

# HINT-Bench: Human INTention Recognition Benchmark for Social Robotics

Valerio Bo<sup>1</sup>, Anais Garrell<sup>1</sup>, and Alberto Sanfeliu<sup>1</sup>

<sup>1</sup> Institut de Robotica y Informatica Industrial (CSIC-UPC) and Universitat Politecnica de Catalunya - BarcelonaTech (UPC), Barcelona, Spain  
{name.surname}@iri.upc.edu

**Abstract.** In modern indoor environments such as hospitals, offices, and homes, service robots must move beyond reactive behaviors and anticipate user needs by inferring human intent. Early intention recognition enables proactive assistance, thereby enhancing efficiency, safety, and user experience. We present an open-source benchmark suite for early human intention recognition that integrates (1) a high-fidelity Gazebo simulation with ROS 1, featuring three Soft Actor–Critic (SAC)-trained agents modeling *collaborative*, *neutral*, and *adversarial* behaviors; (2) multimodal perception comprising 9D LiDAR/odometry state vectors and 135D MediaPipe skeleton keypoints; and (3) two curated datasets: a 300-episode training set and a 300-episode test set pre-sliced into 500 spatial (1–5 *m*) and 500 temporal (1–9 *s*) trigger snapshots per class. We benchmark six baseline methods, including approaches based on trajectory or skeleton data. Our unified evaluation toolkit computes accuracy, precision, recall, F1 score, mean time-to-correct-prediction, noise robustness, and inference latency (CPU/GPU). All code, data, and scripts are available at <https://github.com/valerio-bo/HINT-Bench>, offering a reproducible platform to accelerate research in anticipative human–robot collaboration.

**Keywords:** intent prediction, human-robot interaction, social robotics

## 1 Introduction

As robots become ubiquitous in indoor settings, from hospital corridors [1] and office spaces [2] to domestic living rooms [3], they must anticipate human intentions rather than merely react to presence. Inferring whether a person intends to interact with the robot (*collaborative*), simply pass by (*neutral*), or avoid the robot (*adversarial*) is crucial for planning safe, efficient, and context-aware robot behaviors [4,5]. Early intention recognition, defined as correctly predicting intent before task completion [6], enables robots to initiate assistance at precisely the right moment [7,8].

Existing benchmarks for human action or trajectory forecasting (e.g., NTU RGB+D [9], ETH/UCY [10,11]) focus on general activity recognition or crowd

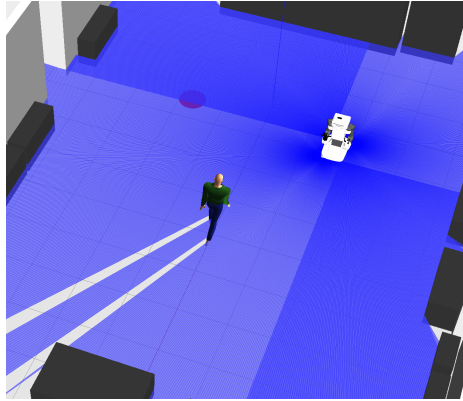


Fig. 1: **Simulated environment setup.** Laboratory environment with the human agent, the robot, and the goal location. The human’s trajectory toward or away from the robot defines collaborative, neutral, and adversarial behaviors.

navigation but do not capture the timing requirements or multimodal perceptions necessary for early intention prediction in HRI [12]. Recent works have explored intention-aware navigation [13, 14] and human motion prediction [15, 16], yet lack standardized benchmarks for evaluating anticipatory capabilities in close-proximity human-robot encounters. To address this gap, we propose a simulation-based benchmark combining:

1. *Behavioral Agents*: Three SAC-trained policies [17] encoding collaborative, neutral, and adversarial human behaviors in a Gazebo 11/ROS Noetic environment.
2. *Multi-Modal Sensing*: 9D LiDAR/odometry state vectors and 135D MediaPipe skeleton keypoints.
3. *Curated Datasets*: A 300-episode training set for model development and a 300-episode test set pre-sliced into spatial (1–5 m) and temporal (1–9 s) triggers, yielding 500 snapshots per class per trigger type.
4. *Baseline Methods*: Six models spanning zero-training heuristics (Geometric Alignment), deep sequential learners on LiDAR (Transformer [18], LSTM [19]), deep sequential learners on skeletons (STGCNN [20], siMLPe [21]), and a model fusing LiDAR and skeleton outputs [22].
5. *Evaluation Toolkit*: A unified script to compute accuracy, precision, recall, F1, mean time-to-correct-prediction, robustness to skeleton jitter, and inference latency on CPU and GPU.

To develop and validate these capabilities, we build a simulation environment comprising a human agent, a mobile robot, and a designated goal, as shown in Fig. 1. By releasing all code, data, environment configurations, and evaluation scripts at <https://github.com/valerio-bo/HINT-Bench>, we provide the community with a reproducible and extensible platform for developing and comparing early intention recognition methods in human–robot collaboration [23].

## 2 Simulation Environment and Behavior Modeling

We built our benchmark upon a high-fidelity Gazebo11 simulation seamlessly integrated with ROS Noetic, providing both realistic physics and sensor emulation while preserving reproducibility and ease of extension. We model a  $6m \times 10m$  laboratory environment complete with walls, a walking human agent, and a marked delivery zone. We defined all world geometry and visual assets in `environments/lab.world`. The benchmark centers on an approaching task in which a mobile robot and a human share the same laboratory environment.

The human agent moves toward a designated goal, toward the robot itself or stays in a generic area, while the robot standing still must infer the human’s underlying intention in real time and act accordingly. Sensor noise parameters, robot and human spawn poses are all configurable via `config/sim_params.yaml`, allowing easy replication or modification of experimental conditions.

The human agent is modeled with realistic mass properties. Rather than scripting its path, we control its motion through velocity commands published on `/human_cmd_vel`. A dedicated reinforcement-learning node subscribes to the shared environment state topic `/env_state`, which publishes a 12-dimensional vector and issues continuous actions  $(a_1, a_2)$  that drive the agent via the ROS navigation stack. This modular approach separates perception, state representation, and control, and enables swapping in alternative policy implementations without altering the core simulation.

To capture a spectrum of human–robot interactions, we define three prototypical intention classes:

- **Collaborative:** The agent’s primary objective is to get close to the robot’s current location. Even if this deviates from the shortest path to the delivery area, the agent will adjust its trajectory to approach the robot.
- **Neutral:** The agent performs a random-waypoint exploration, ignoring both the robot and any delivery tasks. This models bystanders or passers-through.
- **Adversarial:** The agent must get close to a fixed drop-off point while actively maintaining a minimum 1.0 m distance from the robot at all times, simulating a user who wishes to avoid contact.

We train three distinct policies, one per intent, using the Soft Actor–Critic (SAC) algorithm [17]. SAC learns a stochastic policy  $\pi(a | s)$  by optimizing a trade-off between expected return and entropy, which encourages robust exploration. At each timestep, the policy observes  $s_t \in \mathbb{R}^{12}$  and outputs continuous linear and angular velocity commands  $(a_1, a_2)$ . The dense reward is defined by

$$r_t = \begin{cases} +5 \Delta_{\text{goal}} - 2|\omega_A|, & \text{if not neutral,} \\ -2(|v_A| + |\omega_A|), & \text{otherwise} \end{cases}$$

where  $\Delta_{\text{goal}}$  is the reduction in Euclidean distance toward the current target (robot or drop-off), and  $(v_A, \omega_A)$  are the agent’s commanded speeds. We also apply a terminal reward of  $\pm 100$  for successful and failed deliveries, as well as for exiting the environment bounds.

To improve learning efficiency, 20% of early successful trajectories are seeded into the replay buffer. Each policy is trained for 3k environment steps, after which performance saturates. Fig. 2 shows the actor loss, critic Q-value estimates, and episodic returns over training time. This simulation and behavior

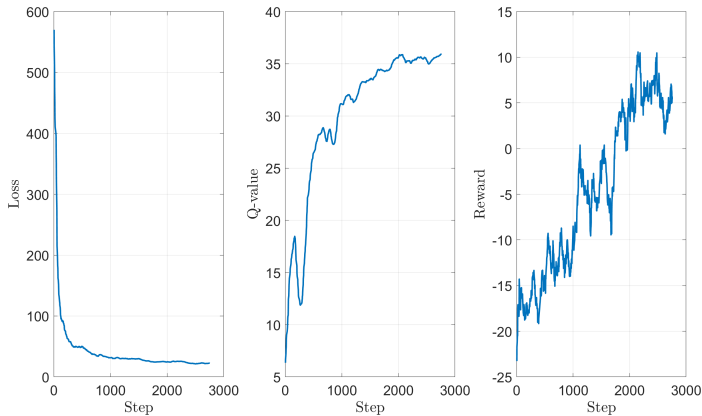


Fig. 2: **SAC training curves.** Average of the losses, Q-values, and rewards for the trained agents.

modeling pipeline yields richly annotated, multimodal datasets that serve as the foundation for our intention recognition benchmark. In the next section, we describe how this data is harvested into training sets and carefully pre-sliced trigger snapshots for rigorous evaluation.

### 3 Datasets and Modalities

To comprehensively evaluate early intention recognition methods, we curate three interrelated datasets: one for LiDAR state sequences, one for skeleton observations, and a specialized test set aligned with spatial and temporal triggers. Together, they provide both continuous trajectories for training and targeted snapshots for rigorous, early-prediction benchmarking.

The first step for training a model for predicting the intention consisted of the creation of the first dataset. This dataset must contain features related to human motion derived from LiDAR. For this reason, we adopted the SPENCER algorithm to detect people in the 2D point cloud [24]. Once detected, we tracked the person’s position, orientation, and velocities. Separately, we used the robot’s odometry to record its position during the simulations. Thus, we obtained a state vector at 10 Hz:

$$l_t = [x_R, y_R, \gamma_R, x_H, y_H, \gamma_H, v_H^x, v_H^y, \omega_H^z] \in \mathbb{R}^9,$$

where  $(x, y, \gamma)$  denote planar position and heading for robot ( $R$ ) and human ( $H$ ), and  $(v_H^x, v_H^y, \omega_H^z)$  capture the human’s instantaneous linear and angular velocities. Over 300 episodes, with 100 episodes for each policy, we log  $l_t$ , the action tuple  $(a_1, a_2)$  issued by the policy, the precise timestamp  $t$ , and the ground-truth intent label  $y \in \{0, 1, 2\}$  for adversarial, collaborative, and neutral intent respectively. This yields approximately 180,000 timesteps of synchronized data. We partition this corpus into 60% training, 20% validation, and 20% hold-out test sets, ensuring that episodes from each intent class are evenly distributed. The entire dataset derived from the LiDAR resides in `datasets/bench_lidar.csv`.

In parallel with LiDAR logging, we simulate a MediaPipe skeleton tracker on the TIAGO’s RGB-D feed. Each camera frame yields up to 33 keypoints with 3D coordinates and a confidence score. After flattening and padding to a fixed 135-dimensional vector, we time-synchronize these skeleton observations with the LiDAR state stream. The resulting skeleton dataset, also collected over the same 300 episodes, is likewise split 60/20/20 for training, validation, and test. This paired LiDAR + skeleton dataset enables multimodal model development and ablation studies, with all related files stored under `data/skeleton_dataset/`.

To probe early-prediction performance under well-defined conditions, we generate an additional 300 episodes solely for evaluation. From each episode, we extract two complementary sets of *trigger snapshots*:

1. *Spatial Triggers*: the first simulation frame at which the Euclidean distance  $d_{HR} = \|[x_H, y_H] - [x_R, y_R]\|_m$  crosses each threshold in  $\{1, 2, 3, 4, 5\} m$ .
2. *Temporal Triggers*: the frame occurring at each fixed elapsed time in  $\{1, 3, 5, 7, 9\} s$  from the episode start.

Because there are 100 episodes per intent class, each trigger type yields  $5 \times 100 = 500$  samples per class, for a total of 1500 spatial and 1500 temporal snapshots. Each snapshot packs the LiDAR-state vector, the corresponding skeleton keypoints, and the true intent label into a single `.pkl` file, organized under `data/triggers/`. This targeted dataset enables a direct comparison of methods at precisely the same decision points, facilitating a fair evaluation of how early and accurately models can infer human intent.

## 4 Baseline Models

To characterize the landscape of early intention recognition, we compare six baseline methods that span from zero-training heuristics to deep temporal architectures processing LiDAR-state, skeleton, or both:

- **Geometric Alignment (Geo)**: A zero-training heuristic computing

$$c_R = \frac{1 + \mathbf{u}_R \cdot \mathbf{v}}{2},$$

where  $\mathbf{u}_R$  is the unit vector from human to robot and  $\mathbf{v}$  is the recent human velocity direction. The method additionally evaluates the consistency

of the movement direction across a temporal window: if the velocity direction fluctuates significantly over time, the behavior is considered neutral; on the other hand, if the movement is consistent, the snapshot is labeled as adversarial if  $c_R$  is low, while  $c_R \approx 1$  implies collaborative labels.

- **Transformer on LiDAR (Trans)**: Adapted from Aksan et al. [18], this model applies self-attention over past 10 LiDAR-state vectors  $l_{t-9}, \dots, l_t \in \mathbb{R}^{10}$  and classifies intent via a final softmax head.
- **LSTM on LiDAR (LSTM)**: Following Zhao et al. [19], a two-layer LSTM ingests 8-step overlapping windows of  $s_t$ , then splits into a binary “engagement” branch (sigmoid) and a 3-way “intent” branch (softmax), trained with combined cross-entropy losses.
- **STGCNN on Skeleton (STGCNN)**: Inspired by Yan et al. [20], this model learns spatio-temporal graph convolutions over human joint trajectories to predict intent from motion dynamics across time.
- **siMLPe on skeleton (siMLPe)**: Inspired by Gomez-Izquierdo et al. [21], 20 past skeleton frames undergo DCT encoding, shared Transformer layers, and dual heads for future-pose reconstruction (L2 loss) and intent classification (cross-entropy).
- **Fusion (Fusion)**: A late fusion averaging the probabilities from a LiDAR-LSTM and a skeleton-STGCNN model to produce an intent prediction.

Table 1 summarizes the input modality, model type, and key characteristics of each baseline.

Table 1: **Baseline model summary.** Description of modality, architecture, trainability and latency for each model.

Model	Modality	Architecture	Trainable	Latency (ms)	
				CPU	GPU
Geo	LiDAR	Dot-product heuristic	No	0.01	0.005
Trans	LiDAR	Transformer encoder	Yes	0.45	0.08
LSTM	LiDAR	2-layer LSTM	Yes	0.68	0.12
STGCNN	Skeleton	Spatio-temporal+GCNN	Yes	0.82	0.15
siMLPe	Skeleton	DCT+Transformer	Yes	1.20	0.18
Fusion	Both	Probability average	Yes	1.95	0.28

## 5 Evaluation Protocol

We evaluate each model’s performance using multiple metrics, measured across both spatial triggers (e.g., distance thresholds to the robot or object) and temporal triggers (e.g., time-based checkpoints before or during interaction). The metrics aim to capture not only classification performance but also prediction responsiveness and system robustness under varying conditions.

- **Accuracy, Precision, Recall, F1-score:** These standard classification metrics are first computed per class and then macro-averaged. This enables the evaluation of performance in balanced and unbalanced class scenarios.
- **Mean Time-to-Correct Prediction (MTCP):** Measures the average delay (in seconds) between a trigger (spatial or temporal) and the model’s first correct prediction that remains stable (i.e., sustained over five subsequent frames), assessing the responsiveness and temporal stability of the models.
- **Noise Robustness:** Evaluates how well each model performs under synthetic perturbations. Gaussian noise with standard deviation  $\sigma \in 0.0, 0.05, 0.1$  meters is added to LiDAR-state and skeleton joint coordinates to simulate sensor jitter. Accuracy is reported at each noise level to measure robustness to input uncertainty.
- **Inference Latency:** Reports the average forward-pass runtime per frame (in milliseconds), measured separately on CPU (Intel i7-9700K) and GPU (NVIDIA RTX 2070). This metric reflects the real-time suitability of each model on resource-constrained platforms.

All evaluations are performed using a single, unified script that processes predictions for each model under various settings. The script supports batch evaluation over multiple noise levels, trigger types, and compute devices. The output is a structured CSV summary of all results.

```
python scripts/evaluate.py --triggers data/triggers --models geo
tran lstm stg mlp fuse --noise 0.0 0.05 0.1 --devices cpu gpu
--out results/summary.csv
```

## 6 Results

We report results across four evaluation axes: classification accuracy, timeliness, robustness to sensors noise, and inference latency. Models were assessed at spatial and temporal triggers using the evaluation protocol detailed in Sec. 5.

### 6.1 Overall Classification Performance

Classification performance at spatial and temporal triggers reveals clear distinctions among the baselines. We report in Table 2 the condition where most models achieved highest performance (3 meters and 5 seconds). The Fusion model achieves the highest accuracy at 89.4%, demonstrating the power of multimodal integration. Among single-modality approaches, siMLPe leads with 85.1%, leveraging DCT-based temporal encoding, followed closely by STGCNN (83.7%). The Transformer model demonstrates strong performance at 82.3%, substantially outperforming LSTM (74.5%), highlighting the advantages of attention mechanisms over recurrent architectures for this task. Geo lags significantly behind at 58.2%, confirming that simple velocity-direction alignment heuristics are insufficient for robust intent discrimination. These results underscore the advantage of sophisticated temporal modeling, particularly skeleton-based architectures, with multimodal fusion providing the strongest performance.

Table 2: **Spatial-trigger metrics.** Spatial-trigger classification metrics (%) at 3 *m* and 5 *s* and Spatial-trigger MTCP (*s*) at each distance.

Model	Acc	Prec	Rec	F1	MTCP ( <i>s</i> )				
					1 <i>m</i>	2 <i>m</i>	3 <i>m</i>	4 <i>m</i>	5 <i>m</i>
Geo	58.2	59.1	56.4	57.7	4.85	5.42	6.10	6.78	7.55
Trans	82.3	83.0	81.2	82.1	2.45	2.78	3.15	3.55	3.98
LSTM	74.5	75.8	71.3	73.5	3.10	3.48	3.92	4.38	4.87
STGCNN	83.7	82.1	82.9	82.5	2.28	2.59	2.93	3.30	3.69
siMLPe	85.1	86.3	84.2	85.2	2.12	2.41	2.72	3.06	3.42
Fusion	89.4	88.7	88.1	88.4	1.85	2.10	2.38	2.67	2.98

## 6.2 Timeliness via MTCP

Timeliness, measured by Mean Time-to-Correct Prediction (MTCP), provides a complementary view of model performance by assessing how quickly models converge on correct predictions. Table 2 and Table 3 report MTCP under spatial and temporal trigger conditions, respectively. Under spatial triggers, MTCP rankings closely mirror accuracy performance, with the Fusion model achieving the fastest convergence at 1.85 seconds at 1 *m*, extending to 2.98 seconds at 5 *m*. The skeleton-based siMLPe demonstrates excellent temporal stability with MTCPs ranging from 2.12 to 3.42 seconds, outperforming STGCNN (2.28–3.69 *s*) despite similar accuracy levels, suggesting that DCT-based temporal encoding provides smoother predictions over time. The Transformer model achieves respectable convergence times (2.45–3.98 *s*), significantly faster than LSTM (3.10–4.87 *s*), indicating that attention mechanisms not only improve accuracy but also prediction consistency. Geo exhibits the poorest performance with MTCPs from 4.85 to 7.55 *s*, reflecting both low accuracy and high prediction instability.

Temporal-triggered MTCP results in Table 3 reveal a consistent decreasing trend across all models as the trigger occurs later. This pattern confirms that longer observation windows provide richer motion cues, enabling faster convergence to stable predictions. Fusion maintains its advantage throughout, with MTCPs decreasing from 3.01 *s* at 1 *s* to 1.89 *s* at 9 *s*. The skeleton-based models exhibit significant improvement over time, with siMLPe achieving 2.17 seconds and STGCNN reaching 2.33 seconds at the 9-second mark. Notably, the gap between high and low-performing models narrows as more temporal context becomes available, though Geo remains substantially slower (4.88 *s* at 9 *s*). These findings demonstrate that both modality choice and architectural design significantly impact also the speed and stability of predictions.

## 6.3 Noise Robustness

To evaluate resilience to perception noise, we injected Gaussian jitter into LiDAR-state data and skeleton joints and measured accuracy at 3 *m* and 5 *s* under

Table 3: **Temporal-trigger metrics and noise robustness.** Temporal-trigger MTCP (s) at each time and accuracy (%) vs. data noise  $\sigma$  at 3 m and 5 s.

Model	MTCP (s)					Accuracy		
	1 s	3 s	5 s	7 s	9 s	$\sigma = 0.0$	$\sigma = 0.05$	$\sigma = 0.10$
Geo	7.62	6.89	6.08	5.42	4.88	58.2	53.1	47.8
Trans	3.98	3.54	3.12	2.78	2.49	82.3	79.7	76.9
LSTM	4.92	4.38	3.87	3.45	3.10	74.5	71.2	67.8
STGCNN	3.71	3.31	2.92	2.60	2.33	83.7	79.8	75.3
siMLPe	3.45	3.07	2.71	2.42	2.17	85.1	82.3	79.2
Fusion	3.01	2.68	2.37	2.11	1.89	89.4	87.8	86.1

increasing noise levels (Table 3). The results reveal striking differences in robustness across architectures. Geo shows catastrophic degradation, dropping from 58.2% to 47.8% accuracy ( $-10.4\%$ ) at  $\sigma = 0.1$ , highlighting the brittleness of heuristic approaches. In contrast, Fusion demonstrates exceptional robustness with only a 3.3% drop (89.4% to 86.1%), benefiting from redundancy across modalities. Among single-modality models, siMLPe exhibits moderate resilience ( $-5.9\%$ ), likely due to DCT’s frequency-domain representation, which filters out high-frequency noise. The Transformer exhibits similar robustness ( $-5.4\%$ ), with attention mechanisms potentially focusing on more reliable features. STGCNN and LSTM show larger drops ( $-8.4\%$  and  $-6.7\%$  respectively), suggesting that spatial graph structures and sequential processing are more sensitive to input perturbations. These results confirm that multimodal fusion not only enhances baseline accuracy but provides crucial robustness for real-world deployment where sensor noise is inevitable.

#### 6.4 Inference Latency

Latency was measured on 10000 forward passes per model and hardware configuration. As shown in Table 1, computational requirements vary dramatically across architectures. Geo achieves negligible latency (0.01 ms CPU) due to simple dot-product computation, while LSTM offers the best efficiency among learned models (0.45 ms CPU). Skeleton-based models exhibit moderate latency (STGCNN: 0.68 ms, siMLPe: 0.82 ms), with the Transformer being higher (1.20 ms) due to the complexity of its attention mechanism. Fusion incurs the highest cost (1.95 ms) from dual-modality processing. GPU acceleration provides substantial speedups, particularly for parallelizable architectures. All models operate well within real-time constraints, confirming their suitability for interactive robotics. The accuracy-latency trade-off favors sophisticated architectures, with Fusion providing best performance at acceptable computational cost.

## 7 Discussion

The comparative analysis of baseline models reveals key insights into the trade-offs between model complexity, modality, and early intent recognition. The geometric baseline (**Geo**) offers negligible latency (0.01 *ms*) but low accuracy (58.2%) and poor robustness (−10.4% at  $\sigma = 0.1$ ). Its MTCP of 6.10 *s* at 3 *m* confirms its inadequacy for timely prediction in social navigation. Among LiDAR models, **Trans** outperforms **LSTM** in accuracy (82.3% vs. 74.5%), benefiting from long-range attention, though with higher latency (1.20 *ms* vs. 0.45 *ms*). Interestingly, LSTM provides competitive MTCP at short ranges, indicating temporally stable predictions despite lower peak accuracy. Trans also exhibits better noise resilience (−5.4% vs. −6.7%). Skeleton-based models underscore the relevance of human pose. **siMLPe** achieves the highest accuracy (85.1%) and better noise handling (−5.9%) due to DCT-based encoding. **STGCNN**, though slightly less accurate (83.7%), suffers more from noise (−8.4%), indicating higher sensitivity in real-world conditions. **Fusion** achieves top performance across metrics: 89.4% accuracy, fastest MTCP (2.38 *s*), and best robustness (−3.3%). Despite having the highest latency (1.95 *ms*), it remains capable of real-time performance. Fusion benefits from complementary modalities, where LiDAR ensures stable global motion, while skeletons capture fine behavioral cues. MTCP analysis shows that high accuracy often aligns with faster intent recognition. Still, architectural features like DCT can enhance temporal smoothness, improving responsiveness. From a deployment perspective, **siMLPe** suits single-modality scenarios needing low latency and high accuracy. For safety-critical applications, **Fusion**’s superior reliability justifies its computational cost. These results set strong baselines for intent recognition and highlight priorities: enhancing skeleton robustness, simplifying fusion, and optimizing both accuracy and temporal consistency. The gap between Geo and learned models underscores the need for sophisticated temporal reasoning over simple heuristics.

## 8 Conclusion and Future Works

In this work, we introduced a comprehensive, open-source benchmark suite for early human intention recognition in human–robot interaction scenarios. Our framework leverages Gazebo 11 and ROS Noetic to simulate realistic indoor environments with a TIAGo robot and reinforcement-learned human agents exhibiting *collaborative*, *neutral*, and *adversarial* behaviors, yielding over 360,000 timesteps of synchronized LiDAR and skeleton data. We curated continuous sequence datasets for training and pre-sliced trigger sets (500 spatial and 500 temporal snapshots per intent) for early-prediction evaluation. Our baseline suite spans from simple heuristics (Geo) to deep temporal architectures (LSTM, Transformer, STGCNN, siMLPe) and their multimodal fusion. The evaluation protocol measures classification accuracy, Mean Time-to-Correct Prediction (MTCP), noise robustness, and inference latency. Results demonstrate that multimodal fusion achieves 89.4% accuracy with exceptional noise robustness and

fastest MTCP convergence. Among single-modality models, skeleton-based siMLPe (85.1%) outperforms LiDAR-based approaches, though with higher noise sensitivity. All models operate within real-time constraints. By open-sourcing all resources at <https://github.com/valerio-bo/HINT-Bench>, we aim to accelerate research in anticipatory robot behavior.

Several promising future directions emerge from this work. First, sim-to-real transfer validation remains crucial, considering that deploying trained models on physical robots will reveal the domain gap between simulated and real sensor data, particularly for skeleton tracking under occlusions. Second, extending to multi-human scenarios would better reflect crowded environments where robots must simultaneously track and predict intentions of multiple agents with potential inter-human interactions. Finally, we plan to extend the simulation environment by introducing static and dynamic obstacles to better reflect the complexity of real-world human-robot interaction scenarios. These elements will constrained agent motion adding more variability and unpredictability to the environment.

## Acknowledgements

This work was supported by JST Moonshot R&D (grant JPMJMS2011) and the European project TORNADO (HORIZON-CL4-2024-DIGITAL-EMERGING-01-101189557).

## References

1. Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726–1743, 2013.
2. Rudolph Triebel, Kai Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, Daniel Cremers, Vanessa Evers, Michelangelo Fiore, et al. Spencer: A socially aware service robot for passenger guidance and help in busy airports. *The International Journal of Robotics Research*, 35(14):1587–1607, 2016.
3. David Feil-Seifer and Maja J Matarić. Defining socially assistive robotics. In *9th International Conference on Rehabilitation Robotics*, pages 465–468. IEEE, 2005.
4. Phani Teja Singamaneni, Pilar Bachiller-Burgos, Luis J Manso, Anaís Garrell, Alberto Sanfeliu, Anne Spalanzani, and Rachid Alami. A survey on socially aware robot navigation: Taxonomy and future challenges. *The International Journal of Robotics Research*, 43(10):1533–1572, 2024.
5. Guy Hoffman. Evaluating fluency in human-robot collaboration. *IEEE Transactions on Human-Machine Systems*, 49(3):209–218, 2019.
6. Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. In *Robotics: Science and Systems*, 2013.
7. Chien-Ming Huang and Bilge Mutlu. Anticipatory robot control for efficient human-robot collaboration. In *11th ACM/IEEE International Conference on Human-Robot Interaction*, pages 83–90. IEEE, 2016.
8. Jim Mainprice and Dmitry Berenson. Goal set inference from human demonstrations. *The International Journal of Robotics Research*, 35(9):1055–1075, 2016.

9. Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016.
10. Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You’ll never walk alone: Modeling social behavior for multi-target tracking. In *IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009.
11. Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.
12. Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Darius M Gavrilu, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020.
13. Yuhang Che, Ahmed M Allam, Masayoshi Okada, and Zsolt Kira. Efficient and trustworthy social navigation via explicit and implicit robot–human communication. *IEEE Robotics and Automation Letters*, 5(2):2675–2682, 2020.
14. Weitian Wang, Rui Li, Yi Chen, and Yunyi Jia. Human intention prediction in human-robot collaborative tasks. In *Companion of the 2018 ACM/IEEE international conference on human-robot interaction*, pages 279–280, 2018.
15. Eike Rehder, Florian Wirth, Martin Lauer, and Christoph Stiller. Pedestrian prediction by planning using deep neural networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5903–5908. IEEE, 2018.
16. Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.
17. Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.
18. Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction. In *2021 International Conference on 3D Vision (3DV)*, pages 565–574. IEEE, 2021.
19. Yu Zhao, Rennong Yang, Guillaume Chevalier, Ximeng Xu, and Zhenxing Zhang. Deep residual bidir-lstm for human activity recognition using wearable sensors. *Mathematical problems in engineering*, (1):7316954, 2018.
20. Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.
21. Gerard Gómez-Izquierdo, Javier Laplaza, Alberto Sanfeliu, and Anaís Garrell. Enhancing context-aware human motion prediction for efficient robot handovers. *arXiv preprint arXiv:2503.00576*, 2025.
22. Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 37(1):50–61, 2020.
23. Guy Hoffman and Cynthia Breazeal. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, pages 1–8, 2007.
24. Dan Jia, Alexander Hermans, and Bastian Leibe. Dr-spaam: A spatial-attention and auto-regressive model for person detection in 2d range data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10270–10277. IEEE, 2020.