Uncertainty-aware Pedestrian Crossing Prediction via Reinforcement Learning

Siyang Dai, Jun Liu, *Senior Member, IEEE,* Ngai-Man Cheung, *Senior Member, IEEE*

***Abstract*—Pedestrian safety is a huge concern for deploying autonomous vehicles in urban environments. Accidents involving pedestrians pose a higher degree of severity, sometimes causing serious injuries and fatalities [1]. It's a challenging task to predict whether a pedestrian will cross the road since they can move in any direction and change motion suddenly. The inherent uncertainty in pedestrian motion has been addressed with probabilistic models in previous works. However, these models are too computationally expensive for real-time predictions. In this paper, we propose a novel reinforcement learning (RL) framework which produces soft labels for the training dataset in order to address the observed data uncertainty. We formulate novel state representations incorporating predictive uncertainty to learn more informative soft labels that improve the model performance and reliability. Finally, we validate the proof of concept with two benchmark datasets and show with extensive experiments on competitive prediction models that our method (even using fewer input modalities) significantly improves the accuracy and f1 score by up to 12% and 13% respectively. We also show that soft labeling as a form of regularization increases model reliability where the model is more accurate when the confidence level is high and more aware of its limitations with indication of low confidence.**

***Index Terms*—Pedestrian action prediction, autonomous vehicle, reinforcement learning, uncertainty estimation.**

I. Introduction

In urban traffic, pedestrians are a major source of concern for autonomous vehicles with the potential to cause severe accidents. According to the Global status report on road safety 2023 [2], Pedestrians constitute a significant portion of traffic-related fatalities worldwide, at a rate of 23%. Road traffic injuries are the leading cause of death for children and young adults aged 5-29 years, according to 2019 data. For autonomous vehicles to be deployed in urban environments, it is crucial to predict pedestrian motion accurately and reliably.

Pedestrian motion prediction is a special case of human motion prediction [3]–[8]. Recently, many studies have been conducted for pedestrian motion prediction, such as [9]–[15] for crossing prediction and [16]–[24] for trajectory prediction. In this paper, we focus on pedestrian crossing prediction, which is a binary classification problem that predicts whether a pedestrian will cross the road at some point in the future. Specifically, the inputs to the prediction model are bounding box, pose, and visual features of the pedestrian as well as vehicle speed in a time sequence, and the output is a binary label indicating whether the pedestrian will cross the road or not. What makes it a challenging problem is the observed multi-modal sensory data including RGB camera, LiDAR and vehicle odometry; and the data pre-processing in the preparatory tasks *i.e.* pedestrian detection or identification [25]–[27] and pose estimation [28]–[30]. Both the observed and post-processed data contain considerable amount of variability and noise that leads to uncertainty in prediction. The type of uncertainty that captures inherent noise in the observed data is named aleatoric (data) uncertainty [31], which we target to address in this paper.

Fig. 1: Example of uncertain pedestrian motion. The pedestrian is walking towards the road during observation but finally turns away thus is labeled *not crossing*.

The aleatoric uncertainty in pedestrian crossing prediction mainly comes from two sources. The first source is sample hardness [32], [33] which is inherent in the ambiguous motion features. For example, the intention of a pedestrian standing near the crosswalk but not looking in the direction of oncoming traffic is hard to infer. Soft labeling for the hard samples provides a form of regularization helping the prediction model better handle the ambiguous samples [34]. The second source is the noisy (incorrect) labels which disagree with the motion features [35]–[37]. For example, a pedestrian could be walking towards the road but turn away at the last moment as illustrated in Fig. 1. The assigned label for this pedestrian is *not crossing* according to his ending action even though the observation exhibits *crossing* characteristic. The contradiction between observation and label (even for a small portion of the dataset) can confuse the prediction model and lead to a degraded performance as discussed in the beginning of Sec. III. We also notice inconsistent labeling in the Pedestrian Intention Estimation (PIE) dataset [20], which provides two types of labels: intention and action. For intention labeling, human subjects were asked to rate a pedestrian's crossing intention by

Siyang Dai, Jun Liu and Ngai-Man Cheung are with Singapore University of Technology and Design. E-mail: siyang_dai@mymail.sutd.edu.sg; jun_liu@sutd.edu.sg; ngaiman_cheung@sutd.edu.sg.

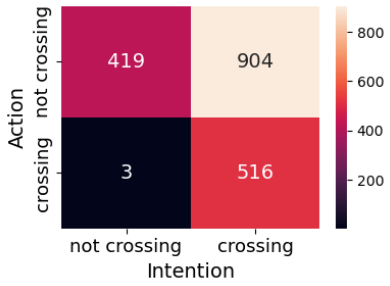


Fig. 2: Inconsistency between "intention" and "action" labels provided by PIE dataset [20].

viewing an initial footage of the pedestrian. Action label is the ending action given the complete observation of the pedestrian. When comparing intention to action labels illustrated in Fig. 2, we find that half of the pedestrian samples have inconsistent labels and the majority of them are labeled to have intention to cross but not cross in the end. This finding suggests the existence of inherent uncertainty in the dataset.

Most of the existing works concentrate on designing network architectures and use the given labels as is. However, aleatoric uncertainty is inherent in the observed data and cannot be eliminated by changing networks. To solve this issue, we directly work on the dataset by soft labeling the uncertain samples with a reinforcement learning framework which is guided by the predictive uncertainty as part of the state representation. In summary, our main contributions are: 1. We propose a novel reinforcement learning framework where soft labels are learned with hidden features from the prediction model and used to train the model itself, with the RL network using the model's performance change as a reward signal, creating a continuous improvement loop. 2. We formulate novel state representations by incorporating an uncertainty metric, leading to better selection of uncertain samples and improvement of model reliability with uncertainty estimation. 3. When applied to competitive models for pedestrian crossing prediction, our framework (even using fewer input modalities) makes significant improvements over the original models on two benchmark datasets.

II. Related work

A. Pedestrian crossing prediction

Pedestrian crossing prediction is a binary classification problem which predicts whether a pedestrian is going to cross the road at some point in the future. Many prior works [11], [12], [38]–[41] employ RNN based methods to process temporal inputs of different modalities followed by a fully connected layer to predict crossing action. [9] and [42] both use hierarchical GRU layers to fuse input features for prediction. A few convolutional methods are attempted such as ConvLSTM for intention prediction [20], and graph convolutional network for reasoning changes in pedestrian pose over time [21]. The hybrid method in [10] encodes visual features by a 3D convolutional network and other input modalities by RNNs followed by a temporal attention module. All encodings are then concatenated and fed into a spatial attention module for prediction. Our method is not limited to input modalities or model architectures while we adapt to existing models to further improve their performance and reliability by considering data uncertainty.

B. Uncertainty in pedestrian motion prediction

A few works have addressed uncertainty in pedestrian motion prediction categorized into two groups: crossing prediction and trajectory prediction. [43] is a close match to our problem as it uses uncertainty estimation to improve robustness of crossing predictions. However, the author uses Monte Carlo dropout to estimate epistemic (model) uncertainty while we use soft labels to handle aleatoric (data) uncertainty. [15] models uncertainty with conditional generative model to conduct probabilistic predictions. We instead work on deterministic models with the goal of producing better labels leveraging observed uncertainty. For trajectory prediction which is a different problem from ours, [23] addresses the uncertainty in pedestrian trajectory by a Bayesian approach and estimates an empirical error bound for the predictive distribution. [18] uses Monte Carlo dropout to quantify the uncertainty in pedestrian trajectory prediction. [17] estimates uncertainty of the trajectory by applying a Kalman Filter with a dynamically adjusted process noise matrix. There are a few works on the uncertainty in general human motion prediction. [44] predicts motion uncertainty by learning a distribution of possible future destinations. [45] employs latent space to capture the inherent uncertainty for predicting multiple feasible trajectories. [46], [47] solve action prediction by estimating soft labels for subsequences at different progress levels using soft regression. [48] also learns soft labels but with a novel annotation strategy allowing the annotator to assign multiple weighted labels. Our novel RL framework with aleatoric uncertainty incorporated into soft label learning is the first of its kind in crossing prediction.

C. Reinforcement learning

Reinforcement learning (RL) is a learning paradigm that learns to take actions in an environment to maximize a reward signal. We find some works in computer vision [49]–[52] which deal with region selection in the object detection task. In action prediction, [53] learns a policy to activate action-related skeleton proposals with deep reinforcement learning. In active learning, [54] and [55] adopt the Deep Q-Learning (DQN) [56] to actively learn an annotation policy for the task of semantic segmentation and pose estimation respectively. [57] applies a policy gradient method named Reinforce [58] to learn an acquisition function. We evaluated both DQN and Reinforce and found they are not suitable for our task. DQN works only with discrete action space and is not friendly to a multi-action setting. Reinforce has high variance and noisy gradients. Our method leverages Deep Deterministic Policy Gradient (DDPG) [59] which can handle continuous action space and multi-action outputs. Moreover, the use of target networks leads to more stable policy learning.

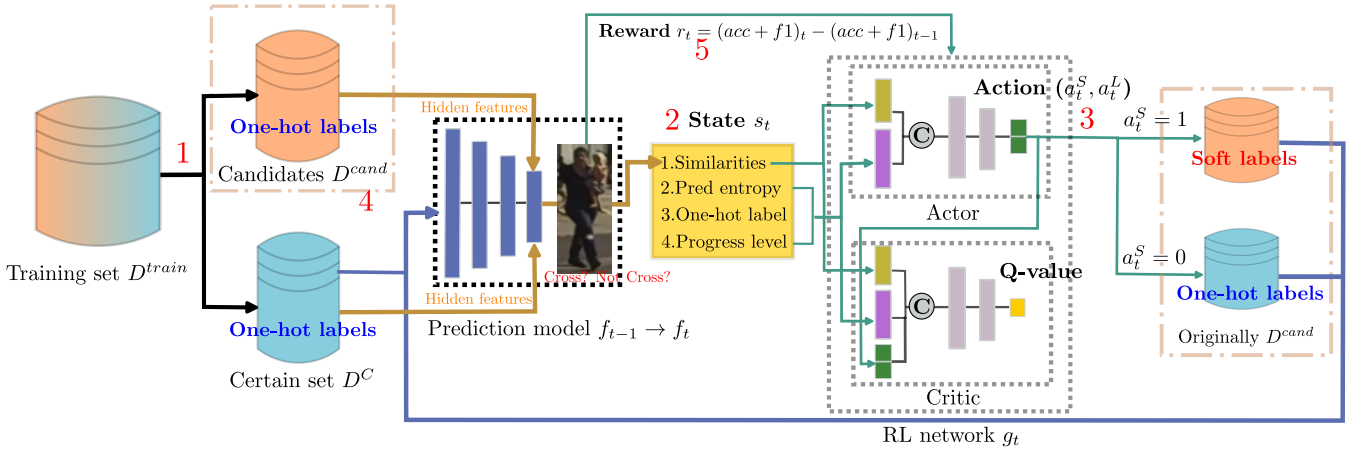


Fig. 3: Workflow of our reinforcement learning framework in selecting and soft labeling uncertain pedestrian samples. Orange line indicates the generation of similarity features; Green line represents training of the RL framework; blue line is training of the prediction model. The detailed procedures are as follows. 1. Split training set D^{train} into candidate set D^{cand} and certain set D^C as described in Sec. III-A Candidates selection. 2. Compute state s_t for the similarity feature by calculating cosine similarities between individual candidate and respective *crossing* and *not crossing* samples in D^C (indicated by orange lines) ; predictive entropy feature is computed from prediction model f_{t-1}; one-hot label and progress level features are from the annotation. 3. Learn dual-action *i.e.* selection and soft labeling (a_t^S, a_t^L) by the Actor network and apply actions to D^{cand} to yield two subsets: one is updated with soft labels and the other remains with one-hot labels. 4. Use updated D^{train} to train f_{t-1} and obtain model f_t. 5. Compute reward r_t on a held-out validation set D^{val} by comparing the sum of accuracy and f1 between f_t and f_{t-1}. Repeat procedure 2-5 for T steps to complete one episode of the RL process.

TABLE I: Illustration of the effect on prediction model PCPA after removing uncertain (simulated with top-loss) samples from the PIE dataset

Method	acc	auc	f1
PCPA [10]	0.87	0.86	0.77
PCPA w/o high-loss samples	**0.89**	**0.87**	**0.81**

III. Method

To show that uncertain samples (*e.g.* Fig. 1) can degrade model performance, we define a hand-crafted rule to filter uncertain samples by using the top-loss samples [32] (since high loss means difficulty to fit the sample indicating ambiguity). We use an empirical 10% as the removal rate and show a comparison of prediction model PCPA [10] trained with and without high-loss samples in Tab. I. We observe an improvement in model performance by 2% gain in accuracy and 4% in f1 score. This simple experiment shows that uncertain samples are indeed harmful.

Instead of simply removing the samples, we handle uncertainty in pedestrian motion by soft labeling. We choose reinforcement learning for generation of soft labels because the RL agent can adapt to the prediction model through maximizing the reward signal which directly links to model performance. In the following subsections, we show how we quantify uncertainty for candidate selection and measuring reliability of prediction models in Sec. III-A, then we discuss the formulation of RL steps as a Markov decision process (MDP) inspired by [54], [55] and how we adopt an Actor-Critic based RL method to solve the MDP problem in Sec. III-B.

A. Quantification of uncertainty

Predictive entropy. In order to select the most uncertain samples for soft labeling, we need to quantify uncertainty of the dataset, indicated by the model confidence. In this paper, we take predictive entropy as the uncertainty metric to represent aleatoric uncertainty. It is defined as the entropy of predicted probability distribution over classes. For each input sample x, the predicted entropy is calculated with Eq. (1) where $\hat{y}_x$ is the prediction.

$$u_x = -\sum_k p(\hat{y}_x = k|x) \log p(\hat{y}_x = k|x) \tag{1}$$

The key idea of quantifying uncertainty for a safety-critical application, such as pedestrian crossing prediction, is to increase robustness and reliability of the model, which is expected to be accurate when it's certain about the predictions, and to indicate high uncertainty when making wrong predictions. The former means the model is reliable given that it is certain about the predictions and therefore more trustworthy. The latter means the model is more likely to be aware of its limitations and not provide overconfident predictions. This is especially important for safety-critical applications, where misleading predictions can result in severe consequences. Based on the aforementioned, we use two conditional probabilities $P(accurate|certain)$ and $P(uncertain|inaccurate)$ to measure the quality of uncertainty estimates across various entropy thresholds as proposed in [60]. The effectiveness of our method is evaluated on both metrics in Sec. IV.

Candidates selection. In Fig. 3, we first split the training data into candidate set D^{cand} and certain set D^C. Then D^{cand}

is fed to the reinforcement learning network for soft labeling. This step narrows down the sample range for the RL network, enabling more efficient training. We filter the candidate set D^{cand} based on the uncertainty level and accuracy of the prediction model. The first rule is to select samples that are highly uncertain, following the idea from the active learning strategy [54], [61], [62]. Those samples with high uncertainty are expected to contain more ambiguous features, and can serve as suitable candidates for soft labeling. Using predictive entropy as the uncertainty metric, we select samples with high entropy scores as candidates, *i.e.* $\{x \in D^{train} : u_x > u_{high}\}$. Secondly, as mentioned before, our target is to be more confident with accurate predictions and more uncertain about inaccurate ones. Therefore, samples that are inaccurately predicted with a high confidence *i.e.* low uncertainty (lower than u_{low}) are contrary to our target. These samples are also included as candidates for soft labeling, *i.e.* $\{x \in D^{train} : \hat{y}_x \neq y_x \wedge u_x < u_{low}\}$. The entropy thresholds u_{low} and u_{high} are hyperparameters that can be tuned empirically.

B. Dual-action reinforcement learning

The core of our method is a reinforcement learning network g that learns to select the most uncertain samples and assign soft labels for training the prediction model f as depicted in Fig. 3. The process is iterated until f maximizes its performance on the validation set.

The reinforcement learning steps (in Alg. 1) cast in an MDP formulation (s_t, a_t, r_t, s_{t+1}) are detailed at each iteration t as: 1) Compute the state s_t for each candidate sample, which characterizes the sample's ambiguous features, uncertainty and other information. 2) Evaluate the state s_t with Actor network μ to generate dual-action (a_t^S, a_t^L) for all candidates and assign the learned soft labels a_t^L to samples whose selection action $a_t^S = 1$ 3) Re-train the prediction model f_{t-1} on the updated training set where selected samples are re-labeled with a_t^L and obtain f_t. 4) Update the state to s_{t+1} based on f_t. 5) Compute the reward r_t as the performance change between f_t and f_{t-1} evaluated on a held-out validation set D^{val}. In following sections, we will elaborate on the RL algorithm used to solve the MDP problem followed by a detailed definition for state, action and reward.

Deep deterministic policy gradient. We take the Actor-Critic based DDPG [59] as our reinforcement learning algorithm for its continuous action space and stable performance. To evaluate action a_t, the Critic $Q(s_t, a_t|\phi^Q)$ produces a Q-value by taking as input the state-action pair (s_t, a_t). Target networks Q' and μ' are used to update the Critic by computing a target Q-value y_t with Eq. (2) then minimizing a temporal difference (TD) loss in Eq. (3).

$$y_t = r_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1}|\phi^{\mu'})|\phi^{Q'}) \tag{2}$$

$$L_{Critic} = \frac{1}{N}\sum_t (y_t - Q(s_t, a_t|\phi^Q))^2 \tag{3}$$

Algorithm 1 Dual-action reinforced learning

Input: behaviour prediction model f_{init}, training set D^{train}, validation set D^{val}, Actor μ_{init}, Critic Q_{init}
Output: Updated prediction model f_T
Function:
def UPDATE(f, D):
  return trained f on D
def COMPUTE(D^{cand}, D^C, f):
  $sim_{cross} = hist(D_x^{cand} \cdot D_{cross}^C)$
  $sim_{nocross} = hist(D_x^{cand} \cdot D_{noross}^C)$
  pred_entropy = Eq. (1)
  s_t = concat(sim_{cross}, $sim_{nocross}$, pred_entropy, gt_label, progress_level)
  return s_t for each candidate sample x
Init: $f_0 \leftarrow$ UPDATE(f_{init}, D^{train}); $D^{cand}, D^C \leftarrow$ SPLIT (D^{train}); $\mu_0 \leftarrow$COLDSTART(μ_{init}); $Q_0 \leftarrow Q_{init}$
for each episode **do**
  for each t **in** $(0, T-1)$ **do**
    $s_t \leftarrow$ COMPUTE(D^{cand}, D^C, f_{t-1})
    $a_t^S, a_t^L \leftarrow \mu_t(s_t)$ for each sample x
    Update D_t^U, D_t^C, f_t and s_{t+1}:
      $D_t^U \leftarrow$ samples whose $a_t^S = 1$; re-label with a_t^L
      $D_t^C \leftarrow$ augment D^C with samples whose $a_t^S = 0$
      $f_t \leftarrow$ UPDATE($f_{t-1}, (D_t^U, D_t^C)$)
      $s_{t+1} \leftarrow$ COMPUTE(D^{cand}, D^C, f_t)
    Compute reward on D^{val}:
      $r_t = Score(f_t) - Score(f_{t-1})$
    Push transition (s_t, a_t, r_t, s_{t+1}) to Replay Buffer R
    Update μ_t and Q_t following Eq. (3) and Eq. (4)
  end
end

N in Eq. (3) is the size of the mini-batch sampled from the replay buffer R and target Q-value y_t in Eq. (2) is calculated as the sum of the reward signal r_t and the output Q' of the target Critic network multiplied by a discount factor γ. Q' is derived from the next state s_{t+1} and next action $\mu'(s_{t+1}|\phi^{\mu'})$ by the target Actor network.

The Actor network is then updated by taking as loss the negative Q-value in Eq. (4), which drives the policy learning to select the most uncertain samples for soft labeling.

$$L_{Actor} = -Q(s_t, a_t|\phi^Q) \tag{4}$$

For updating parameters of target networks Q' and μ', we use Eqs. (5) and (6) *i.e.* weighted sum of the updated network parameters ϕ^Q and ϕ^μ and the targets' last parameters. Following [59], we put more weight on the target itself *i.e.* $\tau \ll 1$.

$$\phi^{Q'} = \tau\phi^Q + (1-\tau)\phi^{Q'} \tag{5}$$

$$\phi^{\mu'} = \tau\phi^\mu + (1-\tau)\phi^{\mu'} \tag{6}$$

State. The state s_t for each sample x in the candidate set D^{cand} serves as input to the policy network *i.e.* the Actor,

which generates dual-action output. To learn useful sample selection and soft labeling actions, we formulate four state features: 1) the similarity of x in hidden features with the respective *crossing* and *not crossing* samples from the certain set D^C; 2) the predictive entropy with respect to the prediction model. 3) the original one-hot label; 4) the progress level *i.e*. temporal position of the subsequence; These four features need to support sample selection a^S and provide cues for soft labeling a^L.

The first state we choose is based on the requirement to capture the most representative information of sample x in discriminating the *crossing* and *not crossing* features. Instead of directly using hidden features before the classification layer of the prediction model, which has been verified in experiments not a useful indicator for class discrimination, we choose to use the closeness in hidden features of sample x to the respective *crossing* and *not crossing* samples from the certain set D^C. As the certain set D^C are guaranteed to have the most representative *crossing* and *not crossing* features, comparing candidate sample x to the respective classes of D^C can straight away tell the bias in hidden features for x. Specifically, we compute the cosine similarity between x and individual samples in D^C and group the cosine similarities by class. To obtain a compact representation, instead of taking the average for all similarity values for each class, we compute a histogram of similarities, which is more informative. For instance, a right skewed distribution indicates that x is closer in hidden features to the particular class it is compared with and vice versa. The similarity state feature makes a significant contribution to soft label learning since it's directly dealing with motion features. It also incorporates the important features of the certain set D^C by correlating the hidden features of candidate samples with the certain set.

The second state feature is the predictive entropy. As discussed in Sec. III-A, predictive entropy is a good indicator of uncertainty in the prediction model output. The rationale behind including predictive entropy in the state representation is to provide a measure of uncertainty in the prediction model, which is crucial for selecting uncertain samples for soft labeling. According to Eq. (1), the maximum entropy is 1 when the predicted probability is 0.5, indicating complete uncertainty in the prediction. Towards the extremes (0 or 1), the entropy decreases to lowest, implying the model is certain about the prediction. Predictive entropy can guide the agent to produce soft labels that accounts for the uncertainty in the prediction model (*e.g*. if the model produces a high entropy, the RL agent can then learn a soft label that is more evenly distributed across classes). Predictive entropy also encourages exploratory actions which the prediction model is less confident about. The experience from these exploratory actions can potentially improve the quality of the soft labels learned.

The third state feature is the original one-hot label. Together with the similarity feature, one-hot labels can assist in identification of noisy samples that either have a disagreement between motion features and the one-hot label or exhibit hard features. For instance, if the hidden features of a candidate sample x whose one-hot label is *crossing* appears more similar to the *not crossing* samples in the certain set D^C, sample x may be having an inaccurate one-hot label. If x doesn't show an evident bias in hidden features to either class, it's likely to have ambiguous features. Both cases should be selected for soft labeling.

We take candidate sample x's progress level as the final state feature. The position of subsequence x in the entire observation time of a pedestrian reflects his uncertainty in action. Intuitively, as the observation draws closer to a pedestrian's final action (which defines the one-hot label), it should be more obvious to conclude his crossing intention.

Action. In this paper, we define a dual-action game in the reinforcement learning setting. The first action a^S is used to select uncertain samples from candidate set D^{cand}; and the second action a^L is used to provide soft labels. a^S is a binary value obtained by thresholding a sigmoid output; and a^L is a continuous real-valued output $\in (0, 1)$ produced by sigmoid. Initially, we consider two ways to learn the actions, *i.e*. sequentially and simultaneously. In sequential manner, we are faced with selecting inappropriate samples, and regardless of the soft labels assigned, they are not useful for training. Therefore, we choose to learn both selection and soft labeling actions simultaneously and through re-training the prediction model with both actions applied, we are able to identify the most appropriate action pairs guided by the reward signal.

At the beginning of each RL episode, we initialize the environment by using the same candidate set D^{cand} and original weights for the prediction model to ensure stability and same initial states. As we step through an episode, we practice Alg. 1, to produce two actions a_t^S and a_t^L at t-th iteration for each sample in the candidate set. We introduce more drastic changes to the actions by applying exploration noise at random times (a tunable hyperparameter). In conclusion, we ensure stability and exploration at the same time during training of the RL network.

Reward. We use the reward as feedback to the RL network to guide its dual-action learning. Since our goal is to enhance performance of the prediction model, we use accuracy and f1 score to achieve balanced performance for both positive and negative classes while keeping a high overall accuracy. The dataset used for reward calculation is the validation set following data split rules from the PIE [20] and JAAD [63] datasets respectively. The reward is calculated as the difference in the sum of accuracy and f1 score between f_t and f_{t-1}. Note that the validation set is not involved in any training process. We pass the performance change of the prediction network to the RL network as a reward signal which the latter tries to maximize by generating more promising actions to further boost the former's prediction performance.

C. Cold start for the actor network

Due to the dual-action setup and the continuous action for soft labeling, we have a large action space to search from,

TABLE II: Performance of the proposed method on original prediction models on PIE and JAAD$_{beh}$ datasets. Last four rows show our reinforcement learning approach applied to the original models. For modality, I: image, B: bounding box, P: pose, S: vehicle speed.

Method	Modality	PIE			JAAD$_{beh}$		
		acc	auc	f1	acc	auc	f1
C3D [64]	I	0.77	0.67	0.52	0.61	0.51	0.75
SF-GRU [9]	I,B,P,S	0.84	0.83	0.72	0.51	0.45	0.63
PCPA [10]	I,B,P,S	0.87	0.86	0.77	0.58	0.5	0.71
VMI [65]	I,B,P,S	0.92	0.91	0.87	0.62	0.53	0.73
Ours (C3D)	I	0.8	0.76	0.65	0.63	0.5	**0.77**
Ours (SF-GRU)	B,P,S	0.88	0.86	0.8	0.63	0.52	0.76
Ours (PCPA)	B,P,S	0.91	0.88	0.83	0.64	0.55	0.76
Ours (VMI)	B,P,S	**0.93**	**0.92**	**0.89**	**0.66**	**0.56**	0.76

and it can be very time-consuming for the RL framework to converge. To be more efficient, we propose to cold start the Actor network in a supervised fashion.

Inspired by state features discussed in Sec. III-B, we create a small dataset for supervised learning of the Actor network. Initially, we choose the small dataset to roughly contain 50% negative and 50% positive samples by ranking the training loss of model f_0 (the bottom 50% low-loss samples are taken as certain set, the next 10% higher-loss samples negative set and top 10% high-loss samples positive set). For the negative set, we assign action $a^S = 0$ and action a^L the same as their one-hot labels. For the positive set, we evaluate their cosine similarity of hidden features with *crossing* and *not crossing* samples from the certain set. We define rules below based on similarity features and one-hot labels to decide on supervised signals. We introduce a distribution measure called skewness to tell whether a sample is more similar to *crossing* or *not crossing* features. A negative skewness means there is more weight in the right tail of the distribution, indicating that more samples agree with the given sample in hidden features than those disagree. Through comparing skewness values in similarity distributions with each class, we can tell the similarity signature for each candidate sample. For a candidate sample whose hidden features are more similar to a particular class, if that class is the same with the sample's one-hot label, we assign for the sample supervised actions $a^S = 0$ and a^L = one-hot label (no change is applied). If the class is different from the given sample's one-hot label, we consider the sample to be uncertain and assign $a^S = 1$. As for a^L, depending on which class the sample is skewed to, we assign a random scalar in the range $(0.5, 1)$ if the sample is biased to *crossing* features and use the range $(0, 0.5)$ for *not crossing*. For all other cases, we assign $a^S = 1$ and a^L to be a random scalar around 0.5.

IV. Experiment

We conduct extensive experiments on two benchmark datasets PIE and JAAD for pedestrian crossing prediction to evaluate the effectiveness of our uncertainty-aware approach. For each dataset, we perform the reinforcement learning steps illustrated in Fig. 3 on three prediction models: C3D [64], SF-GRU [9] and PCPA [10]. We are able to boost the performance of all three models and improve their reliability as elaborated in the following sections.

A. Experimental setup

We follow the same experimental settings with [10] in formulating the pedestrian crossing prediction as a binary classification problem. The objective is to predict whether a pedestrian will start crossing the street at some future time. The prediction relies on observed features including RGB image, bounding box locations, pedestrian pose and ego-vehicle speed. The input modalities used also depend on the model architecture.

Datasets. In this paper, we use two public datasets which are created for studying pedestrian behaviour in traffic: Pedestrian Intention Estimation (PIE) [20] and Joint Attention for Autonomous Driving (JAAD) [63]. PIE contains 6 hours of HD videos recorded in urban environments of Toronto with per-frame behavioural annotations for 1842 pedestrians. PIE also provides bounding box annotations and ego-vehicle information such as speed and heading direction. We follow the same data splits as [10]. Specifically, we take videos from set01, set02 and set04 as training set; for reward calculation, we use the validation set videos from set05 and set06; finally for measuring performance, we use videos from set03 as testing set. JAAD, on the other hand, contains 346 HD videos filmed in North American and Eastern Europe with a focus on pedestrian detection task. Behavioural annotations are also provided but only for 25% of all pedestrians, who are close to the road and will potentially interact with drivers. Further-away pedestrians without behavioural annotations are implicitly considered as not crossing the road. To this end, we use JAAD$_{beh}$ to represent the subset of JAAD with behavioural annotations. In this paper, we only report experimental results on JAAD$_{beh}$ since they are more relevant in the autonomous driving context. Learning soft labels for pedestrians far away from roads doesn't contribute much to the theory this paper is trying to prove. We use the same data splits as [66]: 324, 48 and 276 pedestrian tracks for training, validation and testing set respectively.

To break down a pedestrian track into subsequences, we follow the same observation length, overlap ratio and time-to-event (TTE) configurations from [10]. In particular, we set subsequence length to 16 frames for both datasets. A subsequence is sampled such that its last observed frame falls between 1 and 2 seconds prior the event frame (the frame a pedestrian starts to cross or the third to last frame for not crossing cases) and using an overlap ratio 0.6 for PIE and 0.8 for JAAD. This results in 6 and 11 subsequences per pedestrian track for PIE and JAAD respectively.

Implementation details. For model training, we take different input features for C3D, SF-GRU and PCPA. As discussed in a few studies [9], [23], [71] on pedestrian crossing prediction, the input features have various choices and combinations out of which visual features, pedestrian bounding box & pose, ego-vehicle speed are commonly used. In this paper,

TABLE III: Performance of the proposed method compared with state-of-the-art methods on PIE and JAAD$_{beh}$ datasets. Last row shows our best result which is applied to VMI model.

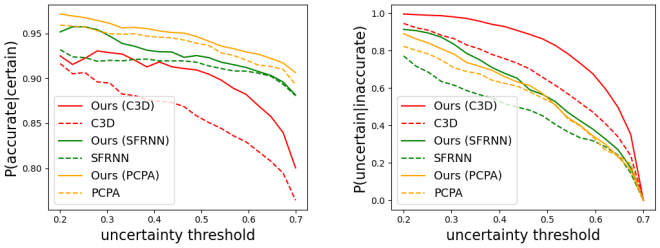
Method	PIE			JAAD$_{beh}$		
	acc	auc	f1	acc	auc	f1
BiPed [67]	0.91	0.9	0.85	-	-	-
MMH [68]	0.89	0.88	0.81	-	-	-
STA [69]	-	-	-	0.62	0.54	0.74
IntFormer [70]	0.89	**0.92**	0.81	0.59	0.54	0.69
CIA [13]	0.84	0.88	**0.9**	-	-	-
ATBB [14]	0.91	0.91	0.83	-	-	-
Ours (VMI)	**0.93**	**0.92**	0.89	**0.66**	**0.56**	**0.76**

considering training with reinforcement learning is more time-consuming since each step in an episode requires re-training of the prediction model, we decide to reduce the complexity by removing visual features for the SF-GRU and PCPA models. Though fewer input modalities are used, we prove our uncertainty-aware approach can boost both models compared to the originally trained models with visual inputs. While for C3D, we keep the same visual inputs since the network is intended to process RGB frames.

For candidates selection (first step in Fig. 3), we empirically set the thresholds for predictive entropy u_{low} and u_{high} as 0.2 and 0.9 respectively, for both PIE and JAAD$_{beh}$ datasets. These thresholds are chosen to ensure a balanced number of samples for soft labeling and to prevent the over-selection of samples with low uncertainty. We include ablation studies in Tab. V and Tab. VI to show the effects of different values of u_{low} and u_{high}. We take the entropy and accuracy from training a randomly initialized model f_{init} by using the same configurations as the original papers.

For all models, we extract the feature map just before the fully connected classification layer as hidden features, in particular a 256-D feature vector for SF-GRU and PCPA and 4096-D for C3D, to compute the cosine similarity for states. The two state features *i.e.* one-hot label and progress level are directly obtained from annotations. Predictive entropy is obtained from training the model f_{t-1}, for individual candidate sample.

We compute a 16-bin histogram for similarities to the respective *crossing* and *not crossing* hidden features per candidate sample. For both Actor and Critic networks, we take as input the two 16-D vectors and process with batch normalization followed by 1D convolution to output two 1-D similarity values. We then concatenate the two outputs and rest of the state features for the Actor and additional dual-action values produced by the Actor for the Critic (illustrated in Fig. 3). Finally we pass the concatenated features to three fully connected layers containing 64, 32, 16 hidden neurons respectively to yield two action outputs for the Actor and one output for Critic *i.e.* the Q-value. We start with a randomly initialized Critic network Q_0 and a cold started Actor network μ_0 described in Sec. III-C and train with a learning rate $\alpha = 10^{-5}$ for the Actor and $\beta = 2.0 \times 10^{-5}$ for the Critic. The discount factor γ in Eq. (2) is set to 0.99 We sample 128-sized mini batch from the Replay Buffer of capacity 10^5 for the learning of Actor and Critic networks.



(a) Accurate when certain (b) Uncertain given inaccurate

Fig. 4: Uncertainty evaluation on prediction models. The higher the metrics, the more reliable the model is.

For training, we set $T = 10$ steps for the agent to explore action space in one game episode. We use 50 episodes which is sufficiently high for training the Actor and Critic networks to convergence. In each step, we re-train the prediction model with the respective hyperparameters specified in each model, but we reduce the epoch size to 10 for faster feedback. We use standard metrics for classification problems: accuracy, auc and f1 to measure the performance of pedestrian crossing prediction.

Results. Tab. II shows the performance of our method on all four prediction models and in Tab. III we compare with other state-of-the-art methods on both PIE and JAAD$_{beh}$ datasets. Our results are obtained by training the model with the updated soft labels which yield the best performance on validation set D^{val}. The results from the original C3D, SF-GRU, PCPA and VMI methods trained with one-hot labels are reported based on [9], [10], [65]. Our method significantly boosts the performance on the original models by a gain of 3%, 4%, 3% in accuracy for C3D, SF-GRU and PCPA respectively on the PIE dataset. For JAAD$_{beh}$, the gains are 2%, 12%, 6% for C3D, SF-GRU, PCPA respectively. It's worth noting that improvements in f1 score (13%, 8%, 6% for PIE and 2%, 13%, 5% for JAAD$_{beh}$) are more significant than accuracy especially for PIE, which shows our method can better handle the imbalanced dataset (*not crossing* samples are 3 times of *crossing* samples for PIE dataset). Though we achieve a smaller gain in accuracy (1%) and f1 score (2%) for VMI on the PIE dataset, we believe it's more challenging to further improve an already strong model such as VMI (0.92 in accuracy and 0.87 in f1). For the JAAD$_{beh}$ dataset, we achieve significant improvement in accuracy (4%) and f1 score (3%) for VMI since the space for improvement is larger. It's worth noting that we use fewer input modalities for SF-GRU, PCPA and VMI models while achieving better performance. This further demonstrates the effectiveness of our method.

In Tab. III, we also compare our results with other methods *i.e.* BiPed [67], Multi-Model Hybrid [68], Spatio-Temporal Attention [69], IntFormer [70], Coupling Intent & Action [13] and Attention-to-Bounding-Box [14]. Due to the lack of public code, we are unable to reproduce the results of these models and cannot verify the boosting effects our method could bring to them. Our method achieves better and on par performance in accuracy and AUC on the PIE dataset and performs the best on the JAAD$_{beh}$ dataset across all metrics. Even though we don't

TABLE IV: Ablation study (for PCPA model trained on PIE dataset) on the effects of different state features. We compare the model performance by removal of each state feature. Abbreviations - sim: similarity with crossing & not crossing hidden features, oh: one-hot label, pl: progressive level, en: entropy.

State features	acc	auc	f1
oh+pl+en	0.87	0.86	0.79
sim+oh+pl	0.88	0.87	0.81
sim+pl+en	0.89	0.86	0.80
sim+oh+en	0.90	0.87	0.82
sim+oh+pl+en	**0.91**	**0.88**	**0.83**

TABLE V: Ablation study on u_{low} and u_{high} for dataset PIE trained on PCPA model.

u_low, u_high	acc	auc	f1
0.1, 0.9	0.91	0.86	0.79
0.1, 0.8	0.88	0.87	0.82
0.2, 0.8	0.89	0.88	0.83
0.2, 0.9	**0.91**	**0.88**	**0.83**

achieve the best f1 score (which is from CIA [13]) on the PIE dataset, we significantly outperform CIA in terms of accuracy (0.93 vs 0.84) and AUC (0.92 vs 0.88). Therefore, the overall performance of our method is better. In an imbalanced dataset like PIE, it's possible that a model like CIA maintains a better balance between precision and recall which results in a high f1 score. However, our method is focusing on improving the overall performance of the model and making it more reliable by learning soft labels for uncertain samples.

We show the improvement of model reliability over the original prediction models in Fig. 4. We present the plots for two metrics on quality of uncertainty estimates $P(accurate|certain)$ and $P(uncertain|inaccurate)$ as explained in Sec. III-A. It's shown in Fig. 4a that for all three prediction models, our method (solid line) achieves higher accuracy when the model predictions are certain for different uncertainty thresholds. This means we can trust the correctness of those predictions when the model has high confidence in the predictions. Fig. 4b shows higher uncertainty in the inaccurately predicted samples using our method. It suggests that the model is more responsible in its predictions and less likely to make wrong or harmful predictions. The improvement on these two metrics confirms the effectiveness of our uncertainty-aware method. Furthermore, models re-trained with soft labels yield more responsible predictions and are more aware of their limitations on those samples incorrectly predicted than the original models. Out of all incorrectly predicted samples by the re-trained models after applying our method, 82% are having high-entropy (above 0.7) While for the original models, only 56% of the incorrectly predicted samples have high-entropy. This indicates our model is more responsible in making inaccurate predictions than the original models.

TABLE VI: Ablation study on u_{low} and u_{high} for dataset JAAD$_{beh}$ trained on PCPA model.

u_low, u_high	acc	auc	f1
0.1, 0.9	0.64	0.54	0.75
0.1, 0.8	0.62	0.53	0.74
0.2, 0.8	0.63	0.52	0.76
0.2, 0.9	**0.64**	**0.55**	**0.76**

To illustrate the improvement in model reliability, we take the pedestrian in Fig. 1 as an example. The sample pedestrian's one-hot label is 0 (*not crossing*), and we learn a soft label of 0.64 which better represents the observed *crossing* cues of the pedestrian. After re-training the model with the soft labels, the probability of the pedestrian crossing is decreased from 0.9 to 0.64, which corresponds to a higher entropy. This shows that the model re-trained with soft labels is more aware of its limitations on those samples incorrectly predicted than the original model.

On computational performance, compared to other probabilistic methods like Monte Carlo Dropout [18], [43] which require multiple forward passes for each sampled dropout mask; and Bayesian Neural Networks [23] which require drawing samples from posterior distributions of weights, our method is more efficient in directly improving the prediction model's performance and reliability through reinforcement learning without adding extra computational cost during inference.

B. Ablation study

This section presents the ablation study of our method. We take PCPA model trained on the PIE dataset as an example to demonstrate the effects of different settings.

Ablation on state features. Tab. IV summarizes the results with individual feature removed from state s_t. We have the best performance with full state features namely similarity features, one-hot label, predictive entropy and progress level. Removing similarity features from state s_t leads to a big drop in performance since similarities are computed from the last hidden layer of the prediction model thus are directly related to the quality of soft labels learned. That's also the reason we use similarities in cold start of the Actor network. Similarly, predictive entropy contributes significantly to the performance as it indicates the uncertainty of the candidate samples. When removing one-hot label, we see a performance drop because when combined with similarity features, one-hot label assists with indicating mismatch between feature and label to some extent. With removal of progress level, the performance is just slightly worse than the full features. The reason is perhaps the overlapping between subsequences are big for PIE dataset thus not significant for learning soft labels.

Ablation on u_{low} and u_{high}. We also conduct ablation studies on the effects of different values of u_{low} and u_{high} for both PIE and JAAD$_{beh}$ datasets. In Tab. V and Tab. VI, we show the performance of the PCPA model with different combinations of u_{low} and u_{high}. It turns out that when $u_{low} = 0.2$ and $u_{high} = 0.9$ the performance is the best for both datasets.

TABLE VII: Comparison of different reinforcement learning algorithms for dataset PIE trained on PCPA model.

RL algorithm	acc	auc	f1
DQN [56]	0.87	0.85	0.79
Reinforce [58]	0.89	0.86	0.81
DDPG [59]	**0.91**	**0.88**	**0.83**

C. Other reinforcement learning algorithms

We also compare our method with other reinforcement learning algorithms: DQN [56] and Reinforce [58]. Since DQN is not designed for continuous action space, we use a discretized version of the action space for DQN. DQN optimizes the Q-value function by minimizing the temporal difference error which is similar to the Critic network in our method. Reinforce is a policy gradient method which directly optimizes the policy function, which is similar to the Actor network in our method. In Tab. VII, we show the performance of the PCPA model trained with our method and the other two algorithms. We show that our method outperforms the other two algorithms in terms of accuracy, auc and f1 score. This demonstrates the effectiveness of utilizing both Actor and Critic networks of DDPG to learn soft labels for uncertain samples.

V. Conclusion

In this paper we present uncertainty-aware pedestrian crossing prediction by a reinforcement learning approach. We propose a novel state representation and an adaptation of the Actor-Critic framework to select and learn soft labels for uncertain pedestrian samples. By maximizing the reward signal which directly reflects the prediction model's performance, the RL agent is able to produce informative soft labels for the prediction model. We conduct extensive experiments on two pedestrian datasets and demonstrate that our method (even with fewer input modalities) outperforms the original models. Moreover, our method leads to more reliable and trustworthy models for providing higher accuracy when the model is certain and higher uncertainty when the model is inaccurate. The uncertainty estimates enable safer decision-making for autonomous vehicles for being more responsible in complex traffic scenarios.

As future work, we believe our method can be extended to other tasks in computer vision, such as weakly supervised learning. It is possible to apply our uncertainty-aware learning to other tasks such as object detection [72], [73] and action recognition [74], where reinforcement learning agent can be guided by uncertainty level to search for more fine-grained labels given weak labels.

VI. Acknowledgement

This work is supported by Economic Development Board of Singapore (EDB) and Singapore Technologies Engineering Ltd (ST Engineering) under the Industry Postgraduate Program (IPP) Project code RS-EDBIP-00031.

References

[1] S. Mokhtarimousavi, J. C. Anderson, A. Azizinamini, and M. Hadi, "Factors affecting injury severity in vehicle-pedestrian crashes: A day-of-week analysis using random parameter ordered response models and artificial neural networks," *International journal of transportation science and technology*, vol. 9, pp. 100–115, 2020. 1

[2] W. H. Organization, "Global status report on road safety 2023," 2023. [Online]. Available: https://www.who.int/publications/i/item/9789240086517 1

[3] W. Guan, X. Song, K. Wang, H. Wen, H. Ni, Y. Wang, and X. Chang, "Egocentric early action prediction via multimodal transformer-based dual action prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2023. 1

[4] J. Tang, J. Zhang, R. Ding, B. Gu, and J. Yin, "Collaborative multi-dynamic pattern modeling for human motion prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 3689–3700, 2023. 1

[5] Z. Zheng, L. Yang, Y. Wang, M. Zhang, L. He, G. Huang, and F. Li, "Dynamic spatial focus for efficient compressed video action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2023. 1

[6] P. Ding and J. Yin, "Towards more realistic human motion prediction with attention to motion coordination," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 5846–5858, 2022. 1

[7] J. Fu, J. Gao, and C. Xu, "Learning semantic-aware spatial-temporal attention for interpretable action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5213–5224, 2022. 1

[8] A. Tong, C. Tang, and W. Wang, "Semi-supervised action recognition from temporal augmentation using curriculum learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 3, pp. 1305–1319, 2023. 1

[9] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Pedestrian action anticipation using contextual feature fusion in stacked rnns," in *BMVC*, 2019. 1, 2, 6, 7

[10] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Benchmark for evaluating pedestrian action prediction," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1257–1267. 1, 2, 3, 6, 7

[11] K. Saleh, M. Hossny, and S. Nahavandi, "Intent prediction of pedestrians via motion trajectories using stacked recurrent neural networks," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 414–424, 2018. 1, 2

[12] A. Rasouli, T. Yau, M. Rohani, and J. Luo, "Multi-Modal Hybrid Architecture for Pedestrian Action Prediction," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2022, pp. 91–97. 1, 2

[13] Y. Yao, E. M. Atkins, M. Johnson-Roberson, R. Vasudevan, and X. Du, "Coupling intent and action for pedestrian crossing behavior prediction," *ArXiv*, vol. abs/2105.04133, 2021. 1, 7, 8

[14] L. Achaji, J. Moreau, T. Fouqueray, F. Aioun, and F. Charpillet, "Is attention to bounding boxes all you need for pedestrian action prediction?" *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 895–902, 2022. 1, 7

[15] X. Zhai, Z. Hu, D. Yang, L. Zhou, and J. Liu, "Social Aware Multi-Modal Pedestrian Crossing Behavior Prediction." 1, 2

[16] W. Chen, Z. Yang, L. Xue, J. Duan, H. Sun, and N. Zheng, "Multimodal pedestrian trajectory prediction using probabilistic proposal network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 6, pp. 2877–2891, 2023. 1

[17] S. Kerscher, N. Balbierer, S. Kraust, A. Hartmannsgruber, N. Müller, and B. Ludwig, "Intention-based Prediction for Pedestrians and Vehicles in Unstructured Environments:," in *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems*. Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 2018, pp. 307–314. 1, 2

[18] A. Nayak, A. Eskandarian, and Z. Doerzaph, "Uncertainty estimation of pedestrian future trajectory using Bayesian approximation," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 617–630, 2022. 1, 2, 8

[19] K. Chen, X. Song, and X. Ren, "Pedestrian trajectory prediction in heterogeneous traffic using pose keypoints-based convolutional encoder-decoder network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1764–1775, 2021. 1

[20] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, "Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 5, 6

[21] D. Cao and Y. Fu, "Using graph convolutional networks skeleton-based pedestrian intention estimation models for trajectory prediction," in *Journal of Physics: Conference Series*, vol. 1621. IOP Publishing, 2020, p. 012047. 1, 2

[22] Y. Xi, D. Ren, M. Li, Y. Chen, M. Fan, and H. Xia, "Robust trajectory prediction of multiple interacting pedestrians via incremental active learning," in *ICONIP*, 2021. 1

[23] A. Bhattacharyya, M. Fritz, and B. Schiele, "Long-term on-board prediction of people in traffic scenes under uncertainty," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 6, 8

[24] H. Sun, Z. Zhao, Z. Yin, and Z. He, "Reciprocal twin networks for pedestrian motion learning and future path prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1483–1497, 2022. 1

[25] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012. 1

[26] L. Y. Wu, L. Liu, Y. Wang, Z. Zhang, F. Boussaid, M. Bennamoun, and X. Xie, "Learning resolution-adaptive representations for cross-resolution person re-identification," *IEEE Transactions on Image Processing*, vol. 32, pp. 4800–4811, 2023. 1

[27] L. Wu, R. Hong, Y. Wang, and M. Wang, "Cross-entropy adversarial view adaptation for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2081–2092, 2020. 1

[28] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1

[29] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *ACM Comput. Surv.*, vol. 56, no. 1, aug 2023. [Online]. Available: https://doi.org/10.1145/3603618 1

[30] F. Zhang, X. Zhu, and M. Ye, "Fast human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[31] A. D. Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?" *Structural Safety*, vol. 31, no. 2, pp. 105–112, Mar. 2009. 1

[32] A. Katharopoulos and F. Fleuret, "Not all samples are created equal: Deep learning with importance sampling," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 2525–2534. [Online]. Available: https://proceedings.mlr.press/v80/katharopoulos18a.html 1, 3

[33] T. Li, J. Liu, W. Zhang, and L. yu Duan, "Hard-net: Hardness-aware discrimination network for 3d early activity prediction," in *ECCV*, 2020. 1

[34] N. Vyas, S. Saxena, and T. C. Voice, "Learning soft labels via meta learning," *ArXiv*, vol. abs/2009.09496, 2020. 1

[35] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W.-H. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *NeurIPS*, 2018. 1

[36] H. Wei, L. Feng, X. Chen, and B. An, "Combating noisy labels by agreement: A joint training method with co-regularization," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 723–13 732, 2020. 1

[37] X. Xia, T. Liu, B. Han, M. Gong, J. Yu, G. Niu, and M. Sugiyama, "Sample selection with uncertainty of losses for learning with noisy labels," *ArXiv*, vol. abs/2106.00445, 2022. 1

[38] O. Ghori, R. Mackowiak, M. Bautista, N. Beuter, L. Drumond, F. Diego, and B. Ommer, "Learning to forecast pedestrian intention from pose dynamics," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1277–1284. 2

[39] S. A. Bouhsain, S. Saadatnejad, and A. Alahi, "Pedestrian intention prediction: A multi-task perspective," *arXiv preprint arXiv:2010.10270*, 2020. 2

[40] D. A. Ridel, N. Deo, D. Wolf, and M. Trivedi, "Understanding pedestrian-vehicle interactions with vehicle mounted vision: An lstm model and empirical analysis," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 913–918. 2

[41] A. Rasouli, M. Rohani, and J. Luo, "Pedestrian behavior prediction via multitask learning and categorical interaction modeling," *arXiv preprint arXiv:2012.03298*, vol. 2, 2020. 2

[42] D. Yang, H. Zhang, E. Yurtsever, K. A. Redmill, and Ü. Özgüner, "Predicting pedestrian crossing intention with feature fusion and spatio-temporal attention," *CoRR*, vol. abs/2104.05485, 2021. [Online]. Available: https://arxiv.org/abs/2104.05485 2

[43] M. Upreti, J. Ramesh, C. R. Kumar, B. Chakraborty, V. Balisavira, P. Czech, V. Kaiser, and M. Roth, "Uncertainty and Traffic Light Aware Pedestrian Crossing Intention Prediction," Feb. 2023. 2, 8

[44] Z. Cao, H. Gao, K. Mangalam, Q.-Z. Cai, M. Vo, and J. Malik, "Long-Term Human Motion Prediction with Scene Context," in *Computer Vision – ECCV 2020*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 387–404. 2

[45] A. Díaz Berenguer, M. Alioscha-Perez, M. C. Oveneke, and H. Sahli, "Context-aware human trajectories prediction via latent variational model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1876–1889, 2021. 2

[46] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, J. Lai, and J. Zhang, "Early action prediction by soft regression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2568–2583, 2019. 2

[47] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, and J. Lai, "Real-time rgb-d activity prediction by soft regression," in *European Conference on Computer Vision*. Springer, 2016, pp. 280–296. 2

[48] N. Hu, Z. Lou, G. Englebienne, B. J. Kröse *et al.*, "Learning to recognize human activities from soft labeled data." in *Robotics: Science and Systems*, 2014. 2

[49] M. Zhou, R. Wang, C. Xie, L. Liu, R. Li, F. Wang, and D. Li, "Reinforcenet: A reinforcement learning embedded object detection framework with region selection network," *Neurocomputing*, vol. 443, pp. 369–379, 2021. 2

[50] S. Mathe, A. Pirinen, and C. Sminchisescu, "Reinforcement learning for visual object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2894–2902, 2016. 2

[51] S. Nayak and B. Ravindran, "Reinforcement learning for improving object detection," in *ECCV Workshops*, 2020. 2

[52] B. Uzkent, C. Yeh, and S. Ermon, "Efficient object detection in large images using deep reinforcement learning," *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1813–1822, 2020. 2

[53] L. Chen, J. Lu, Z. Song, and J. Zhou, "Part-activated deep reinforcement learning for action prediction," in *ECCV*, 2018. 2

[54] A. Casanova, P. H. O. Pinheiro, N. Rostamzadeh, and C. J. Pal, "Reinforced active learning for image segmentation," *ArXiv*, vol. abs/2002.06583, 2020. 2, 3, 4

[55] J. Gong, Z. Fan, Q. Ke, H. Rahmani, and J. Liu, "Meta agent teaming active learning for pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 079–11 089. 2, 3

[56] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *ArXiv*, vol. abs/1312.5602, 2013. 2, 9

[57] A. Padmakumar, P. Stone, and R. J. Mooney, "Learning a policy for opportunistic active learning," in *EMNLP*, 2018. 2

[58] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 2004. 2, 9

[59] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. M. O. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *CoRR*, vol. abs/1509.02971, 2016. 2, 4, 9

[60] J. Mukhoti and Y. Gal, "Evaluating Bayesian Deep Learning Methods for Semantic Segmentation," Mar. 2019. 3

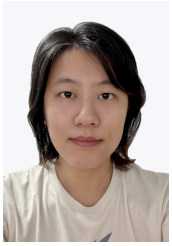
[61] Y. Yang and M. Loog, "Active learning using uncertainty information," 2017. 4

[62] E. Lughofer and M. Pratama, "Online active learning in data stream regression using uncertainty sampling based on evolving generalized fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp. 292–309, 2018. 4

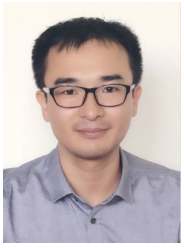
[63] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 206–213. 5, 6

[64] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497. 6

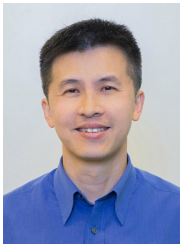
[65] N. Sharma, C. Dhiman, and S. Indu, "Visual–motion–interaction-guided pedestrian intention prediction framework," *IEEE Sensors Journal*, vol. 23, no. 22, pp. 27 540–27 548, 2023. 6, 7
[66] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "It's not all about size: On the role of data properties in pedestrian detection," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0. 6
[67] A. Rasouli, M. Rohani, and J. Luo, "Bifold and semantic reasoning for pedestrian behavior prediction," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 580–15 590. 7
[68] A. Rasouli, T. Yau, M. Rohani, and J. Luo, "Multi-modal hybrid architecture for pedestrian action prediction," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 91–97. 7
[69] D. Yang, H. Zhang, E. Yurtsever, K. A. Redmill, and U. Ozguner, "Predicting pedestrian crossing intention with feature fusion and spatio-temporal attention," *IEEE Transactions on Intelligent Vehicles*, vol. 7, pp. 221–230, 2022. 7
[70] J. Lorenzo, I. Parra, and M. Á. Sotelo, "Intformer: Predicting pedestrian intention with the aid of the transformer architecture," *ArXiv*, vol. abs/2105.08647, 2021. 7
[71] J. Lorenzo, I. Parra, and M. Sotelo, "Intformer: Predicting pedestrian intention with the aid of the transformer architecture," *arXiv preprint arXiv:2105.08647*, 2021. 6
[72] D. Zhang, G. Guo, W. Zeng, L. Li, and J. Han, "Generalized weakly supervised object localization," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022. 9
[73] D. Zhang, J. Han, L. Zhao, and T. Zhao, "From discriminant to complete: Reinforcement searching-agent learning for weakly supervised object detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5549–5560, 2020. 9
[74] T. Zhao, J. Han, L. Yang, and D. Zhang, "Equivalent classification mapping for weakly supervised temporal action localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3019–3031, 2023. 9



Siyang Dai received her B.Eng degree in Electrical and Electronics Engineering from Nanyang Technological University (NTU), Singapore, in 2013 and M.S. degree in Computer Control and Automation from Nanyang Technological University (NTU), Singapore, in 2017. From 2013-2018, she was a software system test engineer with Continental Automotive Singapore. Since 2018, she has been working as a simulation software engineer at ST Engineering Autonomous Solutions on autonomous vehicle simulation tools development. She is currently pursuing her Ph.D. degree at the Singapore University of Technology and Design (SUTD) in the Information Systems Technology and Design (ISTD) pillar under the Industrial Postgraduate Program (IPP), supported by the Economic Development Board (EDB) of Singapore. Her current research interests include autonomous vehicle, computer vision and machine learning.



Jun Liu (Senior Member, IEEE) received the PhD degree from Nanyang Technological University, the MSC degree from Fudan University, and the BEng degree from Central South University. His research interests include computer vision and artificial intelligence. His works have been published in premier computer vision journals and conferences, including TPAMI, CVPR, ICCV, and ECCV. He is listed in the top 2% scientists worldwide identified by Stanford University in 2021 and 2022. He is an Associate Editor of IEEE Transactions on Image Processing and IEEE Transactions on Biometrics, Behavior, and Identity Science, and a regular Area Chair of ICML, NeurIPS, ICLR, CVPR, and WACV, etc.



Ngai-Man Cheung (Senior Member, IEEE) received the Ph.D. in electrical engineering from the University of Southern California, Los Angeles, CA, USA. He is currently an Associate Professor with the Singapore University of Technology and Design. His research interests include image and signal processing, computer vision and AI.