Intention Modelling with Normalizing Flows for User-centric Collaborative Navigation

Kavindie Katuwandeniya, **Stefan H. Kiss**, **Lei Shi** and **Jaime Valls Miro**
{kavindiehansika.katuwandeniya,stefan.h.kiss}@student.uts.edu.au, {lei.shi-1,jaime.vallsmiro}@uts.edu.au
University of Technology Sydney

Abstract

A predictive agent to help the operator of an assistive mobility device like a wheelchair to cooperatively navigate in accordance with the environment is proposed. The framework is predicated on interpreting a user's intended future trajectory to take intervention decisions in real time and collaboratively operate the robotic agent. The work incorporates user control signals alongside information from the surroundings via visual feedback and the recent history of the agent's motions to learn a conditional Normalizing Flow, an advanced deep generative model with the crucial ability to recover exact likelihoods for each of its samples. The integration leads to a uniform probabilistic framework for user intention estimation conditioned on different types of information. Experimental results in an urban navigation simulator (CARLA) demonstrate prediction accuracy increases up to 22.89% when user control inputs are being modelled jointly by the proposed end-to-end framework. A baseline comparison where user controls are considered independent and subsequently fused also suggests that the proposed deep learning based solution provides a stepped improvement. The framework paves the way for a fully functional shared-control navigation strategy for intelligent collaborative control intervention.

1 Motivation

Human-robot interaction (HRI) is a vast field of study which focuses on robotic agents in use or in interaction with humans [Goodrich and Schultz, 2008]. Human-robot collaboration (HRC) is a sub-field of HRI, and the key attribute of HRC is working towards achieving a common goal between a human and a robot [Bauer *et al.*, 2008]. Daily navigation involves a user and an assistive mobility device (robotic agent) to work in tight collaboration to reach the desired destinations. In that regard, agreeing upon a joint policy through understanding the intention of all the agents involved is crucial, such that individual actions can be optimised to reach the common goal [Bauer *et al.*, 2008].

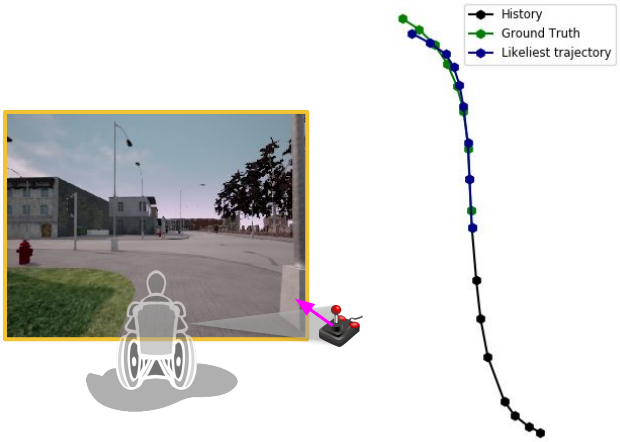

Figure 1: The likeliest trajectory Y from the proposed model $Q(Y|X, I, C)$, conditioned on the **past trajectory** X, scene context I, and the user control C. The executed trajectory by the user is given as ground truth trajectory.

In a human-robot team like an assistive mobility system, usually the human assigns the goal and communicates it through either explicit instructions or implicit actions. The robot interprets the intended goal and acts to achieve it. Comparing with other types of communication methods like speech and gesture, the control signal (if available) is a more natural format in assistive mobility applications.

The motivation for the work presented in this paper is to develop a navigational framework based on human-robot collaboration which can be integrated with existing mobility platforms incurring minimal changes to the

traditional commanding hardware interfaces that users are accustomed to. With this in mind, the aim is to incorporate driving commands from an active human driver as signalled through the typical control interfaces of mobility devices to infer the intended future trajectories a user may choose to follow in a given situation. Without loss of generality, in the case of a power mobility device these are joystick commands, yet other modalities could be equally considered (e.g. voice commands, chin-operated controls, head-rest force sensing devices or Sip-n Puff). Additional advantages of the intended set-up is the readily availability of the user command, user's familiarity with the generation of user commands leading to less diversion of attention. However, this also poses some challenges such as limited information to infer the user intention and assumption of expert user behaviours. To alleviate this, other sensors available in the mobile platform can provide extra information like the past vehicle state and scene context along with the active user control commands, to produce a more accurate intention estimation. Figure 1 is an example scene depicting the outcome from the proposed framework.

2 Related Work and Background

The scope of the work hereby presented sits at the intersection of various domains within the broader realm of HRC.

2.1 User Intention Definitions

The definition of user intention is rather ambiguous. It is application specific, and particularly challenging to estimate due to the complex nature of human behaviour. In the context of assistive mobility devices, the user intention has been historically constrained to specific algorithms such as "follow-corridor", "avoid obstacles", "navigate to *this* place of interest" [Taha *et al.*, 2007]. These systems focus on understanding the environment instead of dealing with the ambiguity of human intention. In the recent research on intention estimation, user intention has been defined as the immediate control for the mobility device [Katuwandeniya *et al.*, 2020] or the final goal pose [Narayanan *et al.*, 2016]. [Demeester *et al.*, 2006] was the first to define the user intention to be a combination of trajectories with a goal state. The intended future trajectory for a predefined time period is also used in the literature as a definition for user intention [Katuwandeniya *et al.*, 2021]. Trajectories capture the spatio-temporal relationship of an entity and conveys more information about the user's traversability preference as opposed to a final goal or an immediate control. Thus for this work the system estimates the intended future trajectory of the user as their intention.

2.2 Multi-Modal Trajectory Prediction

Complex human psychology imposes on disregarding hand-tailored deterministic models for trajectory estimation [Kalman, 1960; Priestley, 1981] and to develop data-driven multi-modal trajectory prediction models. Generative models are capable of determining the underlying data distribution. Generative Adversarial Networks (GANs) [Goodfellow *et al.*, 2020; Gupta *et al.*, 2018; Amirian *et al.*, 2019; Li *et al.*, 2019; Katuwandeniya *et al.*, 2021], Variational Auto-Encoders (VAEs) [Kingma and Welling, 2013; Lucas *et al.*, 2019] and Normalizing Flows (NFs) [Kobyzev *et al.*, 2020; Schöller and Knoll, 2021; Bhattacharyya *et al.*, 2020a; Bhattacharyya *et al.*, 2020b] are widely used deep generative models in the trajectory prediction domain.

Unlike other deep generative models, NFs provide the capability to calculate exact likelihoods for generated samples. They also do not suffer with mode collapse as with GANs [Srivastava *et al.*, 2017] and posterior collapse as with VAEs [Lucas *et al.*, 2019]. In our previous work [Katuwandeniya *et al.*, 2021], a GAN framework was utilised with a rule-based perception framework to counteract the converges issues arising due to training with a small dataset with no visual annotation. In that work we proved the importance of providing scene context, the resulting framework could generate k trajectories complaint with the environment. However, when it comes to executing the intended estimation, a trajectory has to be picked randomly. Modelling the above problem with NFs allows a probabilistic approach for trajectory selection and quantitative evaluation.

2.3 Shared-Control Navigation

Shared-control is a more focused area of research under the field of shared decision-making [Trautman, 2015] where the control commands sent to the platform to be executed are a combination of the constant control of the human and the autonomous system. Since the human operator is physically on the platform, we are dealing with on-board shared-control as opposed to shared teleoperation. Combining the two controls has been reported in the literature based on hand-tailored weighted functions [Urdiales *et al.*, 2007]. The limitation in this set-up is the need of careful tuning to adapt to different scenes and users. The solution is also not suitable for large, map-less environments. Thus, in this work, the focus is on generating controls from an end-to-end framework which accounts the user control as an input.

In this work we advocate for a framework which takes the user input along with the vehicle state and current environment context to develop a distribution of the intended trajectory and recommend picking the likeliest trajectory under the generated distribution to be executed. The work in this paper extends a predictive model

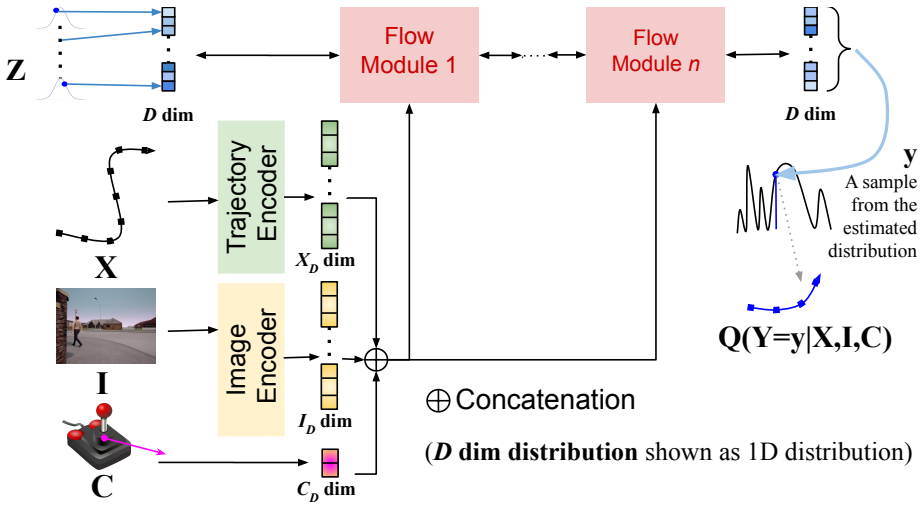

Figure 2: Proposed end-to-end user-centric shared control framework.

to be a user-centric shared-control framework.

We show that frameworks solely based on the past vehicle state have a lower accuracy in comparison to a model utilising all the available observations. We also perform a qualitative evaluation showing how the user signal is correlated with the output trajectory prediction. Additionally an experiment is conducted for justification of integrating user control in an end-to-end framework as opposed to a decoupled approach.

3 Problem Definition and Approach

Our aim is to model the user's intention Y given some observations. We describe the user's intention by their future trajectory: a sequence of 2D positions (x_{t_i}, y_{t_i}) at N discrete time-steps into the future $t_1, \ldots, t_N$.

In this work, we consider three observation modalities to inform the prediction: the past trajectory X, the current scene information I, and the current user control C. The past trajectory X is described similarly to the future trajectory, as a sequence of M past positions up until the current time $t_{1-M}, \ldots, t_0$. The context of the current scene I is provided as a RGB colour image from a forward-facing vehicle-mounted camera. Fundamental to this work, the user's current control C is given from a 2-dimensional joystick command.

We believe the user's intention Y is not fully observable given (X, I, C). As such, we model the intention probabilistically, and attempt to learn a model $Q(Y|X, I, C)$ to approximate the true distribution $\Pr(Y|X, I, C)$. Additionally, we desire an efficient method of sampling from the proposed distribution: a *generative* model. Due to their proven success in approximating complex distributions, and having the desirable properties, we take our model Q to be a *Normalizing Flow*.

Given a training dataset of (Y_i, X_i, I_i, C_i) tuples, the model is optimised by negative log likelihood, thereby maximising the joint probability of the dataset under the model,

$$Q^* = \arg\min_Q - \sum_i \log Q(Y_i|X_i, I_i, C_i)\,. \tag{1}$$

The overall framework is given in Figure 2.

3.1 Normalizing Flows

Normalizing Flows (NFs) convert a known base distribution Z (often Gaussian) to the distribution of interest through a sequence of functions f_i termed *flow modules*. These functions must be both invertible and differentiable, causing the composition to also be invertible and differentiable. This allows the *change of variables* formula to be applied for tractable calculation of the resultant probability density.

$$\begin{aligned} Z &\sim \mathcal{N}(0, 1) \\ f &= f_n \circ f_{n-1} \circ \cdots \circ f_2 \circ f_1 \\ Y &= f(Z) \\ Z &= f^{-1}(Y) \end{aligned} \tag{2}$$

$$\Pr(Y{=}y) = \Pr\left(Z{=}f^{-1}(y)\right) \left| \frac{\mathrm{d}f^{-1}(y)}{\mathrm{d}y} \right| \tag{3}$$

The invertability and differentiability constraints imposed on the flow modules f_i limit the expressivity of the estimated distribution. Monotonic Rational-Quadratic Splines (RQSs) are a family of curves that have been shown to enhance the flexbility of the flow [Durkan *et al.*, 2019]. They are non-linear and expressive, but analytically invertible and differentiable. Each spline is

described by $K+1$ knot points inside a boundary region $[-B, +B]$. It requires $3K-1$ parameters to describe a RQS with $K+1$ points: $2K$ parameters for the bin widths and heights and $K-1$ parameters for the derivatives at knot points inside the boundary (the derivatives at the endpoints are set to 1 so that the function is the identity outside the region). The reader is referred to [Durkan *et al.*, 2019] for a detailed description of RQS NFs.

Coupling

Our problem is multi-dimensional: we wish to model the user's future trajectory, a $2N$-dimensional object. It is important to correlate the dimensions while maintaining the ease of invertibility. Coupling layers have been introduced by [Dinh *et al.*, 2014] in an attempt to model high dimensional, complex distributions using NFs. Coupling layers serve as the main building box for generating triangular Jacobian matrices, resulting in faster computation and inference while maintaining the flexibility of the flow.

For a D-dimensional variable $Y \in \mathbb{R}^D$, the base distribution $Z \in \mathbb{R}^D$ must also be D-dimensional for the flow f to be invertible.

Assume the input to flow module f_i is $Z_i = (Z_i^A, Z_i^B)$ where $Z_i^A \in \mathbb{R}^d$ and $Z_i^B \in \mathbb{R}^{D-d}$ form a disjoint partition of Z_i. A learnable function h_i, termed a *conditioner*, takes one part Z_i^B and generates the parameters $\theta_i = h_i(Z_i^B)$ required to transform the other part through an invertible mapping $g_{\theta_i}(Z_i^A) = Z_{i+1}^A$. The second part Z_i^B is passed through identically.

One important thing to note under this scheme is that g is an invertible function, whereas h is not required to be, and can be learnt with the full flexibility of a deep neural network.

3.2 Conditioning

Ultimately, we do not just wish to model the distribution $\Pr(Y)$, but the conditional distribution $\Pr(Y|X, I, C)$ given the observations of the current scenario. The conditioning function h can be developed to accept not just Z_i^B, but also additional variables that we wish to condition the output distribution on. As h is not required to be efficiently invertible, flexible deep neural networks appropriate to the input data type can be used. The output is thereby conditioned on X, I, and C. Each of these inputs can be initially encoded or feature-engineered to improve the learning process.

Conditioning on Past Trajectory X

The idea of conditioning on the past trajectory X has been exploited in many deep generative models [Gupta *et al.*, 2018; Schöller and Knoll, 2021; Katuwandeniya *et al.*, 2021]. Inspired by these, a recurrent neural network is used to encode the past vehicle state which outputs a X_D-dimensional vector. The specific implementation was performed as per [Schöller and Knoll, 2021], readers are referred there for further details.

GPS and/or IMU sensors mounted on the mobile device can be used to provide odometry and extract a sufficiently accurate X.

Conditioning on Scene Information I

[Katuwandeniya *et al.*, 2021; Li *et al.*, 2019] have proven the importance of scene context in future trajectory prediction when modelled using deep generative models. A low cost RGB camera can be easily integrated to a mobile device (if not already equipped with one) to obtain visual information from a first-person view. An image encoder is trained to extract visual cues and reduce the dimensionality of the image to a I_D-dimensional vector. For this work we used ResNet [He *et al.*, 2016] (initialised using a model trained on the ImageNet dataset [Deng *et al.*, 2009]) as the image encoder. The RGB image was concatenated with 2 additional channels, specifying the pixel position (u, v) so that the image encoder can easily maintain the spatial awareness. It was observed empirically that the addition of the (u, v) channels improved the accuracy of the results.

Conditioning on User Control C

For the developed framework to be a human-robot collaboration framework the most crucial input is the user control. Depending on the mobile platform the user control could be the joystick axes values or steering wheel angle, acceleration and brake pedal state. Since the dimensionality of the user control is low, it was decided to integrate it without encoding further ($C_D = 2$). However, the authors note that the control could be further encoded or feature-engineered for potentially higher accuracy.

Conditioner Networks

Each conditioner h_i is a multi-layer perceptron which takes an input of size $(D-d) + X_D + I_D + C_D$ and outputs a vector θ_i of size $(3K-1)d$ giving the parameters for the RQS of each dimension in Z_i^A. The multi-layer perceptron was constructed with 5 hidden layers, each followed by an ELU activation as suggested by [Schöller and Knoll, 2021].

4 Implementation

CARLA Dataset The training and evaluation was carried out using a dataset collected from a wheelchair platform navigated through realistic urban scenarios in the CARLA [Dosovitskiy *et al.*, 2017] simulation environment. The environment also consisted of pedestrians and vehicles that obey traffic lights and road rules, and react to other obstacles in the road. A physical joystick was used to control the wheelchair through a Robotic

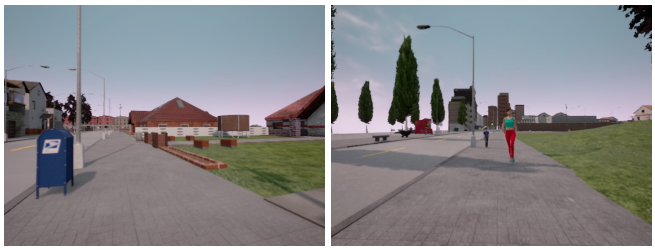

Figure 3: Sample images from the CARLA dataset.

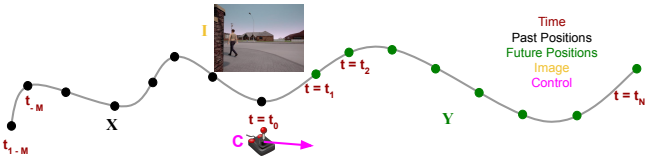

Figure 4: A data instance, showing the past and future trajectories X and Y, the scene information I, the user control C, and their corresponding times.

Operating System (ROS) [Quigley *et al.*, 2009] bridge. Out of the 6 (rosbag) data recordings, each with a duration of about 10 minutes, one was used for validation, one for testing, and the remaining 4 for training. Figure 3 shows two sample images from the simulated dataset. 5252 data instances were used for training, 1308 for validating and 1261 for testing.

Data Instances for Training, Validating, and Testing A data instance representative of the data used for training, testing and validating is given in Figure 4. A trajectory of $M+N$ time-steps is selected where the sampling rate is chosen to be $0.5s$. With the observation and prediction lengths selected as $4s$ each, the trajectory consists of 16 (x, y) positions where $M = 8$ and $N = 8$. For numerical stability, the initial position $X_{t_{1-M}}$ was subtracted from all points and the entire trajectory was rotated around the last observed point X_{t_0} such that the global orientation of the trajectory does not add unnecessary complications for the network to learn. This results in the estimated distribution's dimension being $|Y| = D = 2N = 16$. The flow coupling parameter was chosen to split in half, such that $d = D/2 = 8$.

The current RGB image I from the front facing camera gives the environment context. The user input is the joystick control commands $C = [\text{axis}_0, \text{axis}_1]$. The reasoning for using only the current joystick command is that the past trajectory is representative of the past joystick commands and thus is redundant, but the current active user control is the most up-to-date representation of the user intention.

The vehicle's past trajectory X was encoded to a X_D=16-dimensional vector, the image I encoded to a I_D=16-dimensional vector, and the user control is C_D=2-dimensional. $K + 1 = 8$ knot points were chosen to represent the RQS in each dimension and the boundary was chosen to be $B = 15$. Boundary points need to be chosen to match the dataset. Higher the K, the better the representation at the cost of computation.

The network is trained to minimise the objective function mentioned in equation (1) using the ADAM optimiser for 150 epochs with a learning rate scheduler which updates the learning rate based on the validation loss.

5 Evaluation

The improvement in accuracy of future trajectory prediction is quantitatively justified. A qualitative analysis of how the framework adapts to an active user control to comply with user intention is also carried out. Additionally, the results from an experiment where the user control was decoupled and was later fused probabilistically is also given as a justification for the line of thought for the proposed framework.

5.1 Quantitative Evaluation

It is common in trajectory prediction research to use Average Displacement Error (ADE) and Final Displacement Error (FDE) as evaluation metrics. ADE measures the average Euclidean distance between the ground truth trajectory and the trajectory of interest over the predicted time horizon, while FDE measures the Euclidean distance between the final positions. For selection and execution purposes in an online platform where the ground truth is not available, measuring the ADE and FDE of the likeliest trajectory out of the k samples generated from the estimated distribution: $\text{ADE}_{\texttt{like}}$, $\text{FDE}_{\texttt{like}}$ against a randomly generated sample ($\text{ADE}_{\texttt{rand}}$ and $\text{FDE}_{\texttt{rand}}$) is the most reasonable comparison. For the purpose of completeness, $\text{ADE}_{\texttt{min}}$ and $\text{FDE}_{\texttt{min}}$ which is the ADE and FDE of the trajectory closest to the ground truth (out of the k generated) are included. However, the selection of the minimum-error trajectory is not available at runtime and cannot be executed. k was chosen to be 20. All values are given in *meters*.

We trained 3 models to show the benefits of incorporating the user's control input in predicting the future trajectory of the mobility device. Each model approximates a distribution $Q(Y|B)$ where B is the random variable the model is conditioned on. The results of these models are shown in Table 1. It is worth noting that modelling $Q(Y|X)$ using NFs with RQS as the coupling function is proven to do better (equivalently only to Trajectron++ [Salzmann *et al.*, 2020]) in comparison to state-of-the-art generative models [Schöller and Knoll, 2021]. The authors have compared against tractable and

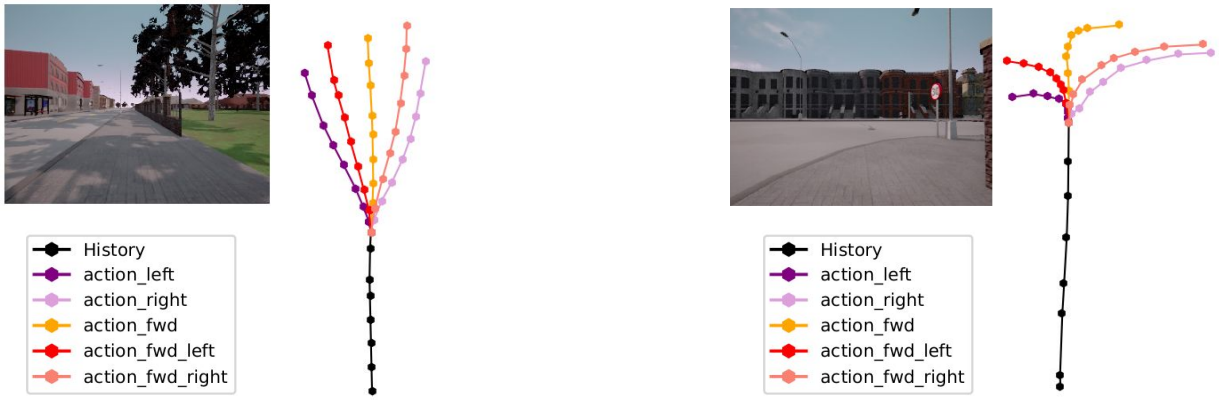

Figure 5: Qualitative Analysis of active user control. On a straight road (left) and at a corner (right).

Metric	$Q(Y\|X)$	$Q(Y\|X,I)$	$Q(Y\|X,I,C)$	Improv.
$\text{ADE}_{\texttt{min}}$	0.62	0.55	0.45	18.18%
$\text{ADE}_{\texttt{like}}$	1.29	0.97	0.80	17.53%
$\text{ADE}_{\texttt{rand}}$	1.57	1.11	0.96	13.51%
$\text{FDE}_{\texttt{min}}$	0.98	0.83	0.64	22.89%
$\text{FDE}_{\texttt{like}}$	2.55	1.80	1.51	16.11%
$\text{FDE}_{\texttt{rand}}$	3.12	2.03	1.74	14.29%

Table 1: Comparative results of the proposed framework conditioned with different observation modalities. Improv. refers to the improvement of the $Q(Y|X,I,C)$ model over $Q(Y|X,I)$.

non-tractable generative models, which motivates the architecture adopted in the proposed framework in this work.

It is clear from columns 2 and 3 that when conditioned on more information (scene context), the results improve. With the integration of C, in addition to converting a predictive framework to a human-robot collaboration framework, it improves the accuracy of the considered metrics as shown in column 4. The improvement: a percentage of error reduction from modelling $Q(Y|X,I)$ to $Q(Y|X,I,C)$ is given in the last column.

For execution, the results justify the selection of the likeliest trajectory against picking a random trajectory. In the proposed framework, the improvement of picking the likeliest over random is 16.67% with regards to ADE and 13.22% with the FDE.

Table 2 shows a comparison of $Q(Y|X,I)$ with the proposed NFs work when not yet condition on the user input, i.e. only condition on (X,I): $Q_{\text{NF}}(Y|X,I)$, with respect to our earlier results from the model $Q_{\text{GAN}}(Y|X,I)$ when k number of equally probable trajectories were generated. $Q_{\text{GAN}}(Y|X,I)$ was modelled using a GAN, and later probabilistically fused with a segmented image [Katuwandeniya *et al.*, 2021]. Since all k generated trajectories (also set to 20) are equally probable, it is

Metric	$Q_{\text{GAN}}(Y\|X,I)$	$Q_{\text{NF}}(Y\|X,I)$	Improv.
ADE	1.61	0.97	39.75%
FDE	3.35	1.80	46.27%

Table 2: The improvement of the NF model Q_{NF} over a purely-generative GAN model Q_{GAN}.

only fair to compare a random trajectory out of the 20 against the likeliest trajectory from the 20 generated under the model proposed in this paper $Q_{\text{NF}}(Y|X,I)$ The significant improvement justifies the utilisation of NFs for estimating the user's intention. In addition, unlike GANs, there are no convergence issues with NFs when considering high dimensional image data. Thus, the proposed framework is also more suitable for small datasets with no visual annotations.

5.2 Qualitative Evaluation

It is interesting to visualise how the estimation of user intention adapts to different user control commands given the same X and I. A qualitative analysis is thus presented whereby the actual joystick command is replaced with 5 preset values: *action_left, action_fwd_left, action_fwd, action_fwd_right, and action_right.* Results from a couple of token examples are given in Figure 5.

It is apparent how user control C has a predictable impact on the generated trajectory: *action_left* trends left, *action_fwd_left* trends left while maintaining forward speed, etc. Crucially however, all generated trajectories are compliant with the other observations: history X and scene information I. Figure 5 left shows a straight road scenario, where the system generates largely forward-directed trajectories under all controls, whereas the figure on the right shows a right-curved scenario where the system shows a stronger bias for taking the right corner with confidence. The user is nevertheless not prevented from turning *against* the corner, albeit trajectories leading off the footpath are taken at

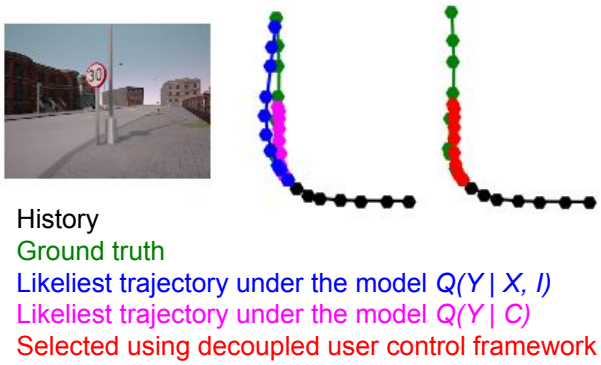

Figure 6: Example scenario from the framework developed with the decoupled user control.

a considerably reduced speed. Notably, the *action_fwd* trajectory still anticipates a later turn around the corner.

5.3 Decoupled User Control

The effectiveness of integrating user controls in an end-to-end framework as opposed to fusing user directives later with a model trained to estimate $Q(Y|X,I)$ was also carried out as a comparative baseline. Two assumptions are required for this experiment to be meaningful in probabilistic terms: (X,I) and C are independent, and also conditionally independent given Y:

$$\begin{aligned} &(X,I) \perp\!\!\!\perp C\,, \\ &((X,I) \perp\!\!\!\perp C) \mid Y\,. \end{aligned} \tag{4}$$

This allows $Q(Y|X,I,C)$ to be decomposed and the likelihood of trajectory y can thus be calculated as

$$\Pr(Y{=}y|X,I,C) \propto \Pr(Y{=}y|X,I) \cdot \Pr(Y{=}y|C) \tag{5}$$

In addition to the model trained to estimate $\Pr(Y|X,I)$, a separate model was trained to estimate $\Pr(Y|C)$, similarly using NFs. A sampler was developed which generates k trajectories from each model and calculates the total likelihood as per equation (5). It is precisely the ability of NFs to calculate the likelihood of samples that makes this feasible. The likeliest trajectory is the trajectory with the highest likelihood under both models. An example is shown in Figure 6.

For the considered 1261 test instances, the sampler selected 653 from the model $Q(Y|X,I)$ and the rest from $Q(Y|C)$. The ADE and FDE of the likeliest trajectory chosen under this method is compared in Table 3 against those using the proposed end-to-end model directly incorporating user controls $Q(Y|X,I,C)$. The improved performance for the end-to-end framework is self-evident.

It is arguable that the independence assumptions made may be behind the deterioration in the results. A more intuitive reasoning may also point towards the fact that without having all the available information (C in the case of $Q(Y|X,I)$, and (X,I) in the case $Q(Y|C)$), it is difficult to make informed decisions.

Metric	$\arg\max_Y$ $Q(Y\|X,I)\cdot Q(Y\|C)$	$\arg\max_Y$ $Q(Y\|X,I,C)$
ADE	1.07	0.80
FDE	1.91	1.51

Table 3: Decoupled learning vs end-to-end learning

6 Conclusion

We present a probabilistic framework for user intention estimation applied to the shared control of an assistive power mobility device like a wheelchair. The scheme is built on top of a state-of-the-art deep generative model, namely Normalizing Flows. The input to the proposed algorithm includes the device's historical trajectory, current visual information, and the current user controls via a joystick input device. The algorithm's output is a high dimensional multivariate conditional distribution, representing the device's predicted future trajectory as a proxy for user intent. The framework is modular and can be expanded further to incorporate additional inputs that may be deemed suitable to the application of interest (platform accelerations, other agent's paths, etc).

The proposed framework is evaluated on a simulated dataset hence allowing for controlled oracle comparatives. Data was collected from driving a simulated wheelchair in a realistic urban navigation simulator (CARLA) with a real joystick. Experimental results demonstrate that prediction accuracy increases by a large margin (up to 22.89% depending on the evaluation metrics) when user control inputs are being modelled jointly with other modalities. In addition, anecdotal evidence from a qualitative evaluation is supplied to be able to rationalise the significance of incorporating user controls into the model, whereby updated predictions can be asserted to align with expectations.

A separate validation considering the user's control as independent variables for probabilistic fusion suggests that the proposed deep learning based solution indeed provides more accurate results. The proposed data-driven framework has thus been proven able to infer most promising user intention predictions allowing for the development of a fully functional shared navigational strategy for joint control intervention.

The future work is twofold: explore other formats of user control signals, including richer information such as raw predictions from accurate motion models, and transferring the intervention strategy from simulation to the real platform, testing with users of varying abilities under a range of different scenarios.

References

[Amirian *et al.*, 2019] Javad Amirian, Jean Bernard Hayet, and Julien Pettre. Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs. *IEEE Computer Society Conference on CVPR Workshops*, 2019-June:2964–2972, 2019.

[Bauer *et al.*, 2008] Andrea Bauer, Dirk Wollherr, and Martin Buss. Human–robot collaboration: a survey. *International Journal of Humanoid Robotics*, 5(01):47–66, 2008.

[Bhattacharyya *et al.*, 2020a] Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional flow variational autoencoders for structured sequence prediction, 2020.

[Bhattacharyya *et al.*, 2020b] Apratim Bhattacharyya, Christoph-Nikolas Straehle, Mario Fritz, and Bernt Schiele. Haar wavelet based block autoregressive flows for trajectories. In *DAGM German Conference on Pattern Recognition*, pages 275–288. Springer, 2020.

[Demeester *et al.*, 2006] Eric Demeester, Alexander Huntemann, Dirk Vanhooydonck, Gerolf Vanacker, Alexandra Degeest, Hendrik Van Brussel, and Marnix Nuttin. Bayesian estimation of wheelchair driver intents: Modeling intents as geometric paths tracked by the driver. In *2006 IEEE/RSJ IROS*, pages 5775–5780. IEEE, 2006.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[Dinh *et al.*, 2014] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[Dosovitskiy *et al.*, 2017] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[Durkan *et al.*, 2019] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Adv. Neural Inf. Process. Syst.*, volume 32, 2019.

[Goodfellow *et al.*, 2020] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[Goodrich and Schultz, 2008] Michael A Goodrich and Alan C Schultz. *Human-robot interaction: a survey*. Now Publishers Inc, 2008.

[Gupta *et al.*, 2018] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. *Proc. of the IEEE Computer Society Conference on CVPR*, pages 2255–2264, 2018.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[Kalman, 1960] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[Katuwandeniya *et al.*, 2020] Kavindie Katuwandeniya, Jaime Valls Miro, and Lakshitha Dantanarayana. End-to-end joint intention estimation for shared control personal mobility navigation. In *2020 16th ICARCV*, pages 1–6. IEEE, 2020.

[Katuwandeniya *et al.*, 2021] Kavindie Katuwandeniya, Stefan H. Kiss, Lei Shi, and Jaime Valls Miro. Multi-modal Scene-compliant User Intention Estimation for Navigation. In *IEEE International Conference on Intelligent Robots and Systems*. (to appear), 2021.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Kobyzev *et al.*, 2020] Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[Li *et al.*, 2019] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional Generative Neural System for Probabilistic Trajectory Prediction. In *IEEE IROS*, pages 6150–6156, 2019.

[Lucas *et al.*, 2019] James Lucas, George Tucker, Roger Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. *arXiv preprint arXiv:1911.02469*, 2019.

[Narayanan *et al.*, 2016] Vishnu K Narayanan, Anne Spalanzani, and Marie Babel. A semi-autonomous framework for human-aware and user intention driven wheelchair mobility assistance. In *2016 IEEE/RSJ IROS*, pages 4700–4707. IEEE, 2016.

[Priestley, 1981] Maurice Bertram Priestley. *Spectral analysis and time series: probability and mathematical statistics*. Number 04; QA280, P7. Academic Press, 1981.

[Quigley *et al.*, 2009] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[Salzmann *et al.*, 2020] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *arXiv preprint arXiv:2001.03093*, 2020.

[Schöller and Knoll, 2021] Christoph Schöller and Alois Knoll. Flomo: Tractable motion prediction with normalizing flows. *arXiv preprint arXiv:2103.03614*, 2021.

[Srivastava *et al.*, 2017] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3310–3320, 2017.

[Taha *et al.*, 2007] Tarek Taha, Jaime Valls Miro, and Gamini Dissanayake. Wheelchair driver assistance and intention prediction using pomdps. In *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, pages 449–454, 2007.

[Trautman, 2015] Pete Trautman. Assistive planning in complex, dynamic environments: a probabilistic approach. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3072–3078. IEEE, 2015.

[Urdiales *et al.*, 2007] C. Urdiales, A. Poncela, I. Sanchez-Tato, F. Galluppi, M. Olivetti, and F. Sandoval. Efficiency based reactive shared control for collaborative human/robot navigation. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3586–3591, 2007.