

Robots That Can See: Leveraging Human Pose for Trajectory Prediction

Tim Salzmann^{1,2}, Hao-Tien Lewis Chiang¹, Markus Ryll², Dorsa Sadigh^{1,3}, Carolina Parada¹, and Alex Bewley¹

Abstract—Anticipating the motion of all humans in dynamic environments such as homes and offices is critical to enable safe and effective robot navigation. Such spaces remain challenging as humans do not follow strict rules of motion and there are often multiple occluded entry points such as corners and doors that create opportunities for sudden encounters. In this work, we present a Transformer based architecture to predict human future trajectories in human-centric environments from input features including human positions, head orientations, and 3D skeletal keypoints from onboard in-the-wild sensory information. The resulting model captures the inherent uncertainty for future human trajectory prediction and achieves state-of-the-art performance on common prediction benchmarks and a human tracking dataset captured from a mobile robot adapted for the prediction task. Furthermore, we identify new agents with limited historical data as a major contributor to error and demonstrate the complementary nature of 3D skeletal poses in reducing prediction error in such challenging scenarios.

Project page: <https://human-scene-transformer.github.io/>

Index Terms—Autonomous Vehicle Navigation; Deep Learning Methods; Human-Aware Motion Planning

I. INTRODUCTION

THE presence of robots sharing the environment with humans has led to a need for effective methods to understand the human’s intention in order to avoid collisions and ensure smooth interactions between humans and robots. A series of steps are required for a robot to successfully navigate around humans in dynamic scenes, namely perception, prediction and planning. Perception is responsible for detecting the presence of humans and extracting other features of the environment around the robot. Prediction aims to model how humans will move into the future, and finally planning future actions towards a goal while avoiding collisions. In this work we focus on the prediction step where the inputs include perceived visual features and the outputs are predicted trajectory distributions for motion planning.

While the trajectory prediction topic has been extensively studied in the context of autonomous driving [1]–[4], predicting human trajectories in other environments where service robots could have a profound impact, such as offices, homes, hospitals, and elderly care facilities, have received less attention. In contrast to street scenes, the main dynamic agent in



Fig. 1: A service robot navigating a busy office space. To do so it anticipates human motion using human-position and visual features such as head orientation or 3D skeletal keypoints.

service robotics environments are humans carrying out a wider range of tasks that require diverse motions in unstructured environments, compared to the setting of driving which has significant structure in place such as staying within lanes or following the traffic rules. Additionally, the spatial environment is generally smaller with a higher degree of perceptual obstructions (e.g. blind corners or internal walls) resulting in a closer proximity upon first observation due to occluded entry-points. Our goal is to enable more natural, safe, smooth, and predictable navigation by anticipating where humans will be moving in the near future using the robot’s onboard sensors. Several existing methods try to generalize vehicle trajectory prediction ideas to human trajectory prediction by representing humans as 2D bounding boxes [5]–[9]. While bounding boxes can be sufficient for predicting vehicle trajectories in outdoor environments [1], [3], [4], [10], the reduction of human agents to bounding boxes neglects the plethora of perceptual information, present in a human-centric scene – scenes with one or multiple humans sharing limited space with the robot — such as a robot navigating in a hallway of a busy office as in Fig. 1. In such settings, humans tend to take advantage of information beyond each other’s position: There is substantial information about people’s intent when *they turn their head* or *look at where to walk next* — humans predict and anticipate each other’s intentions through vision-based features: pose, gaze, and gestures. These visual cues are also apparent for a human viewing a static image absent of temporal cues. Similarly, we will demonstrate that a robot navigating in close proximity to humans as in Fig. 1 enables a more detailed model of the human beyond simple bounding boxes and leads to more accurate prediction of human trajectories. Specifically, we posit the research question: “*Can information from human visual features lead to improved prediction accuracy?*”

Manuscript received: May 2nd, 2023; Revised: July 21st, 2023; Accepted: August 21st, 2023.

This paper was recommended for publication by Gentiane Venture upon evaluation of the Associate Editor and Reviewers’ comments.

¹Tim Salzmann, Hao-Tien Lewis Chiang, Dorsa Sadigh, Carolina Parada, and Alex Bewley are with Google DeepMind {tsal, lewispro, dorsas, carolinap, bewley}@google.com.

²Tim Salzmann and Markus Ryll are with Technical University Munich {tim.salzmann, markus.ryll}@tum.de.

³Dorsa Sadigh is with Stanford University dorsa@cs.stanford.edu

Digital Object Identifier (DOI): see top of this page.

We present the Human Scene Transformer (HST) which leverages different feature streams: Historic positions of each human, vision-based features such as skeletal keypoints (see Figure 1, joints of the human skeleton) or head orientation when available. We specifically focus on demonstrating the usefulness of noisy in-the-wild human skeletal information from a 3D human pose estimator.

As such our contribution is threefold:

- To the best of our knowledge, we are the first to adapt the trajectory prediction task to the domain of human-centric navigation and demonstrate that 3D vision-based features improve prediction performance in a service robot context notwithstanding imperfect in-the-wild data (without a dedicated motion capture system). Specifically in the regime of limited historical data, which is particularly under-explored, but a common scenario in indoor robot navigation our key idea is to use 3D vision features as complimentary predictive cues.
- We present a prediction architecture, which flexibly processes and includes detailed vision-based human features such as skeletal keypoints and head orientation. To target crowded human-centric environments, we define HST within a system of components from fields of Computer Vision, Machine Learning, and Autonomous Driving to make use of real-world sensor data instead of relying on ground truth labels. We demonstrate HST’s capability to consistently model interactions which is critical in human-centric environments.
- We highlight a gap in prior work by showing the limitations of existing datasets for human trajectory prediction in indoor navigation. We propose an adaptation of existing datasets that can enable this new way of future trajectory prediction. Using this adaptation, we demonstrate the feasibility of our approach in human-centric environments. Simultaneously, we display state-of-the-art performance on a common outdoor pedestrian prediction dataset.

II. RELATED WORK

Predicting the future trajectory of humans is a challenging task where prior work has considered various motion models, scene context, and social interaction. We will revisit three research fields which influence our targeted domain of service robots and the use of vision-based human skeletal (human pose) features. We will first outline current research in *Trajectory Prediction*, where the center position of an agent (which can be a human but also a human driven vehicle) is forecasted over time. Subsequently, we will introduce work in the field of *Human Pose Prediction*, which uses a skeletal representation of the human and predicts this pose over time. Finally, we outline approaches in *Pose Estimation*, where the estimated human pose is directly used in trajectory prediction in application domains outside of human-centric navigation.

Trajectory Prediction. From the early work of Pellegrini *et al.* [5] for short-term future locations of humans for next frame data association to the recent longer term multi-second prediction methods that are often used in autonomous driving [1]–[4], [11], trajectory prediction research has played an important role in improving down-stream robotic tasks. Prior approaches consider scene context [4], motion dynamics [3], [4], and the interaction between agents [3]. Salzmann *et al.* [4] combine historic agent positions with scene and dynamics

constraints to make informed predictions. In an autonomous driving context, extracting additional visual information about the human actor (driver, cyclist, pedestrian) is challenging due to occlusion (driver only partially visible in the car) or distance. Therefore, the information representing each agent is reduced to a position estimate per observation timestep [1]–[4]. Unlike the large prediction range required for self-driving [12], we focus on service-robot environments where people are generally close enough to the robot to obtain a richer visual representation of the human. As such, our work can benefit from recent approaches that fuse LiDAR information adapting the human pose estimation problem to a robotic sensor suit [13]. While prior works in trajectory prediction rely on Generative Adversarial Networks (GANs) [8], [14] or Conditional Variational Autoencoders (CVAEs) [4], [7], [10], [15], [16], this work follows the recent trend towards Transformer architectures as they naturally lend themselves to the set-to-set prediction problems such as multi-agent trajectory prediction and are invariant to a varying number of agents. Specifically, we leverage the fundamental idea of a Transformer based prediction framework [1]–[3] inspired by Ngiam *et al.* [3]. Their Transformer architecture is used for vehicle trajectory prediction in autonomous driving applications and captures joint interactions between vehicles.

Pose Prediction. Another related area is human pose forecasting in 3D [17]–[21]. Corona *et al.* use scene context in the form of an influence graph to refine a future trajectory of 3D human poses for a single subject in a motion capture environment [17]. For multi-person pose prediction, [22] extends DeTR [23] to predict multi-person 2D poses from a single image. Towards social robot navigation, Narayanan *et al.* use a sequence of human poses, also known as gait, to classify a person’s emotional state for setting appropriate proximal distance constraints [24]. However, these approaches commonly consider a single human motion relying on ground truth pose information from a motion capture system, while we target multi-human in-the-wild scenarios which are not limited to spaces with a motion capture system.

Pose Estimation in Trajectory Prediction. There have been prior efforts to combine pose estimation with trajectory prediction, i.e., informing forecasted trajectories by incorporating historic pose information. However, these works are either limited to prediction in 2D image space [11], [25], [26] or operating on motion capture datasets which do not exhibit diverse positional movement of the human [17], [27]. Yagi *et al.* [25] showed that augmenting a convolutional auto-encoder style model with scale and pose encoders reduces prediction error compared to position only; however, their approach is applied to first-person video using 2D pose detection and limiting the prediction to the 2D image space. Similarly, Chen *et al.* [26] use a convolutional and recurrent architecture to segment an image into heterogeneous traffic objects and body parts, before using a Transformer decoder to attend to feature maps and extract objects leading to improvements in 2D image space prediction. When considering predicting multiple poses, Adeli *et al.* [28] use a form of graph attention to capture dependencies between interacting agents but this work is, again, limited to 2D first-person videos. Other works have explored attention mechanisms between multiple human features in the image-view [11]. However, for robotic navigation it is desired

to obtain predictions for agents across multiple sensors and ideally in a 3D or bird's-eye metric space. In this work, we follow these requirements by solely relying on onboard sensor information of a robotic platform and predict in the metric global frame rather than in image space.

III. HUMAN SCENE TRANSFORMER

Our proposed method Human Scene Transformer (HST) follows the concept of masked sequence to sequence prediction using an architecture with Transformer blocks (see Figure 3 - top right). This approach has shown promising vehicle prediction results in the autonomous driving domain [3]. HST introduces multiple important ideas extending the general Transformer architecture which makes it suitable for trajectory prediction in human-centric environments. These include the utilization of vision-based human features (Section III-A), a feature attention mechanism to merge multiple potentially incomplete features (Section III-B - Input Embedding), an improved attention mechanism facilitating a more complete information flow (Section III-B - Full Self-Attention), and a self-alignment layer which elegantly solves the problem of discriminating between multiple masked agent timesteps while keeping permutation equivariance (Section III-B - Agent Self-Alignment). Notably, implementing an attention based architecture end-to-end, the model is agnostic to the number of humans per frame. This means the model can dynamically handle a varying number of humans in different timesteps during inference. The maximum number of jointly (single forward pass) predicted humans is only limited by available memory.

A. Model Inputs: Incorporating Vision-based Features

The robot's observations for the last $H + 1$ timesteps are processed as agent features and scene context (Figure 3 blue box). The scene context can be an occupancy grid or a raw point cloud at the current timestep, containing information common to all nearby agents (e.g. static obstacles). Agent features include the centroid position and vision-based features: skeletal keypoints, and head orientation for each agent.

To extract these vision-based features from the raw data, image patches for all agents are first obtained by projecting their detected 3D bounding boxes into the 360 degree image using ex- and intrinsic camera calibrations (see Figure 2-a). To extract skeletal key points from these patches, one could choose from a plethora of off-the-shelf skeletal keypoints extractor for images [29]–[33]. However, these extractors commonly output keypoints in a 2D image coordinate frame. To produce 3D keypoints, we follow the work of Grishchenko *et al.* [34] to estimate 3D keypoints from images using a pre-trained model: As existing datasets commonly only include 2D keypoint annotations, the 3D label required for supervised pre-training is generated by fitting a parametric human shape model to available 2D keypoints solving the following optimization problem:

$$\operatorname{argmin}_{\mathbf{k}} \left(\|\mathbf{r}(\mathbf{k}) - \hat{\mathbf{k}}_2\|_2 + \lambda p(\mathbf{k}) \right), \quad (1)$$

where \mathbf{k} are the 3D skeletal keypoints, $\hat{\mathbf{k}}_2$ is the 2D keypoints label, $\mathbf{r} : \mathbb{R}^{33 \times 3} \rightarrow \mathbb{R}^{33 \times 2}$ is the re-projection function projecting 3D points into the 2D image space using the

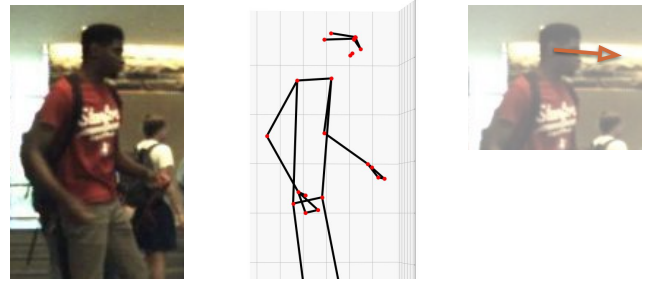


Fig. 2: **Process of estimating three dimensional vision-based features of the human.** (Left) Image of the human cropped based on the bounding box detection. Lighting conditions can be sub-optimal in a human-centric environment. (Middle) Inferred three dimensional pose from the trained pose estimator. (Right) Head orientation post-processed from the pose keypoints.

camera calibrations. Many representations (from < 20 [35] to > 500 [36] keypoints) are present in the literature; we settled for a 33 keypoints skeleton representation [34], [37] as it presents itself as a minimal representation capturing *both* head information and limb articulation. The learned prior distribution over human pose configuration $p(\mathbf{k})$ penalizes infeasible poses which can arise in optimization for the underdetermined 3D-2D-projection problem, as multiple 3D poses can result in equivalent 2D projections. A infeasible pose would be a configuration of human joints which is physically infeasible for a human to achieve (e.g. head rotated full 180°). The prior distribution is learned by fitting a variational autoencoder to a dataset of feasible 3D human poses. The 3D keypoints in the camera space are transformed to the global coordinate frame using extrinsic camera calibration. Given the skeletal keypoints \mathbf{k} from optimizing Equation (1), we can easily extract the head orientation as the vector from in-between ears to in-between eyes. Note that vision-based agent features may not be available for all agents at all timesteps. This can be due to the agent being too far away to reliably extract keypoints (e.g., the picture being too small). We will show in Section III-B - Input Embedding how we can leverage inherent properties of our HST Transformer architecture to deal with these situations.

B. Model Architecture

Figure 3 outlines the HST architecture. We will explain the individual components in this section by first introducing the Transformer layer as core concept and subsequently following the data flow depicted in Figure 3.

Transformer Layer. The primary building block of the model's architecture is the Transformer layer (shown in Figure 3 top right), which itself is comprised of a Multi-Head Attention layer [38] and multiple dense and normalization layers. The Transformer layer receives three tensors as input: Query (Q), Key (K), and Value (V). However, a single tensor may be used for multiple of the inputs. Consequently, we define a self-attention (SA) operation as a Transformer Layer where inputs Q, K, and V are the same tensor: The tensor attends to itself and conveys its information along one or more dimensions. Similarly, we define cross-attention (XA) as a Transformer Layer where the Q input is distinct from the K/V inputs. Intuitively the query attends to additional information from a different tensor as means of merging multiple streams of information. For a comprehensive explanation on the Attention mechanism and its inputs we refer the reader to Vaswani *et al.* [38].

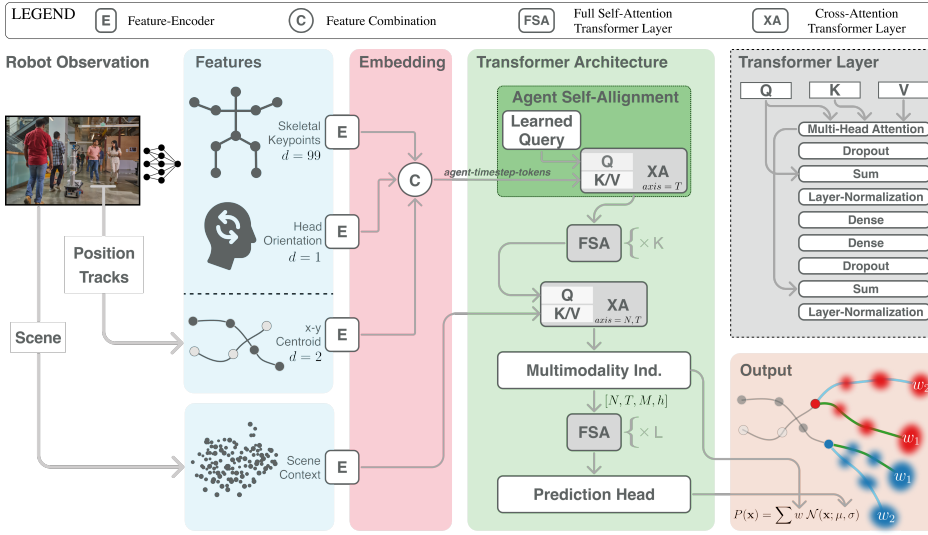


Fig. 3: **Overview of the HST architecture.** From the robot’s sensors we extract the scene context, the historic position tracks of each agent, and vision based skeletal keypoints/head orientation when feasible. All features are encoded individually before the agent features are combined via cross-attention (XA) using a learned query tensor. The resulting agent-timestep-tokens are passed to our Agent Self-Alignment layer which enables the use of subsequent full self-attention (FSA) layers. Embedded scene context is attended to via cross-attention (XA). After multimodality is induced and further FSA layers the model outputs the parameters of a Normal distribution for each agent at each prediction timestep. We can represent the full output structure as a Gaussian Mixture Model (formula in bottom right) over all possible futures where the mixture coefficients w come from the Multimodality Induction. Both cross-attention (XA) and full self-attention layers use the Transformer layer (top right) with different input configurations for Query (Q), Key (K), and Value (V).

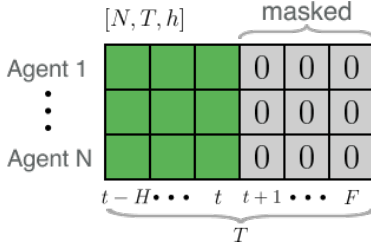


Fig. 4: Structure and masking of embedded tensors. Future agent-timestep-tokens are masked and subsequently filled by the Transformer structure, iteratively with updated latent representations and finally with position distribution information on the output level.

Input Embedding. The input agent features (blue) are tensors of shape $[N, T, d]$, where $d = 2$ for the x-y centroid position, $d = 99$ for the x-y-z position of 33 skeletal keypoints, and $d = 1$ for the head orientation. These tensors contain information of all N nearby agents for all $H + 1$ historic and current input timesteps. If an agent’s feature is not observed at specific timesteps, we mask those timesteps with 0. As depicted in Figure 4, we also mask all future F timesteps for all agents by setting their feature value to 0, thus making only historical and current information available to the model. This masking approach is a well known technique in missing-data problems such as future prediction using Transformer based architectures [1], [3], [38]. Masking exploits the inductive bias inherent in the prediction problem, which allows for the filling of the missing information using available context in the vicinity of the gaps. As such, our approach allows for missing keypoints in frames due to bad lighting or other influences as the Transformer effectively “fills” in for the missing information.

The agent features are encoded independently and are combined by a learned attention query. This masked attention mechanism offers scalability to systems that have a large number of features with limited availability. The combined agent features result in a latent tensor for each agent and timestep of shape $[N, T, h]$ where $T = H + 1 + F$ and h is the size of the token dimension. For simplicity we will refer to such tensors of latent representations throughout the network as *agent-timestep-tokens*.

Full Self-Attention Via Agent Self-Alignment. In contrast to previous methods [3] which alternately attends to agents and time dimensions separately via factorized attention, we propose a *full self-attention* (FSA) operation where each agent-timestep-token attends to all agent-timestep-tokens along *both*

agent and time dimensions. This provides a more direct path of information flow. For example in social interactions, a change in action such as adjustment in walking direction does not have an immediate influence on other humans in proximity but rather influences their future. Following this illustration, an agent at a given timestep in our Transformer architecture should be able to attend not just to other agents at the current timestep (factorized attention) but to *all agents at all timesteps* (full self-attention).

Naïvely using full-self-attention can lead to sub-optimal outcomes. Since all future agent-timestep-tokens are masked out, two agents with the same masked future agent-timestep-tokens will also have the same input (Query) representation to the Transformer layer (Figure 3 top right). This prevents the model from associating future timesteps of an agent with the agent’s history, since all future agents’ timesteps “look” the same (masked). The problem could be addressed by enforcing an innate order on perceived agents, where all agents are enumerated. This, however, would eliminate the permutation invariant set-to-set prediction capabilities [39]; one of the core strengths of Transformers: An agent’s future would be predicted differently based on its enumeration embedding with the same historic features.

Instead, we solve this problem and achieve full self-attention via a simple approach that we refer to as *agent self-alignment*. The agent-timestep-tokens resulting from the feature combination are cross-attended with a learned query tensor *only* in the time dimension. This query, a weight matrix jointly optimized with all other network weights during training, learns to propagate available historic information for each agent to future timesteps, enabling the model to align future masked timesteps of an agent with historic ones during full self-attention without an explicit enumeration embedding. After this process, which is visualized in the dark green box in Figure 3, the previously masked future agent-timestep-tokens hold information from the respective agent’s history; differentiating them from another. As such, the agent self-alignment mechanism preserves agents’ permutation equivariance and enables full self-attention without restricting information flow along matching timesteps [3] or utilizing special attention matrices which explicitly separates agents [1]. The output agent-timestep-tokens of the agent self-alignment then passes

through K transformer layers with full self-attention across agent and time dimensions before cross-attending to the encoded scene features.

Multimodality Induction. Our architecture can predict multiple consistent futures (modes) for a scene. To do so, the Multimodality Induction module repeats the agent-timestep-tokens by the number of future modes (M), resulting in a tensor of shape $[N, T, M, h]$. To discriminate between modes it is combined with a learned *mode-identifier* tensor of shape $[1, 1, M, h]$. Each future’s logit probability w_m ; $m \in 1, \dots, M$ is also inferred here by having the *mode-identifier* attend to the repeated input; resulting in shape $[1, 1, M, h]$ which is subsequently reduced by a MLP to output w_m as $[1, 1, M, 1]$.

Prediction Head. The agent-timestep-tokens updated with the learned mode-identifier go through L Transformer layers, again with full self-attention, before predicting per mode parameters μ, σ using a dense layer as *prediction head*.

C. Producing Multimodal Trajectory Distributions

Combining μ and σ with the mode likelihoods w_m coming from the Multimodality Induction, the Human Scene Transformer models the distribution of the i -th agent’s centroid position at each timestep t with a 2D Gaussian Mixture Model (GMM):

$$P_{\theta}^i(\mathbf{x}_t | O(t), \dots, O(t-H)) = \sum_{m=1}^M w_m \mathcal{N}(\mathbf{x}; \sigma_{m,i,t}, \mu_{m,i,t}), \quad (2)$$

where m is the m -th future mode. Here, we represent the position of an agent at a specific timestep by a GMM with mixture weights w equal to the probability distribution of future modes.

We adopt a joint future loss function, that is, the cumulative negative log-likelihood of the Gaussian mode (m^*) with the smallest mean negative log-likelihood:

$$\mathcal{L}_{\min\text{NLL}} = \sum_{i,t} -\log(\mathcal{N}(\mathbf{x}_{i,t}^*; \sigma_{m^*,i,t}, \mu_{m^*,i,t})), \quad (3)$$

where

$$m^* = \operatorname{argmin}_m \left(\sum_{i,t} -\log(\mathcal{N}(\mathbf{x}_{i,t}^*; \sigma_{m,i,t}, \mu_{m,i,t})) \right), \quad (4)$$

and $\mathbf{x}_{i,t}^*$ is the ground truth agent position. The resulting prediction represents M possible future realizations of all agents at once in a consistent manner, where the mode mixture weights w are shared by all agents in the scene. The most likely future mode during inference is given as

$$m^+ = \operatorname{argmax}_m (w_m). \quad (5)$$

IV. EXPERIMENTS

We structure our experiments to support our contributions: First, we qualitatively and quantitatively demonstrate how our architecture provides accurate predictions for the human-centric service robot domain. We especially demonstrate how HST can leverage and model interactions between humans consistently over multiple possible futures. Further, we show that our approach is cross-domain compatible with unconstrained outdoor pedestrian prediction, where data from a

TABLE I: A summary of different prediction datasets indicating our desiderata addressed by each dataset. Parentheses indicate that a desiderata is only partially fulfilled.

	I Diverse	II Robot	III Unscripted	IV k	V Large
ETH [5] & UCY [6]	✗	✗	✓	(✗)	✗
AD [40]–[43]	✗	✓	✓	(✗)	✓
Motion Capture [44], [45]	✗	✗	✗	(✓)	(✗)
3DPW [46]	✗	✗	✗	✓	✗
Adapted JRDB	✓	✓	✓	(✓)	✗

surveillance camera is used to predict pedestrians in different outdoor squares. Finally, we demonstrate how HST can leverage vision-based features in human-centric environments to improve prediction accuracy, specifically in short history situations where prediction errors are high.

Datasets. To effectively investigate the performance of HST in human-centric environments and the possible benefits that a detailed 3D skeletal representation of the human body can have, a dataset should (I) include a *diverse* range of indoor and outdoor environments, (II) capture humans’ movement from a *robot* viewpoint (III) in a natural *unscripted* environment, (IV) provide labels for the position and skeletal keypoints (k) of all agents at all timesteps, and (V) be sufficiently *large* to prevent over-fitting. The evaluation of different datasets for human trajectory prediction in Table I takes into account the satisfaction of these requirements. Many of existing datasets are collected from a single top-down camera in a limited number of environments, such as the ETH [5] and UCY [6] pedestrian datasets. Others are specific to the autonomous driving domain [40]–[43]. While none of these datasets provide labels for skeleton keypoints, other datasets such as H3.6.M [44], AMASS [45], and 3DPW [46] which are collected using a motion capture system or wearable IMU devices [46], do offer such labels. However, these datasets are limited to artificial environments and often feature stationary or scripted motions. Finally, while all these datasets provide labels for position, these labels are often hand labeled ground truth not representing noisy input data a robot would experience in the real-world during inference.

One dataset which is recorded in diverse human-centric environments using sensors (2 x 16 Channel LIDAR, 5 x Stereo RGB Cameras) on a mobile robotic platform is the JackRabbit Dataset and Benchmark (JRDB) [47]. However, JRDB was created as a detection and tracking dataset rather than a prediction dataset. To make the data suitable for a prediction task, we first extract the robot motion from the raw sensor data to account for the robot’s motion. Tracks are generated for both train and test split using the JRMOT [48] detector and tracker. The ground truth labeled bounding-boxes on the train set were disregarded as they were exposed to filtering during the labeling process to the point where the smoothness eases the prediction task. We were able to increase the number of human tracks for training by associating the JRMOT detections to ground truth track labels via Hungarian matching, while on the test split we solely use JRMOT predictions.

Due to factors such as distance, lighting and occlusion the pre-trained 3D pose estimator model (Section III) is not guaranteed to produce keypoints for all agents at all timesteps. We observed human keypoints information in $\sim 50\%$ of all timesteps for all agents in a distance of up to 7 meters from

the robot. The available data is split using 50% of each scene as training data, 20% as validation data and 30% as test data. We sub-sample the dataset from 15Hz to 3Hz keeping all of the intermediate samples and therefore increasing the number of datapoints by a factor of five.

In addition, we also compare our model to the ETH [5] and UCY [6] datasets. These are standard benchmarks for pedestrian trajectory prediction and enable a fair comparison of our architecture against other methods.

Metrics. In consistency with prior work [1], [4], [7], [8], prediction quality is evaluated using the following metrics:

- 1) Minimum Average Displacement Error (*minADE*): Minimum Mean l_2 distance between the ground truth and all M future mode trajectories.
- 2) Minimum Final Displacement Error (*minFDE*): Minimum l_2 distance between the final positions of the ground truth and all M future mode trajectories.
- 3) Maximum Likelihood Average Displacement Error (*MLADE*): Mean l_2 distance between the ground truth and the most likely mode trajectory.
- 4) Negative Log Likelihood (*NLL*) of the ground truth under the full parametric output distribution.

Lower is better for all metrics.

Baselines. We re-implement the autonomous driving Scene Transformer architecture [3], where we match the number of Transformer layers to our architecture. We further compare to trajectory prediction architectures Trajectron++ [4], AgentFormer [1], SoPhie [8], and PECNet [7].

Evaluation Protocol. For the JRDB prediction dataset we report *minADE*, *MLADE*, and *NLL* for 128k scene snippets from the test split including partially occluded agents. We use up to 2s of history as input and predict the next 4s future of the scene. Note that if the number of agents exceeds $N_{\max} = 16$, we randomly select one agent and only consider and predict it and the $N_{\max} - 1$ closest agents. On the ETH and UCY dataset we follow the standard procedure to train in a leave-one-out fashion and evaluate *minADE* and *minFDE* on 20 trajectories over a prediction horizon of 12 timesteps (4.8s) using 8 historic timesteps as input. We match the evaluation protocol of AgentFormer [1] by setting the number of modes to $M = 20$.

A. Trajectory Prediction in Human-centric Environments

In Table II and Figure 5 we show quantitative and qualitative results of HST’s predictions in the human-centric environment. We show that in crowded human-centric environments the

TABLE II: **Comparison against Scene Transformer on JRDB prediction dataset.** HST outperforms the original Scene Transformer on all metrics. Ablation shows that the interaction attention to other agents improves performance by comparing to a model predicting a single human at time. We also show the positive impact of Full Self-Attention.

Model Configuration		<i>minADE</i>	<i>MLADE</i>	<i>NLL</i>
Scene Transformer [3]		0.53	0.86	0.25
Full Self-Attention				
HST	✗	0.57	0.93	0.89
HST	✓	0.50	0.84	-0.02
HST	✓	0.48	0.80	-0.13

influence of interaction between humans has large benefits on the prediction accuracy of each individual. To show this, we compare against a version of our model which is trained to predict a single human at a time ignoring interactions with other agents. Subsequently, adding our full self-attention via self-alignment mechanism additionally increases the model’s ability to capture interactions across time, leading to improvements across all metrics. The capability to account for interactions between humans is qualitatively demonstrated in Figure 5 where we show multiple predicted futures for a scene of two interacting humans. The two humans approach each other head on. The possible interactions to avoid collisions are modeled *consistently* within each future.

B. Vision-based Features

In this section, we consider the adversarial setting, where the robot encounters a human unexpectedly, i.e., the robot observes a new human with little historical observations. Prediction architectures solely relying on historic position information struggle in scenarios where no or only a limited amount of history of the human position is available to the model. Specifically, at the first instance of human detection, the experimental error is 200% higher compared to full historic information over 2s. Given the specifics of our targeted human-centric environment, where we are mostly interested in humans close to the robot, we are likely able to extract vision-based features for the human in addition to the position. Specifically, we revisit our research question: “Can information from human visual features lead to improved prediction accuracy?”

Before answering this question quantitatively we show a clarifying visual example in Figure 6 where a human just entered the scene through a door and is first detected. When solely relying on historic position information the most likely prediction by the model is a stationary agent. However, when we employ the pre-trained skeleton keypoints estimator to provide pose keypoints as additional input to our model the

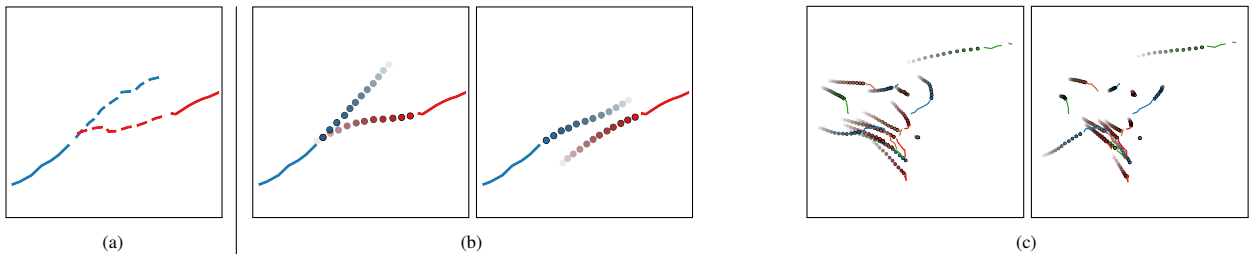
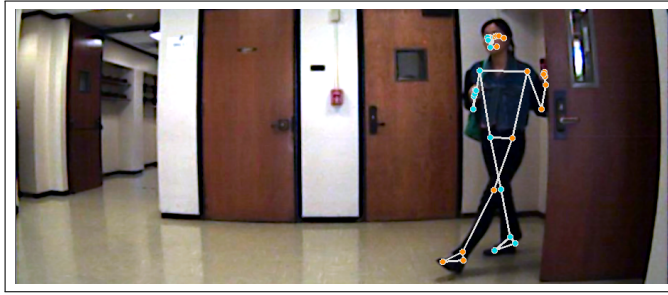
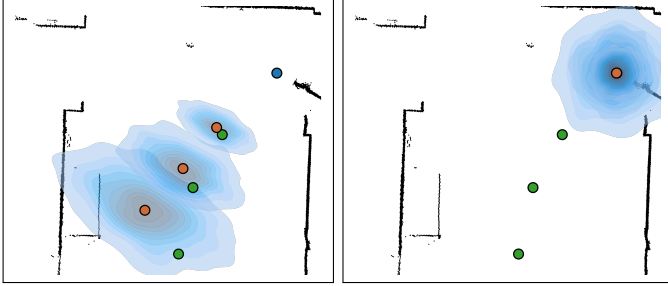


Fig. 5: **Consistently modeled interactions in different predicted futures for a single scene in the x-y-plane.** Two humans approaching each other head on. (a) History (solid) and ground truth future (dashed) of both humans. (b) Two of the M predicted futures (dots - increasing transparency with time) of the scene by HST. Within each mode the influence and reaction of both agents is consistent and reasonable. The humans’ futures are predicted without collisions giving each other space to navigate within the specific predicted future mode of the scene. (c) Two predicted futures of a crowded scene.



(a) First detection of person entering the scene.



(b) Prediction with keypoints.

(c) Prediction without keypoints.

Fig. 6: A visualization of the predicted trajectory distributions for a new human agent entering the scene through the door on the right as viewed in (a). For *both* (b) and (c) the HST model does not have any historic information here and only has access to the current frame. The plot of future trajectory distributions in (b) and (c) show the effect of using and not using skeletal keypoints (respectively) as input in that single frame. Without pose keypoints the HST model predicts the agent to be most-likely stationary while, with keypoints as input, it can reason that the human is moving and correctly anticipates the direction. Blue dot is the detected human at the initial frame, orange dots are our most likely mode predictions with corresponding distribution shown with blue shading, and green dots are the ground truth human future (actually executed trajectory by the human).

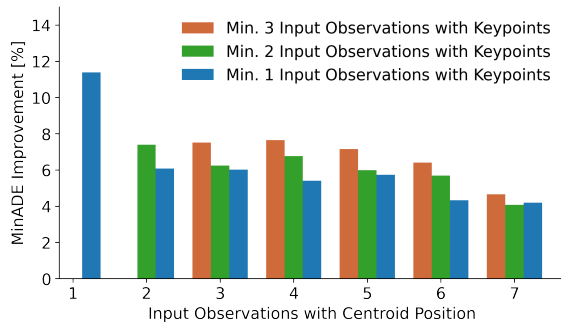


Fig. 7: Impact of vision-based features conditioned on different number of consecutive non occluded input timesteps.

model correctly realizes if the human is in a walking motion and how the human is oriented, accurately predicting the most likely future trajectory.

Quantitatively, during evaluation, when keypoints are available on the first detection we observe a substantial prediction improvement of up to 11% (Figure 7). When additional timesteps with position information are available the improvement using keypoints vs not using keypoints averages between 5% and 10%. The relative improvement generally increases with the number of timesteps with keypoints in the history and decreases with the number of historic position information.

Finally, we want to provide an outlook of how much vision-based features can improve prediction performance if available for all agents at all timesteps. We therefore enforce feature parity between position and skeletal keypoints features by

TABLE III: Vision-based features relatively improve prediction across all metrics on a prediction horizon of 4 s. To create feature parity between position and keypoints features we ignore all position history without detected skeletal keypoints during training and evaluation.

	<i>minADE</i>		<i>NLL</i>		<i>minADE @ 2 s</i>		<i>minADE @ 4 s</i>	
Baseline	0.56	0%	1.04	0%	0.46	0%	0.87	0%
Head Orient.	0.53	-5%	0.89	-14%	0.42	-9%	0.79	-9%
Keypoints	0.51	-9%	0.85	-18%	0.40	-13%	0.77	-11%

disregarding position information without associated keypoints (Table III). We find that a relative improvement of around 10% is achievable using our in-the-wild vision-based features. The baseline is naturally worse than in Table II as we partially disregard historic position information.

C. Pedestrian Dataset

In addition to showing HST’s capabilities in a robotics specific environment we will further validate our architecture against a range of state-of-the-art prediction methods. For this we use ETH/UCY which is prolific in the trajectory prediction community while also being the dataset which we think is the closest to the human-centric environments that we would like to explore: On the ETH and UCY dataset, we either improve current state-of-the-art methods or we are on par with them on 4 out of the 5 scenes.

V. DISCUSSION

Simply representing a human as its spatial position, as commonly done in autonomous driving environments, does deliver a baseline prediction performance in human-centric service robot environments. However, it suffers in challenging settings where the history of a human is limited. Specifically in these situations we demonstrate how the Human Scene Transformer can leverage vision-based features to improve prediction accuracy (Figure 6). Beyond scenarios such as when robot and humans encounter each other in blind corners, general improvement trends using in-the-wild skeletal pose detections were also observed with more observations as shown in Figure 7. Another intuitive assumption which we can support quantitatively in Table III is that the full skeletal keypoints (full human pose) hold additional information over just the head orientation (where is the human looking). This is expected as the head orientation can give away a possible direction for the trajectory, while the full keypoints can be more informative about the speed the human is approaching, e.g., running or slowly walking.

Figure 5 and Table II demonstrate HST’s capability to model consistent interactions between agents and use these influences to improve the overall prediction substantially. This is especially useful for the crowded spaces a service robot navigates and opens opportunities not just modeling human-to-human interactions but robot-human interactions.

Limitations. Exploring this new domain of human-centric environments we recognize that numerous limitations exist: We showed that an on par relationship between positions and keypoint features leads to relative improvements and expect the performance to be similarly correlated to the quality of the detected 3D keypoints. We therefore think that an 3D skeletal keypoints estimator, specifically designed for robotic applications, increasing both the number of successful detections and the quality of each detected skeleton would improve performance for the prediction task as well as other robot tasks

TABLE IV: Results on ETH and UCY dataset. Our method is the best on four of five subsets.

Method	ETH	Hotel	$\min\text{ADE}_{20} / \min\text{FDE}_{20}$ Univ	Zara1	Zara2	Average
SoPhie [8]	0.70 / 1.43	0.76 / 1.67	0.54 / 1.24	0.30 / 0.63	0.38 / 0.78	0.54 / 1.15
PECNet [7]	0.54 / 0.87	0.18 / 0.24	0.35 / 0.60	0.22 / 0.39	0.17 / 0.30	0.29 / 0.48
Trajectron++ [4]	0.43 / 0.86	0.12 / 0.19	0.22 / 0.44	0.17 / 0.32	0.11 / 0.25	0.21 / 0.41
AgentFormer [1] ¹	0.45 / 0.75	0.14 / 0.22	0.25 / 0.45	0.18 / 0.30	0.14 / 0.24	0.23 / 0.39
Scene Transformer [3]	0.50 / 0.76	0.14 / 0.20	0.29 / 0.42	0.22 / 0.36	0.16 / 0.27	0.31 / 0.40
HST	0.41 / 0.73	0.10 / 0.14	0.24 / 0.44	0.17 / 0.30	0.14 / 0.24	0.21 / 0.37

in close human contact (e.g. handover). Such an estimator could make use of the full sensor suite of a robotic platform, fusing camera and LiDAR information. We are happy to see first works [13] in this direction and hope that our findings encourage the research community to pursue this path within the domain of human-centric environments.

In Section III we presented a way to include scene context via a point cloud but were unable to see any predictive benefit from this information. We attribute this to three factors: the small size of the adapted JRDB dataset, the limited number of locations (29) in which the data was recorded, and the raw point lack semantic labels, e.g. a door is indistinguishable from a wall. These factors highlight the potential for further work towards better representations of point clouds.

Conclusion. This work introduced the task of human trajectory prediction into the domain of human-centric service robots. We demonstrated that the proximity of robot and humans in such environments can be leveraged to improve prediction performance by explicitly incorporating vision-based human features. We showed HST achieves strong prediction improvements using in-the-wild 3D pose representations for the critical situation of agents being first detected in close proximity to the robot. A quantitative ablation using paired position and vision-based features highlighted the influence of different visual features with skeletal keypoints providing the highest gains across all metrics. Our Transformer based architecture is flexible in accommodating different feature inputs while also achieving state-of-the-art results on a common pedestrian prediction dataset without visual features outside the domain of human-centric service robot environments. We think that the unstructured and uncompressed nature of environment point-clouds fits nicely with the permutation invariance property of Transformer architectures and are therefore excited to further explore this direction in the future. We hope that our findings will inspire further research in the human-centric domain and in developing improved methods for generating accurate 3D vision-based human representation for service robotics applications.

REFERENCES

- [1] Y. Yuan *et al.*, “Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting,” in *ICCV*, 2021.
- [2] N. Nayakanti *et al.*, “Wayformer: Motion forecasting via simple & efficient attention networks,” *arXiv:2207.05844*, 2022.
- [3] J. Ngiam *et al.*, “Scene Transformer: A unified architecture for predicting future trajectories of multiple agents,” in *ICLR*, 2022.
- [4] T. Salzmann *et al.*, “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data,” in *ECCV*, 2020.
- [5] S. Pellegrini *et al.*, “You’ll never walk alone: Modeling social behavior for multi-target tracking,” in *ICCV*, 2009.
- [6] A. Lerner *et al.*, “Crowds by example,” in *Computer graphics forum*, 2007.
- [7] K. Mangalam *et al.*, “It is not the journey but the destination: Endpoint conditioned trajectory prediction,” in *ECCV*, 2020.
- [8] A. Sadeghian *et al.*, “Sophie: An attentive gan for predicting paths compliant to social and physical constraints,” in *CVPR*, 2019.
- [9] K. Mangalam *et al.*, “From goals, waypoints & paths to long term human trajectory forecasting,” in *ICCV*, 2021.
- [10] B. Ivanovic *et al.*, “The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs,” in *ICCV*, 2019.
- [11] P. Czech *et al.*, “On-Board Pedestrian Trajectory Prediction Using Behavioral Features,” *arXiv:2210.11999*, 2022.
- [12] P. Sun *et al.*, “Rsn: Range sparse net for efficient, accurate lidar 3d object detection,” in *CVPR*, 2021.
- [13] A. Zanfir *et al.*, “HUM3DIL: Semi-supervised Multi-modal 3D Human Pose Estimation for Autonomous Driving,” *arXiv:2212.07729*, 2022.
- [14] A. Gupta *et al.*, “Social gan: Socially acceptable trajectories with generative adversarial networks,” in *CVPR*, 2018.
- [15] B. Ivanovic *et al.*, “Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach,” *RA-L*, 2020.
- [16] B. Ivanovic *et al.*, “Generative modeling of multimodal multi-human behavior,” in *IROS*, 2018.
- [17] E. Corona *et al.*, “Context-aware human motion prediction,” in *CVPR*, 2020.
- [18] Y. Yuan *et al.*, “Dlow: Diversifying latent flows for diverse human motion prediction,” in *ECCV*, 2020.
- [19] Y. Zhang *et al.*, “We are more than our joints: Predicting how 3d bodies move,” in *CVPR*, 2021.
- [20] W. Mao *et al.*, “History repeats itself: Human motion prediction via motion attention,” in *ECCV*, 2020.
- [21] T. Salzmann *et al.*, “Motron: Multimodal Probabilistic Human Motion Forecasting,” in *CVPR*, 2022.
- [22] E. Vendrow *et al.*, “SoMoFormer: Multi-Person Pose Forecasting with Transformers,” *arXiv:2208.14023*, 2022.
- [23] N. Carion *et al.*, “End-to-end object detection with transformers,” in *ECCV*, 2020.
- [24] V. Narayanan *et al.*, “Proxemo: Gait-based emotion learning and multi-view proxemic fusion for socially-aware robot navigation,” in *IROS*, 2020.
- [25] T. Yagi *et al.*, “Future person localization in first-person videos,” in *CVPR*, 2018.
- [26] K. Chen *et al.*, “Pedestrian Trajectory Prediction in Heterogeneous Traffic Using Pose Keypoints-Based Convolutional Encoder-Decoder Network,” *TCSVT*, 2020.
- [27] M. Mahdavian *et al.*, “STPOTR: Simultaneous Human Trajectory and Pose Prediction Using a Non-Autoregressive Transformer for Robot Following Ahead,” *arXiv:2209.07600*, 2022.
- [28] V. Adeli *et al.*, “Tripod: Human trajectory and pose dynamics forecasting in the wild,” in *ICCV*, 2021.
- [29] Z. Cao *et al.*, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” *PAMI*, 2019.
- [30] H.-S. Fang *et al.*, “Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time,” *PAMI*, 2022.
- [31] D. Maji *et al.*, “YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss,” in *CVPR*, 2022.
- [32] tensorflow.org, *MoveNet: Ultra fast and accurate pose detection model*. 2022.
- [33] tensorflow.org, *Real-time human pose estimation in the browser with tensorflow.js*, 2018.
- [34] I. Grishchenko *et al.*, “BlazePose GHUM Holistic: Real-time 3D Human Landmarks and Pose Estimation,” *Computer Vision for AR/VR*, 2022.
- [35] T.-Y. Lin *et al.*, “Microsoft coco: Common objects in context,” in *Computer Vision—ECCV 2014: 13th European Conf., Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, 2014.
- [36] I. Grishchenko *et al.*, *MediaPipe Holistic — Simultaneous Face, Hand and Pose Prediction, on Device*, 2018.
- [37] H. Xu *et al.*, “Ghum & ghumi: Generative 3d human shape and articulated pose models,” in *CVPR*, 2020.
- [38] A. Vaswani *et al.*, “Attention is all you need,” *NeurIPS*, 2017.
- [39] J. Lee *et al.*, “Set transformer: A framework for attention-based permutation-invariant neural networks,” in *Int. conference on machine learning*, 2019.
- [40] S. Eitinger *et al.*, “Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset,” in *ICCV*, 2021.
- [41] H. Caesar *et al.*, “nusenes: A multimodal dataset for autonomous driving,” in *CVPR*, 2020.
- [42] B. Wilson *et al.*, “Argoverse 2: Next Generation Datasets for Self-driving Perception and Forecasting,” in *NeurIPS Datasets and Benchmarks*, 2021.
- [43] J. Houston *et al.*, “One thousand and one hours: Self-driving motion prediction dataset,” in *Conf. on Robot Learning*, 2021.
- [44] C. Ionescu *et al.*, “Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments,” *PAMI*, 2014.
- [45] N. Mahmood *et al.*, “AMASS: Archive of Motion Capture as Surface Shapes,” in *ICCV*, 2019.
- [46] T. von Marcard *et al.*, “Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera,” in *ECCV*, 2018.
- [47] R. Martin-Martin *et al.*, “Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments,” *PAMI*, 2021.
- [48] A. Shenoi *et al.*, “Jrmot: A real-time 3d multi-object tracker and a new large-scale dataset,” in *IROS*, 2020.

¹We report publicly available updated numbers for AgentFormer ([arxiv/github](https://arxiv.org/abs/2207.05844)) which differ from the original publication.