

Model Predictive Control for Crowd Navigation via Learning-Based Trajectory Prediction

Mohamed Parvez Aslam¹, Bojan Derajic^{2,3} ^a, Mohamed-Khalil Bouzidi^{2,4} ^b,
Sebastian Bernhard² ^c and Jan Oliver Ringert¹ ^d

¹Bauhaus-Universität Weimar, Germany

²Continental Automotive GmbH, Germany

³Technical University of Berlin, Germany

⁴Free University of Berlin, Germany

Keywords: Crowd Navigation, Model Predictive Control, Trajectory Prediction.

Abstract: Safe navigation in pedestrian-rich environments remains a key challenge for autonomous robots. This work evaluates the integration of a deep learning-based Social-Implicit (SI) pedestrian trajectory predictor within a Model Predictive Control (MPC) framework on the physical *Continental Corriere* robot. Tested across varied pedestrian densities, the SI-MPC system is compared to a traditional Constant Velocity (CV) model in both open-loop prediction and closed-loop navigation. Results show that SI improves trajectory prediction—reducing errors by up to 76% in low-density settings—and enhances safety and motion smoothness in crowded scenes. Moreover, real-world deployment reveals discrepancies between open-loop metrics and closed-loop performance, as the SI model yields broader, more cautious predictions. These findings emphasize the importance of system-level evaluation and highlight the SI-MPC framework’s promise for safer, more adaptive navigation in dynamic, human-populated environments.

1 INTRODUCTION

The rapid advancement of autonomous systems has transformed mobile robotics, particularly in shared environments where robots and pedestrians coexist (Korbmacher and Tordeux, 2022; Golchoubian et al., 2023). A critical challenge in this domain is ensuring safe navigation in dynamic settings, such as crowded sidewalks and factory floors (Wabersich et al., 2023). Pedestrian trajectory prediction (PTP) is central to this challenge, enabling robots to anticipate human movements and adjust their behavior proactively (Alahi et al., 2016; Salzmann et al., 2021).

Traditional heuristic- or knowledge-based models, while interpretable, often fail to capture the complexity of pedestrian behavior (Sighencea et al., 2021). In contrast, deep learning methods—such as Social LSTM (Alahi et al., 2016), Social GAN (Huang et al., 2021), spatiotemporal graph models (Salzmann et al.,

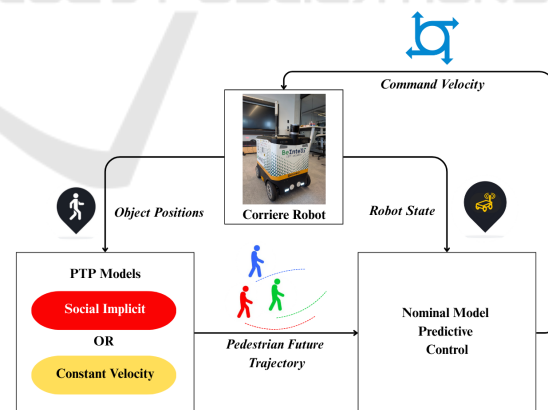


Figure 1: The architecture of an MPC-based motion planning framework with PTP module.

2021; Mohamed et al., 2020; Lian et al., 2023), and hybrid architectures like Y-Net (Mangalam et al., 2020) and Social-Implicit (SI) (Mohamed et al., 2022)—leverage datasets like ETH/UCY (Pellegrini et al., 2009) to model social interactions, generally outperforming traditional approaches in open-loop

^a  <https://orcid.org/0009-0000-6175-535X>

^b  <https://orcid.org/0009-0009-9734-3133>

^c  <https://orcid.org/0000-0002-7194-7539>

^d  <https://orcid.org/0000-0002-3610-3920>

accuracy.

However, open-loop evaluation on static datasets overlooks how these models perform in real-world, dynamic environments, where real-time execution and distribution shift are critical (Alahi et al., 2016). Despite progress in learning-based trajectory prediction, integration into full navigation stacks and closed-loop evaluation remains limited.

Model Predictive Control (MPC) provides a principled framework for dynamic navigation, relying heavily on accurate motion predictions (Bhatt et al., 2023). While MPC aligns well with the output of PTP models, many systems still use simplistic predictors like constant velocity (de Groot et al., 2023; Brito et al., 2019; Vo et al., 2024) or other heuristic-based methods (Chen et al., 2022; Piccinelli et al., 2023; Sani et al., 2021).

Our contributions in this paper are summarized below:

- First, we identify the criteria required for a learning-based PTP model to be integrated within an MPC framework and deploy it on a physical robot, as shown in Fig. 1. The resulting navigation behavior is evaluated by quantifying improvements in both safety metrics—such as minimum collision distance—and trajectory smoothness, measured via acceleration rate (Francis et al., 2023).
- Second, we compare the open-loop performance of predictors on standard datasets with their behavior when integrated into a full navigation stack, thereby assessing the generalization capacity of learned models.
- Third, we demonstrate through extensive experiments that learning-based predictors provide tangible improvements over the baseline, leading to smoother trajectories and improved safety. The evaluation assessment is presented, measuring prediction accuracy, spread, smoothness, and safety across varying pedestrian densities.

2 PRELIMINARIES

2.1 Problem Formulation

We consider a mobile robot navigating in a dynamic, pedestrian-populated environment. The goal is to generate a collision-free, dynamically feasible trajectory that minimizes a cost function over a finite horizon. The optimization incorporates the robot's constraints and learning-based pedestrian motion predictions that model human interactions. The robot adapts

its trajectory in real time to ensure safe and socially compliant navigation.

2.2 Trajectory Prediction Metrics

To evaluate predicted trajectories, we use four key metrics:

Average Displacement Error (ADE) measures the mean Euclidean distance between predicted and ground truth positions over the prediction horizon:

$$\text{ADE} = \frac{1}{N} \sum_{t=1}^N \|\hat{p}_t - p_t\| \quad (1)$$

where $\hat{p}_t$ and p_t are the predicted and true positions at time t , and N is the number of future steps. Final Displacement Error (FDE) measures the distance at the final prediction step:

$$\text{FDE} = \|\hat{p}_N - p_N\| \quad (2)$$

where $\hat{p}_N$ and p_N are the predicted and true final positions.

Average Mahalanobis Distance (AMD) (Mohamed et al., 2022) compares predicted samples to ground truth, accounting for the shape of the predicted distribution:

$$\text{AMD} = \frac{1}{MN} \sum_{i=1}^M \sum_{t=1}^N \sqrt{(\hat{p}_t^i - p_t^i)^\top \Sigma^{-1} (\hat{p}_t^i - p_t^i)} \quad (3)$$

where $\hat{p}_t^i$ is the i -th predicted sample at time t , p_t^i is the corresponding ground truth, Σ is the covariance matrix of the predictions, M is the number of samples, and N is the number of time steps.

Average Maximum Eigenvalue (AMV) (Mohamed et al., 2022) quantifies prediction uncertainty by measuring spread in the predicted samples:

$$\text{AMV} = \frac{1}{MN} \sum_{i=1}^M \sum_{t=1}^N \lambda_{\max}(\Sigma_t^i) \quad (4)$$

where $\lambda_{\max}(\Sigma_t^i)$ is the largest eigenvalue of the covariance matrix Σ_t^i for sample i at time t .

2.3 Robot Model

The robot's motion is modeled using a non-holonomic kinematic unicycle model, which is commonly used in practice for robots with differential drive locomotion. The state of the robot is defined as $\mathbf{x} = [x, y, \theta]^\top$, where (x, y) is the position and θ is the heading angle. The control inputs are linear and angular velocities $\mathbf{u} = [v, \omega]^\top$. The robot dynamics is:

$$\dot{x} = v \cos(\theta), \quad (5a)$$

$$\dot{y} = v \sin(\theta), \quad (5b)$$

$$\dot{\theta} = \omega, \quad (5c)$$

with control limits $v \in [-1.0, 1.0]$ m/s and $\omega \in [-1.0, 1.0]$ rad/s.

3 METHODOLOGY

This section describes how we combine the SI predictive model with an MPC local planner for mobile robot navigation in crowds.

3.1 Social-Implicit: A Lightweight and Distribution-Aware Pedestrian Motion Prediction Model

The *Social-Implicit* model (Mohamed et al., 2022) presents a state-of-the-art approach to pedestrian trajectory prediction, addressing common limitations in evaluation methods while offering a compact and efficient architecture suitable for real-time applications. Traditional metrics such as Best-of-N (BoN), ADE and FDE primarily focus on the single best trajectory prediction, often ignoring the distributional characteristics of the output. On the other hand, the SI model is trained using Implicit Maximum Likelihood Estimation (IMLE), which encourages the predicted trajectory samples to cluster around the ground truth distribution tightly. This results in improved prediction accuracy and reliability, which are very important characteristics for downstream modules such as the MPC local planner.

The SI model was chosen as a motion prediction module because it achieves a strong balance of prediction accuracy (lowest ADE/FDE), computational efficiency (5.8K parameters, sub-10ms inference), and ease of integration. SI's simple architecture enables direct, parallelizable use in MPC frameworks without preprocessing or social pooling, while maintaining robust performance even in dense crowds. Also, smaller models are less prone to overfitting (Caruana et al., 2000). The model used for this paper is trained on the ETH dataset (Pellegrini et al., 2009).

In the following, we describe how the SI model's output is integrated with an MPC, which is necessary for the setup. For each pedestrian $i = 1, \dots, M$, the observed history h^i is a sequence of N_h past positions

$$h^i = ((x_{t-N_h+1}^i, y_{t-N_h+1}^i), \dots, (x_t^i, y_t^i))$$

This means that for pedestrian i , their 2D positions (x, y) at each timestep from $t - N_h + 1$ up to the current time t exist¹. Each h^i is a matrix in $\mathbb{R}^{N_h \times 2}$, where each row is a (x, y) position at a past timestep. All pedestrian histories are stacked together into a single tensor $H = [h^1, \dots, h^M] \in \mathbb{R}^{M \times N_h \times 2}$.

¹If the full history of a pedestrian is not observed at the current time, the first observed position is used to pad the history sequence to the necessary length.

The model predicts, for each pedestrian i , a future trajectory of length N :

$$\hat{p}^i = ((\hat{x}_{t+1}^i, \hat{y}_{t+1}^i), \dots, (\hat{x}_{t+N}^i, \hat{y}_{t+N}^i)) \in \mathbb{R}^{N \times 2}$$

This is the sequence of predicted future 2D positions for pedestrian i .

All predicted futures are stacked as:

$$P = [\hat{p}^1, \dots, \hat{p}^M] \in \mathbb{R}^{M \times N \times 2},$$

where P is a tensor containing the predicted future trajectories for all pedestrians. We denote

$$P \sim \tilde{p}_\Theta(\cdot | H)$$

to indicate that $\tilde{p}_\Theta$ is our predictive model, parameterized by Θ , which defines a probability distribution over possible future trajectories P conditioned on the observed history H .

3.2 Model Predictive Control

The objective is to compute a collision-free and dynamically feasible trajectory that minimizes a cost function over a finite prediction horizon while accounting for the predicted motion of surrounding pedestrians. Let the robot's state at time t be denoted by $\mathbf{x}_t \in \mathbb{R}^n$, and control inputs $\mathbf{u}_t \in \mathbb{R}^m$. The robot aims to find a trajectory $\{\mathbf{x}_t\}_{t=0}^N$ by solving an optimization problem of the form:

$$\min_{\mathbf{u}_{0:N-1}} \sum_{t=0}^{N-1} \ell(\mathbf{x}_t, \mathbf{u}_t) + \ell_N(\mathbf{x}_N) \quad (6)$$

subject to:

$$\begin{aligned} \mathbf{x}_{t+1} &= f(\mathbf{x}_t, \mathbf{u}_t), \\ \mathbf{x}_t &\in \mathcal{X}_{\text{free}}(t), \\ \mathbf{u}_t &\in \mathcal{U}, \\ \forall t &\in \{0, \dots, N-1\}, \end{aligned}$$

where $\ell(\cdot)$ and $\ell_N(\cdot)$ denote the stage and terminal costs, $f(\cdot)$ is the discretized system dynamics, $\mathcal{U}$ is the set of admissible controls, and $\mathcal{X}_{\text{free}}(t)$ represents the obstacle-free space, which is time-varying due to dynamic agents such as pedestrians.

We define the stage cost $\ell(\mathbf{x}_t, \mathbf{u}_t)$ as the following weighted quadratic form:

$$\ell(\mathbf{x}_t, \mathbf{u}_t) = \|\mathbf{x}_t - \mathbf{x}_{\text{goal}}\|_Q^2 + \|\mathbf{u}_t\|_R^2 \quad (7)$$

where $\mathbf{x}_{\text{goal}}$ is the goal state, Q and R are weight matrices and $\|\cdot\|$ is the L2 norm. The terminal cost $\ell_N(\mathbf{x}_N) = \|\mathbf{x}_N - \mathbf{x}_{\text{goal}}\|_{Q_T}^2$ emphasizes reaching the goal, where Q_T is a terminal weight matrix.

The predicted pedestrian trajectories are used to define $\mathcal{X}_{\text{free}}(t)$. To ensure collision avoidance, we impose circular safety zones of radii r_r (robot) and r_p (pedestrian) in the following way:

$$\| [x_t, y_t] - \hat{p}_t^i \| \geq r_r + r_p, \forall i \in [1, M], \forall t \in [0, N]. \quad (8)$$

4 PERFORMANCE EVALUATION

This section describes what hardware is used followed by the test setup and analysis. It presents the evaluation of the combined SI predictive model with an MPC local planner for mobile robot navigation.

4.1 Hardware Setup

The Continental's delivery robot is equipped with a comprehensive sensor suite and high-performance computing hardware to support real-time PTP and MPC. Key perception sensors include a 3D LiDAR Robosense RS-Helios 5515 for high-resolution environmental mapping and two Intel RealSense D435 depth cameras for RGB-D data acquisition. Computation is handled by an Intel NUC11PHKi7C Mini PC with an Intel Core i7 processor, 16GB RAM, and an NVIDIA RTX2060 GPU, enabling the execution of complex PTP models and MPC algorithms.

The MPC architecture is implemented using the CasADi optimization framework and ROS for system integration. The control loop operates at a 100 ms sampling time, using a 4th-order Runge-Kutta discretization of the robot's dynamics. Optimization is performed with IPOPT NLP solver, and CasADi's code generation is used to compile efficient solver code for real-time execution. This setup ensures responsive control while managing the computational demands of integrating PTP models into dynamic navigation tasks.

4.2 Test Environment and Layout

To evaluate the navigation performance in crowds of the MPC planner with the learning-based predictive model, we set up different scenarios that represent real situations encountered during deployment. The robot begins at a fixed central start position (SP in Fig. 2) in all test cases. From SP, it navigates toward one of three goal positions (GP1, GP2, GP3), representing distinct real-world destinations as shown in Fig. 2:

- GP1 (top-left): Simulates a leftward navigation.
- GP2 (top-center): Represents a straightforward corridor-like route.
- GP3 (top-right): Simulates a rightward navigation.

Pedestrians follow predefined routes to introduce structured interaction scenarios. Their paths are categorized by movement direction (see Fig. 2):

- Red paths: Horizontal (left-right or right-left).

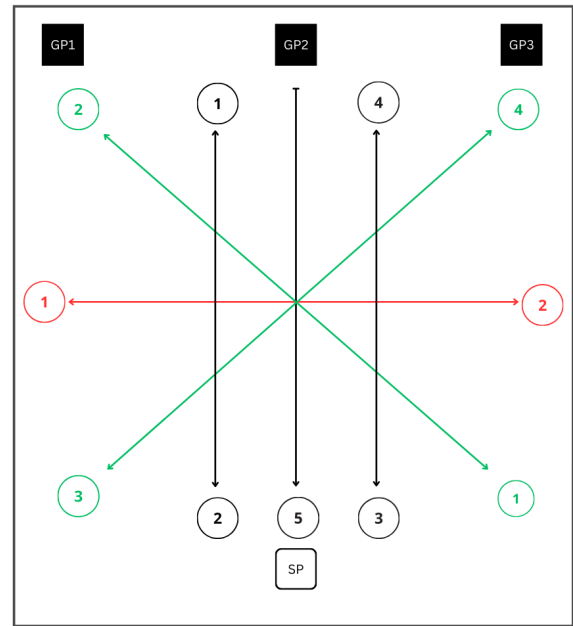


Figure 2: Schematic of the test environment showing robot paths (SP to GP) and pedestrian routes (color-coded).

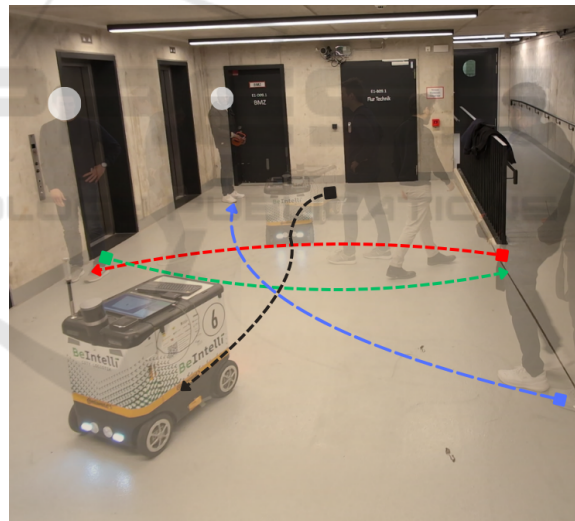


Figure 3: Visualization of the robot navigation with the SI-MPC planner from the 8th experiment. The black arrow represents the robot's motion, while the motion of pedestrians is illustrated by the red, green, and blue arrows.

- Green paths: Diagonal.
- Black paths: Vertical (toward or away from the robot).

Fig. 3 visualizes one particular scenario from the set of conducted experiments, while a more detailed description of the experiments is provided in Table 1.

Test cases systematically varied the robot's goal position and pedestrian density. Pedestrians crossed

Table 1: Description of the conducted experiments.

Test Case	Robot Path	Pedestrian 1	Pedestrian 2	Pedestrian 3
Scene 1	SP → GP1	Red 1 → Red 2	—	—
Scene 2	SP → GP2	GP2 → Black 5	—	—
Scene 3	SP → GP3	Green 2 → Green 1	—	—
Scene 4	SP → GP1	Red 2 → Red 1	Black 1 → Black 2	—
Scene 5	SP → GP2	Red 1 → Red 2	Green 4 → Green 3	—
Scene 6	SP → GP3	Black 4 → Black 3	Green 3 → Green 4	—
Scene 7	SP → GP1	Black 1 → Black 2	GP2 → Black 5	Green 4 → Green 3
Scene 8	SP → GP2	Red 2 → Red 1	Red 1 → Red 2	Black 1 → Black 2
Scene 9	SP → GP1	Black 1 → Black 2	Green 2 → Black 1	Red 1 → Red 2
Scene 10	SP → GP2	Black 4 → Black 3	Green 1 → Black 2	GP2 → Black 5

paths with the robot, moved in parallel, or approached from the opposite direction. Also, the pedestrians always tend to reach their goal positions that are specified at the current test scenario. The tests aim to evaluate the robot’s navigation performance under different interaction scenarios and pedestrian densities (1–3 pedestrians). The experiments consist of 20 test cases, divided into two categories – 10 experiments using the SI model and 10 experiments with the CV model.

We analyze several aspects of navigation performance with this setup, including the comparison of robot behavior under SI and CV models using metrics such as ADE, FDE, AMD, and AMV. Also, we assess how the choice of predictive model influences time-to-goal and minimum distance to pedestrians. This controlled test environment ensures a fair comparison of navigation strategies and highlights the robot’s capacity to adapt to varying pedestrian interactions.

4.3 Quantitative Analysis

The performance evaluation of the SI and CV models integrated in MPC for motion planning of a robot provides critical insights into their effectiveness in dynamic pedestrian environments. This part of the analysis focuses on quantitative metrics (prediction accuracy, motion planning efficiency, safety margins) and qualitative observations (smoothness, adaptability, real-world applicability). The results highlight key trade-offs between safety, efficiency, and motion stability, offering actionable recommendations for deploying these models in real-world robotics.

In low-density environments, the SI model shows a clear advantage over the CV model, as shown in Table 2. It reduces ADE (0.455 vs. 1.655) and FDE (0.687 vs. 2.849) by 72.5% and 75.9%, respectively, reflecting significantly more accurate trajectory predictions. This stems from the SI model’s ability to anticipate pedestrian intent through social navigation behaviors, rather than assuming constant motion. Additionally, the SI model yields an 84.8% lower AMV (0.018 vs. 0.122), indicating greater confidence and

Table 2: Trajectory prediction metrics for varying numbers of pedestrians.

# Ped.	Metric	SI	CV	Improv. (%)
1	ADE	0.455	1.655	72.5
	FDE	0.687	2.849	75.9
	AMD	4.430	8.151	45.6
	AMV	0.018	0.122	84.8
2	ADE	0.561	0.746	24.7
	FDE	0.859	1.338	35.8
	AMD	5.367	3.226	-66.4
	AMV	0.018	0.120	84.5
3	ADE	0.560	0.938	40.3
	FDE	0.873	1.654	47.2
	AMD	5.627	4.443	-26.7
	AMV	0.018	0.121	84.6

reduced uncertainty, and a 45.6% lower AMD (4.430 vs. 8.151), showing closer alignment with true pedestrian distributions. These results highlight the SI model’s superior reliability in simple, predictable settings.

As interaction complexity increases, the SI model maintains better results in ADE (0.561 vs. 0.746, 24.7% improvement) and FDE (0.859 vs. 1.338, 35.8% improvement), though the margin narrows compared to the 1-pedestrian case. This is expected, as multi-pedestrian interactions introduce more variables such as crossing paths and group dynamics. However, the AMD (5.367 vs. 3.226) increases by 66.4% for the SI model, indicating a broader, more conservative prediction space. This suggests the SI model accounts for a wider range of possible pedestrian behaviors to avoid collisions, even if it slightly overestimates uncertainty. The AMV remains 84.5% lower (0.018 vs. 0.120), reinforcing that the SI model’s predictions are still more stable and less erratic than the CV model’s.

In high-density environments, the SI model continues to outperform the CV model in ADE (0.560 vs. 0.938, 40.3% improvement) and FDE (0.873 vs. 1.654, 47.2% improvement). These reductions in er-

ror demonstrate the SI model's robustness in highly dynamic settings, where pedestrians may abruptly change direction or speed. Like the 2-pedestrian case, the AMD (5.627 vs. 4.443) is 26.7% higher for the SI model, further emphasizing its cautious approach in crowded scenarios. This conservative behavior ensures safer navigation but may slightly reduce path optimality. The AMV (0.018 vs. 0.121) remains 84.6% lower, confirming that the SI model's predictions are statistically more reliable despite the increased complexity.

Table 3: Comparison of ADE, FDE, AMD, and AMV for open- and closed-loop evaluation.

Method	ADE	FDE	AMD	AMV
Open Loop (ETH data)	0.642	1.379	3.047	0.127
Closed Loop (SI-MPC)	0.525	0.806	5.141	0.018

Table 3 reveals a nuanced relationship between open-loop benchmark performance and real-world behavior in a full navigation stack. The SI model, trained solely on the ETH dataset (early stopping at epoch 45, minimum validation loss 0.053), achieves better trajectory accuracy (ADE: 0.525 vs. 0.642; FDE: 0.806 vs. 1.379) but shows higher AMD (5.141 vs. 3.047), indicating a broader predictive distribution during operation. Its much lower AMV (0.018 vs. 0.127) further suggests a more cautious, stable behavior in real-time planning. This highlights a key insight: strong open-loop metrics do not guarantee equivalent closed-loop performance. Integration into a navigation stack introduces constraints—like real-time decision-making and safety—that alter model behavior in ways not captured by static evaluations. Thus, open-loop metrics, while useful, must be paired with system-level evaluations to assess true operational generalization.

Table 4: Closed-loop performance comparison of SI and CV motion prediction methods.

# Ped.	Metric	SI	CV	Improv. (%)
1	Min. Dist. (m)	0.50	0.31	61.80
	Time Taken (s)	14.80	18.77	21.10
	Jerk (m/s^3)	0.13	0.19	32.35
	MPC MSE	0.02	0.02	12.54
2	Min. Dist. (m)	0.30	0.22	36.70
	Time Taken (s)	24.82	19.76	-25.50
	Jerk (m/s^3)	0.17	0.17	1.08
	MPC MSE	0.024	0.020	-23.08
3	Min. Dist. (m)	0.26	0.19	41.10
	Time Taken (s)	21.22	18.65	-13.80
	Jerk (m/s^3)	0.21	0.30	28.38
	MPC MSE	0.03	0.04	18.61

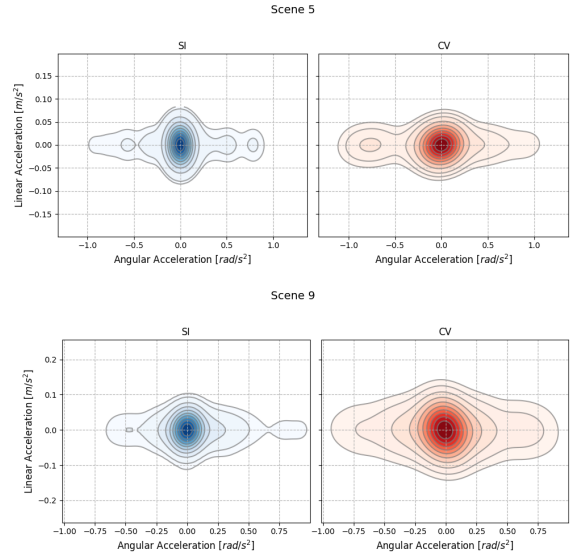


Figure 4: (Top) Linear vs. angular acceleration contour plots for Scene 5. (Bottom) Linear vs. angular acceleration contour plots for Scene 9.

The comparative analysis in Table 4 of minimum distance and time taken reveals key behavioral differences between the SI and CV models. In single-pedestrian scenarios, the SI model maintained a 61.8% larger safety margin (0.50 m vs. 0.31 m) while completing the trajectory 21.1% faster (14.80 s vs. 18.77 s), due to its proactive adaptation to pedestrian movement. As pedestrian density increased, the SI model prioritized safety over speed: in two-pedestrian cases, it preserved a 36.7% larger distance (0.30 m vs. 0.22 m) but took 25.5% more time (24.82 s vs. 19.76 s); in three-pedestrian scenarios, it achieved a 41.1% safety gain (0.268 m vs. 0.190 m) with a 13.8% time increase (21.22 s vs. 18.65 s). These results highlight the practical advantages of an MPC planner using the SI model.

Motion smoothness, measured by jerk (rate of acceleration change), directly influences pedestrian trust. In single-pedestrian tests, the SI model reduced jerk by 31.6% (0.13 m/s^3 vs. 0.19 m/s^3), indicating smoother velocity transitions. For two pedestrians, both models showed similar jerk (0.17 m/s^3), though the SI model had more consistent acceleration patterns (e.g., gradual deceleration in Test 6 vs. abrupt stops in CV's Test 4). The SI model's advantage was more evident in dense scenarios: with three pedestrians, it lowered jerk by 28.4% (0.21 m/s^3 vs. 0.30 m/s^3), avoiding CV's erratic behavior (e.g., sharp deceleration in Test 7). This aligns better with human locomotion norms, reducing discomfort and improving social acceptance.

Prediction reliability, measured via Mean Squared

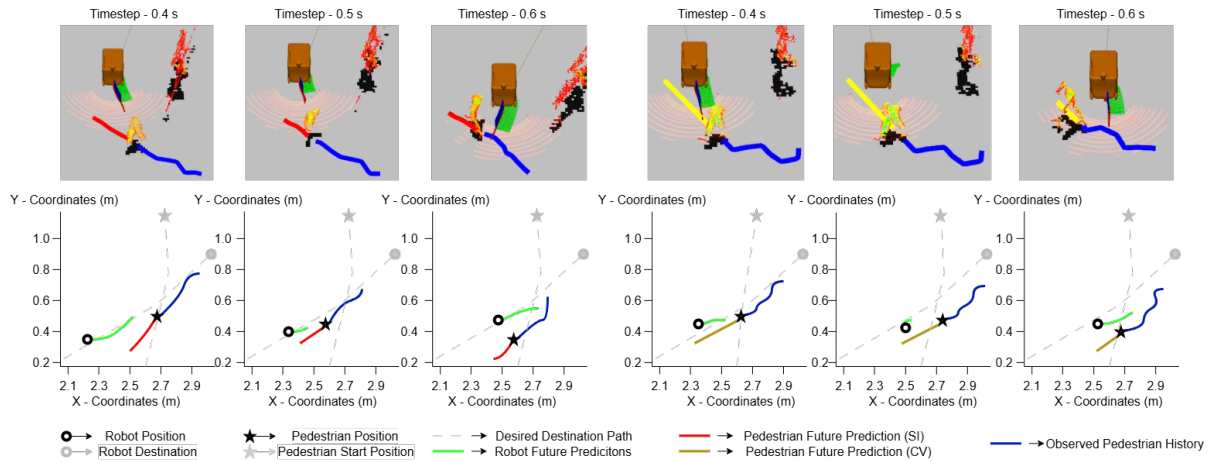


Figure 5: Robot navigation (Scene 1) through crowded environments using SI and CV models. Top row: Runtime snapshots from RVIZ. Bottom row: Corresponding top-down schematic representations of trajectories.

Error (MSE), further distinguishes the models. The SI model outperformed CV in 8 of 10 tests, with a 66.7% lower MSE in Test 1 (0.005 vs. 0.017). This accuracy stems from its socially aware trajectory forecasting, which accounts for pedestrian intent. Notably, the CV model excelled in Test 10 (MSE: 0.026 vs. SI's 0.049), likely due to the SI model's overcompensation in dense crowds. While this outlier suggests occasional over-cautiousness, the SI model's overall lower MSE (e.g., 37.6% improvement in Test 5) confirms its robustness in most real-world conditions.

4.4 Qualitative Analysis

Fig. 4 shows joint linear versus angular acceleration contour plots for two pedestrian interaction scenes. In both, the SI-based MPC produces more compact, elliptical contours centered near the origin, indicating lower control effort and more stable acceleration. This suggests that socially informed predictions help the robot anticipate future states better, reducing the need for corrective maneuvers. In contrast, the CV-based MPC yields broader, more asymmetric distributions, especially in angular acceleration, reflecting more variable and reactive control inputs due to its limited modeling of pedestrian behavior. This effect is pronounced in Scene 9, where higher angular acceleration variance indicates frequent adjustments and poor predictive accuracy. These findings imply that socially aware prediction improves trajectory smoothness and control efficiency, while naive models like CV increase instability and control burden, potentially compromising safety and comfort in close human-robot interactions.

Figure 5 illustrates Scene 1, where the robot navigates from SP to GP1 as a pedestrian moves from

Red 1 to Red 2 under both CV and SI prediction models (see Table 1 and Fig. 2). The visualization shows that the CV model, due to its simplistic and localized predictions, prompts the robot to halt abruptly when a potential collision is detected, particularly at 0.5s and 0.6s. In contrast, the SI model enables smoother, adaptive behavior: the robot reduces speed and adjusts its path gradually, reflecting a more temporally consistent understanding of pedestrian motion. This qualitative difference aligns with quantitative metrics, especially jerk values, where lower values in the SI case confirm its advantage.

Additionally, the overall time taken to reach the goal with SI-MPC is more reasonable and efficient, as it avoids unnecessary stops and abrupt actions, thereby aligning with the observed improvements in time-to-goal metrics presented in the quantitative analysis. Thus, this scene effectively highlights the advantages of the SI model in generating socially compliant and efficient navigation for the robot.

5 CONCLUSION

This study presents three key advances in robot navigation using a deep learning-based Social-Implicit (SI) predictor integrated with MPC on a real robot. First, the closed-loop system demonstrates real-time feasibility at 10 Hz, validating stable MPC performance with neural predictors. Second, comparisons reveal a mismatch between open-loop and closed-loop behavior—despite improved prediction accuracy (e.g., ADE: 0.525 vs. 0.642), closed-loop deployment led to broader, more cautious predictions, emphasizing the need for system-level evaluation. Third, across varied pedestrian densities, the SI-MPC out-

performed the CV-MPC approach, improving safety margins (by 41–75%), reducing motion jerk (28–84%), and achieving faster goal attainment (21–47%), with only moderate efficiency trade-offs in dense crowds. Overall, learned predictors enhance real-world safety and social compliance, though future work should address scalability and conservatism-efficiency trade-offs.

ACKNOWLEDGEMENTS

This work was funded by the German Federal Ministry for Economic Affairs and Climate Action within the project *nxtAIM*.

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 *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971. IEEE.
- Bhatt, N. P., Khajepour, A., and Hashemi, E. (2023). MPC-PF: Socially and Spatially Aware Object Trajectory Prediction for Autonomous Driving Systems Using Potential Fields. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):5351–5361.
- Brito, B., Floor, B., Ferranti, L., and Alonso-Mora, J. (2019). Model predictive contouring control for collision avoidance in unstructured dynamic environments. *IEEE Robotics and Automation Letters*, 4(4):4459–4466.
- Caruana, R., Lawrence, S., and Giles, C. (2000). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press.
- Chen, Y., Rosolia, U., Ubellacker, W., Csomay-Shanklin, N., and Ames, A. D. (2022). Interactive multimodal motion planning with branch model predictive control. *IEEE Robotics and Automation Letters*, 7(2):5365–5372.
- de Groot, O., Ferranti, L., Gavrila, D., and Alonso-Mora, J. (2023). Globally guided trajectory planning in dynamic environments.
- Francis, A., Pérez-D’Arpino, C., Li, C., Xia, F., Alahi, A., Alami, R., Bera, A., Biswas, A., Biswas, J., Chandra, R., Chiang, H.-T. L., Everett, M., Ha, S., Hart, J., How, J. P., Karnan, H., Lee, T.-W. E., Manso, L. J., Mirksy, R., Pirk, S., Singamaneni, P. T., Stone, P., Taylor, A. V., Trautman, P., Tsoi, N., Vázquez, M., Xiao, X., Xu, P., Yokoyama, N., Toshev, A., and Martín-Martín, R. (2023). Principles and guidelines for evaluating social robot navigation algorithms.
- Golchoubian, M., Ghafurian, M., Dautenhahn, K., and Azad, N. L. (2023). Pedestrian trajectory prediction in pedestrian-vehicle mixed environments: A systematic review. *IEEE Transactions on Intelligent Transportation Systems*, 24(11):11544–11567.
- Huang, L., Zhuang, J., Cheng, X., Xu, R., and Ma, H. (2021). STI-GAN: Multimodal pedestrian trajectory prediction using spatiotemporal interactions and a generative adversarial network. *IEEE Access*.
- Korbmacher, R. and Tordeux, A. (2022). Review of pedestrian trajectory prediction methods: Comparing deep learning and knowledge-based approaches.
- Lian, J., Ren, W., Li, L., Zhou, Y., and Zhou, B. (2023). PTP-STGCN: Pedestrian trajectory prediction based on a spatio-temporal graph convolutional neural network. *Applied Intelligence*, 53(3):2862–2878.
- Mangalam, K., An, Y., Girase, H., and Malik, J. (2020). From goals, waypoints & paths to long term human trajectory forecasting.
- Mohamed, A., Qian, K., Elhoseiny, M., and Claudel, C. (2020). Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14412–14420. ISSN: 2575-7075.
- Mohamed, A., Zhu, D., Vu, W., Elhoseiny, M., and Claudel, C. (2022). Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation.
- 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*, pages 261–268.
- Piccinelli, N., Vesentini, F., and Muradore, R. (2023). MPC Based Motion Planning For Mobile Robots Using Velocity Obstacle Paradigm. In *2023 European Control Conference (ECC)*, pages 1–6.
- Salzmann, T., Ivanovic, B., Chakravarty, P., and Pavone, M. (2021). Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data.
- Sani, M., Robu, B., and Hably, A. (2021). Dynamic obstacles avoidance using nonlinear model predictive control. In *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–6.
- Sighencea, B. I., Stanciu, R. I., and Căleanu, C. D. (2021). A review of deep learning-based methods for pedestrian trajectory prediction. *Sensors*, 21(22):7543.
- Vo, V. D., Huy Pham, Q. A., Le, Q. T., Le, M. H., and Tran, D. T. (2024). A model predictive control for dynamic obstacle avoidance of a mobile robot based on ros. In *2024 7th International Conference on Green Technology and Sustainable Development (GTSD)*, pages 299–303.
- Wabersich, K. P., Taylor, A. J., Choi, J. J., Sreenath, K., Tomlin, C. J., Ames, A. D., and Zeilinger, M. N. (2023). Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems*, 43(5):137–177.