. 2024c). *c) Generalization Ability:* Perhaps most critically, they often generalize poorly in out-of-distribution (OOD) scenarios, potentially leading to unsafe behavior in unfamiliar environments (Rudenko et al. 2020). These challenges highlight the continued importance of robust and predictable heuristics in safety-critical systems (Phong et al. 2023), which requires methods that are not only accurate but also fast, explainable, and generalizable.

Therefore, we pose the following research question: *Can we automatically design computationally efficient, accurate, interpretable, and highly generalizable trajectory prediction heuristics?*

Inspired by recent research exploring combinations of Large Language Models (LLMs) with Evolutionary Algorithms (EAs) for automated algorithm design (Dat, Doan, and Binh 2025; Ye et al. 2024; Liu et al. 2024a,b; Zheng et al. 2025; Wu et al. 2024), we propose a novel approach to automatically generate effective prediction heuristics. We hypothesize that coupling the generative and reasoning capabilities of LLMs with the structured search of EAs can overcome the limitations of manual heuristic design to discover novel, high-performance heuristics suitable for real-world deployment.

In this paper, we introduce TRAJEVO (**T**rajectory **E**volution), a novel framework designed to achieve this goal. TRAJEVO leverages LLMs within an evolutionary loop to iteratively generate, evaluate, and refine prediction heuristics

directly from data. Our main contributions are as follows:

- We present TRAJEVO, the first framework, to the best of our knowledge, that integrates LLMs with evolutionary algorithms specifically for the automated discovery and design of fast, explainable, and robust trajectory prediction heuristics for real-world applications.
- We introduce a Cross-Generation Elite Sampling strategy to maintain population diversity and a Statistics Feedback Loop that enables the LLM to analyze heuristic performance and guide the generation of improved candidates based on past trajectory data.
- We demonstrate that TRAJEVO generates heuristics that significantly outperform prior heuristic methods on public-open real-world datasets and exhibit remarkable generalization, achieving an over 20% performance improvement on an unseen OOD dataset against both traditional heuristics and deep learning methods, while remaining computationally fast and interpretable.

Related Work

Heuristic Methods for Trajectory Prediction Traditional heuristic approaches provide interpretable frameworks for trajectory prediction, but often with accuracy limitations. The Constant Velocity Model (CVM) and its sampling variant (CVM-S) (Schöller et al. 2020) represent foundational baselines that assume uniform motion patterns. More sophisticated approaches include Constant Acceleration (Polychronopoulos et al. 2007), CTRV (Constant Turn Rate and Velocity) (Lu et al. 2021), and CSCRCTR (Zhai, Meng, and Wang 2014), which incorporate different kinematic assumptions. The Social Force Model (Helbing and Molnar 1995) pioneered physics-inspired representations of pedestrian dynamics, with numerous extensions incorporating social behaviors (Chen et al. 2018) and group dynamics (Helbing, Farkas, and Vicsek 1997). Despite their efficiency and explainability, they typically struggle with complex real-world, multi-agent interactions due to their limited expressiveness and reliance on manually defined parameters – limitations our proposed TRAJEVO framework specifically addresses through automatic heuristic generation.

Learning-based Trajectory Prediction Deep learning has substantially advanced trajectory prediction accuracy at the cost of computational efficiency and interpretability (Sun et al. 2022; Xie, Li, and Wang 2023). Social-LSTM (Alahi et al. 2016) is a seminal work using a social pooling mechanism to model inter-agent interactions, while Social-GAN (Gupta et al. 2018) leveraged generative approaches to capture trajectory multimodality. Graph-based architectures like STGAT (Huang et al. 2019) and Social-STGCNN (Mohamed et al. 2020) explicitly model the dynamic social graph between pedestrians. Recent approaches include Trajectron++ (Salzmann et al. 2020), which integrates various contextual factors, MemoNet (Xu et al. 2022) with its memory mechanisms, EigenTrajectory (Bae, Oh, and Jeon 2023) focusing on principal motion patterns, HyperSTTN (Wang et al. 2024) utilizing hypergraph-based spatial-temporal transformers for group-aware trajectory predic-

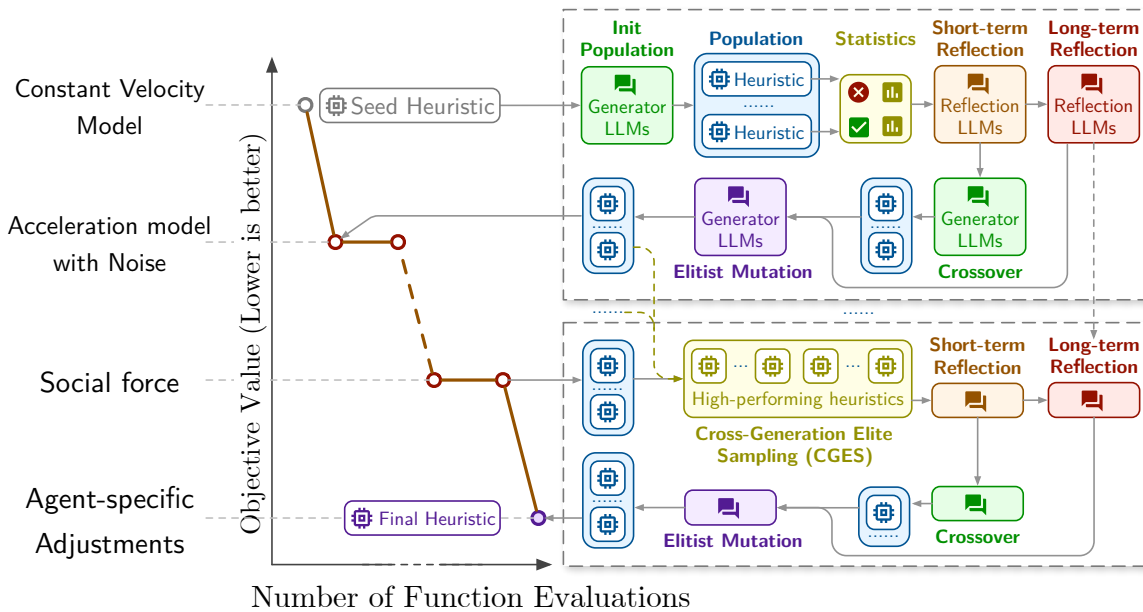


Figure 2: An illustration of the TRAJEVO evolutionary process. **(Left)** The framework continuously discovers and evaluates a variety of heuristic strategies, such as those based on constant velocity or social forces. **(Middle)** As the evolution progresses, the performance of these heuristics steadily improves, evidenced by the decreasing objective value from a simple seed to a final, optimized solution. **(Right)** The overall TRAJEVO pipeline orchestrates this entire discovery and optimization process, using an LLM-driven evolutionary algorithm to automatically generate and refine the heuristics.

tion, and MoFlow (Fu et al. 2025) employing normalizing flows for stochastic prediction. While these methods achieve high in-distribution accuracy, their three well-documented challenges: high computational cost, lack of interpretability, and poor out-of-distribution generalization (Rudenko et al. 2020), motivate the design of our TRAJEVO framework.

LLMs for Algorithmic Design Recent work has combined Large Language Models (LLMs) with Evolutionary Algorithms (EAs) for automated algorithm design (Ye et al. 2024; Dat, Doan, and Binh 2025). However, applying this paradigm to the safety-critical and stochastic domain of trajectory prediction remains unexplored. To the best of our knowledge, TRAJEVO is the first framework to adapt LLM-driven EAs for discovering fast, interpretable, and generalizable trajectory prediction heuristics.

While these studies establish the general potential of LLM-driven evolution, its application to specialized, safety-critical domains remains largely unexplored. The domain of multi-agent trajectory prediction presents unique challenges, including stochastic interactions and the need for both high accuracy and computational efficiency. To the best of our knowledge, TRAJEVO is the first framework to adapt the LLM-driven evolutionary paradigm specifically for this task.

TRAJEVO

Problem Definition

Task We address multi-agent trajectory prediction. The task is, for each agent i , to predict its future path $Y_i = (P_i^{T_{\text{obs}}+1}, \dots, P_i^{T_{\text{obs}}+T_{\text{pred}}})$ given its observed history $H_i =$

$(P_i^1, \dots, P_i^{T_{\text{obs}}})$, where $P_i^t \in \mathbb{R}^2$ is the position at timestep t . Here, T_{obs} is the historical observation span and T_{pred} is the future prediction length.

Metrics We adopt the widely used standard measurement matrix in trajectory prediction to ensure fair comparison with all baselines (Gupta et al. 2018). Performance is measured by the minimum Average Displacement Error (minADE_K) and minimum Final Displacement Error (minFDE_K). This matrix evaluates a model’s ability to capture the multi-modal nature of human motion by generating K trajectory samples and selecting the one with the lowest error against the ground truth. Following the same setting as the baselines, we use $K = 20$ for our method to ensure a fair and direct comparison.

Optimization Objective To guide the evolutionary search, we employ the Mean Squared Error (MSE) and calculate the average squared Euclidean distance between the predicted and ground truth trajectory points, where a lower MSE corresponds to a higher fitness. The choice of MSE is motivated by its simplicity and generality, as well as its capacity to provide a continuous and well-defined error signal, which is crucial for guiding the evolutionary process. Its widespread adoption ensures our results are comparable across the literature, while its parameter-free nature aligns with our goal of automated heuristic design.

Evolutionary Framework

Our evolutionary framework is designed not merely to minimize prediction error on a given dataset, but to discover

heuristics that are fundamentally simple and robust enough to generalize across different environments. Inspired by the Reflective Evolution approach (Ye et al. 2024), the framework is illustrated in Fig. 2 (right). In this algorithm, Large Language Models (LLMs) serve as the core genetic operators, following these steps:

Initial Population The process starts by seeding the generator LLM with a task specification (including problem details, input/output format, and the objective J) alongside a basic heuristic like the Constant Velocity Model (CVM). The LLM then generates an initial population of N diverse heuristics, providing the starting point for the evolutionary search.

Selection for Crossover Parents for crossover are chosen from successfully executed heuristics in the current population. The selection balances exploration and exploitation: 70% of parents are selected uniformly at random from successfully running candidates, while 30% are chosen from the elite performers (those with the lowest objective J).

Reflections TRAJEVO employs two types of reflection as in Ye et al. (2024). *Short-term reflections* compare the performance of selected crossover parents, offering immediate feedback to guide the generation of offspring. *Long-term reflections* accumulate insights across generations, identifying effective design patterns and principles to steer mutation and broader exploration. Both reflection mechanisms produce textual guidance (i.e. “verbal gradients” (Pryzant et al. 2023)) for the generator LLM.

Crossover This operator creates new offspring by combining code from two parent heuristics. Guided by short-term reflections that compare the ‘better’ and ‘worse’ performing parent, the LLM is prompted to mix their effective “genes”, enabling the emergence of potentially superior heuristics.

Elitist Mutation The mutation operator mutates the elitist (best found so far) heuristic. In TRAJEVO, this involves the generator LLM modifying an elite heuristic selected. This mutation step is informed by the insights gathered through long-term reflections.

It is crucial to note that this entire evolutionary process is a one-time, offline procedure. The final output is a standalone Python heuristic that runs with high computational efficiency and does not require the LLM during inference.

Cross-Generation Elite Sampling

Evolving effective heuristics for complex tasks like trajectory prediction poses a significant search challenge, where standard evolutionary processes can easily get trapped in local optima (Osuna and Sudholt 2018). Simple mutations often yield only incremental improvements, failing to explore sufficiently diverse or novel strategies. To address this limitation and enhance exploration, we introduce Cross-Generation Elite Sampling (CGES), a core component influencing the mutation step within the TRAJEVO framework.

In contrast to typical elitism focusing on the current generation’s best, CGES maintains a history archive of high-performing heuristics accumulated across all past genera-

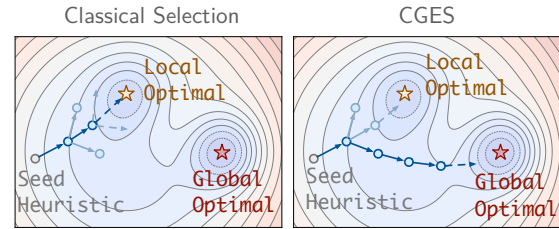


Figure 3: Cross-Generation Elite Sampling (CGES) helps escape local optima by sampling elite individuals from past generations (left), which greatly helps achieve much better objective values (right).

```
...
<stats>
Statistics of trajectory index counts with
the lowest ADE. These help us understand
which heuristics contribute to the
performance for at least some trajectories.
Traj Index: Count: {0: 67, 1: 10, 2: 3, 3:
2, 4: 7, 5: 0, ... 18: 0, 19: 2}
</stats>
...
```

```
...
The better code explicitly includes a
deterministic (linear extrapolation)
trajectory as the first prediction. This
significantly improves ADE, as evidenced by
trajectory 0 having the lowest ADE count by
far. The statistics suggest it's beneficial
to have at least one trajectory that closely
follows the established motion.
...
```

Figure 4: Statistics Feedback Loop: (top) distribution of prediction index effectiveness by minimum ADE frequency; (bottom) LLM prompt incorporating statistical feedback for guided mutation.

tions. Specifically, CGES modifies how the elite individual targeted for mutation is selected: instead of necessarily choosing the top performer from the current population, the heuristic designated to undergo mutation is sampled by CGES directly from this history. This sampling uses a Softmax distribution based on the recorded objectives J of the historical elites, thus prioritizing individuals that have proven effective. By potentially reintroducing and modifying diverse, historically successful strategies during the mutation phase, CGES significantly improves the exploration capability, facilitating escape from local optima and the discovery of more robust heuristics, as demonstrated in Fig. 3.

Statistics Feedback Loop

While the objective J measures overall quality, it does not reveal *which* of a heuristic’s diverse internal prediction strategies are actually effective. To provide this crucial insight for effective refinement, TRAJEVO incorporates a Statistics Feedback Loop (SFL), illustrated in Fig. 4.

This loop specifically analyzes the contribution of the 20 distinct trajectory prediction sets (indexed $k = 0 \dots 19$) generated by a heuristic. After evaluation, we compute a key statistic: a distribution showing how frequently each prediction index k delivered the minimum ADE for individual trajectory instances across the dataset (Fig. 4 top). This directly highlights the empirical utility of the different diversification strategies associated with each index k . This statistical distribution, together with the heuristic’s code, is fed to the reflector and mutation LLMs to identify which strategies (k) contribute most effectively to performance (Fig. 4 bottom). Such feedback provides actionable guidance to the generator LLM, enabling specific improvements to the heuristic’s multi-prediction generation logic based on the observed effectiveness of its constituent strategies. A qualitative example of the full TRAJEVO evolution is provided in Fig. 2.

Experiments

Experimental Setup

Datasets We evaluate TRAJEVO on the ETH-UCY benchmark (Pellegrini et al. 2009), a collection of datasets comprising real-world pedestrian trajectories with the standard leave-one-out protocol where heuristics are evolved on four datasets and tested on the remaining one (Alahi et al. 2016). To specifically assess OOD generalization, we use the SDD dataset (Robicquet et al. 2016) as a completely unseen test set. For all experiments, following the standard setting widely used by baseline methods on these benchmarks, we observe 8 past frames (3.2s) to predict 12 future frames (4.8s).

Baselines We compare heuristics generated by TRAJEVO against both heuristic and deep learning baselines. *Heuristic baselines*, which are well-suited for resource-constrained systems, include kinematic models such as the Constant Velocity Model (CVM) and its sampling variant (CVM-S), Constant Acceleration (ConstantAcc), and Constant Turn Rate and Velocity (CTRV), as well as CSCRCTR, Linear Regression (LinReg), and the physics-inspired Social Force model. To benchmark against more complex data-driven approaches, we also evaluate *deep learning baselines*, including early influential models like Social-LSTM and SocialGAN; graph-based methods such as STGAT and Social-STGCNN; and recent state-of-the-art models like Trajectron++, MemoNet, EigenTrajectory, and MoFlow. To ensure robustness, all reported results for TRAJEVO are averaged over 10 independent evolutionary runs. The standard deviations across runs were consistently low, typically ranging from 0.02 to 0.06.

Hardware and Software All experiments were conducted on a workstation equipped with an AMD Ryzen 9 7950X 16-Core Processor and a single NVIDIA RTX 3090 GPU. The TRAJEVO framework generates trajectory prediction heuristics as executable Python code snippets in a Python 3.12 environment, employing Google’s Gemini 2.0 Flash model (Google AI for Developers 2025) in the main experiment. To evaluate the stability and generalizability of our framework, its performance was systematically tested with a variety of

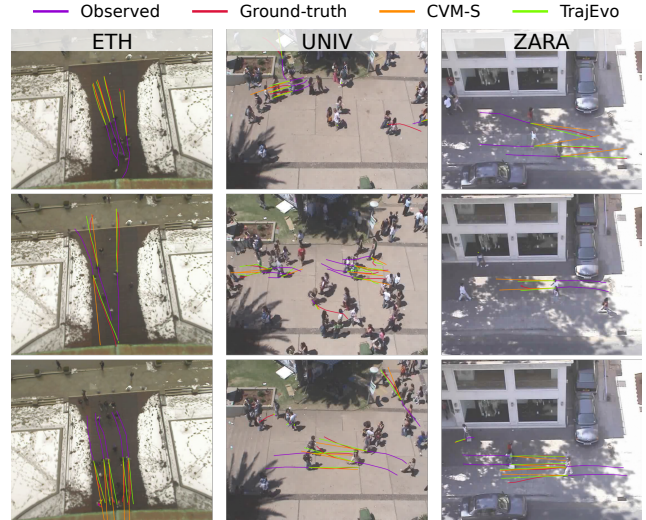


Figure 5: Trajectory prediction results for CVM-S and TRAJEVO across different datasets. Each row illustrates a distinct pattern: (top) linear trajectories, (middle) non-linear trajectories, and (bottom) collision-avoidance cases. We visualize the single best trajectory out of $K = 20$ samples based on the optimization objective.

alternative models, including DeepSeek V3 & R1, Qwen3-32B, ChatGPT-4o, and Claude 3.7 Sonnet.

In-Distribution Performance

Comparison with Heuristic Methods We report results against heuristic baselines in Table 1. TRAJEVO consistently achieves the best performance across all ETH-UCY datasets and significantly outperforms all competitors on average. This establishes TRAJEVO as the new state-of-the-art among heuristic approaches on this benchmark. Interestingly, the general performance trend across baseline methods suggests that heuristics incorporating more complexity beyond basic constant velocity assumptions are generally worse suited for real-world pedestrian data, including SocialForce, with the second-best result obtained by the relatively simple CVM-S. Fig. 5 shows examples of the generated trajectories.

Comparison with Deep Learning Methods Table 2 reports results against deep learning methods. On this in-distribution benchmark, while TRAJEVO does not surpass highly specialized neural networks like MoFlow, it establishes a strong baseline. It consistently outperforms several established models, including Social-LSTM, and even more recent ones like Trajectron++ on certain datasets. This performance is highly competitive for a heuristic-based approach. However, as the next section demonstrates, the primary strength of our method lies not in incremental in-distribution gains, but in its superior generalization capabilities in OOD situations.

Method	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
SocialForce (Helbing and Molnar 1995)	1.46/2.48	0.69/1.23	0.96/1.75	1.37/2.51	0.84/1.53	1.06/1.90
LinReg (Bishop and Nasrabadi 2006)	1.04/2.20	0.26/0.47	0.76/1.48	0.62/1.22	0.47/0.93	0.63/1.26
ConstantAcc (Polychronopoulos et al. 2007)	3.12/7.98	1.64/4.19	1.02/2.60	0.81/2.05	0.60/1.53	1.44/3.67
CSCRCTR (Zhai, Meng, and Wang 2014)	2.27/4.61	1.03/2.18	1.35/3.12	0.96/2.12	0.90/2.10	1.30/2.83
CVM (Schöllner et al. 2020)	1.01/2.24	0.32/0.61	0.54/1.21	0.42/0.95	0.33/0.75	0.52/1.15
CVM-S (Schöllner et al. 2020)	0.92/2.01	0.27/0.51	0.53/1.17	0.37/0.77	0.28/0.63	0.47/1.02
CTRV (Lu et al. 2021)	1.62/3.64	0.72/1.09	0.71/1.59	0.65/1.50	0.48/1.10	0.84/1.78
TRAJEVO	0.47/0.77	0.17/0.30	0.51/1.10	0.35/0.75	0.27/0.57	0.35/0.70

Table 1: Comparison of TRAJEVO with trajectory prediction heuristics across datasets with mean minADE₂₀ / minFDE₂₀ (meters) on the ETH-UCY dataset.

Method	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Social-LSTM (Alahi et al. 2016)	<u>1.09/2.35</u>	0.79/1.76	0.67/1.40	0.56/1.17	0.72/1.54	0.77/1.64
Social-GAN (Gupta et al. 2018)	<u>0.87/1.62</u>	0.67/1.37	<u>0.76/1.52</u>	0.35/0.68	0.42/0.84	0.61/1.21
STGAT (Huang et al. 2019)	<u>0.65/1.12</u>	<u>0.35/0.66</u>	<u>0.52/1.10</u>	0.34/0.69	<u>0.29/0.60</u>	<u>0.43/0.83</u>
Social-STGCNN (Mohamed et al. 2020)	<u>0.64/1.11</u>	0.49/0.85	0.44/0.79	0.34/0.53	0.30/0.48	0.44/0.75
Trajectron++ (Salzmann et al. 2020)	<u>0.61/1.03</u>	0.20/0.28	0.30/0.55	0.24/0.41	0.18/0.32	0.31/0.52
MemoNet (Xu et al. 2022)	0.41/0.61	0.11/0.17	0.24/0.43	0.18/0.32	0.14/0.24	0.21/0.35
EigenTrajectory (Bae, Oh, and Jeon 2023)	0.36/0.53	0.12/0.19	0.24/0.43	0.19/0.33	0.14/0.24	0.21/0.34
MoFlow (Fu et al. 2025)	0.40/0.57	0.11/0.17	0.23/0.39	0.15/0.26	0.12/0.22	0.20/0.32
TRAJEVO	0.47/0.77	0.17/0.30	0.51/1.10	0.35/0.75	0.27/0.57	0.35/0.70

Table 2: Comparison of TRAJEVO with deep learning approaches (mean minADE₂₀/minFDE₂₀ on ETH-UCY). Each underlined number indicates that the result is worse than the corresponding result from TRAJEVO on at least one metric.

Out-of-Distribution Performance

A crucial capability for robotic systems deployed in the real world is generalizing to unseen (OOD) environments during development. We evaluate this on the unseen dataset SDD, directly testing both deep learning models (trained on ETH-UCY) and the heuristics from TRAJEVO (evolved on the ETH-UCY). We report these OOD performance results in Table 3 for three selected heuristics and three recent established neural methods. Remarkably, TRAJEVO demonstrates superior generalization, significantly outperforming not only all heuristic baselines but also all tested deep learning methods, including SOTA models like MoFlow. TRAJEVO performs substantially better than the best deep learning competitor, EigenTrajectory and MoFlow. This suggests that the interpretable and efficient heuristics discovered by TRAJEVO may possess greater robustness to domain shifts compared to complex neural networks trained on specific distributions.

Ablation Study

We conducted an ablation study, reported in Table 4, to validate the effectiveness of the core components introduced in TRAJEVO. The results demonstrate that both the SFL and CGES meaningfully improve performance. Removing either component from the full framework leads to a noticeable degradation in prediction accuracy, confirming their positive contribution.

Analysis and Discussion

Performance of Different LLMs To demonstrate the robustness of the TRAJEVO framework and the stability of its outcomes, we evaluated its performance when driven by several state-of-the-art LLMs, including DeepSeek-V3 & R1, Qwen3-32B, ChatGPT-4o, and Claude 3.7 Sonnet. As detailed in Table 5, a key finding is the consistent effectiveness across all models. Each LLM was capable of generating high-performing heuristics, which underscores the general utility and stability of TRAJEVO. While the overall performance level remained high and stable, we did observe minor variations that highlight opportunities for fine-tuning. For instance, Deepseek-R1 achieved the best average results, while other models excelled on specific datasets. This analysis confirms that the TRAJEVO framework is effective and not reliant on a single specific LLM, ensuring stable, high-quality outcomes regardless of the chosen model.

Evolution Resources A single TRAJEVO evolutionary run takes approximately 5 minutes using Google’s Gemini 2.0 Flash (Google AI for Developers 2025) with an average API cost of \$0.05. In contrast, training neural methods can take a full day on a contemporary GPU, incurring significantly higher estimated compute costs – e.g., on an RTX 3090 as reported by Bae, Oh, and Jeon (2023), around \$4 based on typical rental rates, 80× more than TRAJEVO.

Inference Resources Crucially, the LLM is only used during the offline evolutionary design phase and is not called at inference time. The resulting heuristics are lightweight Python functions. As such, TRAJEVO-generated heuristics

Method (tested on SDD dataset)	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
SocialForce (Helbing and Molnar 1995)	33.64/60.63	33.64/60.63	33.64/60.63	33.64/60.63	33.64/60.63	33.64/60.63
CVM (Schöller et al. 2020)	18.82/37.95	18.82/37.95	18.82/37.95	18.82/37.95	18.82/37.95	18.82/37.95
CVM-S (Schöller et al. 2020)	16.28/31.84	16.28/31.84	16.28/31.84	16.28/31.84	16.28/31.84	16.28/31.84
Trajectron++ (Salzmann et al. 2020)	46.72/69.11	47.30/67.76	46.08/75.90	47.30/72.19	46.78/68.59	46.84/70.71
EigenTrajectory (Bae, Oh, and Jeon 2023)	14.51/25.13	14.69/24.64	14.31/27.60	14.69/26.25	14.53/24.94	14.55/25.71
MoFlow (Fu et al. 2025)	17.00/27.98	17.21/27.43	17.00/30.63	17.24/29.22	17.27/27.56	17.14/28.56
TRAJEVO	12.58/23.82	12.26/23.60	12.78/23.71	13.10/25.23	12.22/23.57	12.59/23.99

Table 3: Out of distribution performance of methods trained on different ETH-UCY splits and tested on the unseen SDD dataset. We report minADE₂₀ / minFDE₂₀ (pixels) on the SDD dataset.

Dataset	- SFL - CGES	- SFL	TRAJEVO
ETH	0.68/1.36	0.59/1.12	0.47/0.77
HOTEL	0.26/0.45	0.19/0.33	0.17/0.30
UNIV	0.59/1.22	0.52/1.13	0.51/1.10
ZARA1	0.37/0.77	0.36/0.76	0.35/0.75
ZARA2	0.31/0.65	0.28/0.59	0.27/0.57

Table 4: Ablation study for the evolution framework removing different components. Lower is better (\downarrow).

Model	ETH		HOTEL		UNIV	
	ADE	FDE	ADE	FDE	ADE	FDE
deepseek-R1	0.46	0.74	0.16	0.29	0.49	1.06
deepseek-V3	0.43	0.71	0.21	0.39	0.50	1.08
Qwen3-32B	0.49	0.74	0.18	0.32	0.51	1.09
ChatGPT-4o	0.51	0.82	0.15	0.30	0.54	1.13
Claude-3.7	0.49	0.93	0.18	0.36	0.49	1.07

Model	ZARA1		ZARA2		AVG	
	ADE	FDE	ADE	FDE	ADE	FDE
deepseek-R1	0.33	0.72	0.26	0.54	0.34	0.67
deepseek-V3	0.34	0.69	0.27	0.56	0.35	0.69
Qwen3-32B	0.35	0.75	0.27	0.59	0.36	0.70
GPT-4o	0.36	0.72	0.31	0.54	0.37	0.70
Claude-3.7	0.32	0.61	0.27	0.57	0.35	0.71

Table 5: Performance of various LLMs on the standard ETH-UCY benchmark datasets. Best results are in bold.

demonstrate significant speed advantages, requiring only 0.65ms per instance on a single CPU core with the Python version and less than 5 μ s for a version of the heuristic rewritten in C++. In contrast, neural baselines need 12-29ms on GPU and 248-375ms on (multi-core) CPU. Notably, TRAJEVO achieves over 10 \times speedup for the original Python version and more than 2400 \times speedup for the C++ version compared to the optimized MoFlow running on a dedicated GPU, highlighting its relevance for resource-constrained, real-time robotic systems.

Explainability Unlike the black-box nature of neural networks, TRAJEVO produces human-readable Python code. For instance, the heuristic evolved for the ZARA1 dataset is not a simple tweak of a known model. It intelligently

combines four distinct strategies to generate its 20 samples. The primary strategy ($k = 0..9$) is a sophisticated form of adaptive linear extrapolation, where the velocity vector is averaged over the last few steps and perturbed by noise scaled to the agent’s current speed—allowing it to model both smooth paths and sudden hesitations. A second strategy ($k = 10..14$) introduces rotational noise, effectively simulating curved paths. A third ($k = 15..17$) implements a memory-based collision avoidance mechanism by checking potential future positions against the recent paths of nearby agents. The final strategy ($k = 18, 19$) acts as a conservative fallback, using a heavily damped extrapolation. This automated discovery of a complex, multi-faceted strategy combining adaptive kinematics and interaction rules is a key advantage, making the model’s logic transparent and verifiable, which is crucial for safety-critical applications.

Limitations While TRAJEVO shows strong performance, limitations remain. Our implementation works on bird’s-eye-view pedestrian datasets using only past 2D positions, so a gap to specialized deep models on ETH-UCY in in-distribution accuracy. The evolutionary objective is defined only on trajectory prediction error rather than downstream planning or safety metrics (e.g., collision rate or comfort); extending TRAJEVO in this direction left to future work.

Conclusion

We introduced TRAJEVO, a framework leveraging Large Language Models and evolutionary algorithms to automate trajectory prediction heuristic design. Our experiments demonstrate that TRAJEVO generates heuristics that outperform traditional methods on standard benchmarks and exhibit superior out-of-distribution generalization, surpassing deep learning models on unseen data while remaining fast and interpretable. TRAJEVO represents a step towards automatically discovering efficient, explainable, and generalizable trajectory prediction heuristics, offering an alternative to black-box models.

Future research will focus on bridging the performance gap to state-of-the-art models by enhancing evolutionary search and incorporating multi-modal inputs. For real-world applicability, the optimization objective will be reframed from predictive accuracy metrics towards performance on downstream tasks like navigation and planning.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2024-00410082), by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2025-00563763), and by the InnoCORE program of the Ministry of Science and ICT(N10250154).

References

- Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; and Savarese, S. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 961–971.
- Bae, I.; Oh, J.; and Jeon, H.-G. 2023. Eigentrajjectory: Low-rank descriptors for multi-modal trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10017–10029.
- Bishop, C. M.; and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 4. Springer.
- Cao, D.; Li, J.; Ma, H.; and Tomizuka, M. 2021. Spectral temporal graph neural network for trajectory prediction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 1839–1845. IEEE.
- Chen, X.; Treiber, M.; Kanagaraj, V.; and Li, H. 2018. Social force models for pedestrian traffic—state of the art. *Transport reviews*, 38(5): 625–653.
- Choi, C.; Choi, J. H.; Li, J.; and Malla, S. 2021. Shared cross-modal trajectory prediction for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 244–253.
- Dat, P. V. T.; Doan, L.; and Binh, H. T. T. 2025. Hsevo: Elevating automatic heuristic design with diversity-driven harmony search and genetic algorithm using llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 26931–26938.
- Dax, V. M.; Li, J.; Sachdeva, E.; Agarwal, N.; and Kochenderfer, M. J. 2023. Disentangled neural relational inference for interpretable motion prediction. *IEEE Robotics and Automation Letters*, 9(2): 1452–1459.
- Farina, F.; Fontanelli, D.; Garulli, A.; Giannitrapani, A.; and Praticchizzo, D. 2017. Walking ahead: The headed social force model. *PloS one*, 12(1): e0169734.
- Fu, Y.; Yan, Q.; Wang, L.; Li, K.; and Liao, R. 2025. MoFlow: One-Step Flow Matching for Human Trajectory Forecasting via Implicit Maximum Likelihood Estimation based Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Google AI for Developers. 2025. Gemini API Pricing — Gemini API — Google AI for Developers. <https://ai.google.dev/gemini-api/docs/pricing>. Last updated: 2025-04-21, Accessed: 2025-05-01.
- Gupta, A.; Johnson, J.; Fei-Fei, L.; Savarese, S.; and Alahi, A. 2018. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2255–2264.
- Helbing, D.; Farkas, I.; and Vicsek, T. 1997. Modeling the dynamics of human behavior in complex systems. *Physical Review E*, 56(4): 4282.
- Helbing, D.; and Molnar, P. 1995. Social force model for pedestrian dynamics. *Physical review E*, 51(5): 4282.
- Huang, Y.; Bi, H.; Li, Z.; Mao, T.; and Wang, Z. 2019. STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Itkina, M.; and Kochenderfer, M. 2023. Interpretable self-aware neural networks for robust trajectory prediction. In *Conference on Robot Learning*, 606–617. PMLR.
- Kim, S.; Baek, J.; Kim, J.; and Lee, J. 2025. GUIDE-CoT: Goal-driven and User-Informed Dynamic Estimation for Pedestrian Trajectory using Chain-of-Thought. *arXiv preprint arXiv:2503.06832*.
- Lange, B.; Li, J.; and Kochenderfer, M. J. 2024. Scene informer: Anchor-based occlusion inference and trajectory prediction in partially observable environments. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 14138–14145. IEEE.
- Li, J.; Hua, C.; Ma, H.; Park, J.; Dax, V.; and Kochenderfer, M. J. 2024a. Multi-agent dynamic relational reasoning for social robot navigation. *arXiv preprint arXiv:2401.12275*.
- Li, J.; Hua, C.; Park, J.; Ma, H.; Dax, V.; and Kochenderfer, M. J. 2022a. Evolvehypergraph: Group-aware dynamic relational reasoning for trajectory prediction. *arXiv preprint arXiv:2208.05470*.
- Li, J.; Li, J.; Bae, S.; and Isele, D. 2024b. Adaptive prediction ensemble: Improving out-of-distribution generalization of motion forecasting. *IEEE Robotics and Automation Letters*.
- Li, J.; Ma, H.; and Tomizuka, M. 2019. Conditional generative neural system for probabilistic trajectory prediction. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6150–6156. IEEE.
- Li, J.; Ma, H.; Zhang, Z.; Li, J.; and Tomizuka, M. 2021. Spatio-temporal graph dual-attention network for multi-agent prediction and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 23(8): 10556–10569.
- Li, J.; Shi, X.; Chen, F.; Stroud, J.; Zhang, Z.; Lan, T.; Mao, J.; Kang, J.; Refaat, K. S.; Yang, W.; et al. 2023a. Pedestrian Crossing Action Recognition and Trajectory Prediction with 3D Human Keypoints. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 1463–1470. IEEE.
- Li, J.; Yang, F.; Tomizuka, M.; and Choi, C. 2020. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *Advances in neural information processing systems*, 33: 19783–19794.
- Li, K.; Chen, Y.; Shan, M.; Li, J.; Worrall, S.; and Nebot, E. 2023b. Game theory-based simultaneous prediction and planning for autonomous vehicle navigation in crowded environments. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2977–2984. IEEE.
- Li, P.; Pei, X.; Chen, Z.; Zhou, X.; and Xu, J. 2022b. Human-like motion planning of autonomous vehicle based on probabilistic trajectory prediction. *Applied Soft Computing*, 118: 108499.
- Liu, F.; Tong, X.; Yuan, M.; Lin, X.; Luo, F.; Wang, Z.; Lu, Z.; and Zhang, Q. 2024a. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. *arXiv preprint arXiv:2401.02051*.
- Liu, F.; Zhang, R.; Xie, Z.; Sun, R.; Li, K.; Lin, X.; Wang, Z.; Lu, Z.; and Zhang, Q. 2024b. Llm4ad: A platform for algorithm design with large language model. *arXiv preprint arXiv:2412.17287*.
- Liu, P.; Liu, H.; Li, Y.; Shi, T.; Zhu, M.; and Pu, Z. 2024c. Traj-Explainer: An Explainable and Robust Multi-modal Trajectory Prediction Approach. *arXiv preprint arXiv:2410.16795*.
- Lu, T.; Watanabe, Y.; Yamada, S.; and Takada, H. 2021. Comparative evaluation of Kalman filters and motion models in vehicular state estimation and path prediction. *Journal of Navigation*, 74.

- Ma, H.; Li, J.; Hosseini, R.; Tomizuka, M.; and Choi, C. 2022. Multi-objective diverse human motion prediction with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8161–8171.
- Ma, H.; Sun, Y.; Li, J.; and Tomizuka, M. 2021. Multi-agent driving behavior prediction across different scenarios with self-supervised domain knowledge. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 3122–3129. IEEE.
- Ma, Y.; Manocha, D.; and Wang, W. 2018. Efficient Reciprocal Collision Avoidance between Heterogeneous Agents Using CT-MAT. In *AAMAS*.
- Madjid, N. A.; Ahmad, A.; Mebrahtu, M.; Babaa, Y.; Nasser, A.; Malik, S.; Hassan, B.; Werghe, N.; Dias, J.; and Khonji, M. 2025. Trajectory Prediction for Autonomous Driving: Progress, Limitations, and Future Directions. *arXiv preprint arXiv:2503.03262*.
- Mohamed, A.; Qian, K.; Elhoseiny, M.; and Claudel, C. 2020. Social-stgcn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14424–14432.
- Osuna, E. C.; and Sudholt, D. 2018. Runtime analysis of probabilistic crowding and restricted tournament selection for bimodal optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 929–936.
- Pellegrini, S.; Ess, A.; Schindler, K.; and van Gool, L. 2009. You’ll Never Walk Alone: Modeling Social Behavior for Multi-target Tracking. In *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, 261–268.
- Phong, T.; Wu, H.; Yu, C.; Cai, P.; Zheng, S.; and Hsu, D. 2023. What truly matters in trajectory prediction for autonomous driving? *Advances in Neural Information Processing Systems*, 36: 71327–71339.
- Polychronopoulos, A.; Tsogas, M.; Amditis, A. J.; and Andreone, L. 2007. Sensor fusion for predicting vehicles’ path for collision avoidance systems. *IEEE Transactions on Intelligent Transportation Systems*, 8(3): 549–562.
- Pryzant, R.; Iter, D.; Li, J.; Lee, Y. T.; Zhu, C.; and Zeng, M. 2023. Automatic prompt optimization with “gradient descent” and beam search. *arXiv preprint arXiv:2305.03495*.
- Rainbow, B. A.; Men, Q.; and Shum, H. P. 2021. Semantics-STGCNN: A semantics-guided spatial-temporal graph convolutional network for multi-class trajectory prediction. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2959–2966. IEEE.
- Robicquet, A.; Sadeghian, A.; Alahi, A.; and Savarese, S. 2016. Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes. In *European Conference on Computer Vision (ECCV)*, 549–565. Springer International Publishing.
- Robla-Gómez, S.; Becerra, V. M.; Llata, J. R.; González-Sarabia, E.; Torre-Ferrero, C.; and Pérez-Oria, J. 2017. Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments. *IEEE Access*, 5: 26754–26773.
- Rudenko, A.; Palmieri, L.; Herman, M.; Kitani, K. M.; Gavrila, D. M.; and Arras, K. O. 2020. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8): 895–935.
- Salzmann, T.; Ivanovic, B.; Chakravarty, P.; and Pavone, M. 2020. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, 683–700. Springer.
- Schöller, C.; Aravantinos, V.; Lay, F.; and Knoll, A. 2020. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 5(2): 1696–1703.
- Sun, F.-Y.; Kauvar, I.; Zhang, R.; Li, J.; Kochenderfer, M. J.; Wu, J.; and Haber, N. 2022. Interaction modeling with multiplex attention. *Advances in Neural Information Processing Systems*, 35: 20038–20050.
- Wang, L.; Lavoie, M.-A.; Papais, S.; Nisar, B.; Chen, Y.; Ding, W.; Ivanovic, B.; Shao, H.; Abuduweili, A.; Cook, E.; Zhou, Y.; Karkus, P.; Li, J.; Liu, C.; Pavone, M.; and Waslander, S. 2025a. Deployable and Generalizable Motion Prediction: Taxonomy, Open Challenges and Future Directions. *arXiv preprint arXiv:2505.09074*.
- Wang, W.; Wang, C.; Yang, B.; Chen, G.; and Min, B.-C. 2024. Hyper-sttn: Social group-aware spatial-temporal transformer network for human trajectory prediction with hypergraph reasoning. *arXiv preprint arXiv:2401.06344*.
- Wang, Y.; Huang, X.; Sun, X.; Yan, M.; Xing, S.; Tu, Z.; and Li, J. 2025b. Uniocc: A unified benchmark for occupancy forecasting and prediction in autonomous driving. *arXiv preprint arXiv:2503.24381*.
- Wang, Z.; Wang, Y.; Wu, Z.; Ma, H.; Li, Z.; Qiu, H.; and Li, J. 2025c. Cmp: Cooperative motion prediction with multi-agent communication. *IEEE Robotics and Automation Letters*.
- Wu, X.; Wu, S.-h.; Wu, J.; Feng, L.; and Tan, K. C. 2024. Evolutionary computation in the era of large language model: Survey and roadmap. *IEEE Transactions on Evolutionary Computation*.
- Xie, S.; Li, J.; and Wang, J. 2023. A cognition-inspired trajectory prediction method for vehicles in interactive scenarios. *IET Intelligent Transport Systems*, 17(8): 1544–1559.
- Xu, C.; Mao, W.; Zhang, W.; and Chen, S. 2022. Remember intentions: Retrospective-memory-based trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6488–6497.
- Xu, Z.; Zhou, R.; Yin, Y.; Gao, H.; Tomizuka, M.; and Li, J. 2024. MATRIX: multi-agent trajectory generation with diverse contexts. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 12650–12657. IEEE.
- Yamaguchi, K.; Berg, A. C.; Ortiz, L. E.; and Berg, T. L. 2011. Who are you with and where are you going? In *CVPR 2011*, 1345–1352.
- Yao, J.; Zhang, X.; Xia, Y.; Wang, Z.; Roy-Chowdhury, A. K.; and Li, J. 2024. Sonic: Safe social navigation with adaptive conformal inference and constrained reinforcement learning. *arXiv preprint arXiv:2407.17460*.
- Ye, H.; Wang, J.; Cao, Z.; Berto, F.; Hua, C.; Kim, H.; Park, J.; and Song, G. 2024. ReEvo: Large Language Models as Hyper-Heuristics with Reflective Evolution. In *Advances in Neural Information Processing Systems*.
- Zhai, G.; Meng, H.; and Wang, X. 2014. A Constant Speed Changing Rate and Constant Turn Rate Model for Maneuvering Target Tracking. *Sensors*, 14(3): 5239–5253.
- Zheng, Z.; Xie, Z.; Wang, Z.; and Hooi, B. 2025. Monte carlo tree search for comprehensive exploration in llm-based automatic heuristic design. *International Conference on Learning Representations*.
- Zhou, R.; Zhou, H.; Gao, H.; Tomizuka, M.; Li, J.; and Xu, Z. 2022. Grouptron: Dynamic multi-scale graph convolutional networks for group-aware dense crowd trajectory forecasting. In *2022 International Conference on Robotics and Automation (ICRA)*, 805–811. IEEE.