In ICCV 2021

Where are you heading? Dynamic Trajectory Prediction with Expert Goal Examples

He Zhao
York University
zhufl@eecs.yorku.ca

Richard P. Wildes
York University
wildes@cse.yorku.ca

Abstract

Goal-conditioned approaches recently have been found very useful to human trajectory prediction, when adequate goal estimates are provided. Yet, goal inference is difficult in itself and often incurs extra learning effort. We propose to predict pedestrian trajectories via the guidance of goal expertise, which can be obtained with modest expense through a novel goal-search mechanism on already seen training examples. There are three key contributions in our study. First, we devise a framework that exploits nearest examples for high-quality goal position inquiry. This approach naturally considers multi-modality, physical constraints, compatibility with existing methods and is nonparametric; it therefore does not require additional learning effort typical in goal inference. Second, we present an end-to-end trajectory predictor that can efficiently associate goal retrievals to past motion information and dynamically infer possible future trajectories. Third, with these two novel techniques in hand, we conduct a series of experiments on two broadly explored datasets (SDD and ETH/UCY) and show that our approach surpasses previous state-of-the-art performance by notable margins and reduces the need for additional parameters. Code can be found at our Project Page.

1. Introduction

Video predictive understanding on motion patterns of human or robotic agents is essential to many real-world intelligent systems. Forecasting the future trajectories of pedestrians in crowded scenes is an example of such research and recently has received considerable attention [1, 12, 51, 20, 25]. It studies the ability of artificial vision systems to anticipate the future motion of individuals from current observations and therefore is of importance to a variety of allied areas, including self-driving vehicles, service robots and surveillance systems [38].

Research on modeling pedestrian walking trajectories has evolved from relatively simple physical motion models (*e.g.*, social force [13] or constant velocity [42]) to more

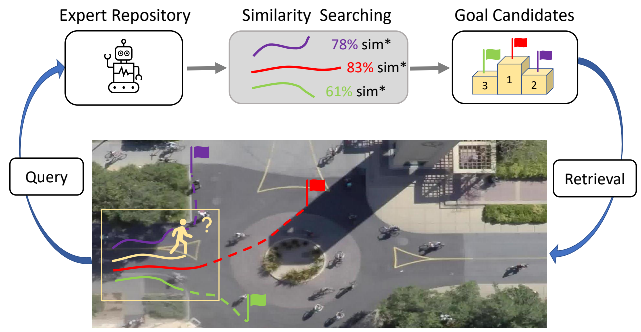

Figure 1: Overview of the Query-Retrieval based framework. A test trajectory with unknown future goal position is matched with expert examples that previously have been seen and stored in an expert repository. Comparisons are based on same observation length test (query) and stored expert example (shown as solid lines) trajectories. This step results in multimodal nearest trajectories with high similarity being retrieved, *i.e.*, those shown in purple, red and green, as well as their goal positions (denoted as colored flags) as potential goals for subsequent full trajectory forecasting.

sophisticated efforts that take into account social compliance [30, 52, 41], enviromental awareness [27, 39, 44, 26] as well as end-goal policies [29, 6]. Recent efforts have found notable performance improvements by encoding goal positions (also dubbed destinations or endpoints) together with historically observed trajectories, with special effectiveness noticed in long-term prediction horizons. These efforts operate essentially in two steps: (i) Inference of goal positions from estimators typically trained in parallel with trajectory estimators; (ii) subsequent trajectory prediction that forecasts unseen movement, conditioned on both the past motion history and the inferred goal information. In nature, this scheme implicitly converts trajectory extrapolation to interpolation (*i.e.*, bridging the pathway between initial trajectories and goal positions).

Goal-based research has been seen in a variety of places, *e.g.*, motion planning [16] and reinforcement learning [17, 32, 8, 9]. These efforts either pre-define the desired goal space with human supervision [7, 33] or leverage a learnable module to obtain that information directly from input, *e.g.*, preliminary states or raw images [32]. The latter is favored by the general trajectory prediction field [29, 6], because typically pedestrians walk through scenes that do not have a priori specified goal positions. However, this choice raises an additional need: Training side models to infer goal positions during testing, which demands extra learnable parameters and goal annotations, if not given by default. Moreover, the learned goals might not be of ideal quality, *e.g*. violating road boundaries or traffic rules.

Contributions. In response to the above challenges, we make three contributions. First, we focus on developing an effective and low-budget approach that automatically explores potential goal positions from a repository of candidate trajectories, namely by making use of expertise based on previous examples, without incurring additional training procedures. Our approach leverages the power of recent advances in data-efficient machine learning, where unlabeled data are self-annotated via metric matching on nearest labelled neighbors. Following this insight, we devise a goal-retrieval algorithm that performs similarity search between partially observed trajectories from a test set and expert examples from a training set, to obtain a small, multi-modal set of candidate goal positions. No previous research has used goal retrieval from an expert repository for trajectory prediction. An overview of our approach to goal retrieval is provided in Fig. 1. Second, we develop a subsequent trajectory predictor that inputs the history of trajectory observations and the queried goal results with a novel low overhead data-shift encoding to jointly infer a diverse, yet accurate set of future trajectories. Third, we conduct extensive experiments showing that our approach surpasses the previous best performance on both the Stanford Drone (SDD) and the ETH/UCY datasets by 15%. Notably, our results are achieved without involving any additional learning components for goal inference. Code is at our Project Page.

Related work. Human trajectory forecasting has seen great recent progress. Exploring the collective dynamics behind a group of walking pedestrians in complex scenes is one of the main focuses in the past few years [1, 12, 3, 30, 52]. Trending methodologies for this purpose include attention [46] and graph neural network [19] frameworks. Meanwhile, modelling the constraints from environments is another direction that has shown solid benefits [39, 24, 27, 41, 44]. Producing multi-modal predictions also has received considerable attention [12, 41, 24]. Major approaches for diversifying outputs include deep generative models [18, 11]. and Gaussian Mixture Models [15, 30]. Our work follows the latter idea to allow for diversity in predicted trajectories

Recently, goal conditioned approaches have shown superior performance over the aforementioned approaches [45, 35, 29, 6]. One such effort models the causal relationship between semantic goals (*e.g*., right-turn or go-straight) and future trajectories [35, 45, 36], while others rely on positional goals (*e.g*., destination coordinates) [29, 6]. Common across these approaches is establishment of a supervised goal estimator to assist later trajectory forecasting. In contrast, while our work exploits goal information, it does so with a novel, nonparametric search-based approach.

Learning from an expert is an established principle. This research direction assumes that a group of representative examples can act as an intelligent system to model versatile real world data. For instance, an earlier effort grouped a set of human walking examples to model crowd trajectories in simulation environments [23]. Some recent work also found it useful to assist multi-modal video frame prediction [49] as well as adaptive robot locomotion generation [50]. Other work has used example extrapolation to remedy data under-representation for robust learning [22].

The intuition of using expert examples also has been used in recent efforts aimed at data efficient learning (*e.g*., one-shot [47], prototype [43] and few-shot [54] learning). Here, research finds that training of intelligent models can depend on only a small amount of annotated examples, as other unobserved data can be self-annotated by matching with adjacent expert examples [2, 10, 47, 43, 48, 22].

Our proposed solution is inspired by techniques seen in expert learning and data efficient machine learning. We apply their insight on use of expert examples to the task of goal conditioned trajectory prediction, with a particular focus on helping the goal inference step. We make use of available trajectory training data to serve as an expert repository that we can index into based on observed test trajectories. Then, the goals of the indexed trajectories are used as input to our full trajectory estimator. We found that running similarity search with a customized dynamic time warping (DTW) [40] metric yields high-quality goal estimations for unseen test trajectories, which further produces superior evaluation results for the overall forecasting. Notably, the searching step can be sped by existing tools [31] to satisfy real-time inference. We are the first to explore a nonparameteric approach to goal inference and show that it leads to state-of-the-art performance in pedestrian trajectory prediction.

2. Technical approach

2.1. Problem formulation

We seek to predict the correct future trajectory of the i^{th} pedestrian in 2D coordinates: $\hat{\mathbf{Y}}_i = \{(\hat{x}_i^t, \hat{y}_i^t) \in \mathbb{R}^2, t = \{t_{obs+1}, ..., t_{end}\}\}$, given M co-existing pedestrians and their observed trajectories $\mathbf{X}_i = \{(x_i^t, y_i^t) \in \mathbb{R}^2, t = \{1, ..., t_{obs}\}\}$ as inputs, where $i \in [1, M]$. More specifically, we assume the predicted coordinates $(\hat{x}^t, \hat{y}^t)$ are random variables that follow a bivariate Gaussian distribution, *i.e*., $(\hat{x}^t, \hat{y}^t) \sim \mathcal{N}(\mu_x, \mu_y, \sigma_x, \sigma_y, \mathrm{corr}_{xy})$, so that diverse outcomes can be sampled to support multi-modality.

Our approach proceeds in the following two steps: First, we query pseudo goal positions $(\hat{x}_i^{t_{end}}, \hat{y}_i^{t_{end}})$ of the testing

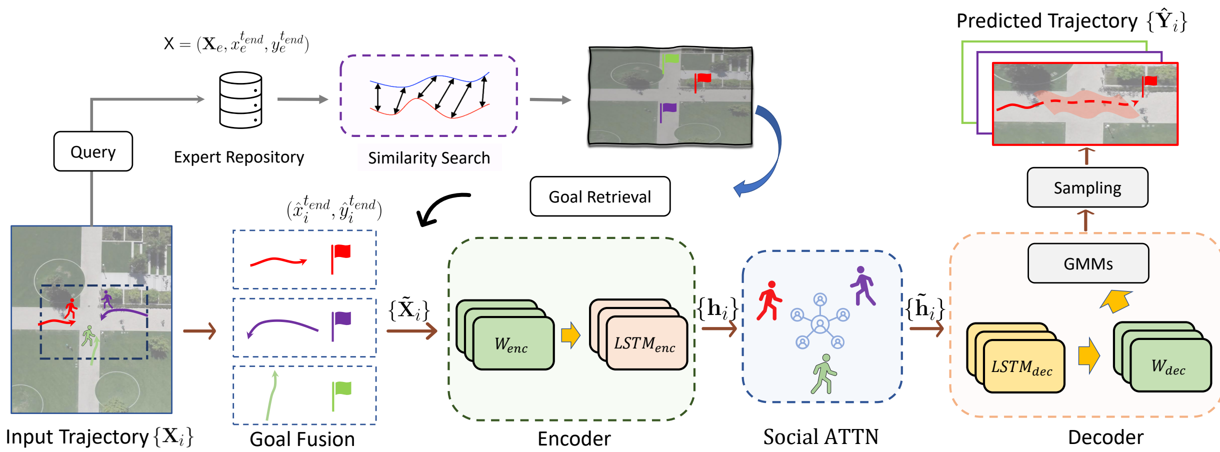

Figure 2: Pipeline of our goal-retrieval based trajectory prediction algorithm. Groups of partially observed trajectories, $\{\mathbf{X}_i\}$, are processed initially by a Query-Search engine operating over an expert repository, X, to produce pseudo goal candidates, $\{\hat{x}_i^{t_{end}}, \hat{y}_i^{t_{end}}\}$ (purple dashed box). Subsequently, the fused information of observed trajectories and estimated goals, $\{\tilde{\mathbf{X}}_i\}$, is forwarded through a sequential encoder (green dashed box) where inputs are embedded in a high dimensional feature, $\{\mathbf{h}_i\}$. Following social attention processing (Social ATTN), a sequential decoder (pink dashed box) recursively predicts a bivariate Gaussian at every time step. Final predicted trajectories, $\{\hat{\mathbf{Y}}_i\}$, are obtained via multiple sampling from the bivariate Gaussian.

input, $\mathbf{X}_i$, through a search in an expert repository of example trajectories, X. Each entry in this repository is comprised of a trajectory sequence, $\mathbf{X_e}$, in the same format as $\mathbf{X}_i$ and its corresponding end positions $(x_e^{t_{end}}, y_e^{t_{end}})$. The end positions of the K_e nearest neighbors of the test trajectory, $\mathbf{X}_i \in$ X, are returned, with K_e the number returned. The repository is built from training data, as detailed in Sec. 3.2. Second, we predict the future trajectory, $\hat{\mathbf{Y}}_i = f(\mathbf{X}_i, \hat{x}_i^{t_{end}}, \hat{y}_i^{t_{end}})$, with $f(\cdot)$ denoting the subsequent trajectory predictor. In the following sections we elaborate how these two steps work in detail. Figure 2 provides a summary of our overall approach.

2.2. Goal retrieval via dynamic time warping

The first component to our approach is a search engine that runs a similarity comparison on testing data and expert examples, *i.e.*, those contained in X. We retrieve useful goal estimates according to

$$\{(\hat{x}_i^{t_{end}}, \hat{y}_i^{t_{end}})\}_{K_e} = \mathcal{S}\left(\underset{\mathbf{X}_e \in \mathrm{X}}{\overset{K_e}{\arg\min}}(\mathcal{D}(\mathbf{X}_i, \mathbf{X}_e))\right), \quad (1)$$

where $\mathcal{D}$ is a distance function between two trajectories, K_e above the $\arg\min$ operator symbolizes that the K_e entries in X that yield the smallest distance are returned and $\mathcal{S}$ selects the end positions of those matches. We select the K_e smallest distance $\mathbf{X}_e$ by calculating the distance between the test trajectory, $\mathbf{X}_i$, and each entry in X, sorting them by distance and taking the K_e with the smallest distance. $\mathcal{S}$ simply selects the end positions associated with each of these trajectories in the repository. In other words, we take the goal position out of the closest K_e expert examples as the pseudo goal for testing data.

For the matching function, $\mathcal{D}(\cdot)$, we find Dynamic Time Warping (DTW) works effectively for our needs. DTW is a well-established approach for measuring the distance between temporal sequences [40]. Originally, it was solved via dynamic programming. Recently, however, it has been relaxed in computational expense, made differentiable and gained in popularity, *e.g.*, [5, 53, 4, 28]. What is particularly interesting to us is its computational efficiency. Specifically, we follow some existing examples [5, 4] to define the matching function γ-Soft-DTW as the following

$$\begin{aligned}\mathcal{D}(\mathbf{X}_i, \mathbf{X}_e) &= \mathrm{DTW}_\gamma(\mathbf{X}_i, \mathbf{X}_e) \\ &= \min{}_\gamma\{\langle A, \Delta(\mathbf{X}_i, \mathbf{X}_e)\rangle, A \in \mathbb{R}^{n\times m}\},\end{aligned} \quad (2)$$

where $\Delta(\cdot)$ is the distance matrix (*e.g.*, Euclidean) measuring element-wise adjacency, A is the alignment matrix that denotes the matching choices and the inner product operator, $\langle\rangle$, yields the similarity score. Here, the soft min, $\min_\gamma$, with $\gamma \geq 0$, is defined as [5]

$$\min{}_\gamma(a_1, ..., a_n) = \begin{cases}\min_{i\leq n} a_i, & \gamma = 0 \\ -\gamma \log\Sigma_{i=1}^n e^{-a_i/\gamma} & \gamma > 0\end{cases} \quad (3)$$

where the a_i represent entries in the distance matrix and γ is a smoothing factor with value set empirically; see Sec. 3.2.

Finally, for better informed matching, we enrich the trajectory descriptors by concatenating their motion information as velocities $(\mathbf{V}_i, \mathbf{V}_e)$, *i.e.* the argument to $\mathcal{D}$ in (1) becomes $(cat(\mathbf{X}_i, \mathbf{V}_i), cat(\mathbf{X}_e, \mathbf{V}_e))$. Thus, similarity considers not only geo-location, but also speed and direction.

Fig. 3 plots goal search results on the evaluated datasets using our approach. It is seen that a large portion of goal retrievals are of high quality, *e.g.*, 83% of test data from the Stanford Drone Dataset [37] (a) yield retrieval error smaller than 10 pixels, amongst which more than half are close to

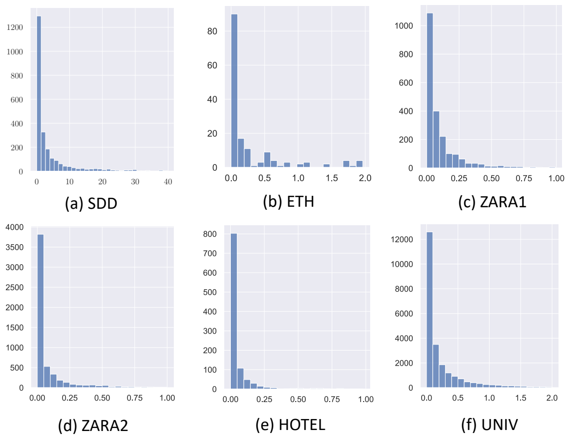

Figure 3: Illustration of goal retrieval quality by plotting the retrieval error (l_2 norm) distribution over test data of all datasets in our experiments (X-axis values closer to 0, the better). These results are achieved via similarity searching across an expert repository using dynamic time warping.

perfection, *i.e.*, $\leq$1 pixel error. Goal searching on another five datasets [23, 34] demonstrates consistently good results (b)-(f). Notably, we are able to achieve this level of performance without the need to learn a model, as the training data serves as its own model in terms of the repository, X. Moreover, our similarity search through the repository can be implemented with modest computational cost; see Sec. 3.5.

Following the protocol for goal conditioned trajectory prediction proposed elsewhere [29], we assess all K_e goal candidates with respect to groundtruth and select the one that provides smallest error. Thus, only a single goal candidate, $(\hat{x}_i^{t_{end}}, \hat{y}_i^{t_{end}})$, is used along with the test trajectory, $\mathbf{X}_i$, as input to the trajectory predictor, as described next.

2.3. Goal conditioned trajectory predictor

We now detail our subsequent trajectory predictor incorporating past observations, $\mathbf{X}_i$, and queried goal positions, $(\hat{x}_i^{t_{end}}, \hat{y}_i^{t_{end}})$, to infer diverse and accurate predictions.

Goal encoding as shifting by goal. Our model handles goal information differently from existing work, where goal positions were concatenated with motion history in high-dimensional feature space [29], or explicitly used to compute the remaining distance as an additional input, *cf*. [6]. Both methods lead to extra embedding efforts.

Instead, we are motivated by the intuition of shifting data according to the mean, adopted by work in machine learning (*e.g.*, batch normalization) and sequence modelling (*e.g.*, temporal substraction for trajectory stationarization in *Trajectron++* [41]), and find it equally sufficient to subtract goal position values from all past motion trajectories before encoding them with Multi-Layer Perceptrons (MLPs). By doing this, we incorporate goal information into feature embedding with zero extra effort; in particular, we define

$$\begin{aligned}\tilde{\mathbf{X}}_i &= \mathbf{X}_i - (\hat{x}_i^{t_{end}}, \hat{y}_i^{t_{end}}) \\ &= \{(x_i^t, y_i^t) - (\hat{x}_i^{t_{end}}, \hat{y}_i^{t_{end}}), t = \{1, ..., t_{obs}\}\},\end{aligned} \tag{4}$$

as our shifted input trajectory and

$$\mathbf{F}_i = \mathrm{W}_{enc}(\tilde{\mathbf{X}}_i) \tag{5}$$

as the shifted encoding. $\mathbf{F}_i$ associates the projected high-dimensional feature of 2D coordinates for every time stamp, *i.e.*, $\mathbf{F}_i \in \mathbb{R}^{D \times t_{obs}}$ and $\mathrm{W}_{enc} \in \mathbb{R}^{2 \times D}$. W_{enc} is realized as a MLP. An ablation study on our choice over concatenating goals with input motion history is provided in Sec. 3.5.

Note that during training, we use the ground-truth goal positions, $(x_i^{t_{end}}, y_i^{t_{end}})$, as input for (4) to prevent the learning process from being disturbed by noisy data, whereas the queried goal positions are used for testing.

Trajectory Prediction. For computing outputs, $\hat{\mathbf{Y}}_i$, given a sequence of input embeddings, $\mathbf{F}_i$, a seq2seq generator implemented as two Long Short-Term Memory (LSTM) units [14] is adopted. Sequence generation proceeds by sequentially encoding and decoding the embedded features, followed by mapping to intermediate results that are used recursively for subsequent prediction according to

$$\mathbf{h}_{enc}^k = \mathrm{LSTM}_{enc}(\mathbf{F}_i^k,\ \mathbf{h}_{enc}^{k-1}),\quad k \in (1, t_{obs}), \tag{6}$$

where $\mathbf{h}_{enc}^k$ is the k^{th} hidden encoder state and the initial hidden state, $\mathbf{h}^0$, is sampled from a normal distribution.

For decoding, another LSTM whose first input is set to the concatenation of the encoded history, $\mathbf{h}_{enc}$, and the last observed coordinates, $\mathbf{X}_i^{t_{obs}}$, is used to produce an output hidden states sequence in a recursive fashion according to

$$\mathbf{h}_{dec}^{k+1} = \mathrm{LSTM}_{dec}\left(\mathrm{cat}(\mathbf{h}_{enc}, \hat{\mathbf{Y}}_i^k), \mathbf{h}_{dec}^k\right), k \in (t_{obs}, t_{end}), \tag{7}$$

where $\hat{\mathbf{Y}}_i^k$ is the next coordinate produced online.

To allow multi-modal forecasting, we set the output to be the parameters of a bivariate Gaussian *cf*. [30, 41]:

$$\mu_x,\ \mu_y,\ \sigma_x,\ \sigma_y,\ \mathrm{corr}_{xy} = \mathrm{W}_{dec}(\mathbf{h}_{dec}^k); \tag{8}$$

$$\hat{\mathbf{v}}_i^k \sim \mathcal{N}(\mu_x, \mu_y, \sigma_x, \sigma_y, \mathrm{corr}_{xy}); \tag{9}$$

$$\hat{\mathbf{Y}}_i^k = \hat{\mathbf{Y}}_i^{k-1} + \hat{\mathbf{v}}_i^k, \tag{10}$$

where W_{dec} is a MLP decoder that projects the decoded LSTM hidden state, $\mathbf{h}_{dec}^k$, to a 5-dimensional vector representing the bivariate Gaussian, $\mathcal{N}(\mu_x, \mu_y, \sigma_x, \sigma_y, \mathrm{corr}_{xy})$. Finally, the full prediction, $\hat{\mathbf{Y}}_i^k$, can be recovered by adding the previous prediction, $\hat{\mathbf{Y}}_i^{k-1}$ and the sampled motion vector, $\hat{\mathbf{v}}_i^k$, according to (9) and (10).

Social Compliance. To consider the collective effect from co-existing pedestrians, we follow recent findings and use an attention mechanism on pedestrians that are near to each other according to a threshold. Within the threshold, neighboring pedestrians, *e.g.*, $(\mathbf{X}_i, \mathbf{X}_j)$, are given a connectivity value, $C_{i,j}$ of 1, otherwise 0, *i.e.*, if $\mathrm{d}(\mathbf{X}_i, \mathbf{X}_j)$ $<$ threshold: $C_{i,j} = 1$; else $C_{i,j} = 0$. We use the l_2

norm as the distance function $d(\cdot)$ and choose thresholds using precedent procedures [29, 41], as detailed in Sec. 3.2. The attention mechanism operates on the last output of LSTM$_{enc}$, here simplified as $\mathbf{h}_i$. In particular, letting

$$e(i,j) = \mathrm{softmax}(\mathrm{W}_\theta(\mathbf{h}_i)\ \mathrm{W}_\phi(\mathbf{h}_j)), \tag{11}$$

the attention weighted output is given as

$$\tilde{\mathbf{h}}_i = \sum_{j \in M} C_{i,j}\ e(i,j)\ \mathrm{W}_g(\mathbf{h}_i), \tag{12}$$

where W_θ and W_ϕ are learned linear transformation matrices on arbitrary pairs of pedestrians prior to normalized weights conversion, $e(i,j)$. Subsequently, a weighted sum operation, (12), is applied on the results of another learned linear transform matrix, W_g, to produce the outputs. This socially attentioned embedding is more informative since it accounts for neighboring agents' motion history as well as their destination plans. We use this output as the input to the trajectory decoder, (7), *i.e.*, $\mathbf{h}_{enc} = \tilde{\mathbf{h}}_i$.

2.4. Learning Scheme

We found it sufficient to train the model end-to-end solely by minimizing the negative log-likelihood of the bivariate Gaussian on all pedestrians and future times,

$$L(\theta) = -\sum_{k=t_{obs+1}}^{t_{pred}} \sum_{i=1}^{M} \log(\mathbf{Y}_i | \mu_x, \mu_y, \sigma_x, \sigma_y, \mathrm{corr}_{xy}), \tag{13}$$

where θ refers to parameters associated with all learnable modules, *i.e.*, W_{enc}, W_{dec}, LSTM$_{enc}$, LSTM$_{dec}$ and attention module weights $\{\mathrm{W}_\phi, \mathrm{W}_\theta, \mathrm{W}_g\}$.

3. Empirical evaluation

3.1. Datasets and evaluation protocol

To evaluate our approach, we choose three widely examined datasets, the Stanford Drone (SDD) [37], ETH [34] and UCY [23] datasets. SDD is a human trajectory prediction dataset that consists of 20 scenes in top down view. We follow the train-test split in the *TrajNet++* challenge [21] and focus on pedestrians. The ETH dataset contains two scenes (ETH and Hotel) and the UCY dataset contains 3 scenes (ZARA1, ZARA2 and UCY). They together consist of 1536 pedestrians. For both datasets, our model takes as input an observation of an eight timestep long trajectory and predicts the trajectory for the next twelve timesteps.

We present prediction accuracy in terms of two well-known metrics, Average Displacement Error (ADE) and Final Displacement Error (FDE), given as

$$\mathrm{ADE} = \frac{\sum_{i=1}^{M}\sum_{k=t_{obs+1}}^{t_{end}} ||\mathbf{Y}_i^k - \hat{\mathbf{Y}}_i^k||_2}{M \times T} \tag{14}$$

and

$$\mathrm{FDE} = \frac{\sum_{i=1}^{M} ||\mathbf{Y}_i^{t_{end}} - \hat{\mathbf{Y}}_i^{t_{end}}||_2}{M}, \tag{15}$$

where M is the number of targets, T is the number of predicted timesteps, $\mathbf{Y}_i^k$ and $\hat{\mathbf{Y}}_i^k$ are the predicted and groundtruth (resp.) positions of target i at time step k and t_{end} is the final predicted timestep.

Goal-based evaluation. Extant protocol for goal-based trajectory prediction assesses an initial set of goal samplings, selects the one closest to the groundtruth final trajectory position and then proceeds to produce midway predictions, *cf*. [29]. We follow the same procedure to evaluate our model, but substitute the goal sampling with our approach to goal retrieval by searching through an expert repository, as detailed in Sec. 2.2. Prior to selection of the single goal candidate passed to the trajectory predictor, the initial set of candidates searched for in the repository is $K_e = 20$, which we found to be effective and efficient, which is validated in the ablation studies; see Sec. 3.5.

Best-of-N Sampling. We report the best ADE and FDE accuracy out of multiple sampled results from our trajectory predictor, using the single selected goal retrieval. In the following evaluations, N is set to 20 for fair comparisons with existing work [1, 12]. We denote this minimizing value as $\mathrm{Min}_{\mathrm{x}}$, *e.g.*, Min_{20} for $N = 20$. Various values, $N \in [5, 10]$, are considered in an ablation study in Sec. 3.5.

3.2. Implementation details

To build our model, we specify that LSTM$_{enc}$ and LSTM$_{dec}$ have hidden states of dimension 128. For the motion history encoder, W_{enc}, we adopt a MLP that consists of sequential activations with shape of $[2 \to 512 \to 256 \to 128]$. A similar MLP that has activations with shape of $[128 \to 64 \to 32 \to 5]$ is used for the bivariate-GMM decoder, W_{dec}. For the attention module, we specify the linear transformation matrices, W_θ and W_ϕ, as two MLPs with the same shape, $[128 \to 256 \to 64]$, and the W_g as the same but with shape $[256 \to 256 \to 128]$. Throughout, the ReLU activation function is used to increase nonlinearity.

For the SDD dataset, training employs the Adam optimizer and a learning rate 0.0003 with $\beta 1$ = 0.9 and $\beta 2$ = 0.99 to minimize the loss (13). The batch size is 512 and the training proceeds 350 epochs. For the ETH and UCY datasets, the same optimizer is adopted to train the model for 250 epochs, with a batch size of 128. The learning rate is initialized as 0.01 for the first 150 epochs, which decays to 0.002 for the rest, *cf*. [30].

We build the expert repository, $\mathsf{X} = \{\mathbf{X}_e, x_e^{t_{end}}, y_e^{t_{end}}\}$, with the same training data introduced in Sec. 3.1 for all datasets. We also enrich the repository of the ETH and UCY datasets by rotating all trajectories in a scene over a range angles from 0° to 360° with an interval of 15°. Random rotation is often used as a data augmentation method

Models	Evaluation Metrics (ADE / FDE) on Min$_{20}$					
	ETH	HOTEL	ZARA1	ZARA2	UNIV	AVG
Linear [1]	1.33 / 2.94	0.39 / 0.72	0.62 / 1.21	0.77 / 1.48	0.82 / 1.59	0.79 / 1.59
Social-GAN [12]	0.81 / 1.52	0.72 / 1.61	0.34 / 0.69	0.42 / 0.84	0.60 / 1.26	0.58 / 1.18
SoPhie [39]	0.70 / 1.43	0.76 / 1.67	0.30 / 0.63	0.38 / 0.78	0.54 / 1.24	0.54 / 1.15
Social-STGCNN [30]	0.64 / 1.11	0.49 / 0.85	0.34 / 0.53	0.30 / 0.48	0.44 / 0.79	0.44 / 0.75
Goal-GAN [6]	0.59 / 1.18	0.19 / 0.35	0.43 / 0.87	0.32 / 0.65	0.60 / 1.19	0.43 / 0.85
PECNet [29]	0.54 / 0.87	0.18 / 0.24	0.22 / 0.39	0.17 / 0.30	0.35 / 0.60	0.29 / 0.48
Trajectron++ [41]	0.43 / 0.86	0.12 / 0.19	0.17 / 0.32	0.12 / 0.25	0.22 / **0.43**	0.20 / 0.39
Ours	**0.30 / 0.56**	**0.09 / 0.13**	**0.15 / 0.28**	0.12 / **0.23**	**0.19** / 0.44	**0.17 / 0.33**

Table 1: Evaluation results on the ETH and UCY datasets for next 12 timestep prediction. Numbers are taken from the minimum ADE/FDE of 20 randomly evaluated samples, denoted as Min$_{20}$. Though **Linear** is deterministic, we list it here as a sanity check. Bolded numbers indicate best performance.

in recent work [42, 41], to combat overfitting. We find this augmentation unnecessary for the SDD dataset, which indicates SDD is more balanced. Empirically, we set the γ-Soft-DTW smoothing parameter to $\gamma = 2$. The social attention threshold in Sec. 2.3 is set to 100 pixel distance for SDD and 3 world distance for ETH/UCY, *cf*. [29, 41].

3.3. Overall prediction results

ETH and UCY datasets. Table 1 shows comparative results for our algorithm $vs.$ various alternatives. Ours perform on-par with the previous best method Trajectron++ on the average ADE (*i.e.*, 0.17 $vs.$ 0.20), while further reducing the FDE by 15% on average, with the biggest improvement happening in the ETH subset (*e.g.*, around 30%). We find the lowest absolute displacement error in both ADE and FDE when evaluated on the HOTEL subset, *i.e.*, 0.09/0.13. The overall relative success of our approach can be explained by the discrepancy in data use. Trajectron++ uses the full future trajectory (*i.e.*, more than just goal positions) to learn a latent structure in training. This structure is supposed to implicitly provide future information for testing. In contrast, we go further to use goal information more explicitly in both training and testing. (We further explore the pecularities of the HOTEL dataset in the ablation studies.)

Especially, when compared with two other goal-based methods, *i.e.*, Goal-GAN [6] and PECNet [29], ours has shown to be more effective, likely for two reasons: First, both methods use deep generative models with a fixed prior distribution (standard Gaussian) to approximate the goal distribution. This paradigm has been found suffering from diversity collapse as well as limited sample quality [49]; second, their methods are constrained to modelling the diversity of goal positions, not that of other trajectory points, which naturally lose the ability to cover a diverse set of mid-way trajectories. Instead, ours uses a nonparameteric approach to goal retrieval, which decouples the goal inference from subsequent trajectory sampling, and therefore repriorритizes the sampling on the overall trajectories.

SDD dataset. The evaluation results on this dataset can be viewed in Table 2 (*i.e.*, Ours). Looking especially at the goal based methods (Goal-GAN [6], PECNet [29] and Ours), it is seen that more desirable performance is observed when compared to all others (*e.g.*, graph neural network based EGraph [25], scene image conditioned CGNS [24] and the rest [12, 39]). These results show solid improvement from incorporating goal information into trajectory forecasting. Notably, our approach again achieves best results overall. Similar to the earlier discussion, we can explain these improvements in terms of goal search with respect to an expert repository being more effective than alternatives, which we further document in Sec. 3.5.

To explore further the possible performance of our model, we also show results from full twelve step trajectory sampling given retrieved goals (denoted as ours-$\mathcal{F}$), rather than the standard protocol we report elsewhere, *i.e.* using the goal prediction (or retrieval) results for FDE and then merging them with the first eleven timestep trajectory sampling for ADE. If allowed, our model produces exceptional results on FDE (*e.g.* 9.03 $vs.$ 14.38) through refinement of initial goal estimates. This result suggests that current goal-based evaluation does not adequately consider the power of goal-based estimators to influence final destinations.

Model	ADE	FDE	MC ↑	F ↑
Goal-GAN [6]	0.55	1.03	92.48	89.47
Ours	0.52	0.97	94.67	91.93

Figure 4: Illustration of the feasibility quality of our results on the SDD Hyang4 scene. Most of our goal retrievals (green flags) are reasonably close to GT goals (yellow flags) and our trajectory predictions (green dotted lines) respect road boundaries. See text for definition of metrics.

Metrics	Evaluation Metrics (ADE / FDE) on Min_{20}							
	S-GAN -[12]	Sophie [39]	CGNS [24]	EGraph [25]	Goal-GAN [6]	PECNet [29]	Ours	Ours-$\mathcal{F}$
ADE	27.23	16.27	15.6	13.9	12.20	9.96	**7.69**	7.51
FDE	41.44	29.38	28.2	22.9	22.10	15.88	**14.38**	9.03

Table 2: Evaluation results on the SDD dataset for the next 12 timesteps trajectory prediction. Numbers are taken from the minimum of 20 random evaluated samples, denoted as Min_{20}. $\mathcal{F}$ denotes the result of sampling all next 12 steps given the retrieved goals, to reveal the full power of proposed approach.

Match $\mathcal{D}(\cdot)$	ADE	Time
DTW-Dual.	7.69	10.9ms
DTW-Vel.	7.95	7.1ms
DTW-Geo.	8.68	7.1ms
Euc.-Vel.	8.43	6.2ms
Euc.-Geo.	9.01	6.2ms

(a) Goal search comparison.

Methods	ADE	Param
Goal-Shift	7.69	✗
Goal-Cat	10.74	✓
Goal-Cat2	9.06	✓
Goal-Res	11.43	✓

(b) Goal use comparison.

Table 3: Ablation studies for accuracy and search speed *vs*. matching function as well as accuracy *vs*. goal-encoding on SDD. See text for details.

3.4. Feasibility of expert examples

To further validate our approach, we provide additional comparisons using the feasibility evaluation protocol [6] on the SDD Hyang-4 scene; see Fig. 4 for results. Notably, two extra metrics are designed for this purpose: mode-coverage (MC) that measures the portion of goal predictions (or goal retrievals in our approach) that are distant to ground-truth goals up to $2m$ (red dotted circle) and, F, denoting the ratio of trajectories lying inside the feasible area (manually segmented road boundary). Without using any goal learning, our results outperform Goal-GAN [6]. We attribute this to expert goal examples respecting environmental constraints, *e.g*. staying on walkways. Details are in the Supplement.

3.5. Ablation studies

Goal search efficiency. Our search engine runs in real-time, thanks to three main factors: First, relaxed soft dynamic time warping that can be computed with CUDA acceleration [5]; second, fast search for the K_e nearest neighbors to a test trajectory in the expert repository [31]; third, the searched data entity is of low dimensionality, *i.e*., each entry is a concatenation of positions and velocities of an eight timestep trajectory. Therefore, each testing entry would cost about 10ms to grab the nearest 20 goal examples. A thorough study of other matching options and their efficiency is provided in Table 3a. Geo., Vel. and Euc. denote geo-locations, velocity and Euclidean, resp. Our proposed approach is denoted DTW-Dual.

Use of goal information. Given that existing work has turned to different strategies for employing goal information, we conduct experiments to systematically validate them. In particular, we study four goal use strategies: Our proposed Goal-Shift (Eq. 4) that subtracts goal positions from input trajectories; Goal-Cat that concatenates goals with raw inputs before encoding; Goal-Cat2 that concatenates encoded goals and inputs in feature space, *cf*. [29]; finally, Goal-Res that concatenates the ongoing prediction and its residual distance to the goal, *cf*. [6]. Results are listed in Table 3b. Check mark indicates that the approach incurs extra parameters. We find that the simplest strategy, *i.e*., Goal-shift, produces the best ADE / FDE and that is what we use for all results reported elsewhere in this paper.

Table 4 shows results of using our predictor without conditioning on goal information. Comparison to results from the full approach (Tables 1 & 2) shows considerable benefit of goal conditioning. Table 4 also shows results when rather than invoking our predictor, trajectory prediction is based simply on the next twelve timesteps of the best matched eight timestep trajectory in the expert repository. Again, it is seen that the full approach provides much better results.

Number of samples. Table 5 has results of different combinations of K_e and N, which always sum to 20, as previous approaches typically rely on a total of 20 samples. We find that a good balance between goal candidates and trajectory prediction samples, (*e.g*., $K_e = 12$, $N = 8$), excels on the ADE; yet, the larger the K_e, the lower the FDE.

Table 6 further shows more generally that while more goal samples yields better results, in most cases there are diminishing returns beyond 20 samples. We also see that the retrieved goal results most favor the HOTEL subset amongst the five, *i.e*., smallest displacement error with groundtruth goals against goal candidate at all levels. This may be explained by a greater portion of its trajectories being linear, *cf*. results of **Linear** in Table 1.

Repository size. Table 7 shows accuracy results as the size of the repository is reduced systematically, *i.e*. few-shot goal retrieval. It is seen that there is only a gradual fall-off in accuracy as fewer entries are made available.

Compatibility. As another comparison, for the single-shot trajectory prediction setup (*i.e*., N=1) in the right-most column of Table 5, we insert our goal retrieval results into the pretrained trajectory predictor of PECNet [29]; we choose that predictor module as it is trained intentionally for deterministic prediction. We see that our goals bring instant improvement, without any modification on either side. This result reaffirms our goal retrieval quality, *e.g*. as shown in Fig. 3, as well as that our goal retrieval module is readily compatible with other approaches.

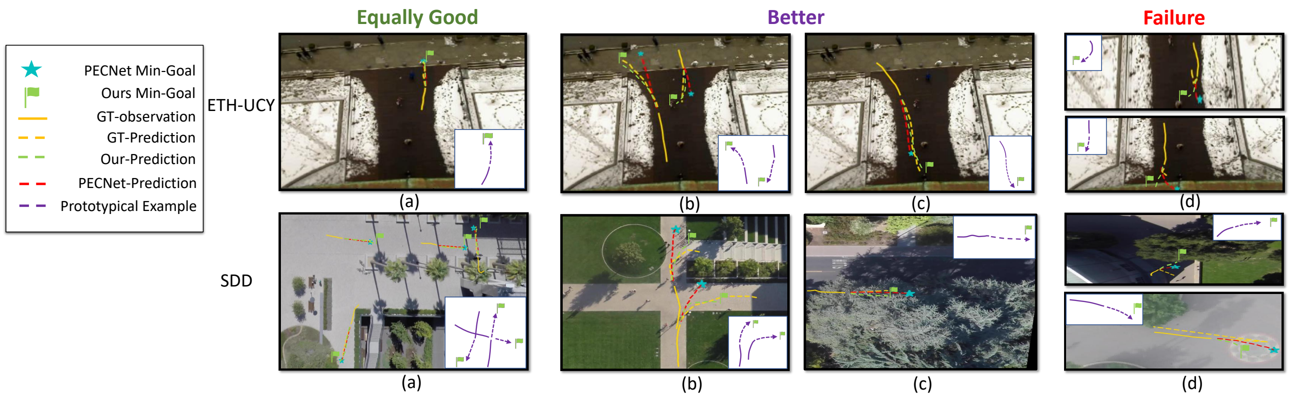

Figure 5: Plotting of evaluation results from the proposed method, ground-truth and a comparison work (PECNet). Top row shows studied cases for the ETH test set and bottom row shows that for the SDD test set. Each column is noted with its comparative category. From left to right: both equally good; ours better; and both failure. We found no cases where the comparison approach was noticeably better than ours. The retrieved best prototypical examples are shown as inset boxes to illustrate why goal examples are helpful (or misleading). Our goal retrievals always follow physical constraints and are interpretable.

Dataset	ETH-UCY		SDD	
	ADE	FDE	ADE	FDE
Predictor w/o goal	0.35	0.65	16.35	20.65
Retrieval only	1.34	1.81	32.35	46.46

Table 4: Ablation results based on our trajectory predictor without goal conditioning and on a retrieval-only approach.

K_e	5	10	12	15	20	20
N	15	10	8	5	1	1 *
ADE	10.39	9.65	9.43	9.89	13.24	9.11
FDE	23.40	18.43	17.32	15.82	14.38	15.20

Table 5: Ablation study on two hyperparameters, *i.e.* K_e and N, which correspond to the top two rows. Each cell shows results of different configuration on the SDD dataset in the ADE / FDE metrics. * denotes results from our goal and pretrained trajectory predictor of PECNet [29].

3.6. Goal and prediction visualization

To understand further why our model exhibits its strong results, we show visualizations of goal retrievals and trajectory predictions in Fig. 5. Results from another goal-based work, *i.e.*, PECNet [29], are also given. For both datasets, we provide three types of visual examples: equally good for both approaches, ours performs better and both fail, to shed light on the reasons behind our results.

For the ETH/UCY datasets, *i.e.*, the top row in Fig. 5, we plot testset trajectories of the ETH subset, from which we have seen the most improvement. We observe that our method performs on-par with PECNet on linear-like trajectories (a), while ours can achieve better predictions on trajectories with relatively high curvature (b and c). The reason might be that DTW is efficient at curvy shape matching, *cf*. [53]. Yet, both methods fail at the U-shape trajectory (d).

For the SDD dataset (bottom row), the same good performance on linear trajectories is observed by both methods (a). However, our approach performs much better when it comes to special road conditions, *e.g.*, 4-way intersections and pedestrian stairs (b). We believe the reason is that a few expert examples have been found behaving similarly at nearby geo-locations. We also see improved results on cases where the alternative fails to match the speed of future trajectories, *e.g.*, either too fast or too slow (c). This result can be attributed to use of velocity in our goal searching module that explicitly considers matching quality in motion. Again, for failures, neither approach captures complex motion dynamics, *e.g.*, 180° turn or unanticipated right-turn (d).

	ETH	HOTEL	ZARA1	ZARA2	UNIV
$K_e = 5$	1.01	0.27	0.45	0.45	0.60
$K_e = 10$	0.71	0.17	0.34	0.31	0.49
$K_e = 20$	0.56	0.09	0.28	0.23	0.44
$K_e = 50$	0.48	0.07	0.25	0.19	0.41

Table 6: Goal retrieval quality *vs.* various number of repository search candidates, K_e, on the ETH and UCY datasets. Lower is better.

$R(\%)$	10	20	30	40	60	80	90
ADE	9.12	8.68	8.10	8.02	7.84	7.73	7.70
FDE	18.83	17.06	16.20	15.80	15.92	14.46	14.40

Table 7: Ablation study on few-shot goal retrieval by having only a percentage, R, of the original repository available for the SDD dataset.

4. Conclusions

We have introduced a novel approach to pedestrian trajectory prediction, where the key innovation is the use of goal search through an expert repository to provide endpoints for goal conditioned prediction. Our approach does not require learning of model parameters for goal generation, yet produces high accuracy goals at modest computational expense. We also propose a novel way to use goal data (shifting by goal) that is simpler and incurs less overhead than current alternatives, yet sets a new state-of-the-art on the SSD, ETH and UCY datasets with the goal conditioned predictor we implemented. Moreover, when using our goals as input to an alternative goal conditioned trajectory predictor (PECNet) its performance also improves, which suggests broad applicability of our approach.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *Proc. CVPR*, 2016. 1, 2, 5, 6

[2] Christopher G Atkeson, Andrew W Moore, and Stefan Schaal. Locally weighted learning. In *Lazy learning*, pages 11–73. Springer, 1997. 2

[3] Niccoló Bisagno, Bo Zhang, and Nicola Conci. Group LSTM: Group trajectory prediction in crowded scenarios. In *Proc. ECCVW*, 2018. 2

[4] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *Proc. CVPR*, 2019. 3

[5] Marco Cuturi and Mathieu Blondel. Soft-DTW: a differentiable loss function for time-series. In *Proc. ICML*, 2017. 3, 7

[6] Patrick Dendorfer, Aljosa Osep, and Laura Leal-Taixé. Goal-GAN: Multimodal trajectory prediction based on goal position estimation. In *Proc. ACCV*, 2020. 1, 2, 4, 6, 7

[7] Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-conditioned imitation learning. In *NIPS*, 2019. 1

[8] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *Proc. ICML*, 2018. 1

[9] Dibya Ghosh, Abhishek Gupta, and Sergey Levine. Learning actionable representations with goal-conditioned policies. *arXiv preprint arXiv:1811.07819*, 2018. 1

[10] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. In *NIPS*, 2004. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[12] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proc. CVPR*, 2018. 1, 2, 5, 6, 7

[13] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 1

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[15] Boris Ivanovic and Marco Pavone. The Trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proc. ICCV*, 2019. 2

[16] Leslie Pack Kaelbling. Hierarchical learning in stochastic domains: Preliminary results. In *Proc. ICML*, 1993. 1

[17] Leslie Pack Kaelbling. Learning to achieve goals. In *Proc. IJCAI*, 1993. 1

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[19] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. ICLR*, 2017. 2

[20] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks. In *NIPS*, 2019. 1

[21] Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human trajectory forecasting in crowds: A deep learning perspective. *arXiv preprint arXiv:2007.03639*, 2020. 5

[22] Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. Neural data augmentation via example extrapolation. *arXiv preprint arXiv:2102.01335*, 2021. 2

[23] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer Graphics Forum*, volume 26, 2007. 2, 4, 5

[24] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional generative neural system for probabilistic trajectory prediction. In *Proc. IROS*, 2019. 2, 6, 7

[25] Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. EvolveGraph: Multi-agent trajectory prediction with dynamic relational reasoning. *NeurIPS*, 2020. 1, 6, 7

[26] Junwei Liang, Lu Jiang, and Alexander Hauptmann. Simaug: Learning robust representations from 3D simulation for pedestrian trajectory prediction in unseen cameras. In *Proc. ECCV*, 2020. 1

[27] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proc. CVPR*, 2019. 1, 2

[28] Mehran Maghoumi, Eugene M Taranta II, and Joseph J LaViola Jr. DeepNAG: Deep non-adversarial gesture generation. *arXiv preprint arXiv:2011.09149*, 2020. 3

[29] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *Proc. ECCV*, 2020. 1, 2, 4, 5, 6, 7, 8

[30] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proc. CVPR*, 2020. 1, 2, 4, 5, 6

[31] Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE TPAMI*, 36(11):2227–2240, 2014. 2, 7

[32] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *NIPS*, 2018. 1

[33] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation. In *Proc. CVPRW*, 2018. 1

[34] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proc. ICCV*. 4, 5

[35] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K Tsotsos. PIE: A large-scale dataset and models for pedes-

trian intention estimation and trajectory prediction. In *Proc. ICCV*, 2019. 2

[36] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proc. ICCV*, 2019. 2

[37] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proc. ECCV*, 2016. 3, 5

[38] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020. 1

[39] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. In *Proc. CVPR*, 2019. 1, 2, 6, 7

[40] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. 2, 3

[41] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. In *Proc. ECCV*, 2020. 1, 2, 4, 5, 6

[42] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. What constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 5(2):1696–1703, 2020. 1, 6

[43] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 2

[44] Shan Su, Cheng Peng, Jianbo Shi, and Chiho Choi. Potential field: Interpretable and unified representation for trajectory prediction. *arXiv preprint arXiv:1911.07414*, 2019. 1, 2

[45] Yichuan Charlie Tang and Ruslan Salakhutdinov. Multiple futures prediction. *arXiv preprint arXiv:1911.00997*, 2019. 2

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2

[47] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016. 2

[48] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves ImageNet classification. In *Proc. CVPR*, 2020. 2

[49] Jingwei Xu, Huazhe Xu, Bingbing Ni, Xiaokang Yang, and Trevor Darrell. Video prediction via example guidance. In *Proc. ICML*, 2020. 2, 6

[50] Chuanyu Yang, Kai Yuan, Qiuguo Zhu, Wanming Yu, and Zhibin Li. Multi-expert learning of adaptive legged locomotion. *Science Robotics*, 5(49), 2020. 2

[51] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *Proc. ECCV*, 2020. 1

[52] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. SR-LSTM: State refinement for LSTM towards pedestrian trajectory prediction. In *Proc. CVPR*, 2019. 1, 2

[53] Jiaping Zhao and Laurent Itti. shapeDTW: Shape dynamic time warping. *Pattern Recognition*, 74:171–184, 2018. 3, 8

[54] Xiatian Zhu, Antoine Toisoul, Juan-Manuel Prez-Ra, Li Zhang, Brais Martinez, and Tao Xiang. Few-shot action recognition with prototype-centered attentive learning. *arXiv preprint arXiv:2101.08085*, 2021. 2