Discrete Residual Flow for Probabilistic Pedestrian Behavior Prediction

Ajay Jain[*3†], **Sergio Casas**[*12], **Renjie Liao**[*12], **Yuwen Xiong**[*12], **Song Feng**[1], **Sean Segal**[12], **Raquel Urtasun**[12]
Uber Advanced Technologies Group[1], University of Toronto[2], UC Berkeley[3]
ajayj@berkeley.edu, {sergio.casas,rjliao,yuwen,songf,ssegal,urtasun}@uber.com

Abstract: Self-driving vehicles plan around both static and dynamic objects, applying predictive models of behavior to estimate future locations of the objects in the environment. However, future behavior is inherently uncertain, and models of motion that produce deterministic outputs are limited to short timescales. Particularly difficult is the prediction of human behavior. In this work, we propose the *discrete residual flow network* (DRF-NET), a convolutional neural network for human motion prediction that captures the uncertainty inherent in long-range motion forecasting. In particular, our learned network effectively captures multimodal posteriors over future human motion by predicting and updating a discretized distribution over spatial locations. We compare our model against several strong competitors and show that our model outperforms all baselines.

Keywords: Deep Learning, Autonomous Driving, Uncertainty, Forecasting

1 Introduction

In order to plan a safe maneuver, a self-driving vehicle must predict the future motion of surrounding vehicles and pedestrians. Motion prediction is challenging in realistic city environments. In Figure 1, we illustrate several challenges for pedestrian prediction. Gaussian distributions often poorly fit state posteriors (Fig. 1-a). Further, pedestrians have inherently multimodal behavior, as they can move in arbitrary directions and have unknown and changing goals, each achievable with multiple trajectories (Fig. 1-b). Even with strong evidence for a particular action, such as a road crossing, partially observed environments increase uncertainty in *the timing* of the action (Fig. 1-c). However, a self-driving vehicle motion planner needs actor predictions to be associated with time. Additional challenges include efficiently integrating spatial and temporal information, the mixed continuous-discrete nature of trajectories and maps, and availability of realistic data.

In the context of self-driving, most prior work represents behaviors through trajectories. Future trajectories can be predicted with a recurrent neural network (RNN) [1, 2, 3], a convolutional neural network (CNN) [4, 5, 6], or with constant velocity, constant acceleration, or expert-designed heuristics. However, a trajectory that minimizes the mean-squared error with respect to the true path can only capture the conditional average of the posterior [7]. The conditional average trajectory does not represent all possible future behaviors and may even be infeasible, lying between feasible trajectories.

To express multiple possible behaviors, a fixed number of future trajectories can be predicted [8], or several can be sampled [3, 9]. Still, in realistic environments, posterior predictive distributions are complex and a large number of samples are needed to capture the space of possibilities. Such models tradeoff prediction completeness and latency from repeated sampling. Further, the number of possible trajectories increases exponentially over long time horizons, and uncertainty grows rapidly.

Instead of predicting trajectories, in this work, we take a probabilistic approach, predicting distributions over pedestrian state at each timestep that can directly be used for cost-based self-driving vehicle planning. Conditioning on a spatio-temporal rasterization of agent histories aligned to the local map, we leverage deep convolutional neural network architectures for implicit multi-agent reasoning, and mimic human dynamics through a *discrete residual flow network*, which we refer to as DRF-NET. We summarize our contributions as follows:

[*] Denotes equal contribution.
[†] Work done while at Uber ATG.

3rd Conference on Robot Learning (CoRL 2019), Osaka, Japan.

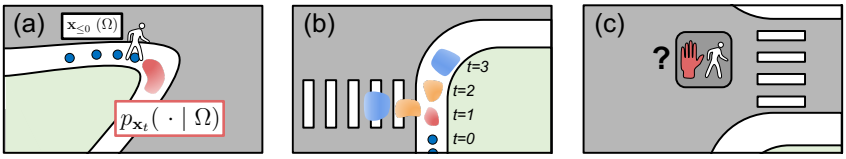

Figure 1: Challenging urban scenarios for pedestrian prediction, depicting pedestrian detections (circles) and future state posteriors colored by time horizon. (a) Gaussian distributions often poorly express scene-sensitive behaviors. (b) Inherent multimodality: the pedestrian may cross a crosswalk or continue along a sidewalk. (c) Partial observability: signals and actors may be occluded.

- We develop a deep probabilistic formulation of actor motion prediction that provides marginal distributions over state at each future timestep without expensive marginalization or sampling. Our *discrete residual flow* equation is motivated by autoregressive generative models, and better captures temporal dependencies than time-independent baselines.
- We propose the convolutional Discrete Residual Flow Network that predicts actor state over long time horizons with highly expressive discretized distributions.
- We thoroughly benchmark model variants and demonstrate the benefit of belief discretization on a large scale, real-world dataset. We evaluate the *likelihood*, *displacement error*, *multimodality*, *entropy*, *semantic mass ratio* and *calibration* of the predictions, using a novel ModePool operator for estimating the number of modes of a discrete distribution.

2 Related work

Prior work on pedestrian prediction has largely modeled trajectories, goals, or high-level intent.

Human trajectory forecasting The pedestrian prediction literature is reviewed in [10, 11]. Multi-pedestrian interactions have been modeled via pooling [1, 3] or game theory [12]. Becker et al. [2] predict future trajectories with a recurrent encoder and MLP decoder, reporting lower error than more elaborate multi-agent schemes, and find that behaviors are multimodal and strongly influenced by the scene. Social GAN [3] is a sequence-to-sequence generative model where trajectory samples vary in speed and turning angle, trained with a variety loss to encourage diversity. However, the runtime of the sampling approach scales with the number of samples (150 ms for 12 trajectories), even without using a local map, and many samples are needed. SoPhie [9] is another sampling strategy integrating external overhead camera imagery. In contrast, we predict entire expressive spatial distributions rather than individual samples and incorporate a local map into prediction.

Goal directed prediction Ziebart et al. [13] use historical paths to pre-compute a prior distribution over pedestrian goals indoors, then develop an MDP to infer a posterior distribution over future trajectories. Wu et al. [14] use a heuristic to identify possible goal locations in a mapped environment and a Markov chain to predict the next-time occupancy grid. Rehder et al. [15, 16] use a two-stage deep model to predict a Gaussian mixture over goals, then construct distributions at intermediate timesteps with a planning network. Still, the number of mixture components must be tuned, and the mixture is discretized during inference, which is computationally expensive. Fisac and Bajcsy [17, 18] specify known goals for each human indoors, then estimate unimodal state distributions by assuming humans approximately maximize utility *i.e*. progress toward the goal measured by Euclidean norm. They estimate prediction confidence from model performance and return uninformative distributions at low confidence. Confidence estimation is complementary to our approach.

Semantic map Pedestrian predictors have separately reasoned about spatially continuous trajectories and discretized world representations [13, 15]. These works either ignore the semantic map or integrate it at an intermediate stage. In vehicle prediction, input map rasterizations are more widely used. IntentNet [5] renders a bird's-eye view of the world to predict vehicle trajectories and high-level intention simultaneously, using a rasterized lane graph and a 2D convolutional architecture to improve over previous work [4]. Similar map rasterizations are used in [6, 19, 20], and this work.

Related modeling techniques The convolutional long short-term memory (ConvLSTM) architecture has been applied to spatio-temporal weather forecasting [21]. A ConvLSTM iteratively updates a hidden feature map, from which outputs are derived. In contrast, DRF-NET sequentially adapts the output space rather than a hidden state. Similarly, the adaptive instance normalization operator [22] uses a shared feature to predict and apply scale/shift parameters to a fixed, discrete image. Normalizing flows [23] apply a series of invertible mappings to samples from a simple prior, *e.g*. a Gaussian, constructing a random variable with a complex PDF. While normalizing flows transform individual samples, we directly transform a probability mass function (PMF) for computational efficiency.

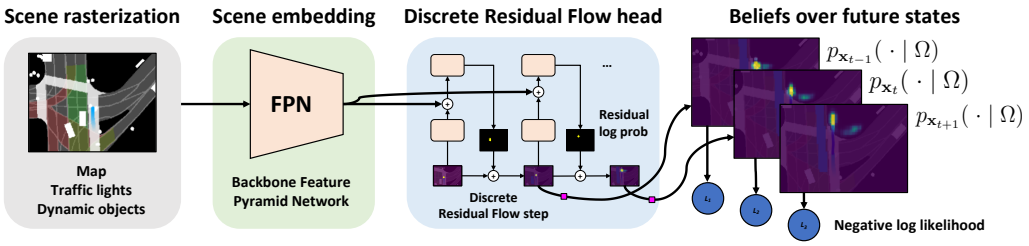

Figure 2: Overview of the Discrete Residual Flow Network. Pedestrian of Interest (PoI) and actor detections are aligned with a semantic map. A multi-scale backbone jointly reasons over spatio-temporal information in the input, embedding context into a feature $\mathcal{F}$. Finally, the DRF head recursively adapts an initial distribution to predict future pedestrian states on long time horizons.

3 Discrete Residual Flow Network

In this paper, we express beliefs over future pedestrian positions through categorical distributions that discretize space. Such distributions can be used for cost-based planning or constrained path optimization in self-driving vehicles. In this section, we explain how we represent historical observations as a multi-channel image encoding both the known map and detected actors, a process we call *rasterization*. We then introduce a backbone deep neural network which extracts features from the rasterized image, followed by the probabilistic framework for our DRF-NET. Finally, we introduce our DRF head which uses the extracted features for prediction.

Encoding Historical Information Future pedestrian actions are highly correlated with historical actions. However, actions are also influenced by factors such as road surface types, traffic signals, static objects, vehicles, and other pedestrians. We *rasterize* all semantic map information and agent observations into a 3D tensor, encoding both spatial and temporal information by automatic rendering. The first two dimensions correspond to the spatial domain and the third dimension forms channels. Each channel is an 576×416 px image encoding specific local bird's eye view (BEV) information at a resolution of 8 px per meter. Figure 3 shows an example rasterization from a real urban scene.

Dynamic agents are detected from LiDAR and camera with the object detector proposed in Liang et al. [24], and are associated over time using a matching algorithm. Resulting trajectories are refined using an Unscented Kalman Filter [25]. DRF-NET renders detected pedestrians in each timestep t for the past 6 seconds in channel D_t and detected non-pedestrians (*e.g.* vehicles) in channel V_t. To discriminate the pedestrian of interest from other actors, a grayscale image R masks their tracklet.

DRF-NET renders the local map in a similar fashion to [5], though centers the map about the PoI. 15 semantic map channels M finely differentiate urban surface labels. These channels mask crosswalks, drivable surfaces, traffic light state, signage, and detailed lane information. Maps are annotated in a semi-automated fashion in cities where the self-driving vehicle may operate, and only polygons and polylines are stored. The final rasterization is $\Omega = [D_{\leq 0}, V_{\leq 0}, R, M]$ where $[\cdot]$ indicates concatenation along the channel dimension, the subscript ≤ 0 indicates a collection of elements from all past timesteps and $t = 0$ is the last timestep. All channels are rotated such that the currently observed PoI is oriented toward the top of the scene.

Backbone Network DRF-NET uses a deep residual network with 18 convolutional layers (ResNet-18) [26] to extract features $\mathcal{F}$ from the rasterization Ω. We extract 4 feature maps at $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ and $\frac{1}{16}$ of the input resolution from ResNet-18. These multi-scale intermediate features are upscaled and aggregated into a $\frac{1}{4}$ resolution global context with a feature pyramid network (FPN) [27].

Probabilistic Actor State Prediction We now introduce a probabilistic formulation of future actor state prediction. Given rasterization Ω, we are interested in inferring a predictive posterior distribution over possible spatial locations of the PoI for each timestep t where $t = 1, \cdots, T_f$. Instead of treating the state as a continuous random variable, we discretize space to permit a one-hot state encoding. Specifically, we divide space into a grid with K bins. The state at time t, $\mathbf{x}_t$, is a discrete random variable which takes one of the K possible bins.

Consider the joint probability of the states in the future T_f timesteps, *i.e.*, $p_{\mathbf{x}_1, \cdots, \mathbf{x}_{T_f}}(x_1, \cdots, x_{T_f} \mid \Omega)$. This distribution can be modeled with several factorizations. The first and the most straightforward factorization assumes conditional independence of future timesteps,

$$p_{\mathbf{x}_1, \cdots, \mathbf{x}_{T_f}}(x_1, \cdots, x_{T_f} \mid \Omega) = \prod_t p_{\mathbf{x}_t}(x_t \mid \Omega) \tag{1}$$

(a) Aggregated rasterization Dynamic objects shown at t=0 **(b) Semantic map channels** Crossings, road mask, lights, lanes **(c) PoI + dynamic object history** Subset of timesteps shown

Figure 3: **Scene history and context representation.** DRF-NET rasterizes map elements into a shared spatial representation (b), augmented with spatio-temporal encodings of actor motion (c).

We can use a neural network, *e.g.*, a CNN, to directly model $p_{\mathbf{x}_t}(x_t \mid \Omega)$. In Section 4.3, we show the performance of a *mixture density network* and *fully-convolutional predictor* that simultaneously predict these factors. Still, conditional independence is a strong assumption. The second factorization follows an autoregressive fashion, providing the foundation for many models in the literature,

$$p_{\mathbf{x}_1,\cdots,\mathbf{x}_{T_f}}(x_1,\cdots,x_{T_f} \mid \Omega) = \prod_t p_{\mathbf{x}_t|\mathbf{x}_{\leq t-1}}(x_t \mid x_{\leq t-1}, \Omega) \tag{2}$$

For example, recurrent encoder-decoder architectures [1, 2, 3] sample trajectories one state at a time and capture the conditional dependencies through a hidden state.

In contrast to the sample-based approach, often we desire access to compact representations of $p_{\mathbf{x}_t}(x_t \mid \Omega)$ for a particular Ω, such as an analytic form or a discrete categorical distribution. As we always condition on Ω, we refer to $p_{\mathbf{x}_t}(x_t \mid \Omega)$ as a marginal distribution. Access to the marginal provides interpretability, parallel sampling and ease of planning as the marginals can be used as occupancy grids. However, direct marginalization is expensive if not intractable as we typically have no simple analytic form of the joint distributions. Approximation is possible with Monte Carlo methods, though many samples are needed to characterize the marginal.

Instead, we propose a *flow* between marginal distributions that resembles an autoregressive model in its iterative nature, but avoids sampling at each step. In contrast to a normalizing flow [23], which approximates a posterior over a single random variable by iteratively transforming its distribution, discrete residual flow transforms between the marginal distributions of different, temporally correlated random variables by exploiting a shared domain.

Discrete Residual Flow Our model recursively constructs $p_{\mathbf{x}_t}(\,\cdot \mid \Omega)$ from $p_{\mathbf{x}_{t-1}}(\,\cdot \mid \Omega)$,

$$\log p_{\mathbf{x}_t}(x_t \mid \Omega) = \log p_{\mathbf{x}_{t-1}}(x_t \mid \Omega) + \underbrace{\log \psi_{t;\theta_t}(x_t, p_{\mathbf{x}_{t-1}}(\,\cdot \mid \Omega), \Omega)}_{\text{Residual}} - \log Z_t, \tag{3}$$

where we refer to the second term on the right hand side as the *residual*. $\psi_{t;\theta_t}$ is a sub-network with parameter θ_t called the *residual predictor* that takes the marginal distribution $p_{\mathbf{x}_{t-1}}(\,\cdot \mid \Omega)$ and context Ω as input, and predicts an elementwise update that is used to construct the subsequent marginal distribution $p_{\mathbf{x}_t}(\,\cdot \mid \Omega)$. Z_t is the normalization constant to ensure $p_{\mathbf{x}_t}(\,\cdot \mid \Omega)$ is a valid distribution. Note that the residual itself is not necessarily a valid probability distribution.

Eq. (3) can be viewed as a discrete probability flow which maps from the distribution of $\mathbf{x}_{t-1}$ to the one of $\mathbf{x}_t$. We use deep neural networks to instantiate the probability distributions under this framework and provide a derivation of Eq. (3) in the appendix, Section 6.5.

For initialization, $p_{\mathbf{x}_0}(\,\cdot \mid \Omega)$ is constructed with high value around our $t = 0$ PoI position and near-zero value over other states. In implementation, the residual predictor is a convolutional architecture that outputs a 2D image, a compact and convenient representation as our states are spatial. This 2D image is queryable at state x_t via indexing, as is the updated marginal. Additionally, in implementation, we normalize all marginals at once and apply residuals to the unnormalized potential $\tilde{p}$,

$$\log \tilde{p}_{\mathbf{x}_t}(x_t \mid \Omega) = \log \tilde{p}_{\mathbf{x}_{t-1}}(x_t \mid \Omega) + \log \psi_{t;\theta_t}(x_t, \tilde{p}_{\mathbf{x}_{t-1}}(\,\cdot \mid \Omega), \Omega) \tag{4}$$

Figure 2 illustrates the overall computation process. The embedding of the rasterization $\mathcal{F}(\Omega)$ is shared at all timesteps, used by each residual predictor. Figure 4 further illustrates the architectural details of the DRF residual predictor for one timestep.

Learning We perform learning by minimizing the negative log likelihood (NLL) of the observed sequences of pedestrian movement. Specifically, we solve the following optimization,

$$\min_{\Theta} \qquad -\mathbb{E}_{\mathbf{x},\Omega}\left[\sum_{t=1}^{T_f} \log p_{\mathbf{x}_t}(x_t \mid \Omega)\right] \tag{5}$$

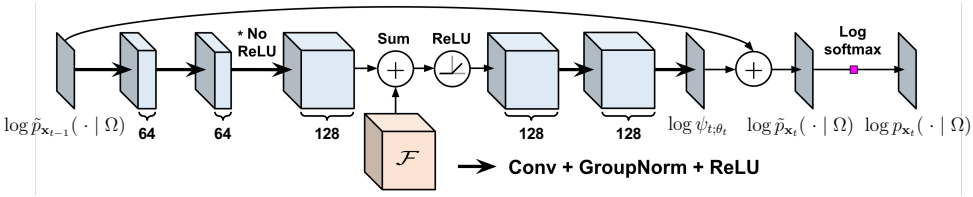

Figure 4: One step of recursive Discrete Residual Flow. The log potential is used to update the global feature map $\mathcal{F}$. DRF then predicts a residual $\psi_{t;\theta_t}$ to flow to the log potential for the next timestep.

Model	Negative log likelihood (NLL) Mean	@ 1 s	@ 3 s	@ 10 s	ADE (m) 0.2-10s	FDE (m) @ 1 s	@ 3 s	@ 10 s	Mass Ratio (%) Acc.	Recall
Density Net	5.39	2.87	3.96	6.74	3.49	0.93	1.72	7.66	77.99	81.33
MDN-4	3.01	1.64	2.00	4.33	1.47	0.38	0.69	3.38	87.85	84.12
MDN-8	3.43	1.60	2.77	4.79	1.78	0.60	0.88	3.91	85.56	84.19
ConvLSTM	2.51	0.89	1.86	4.07	1.58	0.47	1.06	3.20	88.02	85.02
DRF-NET	**2.37**	**0.76**	**1.74**	**3.83**	**1.23**	**0.35**	**0.62**	**2.71**	**89.78**	**85.41**

Table 1: Comparison of the baselines and our proposed model DRF-NET with access to ground-truth observations. Metrics are negative log likelihood in 0.5×0.5 m^2 bin containing future GT position, average displacement error (ADE) and final displacement error (FDE) in meters, and percent of predicted mass. Mean NLL, ADE and the mass ratios are averaged over 50 timesteps, $t = 0.2 - 10$ s.

where the expectation $\mathbb{E}[\cdot]$ is taken over all possible sequences and will be approximated via mini-batches. $\Theta = \{\theta_1, \cdots, \theta_{T_f}, w\}$ where w denotes the parameters of the backbone network.

4 Evaluation

There is not a standard dataset for probabilistic pedestrian prediction with real-world maps and dynamic objects. Thus, we construct a large-scale dataset of real world recordings, object annotations, and online detection-based tracks. We implement baseline pedestrian prediction networks inspired by prior literature [28, 29, 15] and compare DRF-NET against these baselines on standard negative log likelihood and displacement error measures. We propose an evaluation metric for measuring prediction multimodality, which is one of the most characteristic properties of pedestrian behavior. We also analyze the calibration, entropy and semantic interpretation of predictions. Finally, we present qualitative results in complex urban scenarios.

4.1 Dataset

Our dataset consists of 481,927 ground truth pedestrian trajectories gathered in several North-American cities. The dataset is split into 375,700 trajectories for training, 34,571 for validation, and 71,656 held-out trajectories for testing. Dynamic objects are manually annotated in a $360°$, 120 m range view from an on-vehicle LiDAR sensor. Annotations contain 6 s (30 frames) of past observations and 10 s (50 frames) of the future. These 5 Hz, 16 s sliding windows are extracted from longer logs.

We also fine-tune and evaluate DRF-NET with variable length trajectories from an object detector in the same scenarios. The detector is discussed in Section 3. This assesses real-world, on-vehicle prediction performance, reflecting the challenges inherent to real perception such as partial observability, occlusion and identity switches in tracking algorithms. While PoIs are annotated for a full 16 seconds in our ground truth experiments, realistic tracks are of variable length. A self-driving vehicle must predict the behavior of other agents with a very limited set of observations. Thus, we evaluate DRF-NET by predicting 10 seconds (50 frames) into the future, given tracks with as few as 3 historical frames, sufficient for estimating acceleration. Relaxing the requirements about past history avoids skewing our dataset toward easily tracked pedestrians, such as stationary agents.

4.2 Baselines

In this section, we describe two baseline predictor families. These baselines are trained end-to-end to predict distributions given features $\mathcal{F}(\Omega)$ produced by the same backbone as our proposed model.

Model	Real detection data (NLL) Mean	@ 1 s	@ 3 s	@ 10 s
Density Net	5.64	1.88	4.12	7.91
MDN-4	3.21	1.52	2.54	4.71
MDN-8	3.21	1.53	2.55	4.73
ConvLSTM	3.14	1.54	2.51	4.64
DRF-NET	**2.98**	**1.47**	**2.39**	**4.36**

Table 2: Probabilistic prediction comparison of the baselines and our proposed model DRF-NET when noisy detections (online tracks) are observed instead of the ground-truth.

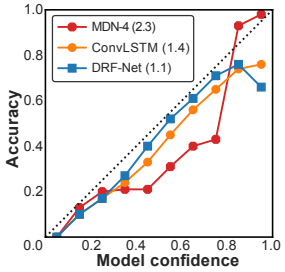

Figure 5: Calibration curves and expected calibration error ($* 10^{-3}\%$)

Mixture Density Networks (MDNs) represent a conditional posterior over continuous targets given continuous inputs with a fully-connected neural network that predicts parameters of Gaussian mixture model [7]. For a baseline, we implement a variant of this architecture that models pedestrian posteriors at multiple time horizons, conditioned on the past history and current location. Inspired by Rehder et al. [15], we generate the i-th mixture component from the neuron outputs $\{m_x, m_y, s_x, s_y, r, p\}_i$ which are then reparameterized as $\sigma_{x,i} = \exp(s_{x,i}) + \epsilon, \sigma_{y,i} = \exp(s_{y,i}) + \epsilon$, and $\rho_i = \tanh(r_i)$ to obtain the mean $\vec{\mu}_i$, covariance matrix Σ_i and the responsibility of the mixture π_i:

$$\vec{\mu}_i = \begin{bmatrix} m_{x,i} \\ m_{y,i} \end{bmatrix}, \Sigma_i = \begin{bmatrix} \sigma_{x,i}^2 & \rho_i \sigma_{x,i} \sigma_{y,i} \\ \rho_i \sigma_{x,i} \sigma_{y,i} & \sigma_{y,i}^2 \end{bmatrix}, \pi_i = \frac{\exp(p_i)}{\sum_{j=1}^{N} \exp(p_j)} \tag{6}$$

Training MDNs is challenging due to a high sensitivity to initialization and parameterization. To avoid numerical instabilities, the minimum standard deviation is ϵ. Even with a careful initialization and parameterization, training can be unstable, which we mitigate by discarding abnormally large losses. Note that Rehder et al. [15] stabilized training by minimizing only the *minimum* of the batchwise negative log likelihood. Minimizing this minimum loss leads to a good performance on easy examples, but catastrophic performance on hard ones. Lastly, conversions from a discretized spatial input to a continuous output can be challenging to learn [30], a problem that our proposed DRF-NET avoids via a discretized output that is spatially aligned with the input.

ConvLSTM In contrast to our DRF-NET that recursively updates output distributions in the log-probability space, one can also recurrently update *hidden state* using a Convolutional LSTM [21] that observes the previous prediction. Output distributions are then predicted from the hidden state.

4.3 Results

We evaluate negative log likelihood (NLL) at short and long prediction horizons, where lower values indicate more accurate predictions, as well as the mean NLL across all 50 future timesteps. In Table 1 and 2, we present results on the held-out test set for ground truth annotated logs and tracked, real-world detections, respectively. Our proposed DRF-NET achieves a superior likelihood over the baselines by introducing a discrete state representation and a probability flow between timesteps.

Likelihood on ground truth tracks In order to evaluate our results under perfect perception, we benchmark on ground truth (annotated) pedestrian trajectories. Table 1 shows that our proposed model reduces the mean NLL by 0.64 when compared to the best performer among the MDNs and by 0.14 with respect to the ConvLSTM baseline. This corresponds to a 90% increase in geometric mean likelihood compared to the best MDN and to a 15% increase when compared to the ConvLSTM.

Likelihood on online tracks Under online, imperfect perception, DRF-NET achieves a reduction of 0.23 in mean NLL over the best MDN and 0.16 over ConvLSTM, *i.e.* a 26% and a 17% increase of the geometric mean likelihood of the future observed pedestrian positions, respectively (Table 2). DRF-NET's sequential residual updates may regularize and smooth predictions despite perception noise. Adding more than 4 components to the density networks does not reduce NLL. Directly predicting occupancy probability over a grid delivers stronger performance than discretizing a continuous spatial density. Using an explicit memory with hidden state updates (ConvLSTM) also has inferior performance to our proposed flow between output distributions.

Displacement error We compute the expected root mean squared error, or expected displacement error, between the ground truth pedestrian position and model predictions. This is approximated by discretizing posteriors, computing the distance from each cell to the ground truth, and taking the average weighted by confidence at each cell. Table 1 reports the error in meters, averaged over 50 timesteps (ADE) and at specific horizons (FDE). DRF-NET significantly outperforms all baselines.

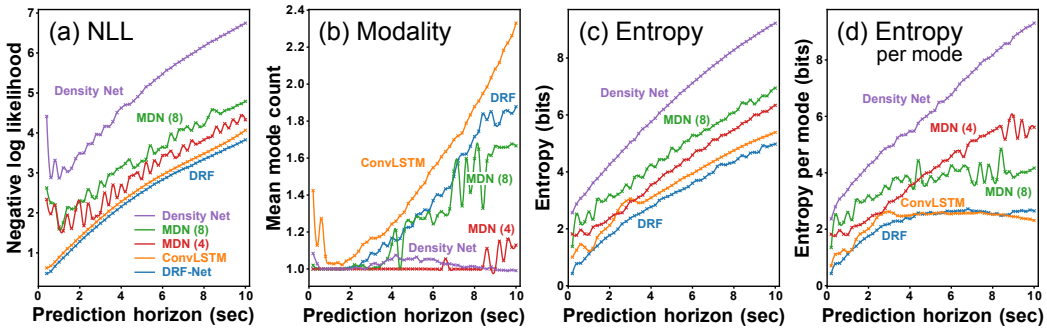

Figure 6: Test metrics. DRF-NET has low NLL (a) and captures the multimodality inherent in long-range futures (b). A discrete state space (DRF and ConvLSTM) yields the lowest NLL and entropy (c), and entropy per mode saturates. However, it increases with horizon for continuous MDNs (d).

Ablative model variant	Ground truth data (NLL)				Real detection data (NLL)			
	Mean	@ 1 s	@ 3 s	@ 10 s	Mean	@ 1 s	@ 3 s	@ 10 s
Independent, categorical (Fully conv)	2.45	0.80	1.83	3.89	3.06	1.49	2.46	4.45
+ Sequential refinement (DRR)	2.40	0.80	1.78	**3.83**	3.02	1.49	2.44	4.40
Discrete Residual Flow	**2.37**	**0.76**	**1.74**	**3.83**	**2.98**	**1.47**	**2.39**	**4.36**

Table 3: Ablation study of multiple probabilistic prediction heads. Metric is NLL as in Table 1.

Model calibration To understand overconfidence of predictive models, we compute calibration curves and expected calibration error (ECE) on the ground truth test set according to Guo et al. [31] by treating models as multi-way classifiers over space. ECE measures miscalibration by approximating the expected difference between model confidence and accuracy. DRF has the lowest calibration error, with accuracy closest to the model confidence on average, as shown in Fig. 5. While somewhat overconfident, these models could be recalibrated with isotonic regression or temperature scaling.

Multimodality and Entropy Analysis We propose a ModePool operator to estimate the number of modes of a discrete spatial distribution. ModePool approximates the number of local maxima in a discrete distribution p as follows, where the max is taken over $|\delta_r|, |\delta_c| \leq \lfloor \frac{k}{2} \rfloor$, *i.e.* $k \times k$ windows:

$$\text{ModePool}_{k,\epsilon}(p) = \sum_{i,j} \mathbb{1}_{p_{i,j} = \max p_{i+\delta_r, j+\delta_c}} \mathbb{1}_{p_{i,j} \geq \epsilon} \tag{7}$$

Only local maxima with mass exceeding a threshold ϵ are counted. ModePool is efficiently implemented on GPU by adapting the MaxPool filter commonly used in CNNs for downsampling. In Figure 6-b, modality is estimated with $k = 5$, $\epsilon = 0.1$. Given our output resolution, at most one mode per 2.5×2.5 m^2 area can be counted. While the baseline MDN-4 predicts multiple Gaussian distributions, we observe strong mode-collapse. In contrast, DRF produces predictive posteriors that have increasingly multimodal predictions over horizons. Though an MDN of 8 mixtures captures some multimodality as well, the mean number of modes is highly inconsistent over time (6-b, middle).

Fig. 6-c shows the mean entropy of the predicted distributions. Entropy for DRF-NET is the lowest. As DRF-NET also achieves lower NLL at all future horizons (6-a), DRF-NET predictions can be interpreted as low bias and low variance. We combine entropy and modality into a single metric in Fig. 6-d. For the discrete heads (DRF, ConvLSTM), the entropy per mode saturates. These models capture inherent future uncertainty by adding distributional modes *e.g.* high level actions rather than increasing per-mode entropy. This is not the case for baselines, where entropy per mode grows over time. Qualitatively, in Fig. 7, DRF-NET predictions remain the most concentrated over long horizons.

Semantic mass ratio Our semantic map can partition the world into three disjoint high-level classes, $\mathcal{C} = \{\text{Crosswalk}, \text{Road}, \text{Off-Road}\}$. To interpret how well models understand the map, we measure *confidence-weighted semantic accuracy*, the mean predicted mass that falls on the correct map class. We also measure *safety-sensitive recall*, the mean mass that falls into a drivable region when the PoI is in a drivable region—performance when a PoI is on-road is very important to a self-driving car. Let $c(\mathbf{x}) \in \mathcal{C}$ be the class of location $\mathbf{x}$, determined by the map, and c_t^* be the ground truth class of the PoI position at time t. Then, we compute metrics as follows, reported in Table 1:

$$\text{Accuracy}(c^*_{\geq 1}, \mathbf{x}_{\leq 0}, \Omega) = \frac{1}{T_f} \sum_{t=1}^{T_f} P(c(\mathbf{x}_t) = c_t^*) \tag{8}$$

$$\text{Recall}(c^*_{\geq 1}, \mathbf{x}_{\leq 0}, \Omega) = \frac{1}{|\text{SS}|} \sum_{t \in \text{SS}} P(c(\mathbf{x}_t) \in \{\text{CW, ROAD}\}), \tag{9}$$

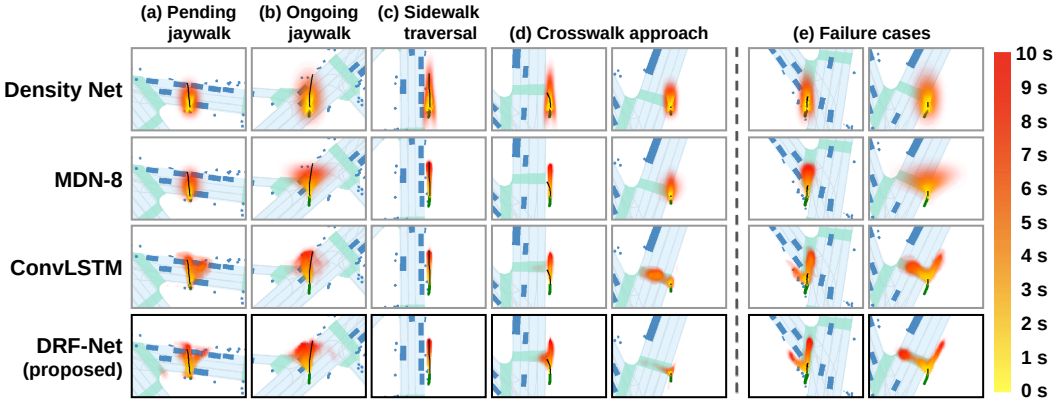

Figure 7: Pedestrian predictions: ground truth past trajectory is green, future is black, opacity shows density, and color shows time horizon. MDN-4 predictions are omitted due to similarity to MDN-8; both are largely unimodal. More results in the supplementary video.

where SS $= \{t : c_t^* \in \{\text{CW, ROAD}\}\}$, the safety-sensitive timesteps. DRF-NET significantly outperforms baselines on semantic mass ratio metrics, and most accurately predicts the type of surface the PoI will traverse. This suggests that DRF-NET better uses the map, and is qualitatively reflected by low-entropy, concentrated mass within map polygons in Figure 7.

Ablation Study We conduct an ablation study that evaluates the value of discrete predictions and our residual flow formulation. We study two variants of the DRF prediction head, a fully convolutional and a discrete residual refinement (DRR) head. MDNs predict continuous mixtures of Gaussians assuming conditional independence of future states, which can be discretized for cost-based planning. We can instead directly predict independent discrete distributions. The fully convolutional predictor projects the spatial feature $\mathcal{F}$ (Section 3) into a 50-channel space representing per-timestep logits with a 1x1 convolution on scene features. Spatial softmax produces valid distributions over the discrete spatial support. The DRR head takes as input the discrete probability distributions output by our fully convolutional predictor and sequentially predicts per-timestep residuals in log-space with per-timestep weights. DRR thereby refines independent predictions sequentially.

Table 3 shows that state space discretization and categorical prediction (fully convolutional head) has significantly better NLL than the best continuous mixture model in Table 1, a 0.56 reduction in NLL. Sequential refinement of independent predictions using DRR improves performance. However, predicting flow in the log probability space with DRF achieves the best likelihood.

Qualitative Results Figure 7-a shows predictions for a pedestrian in a challenging pre-crossing scenario. Predictive posteriors modeled by DRF-NET (4th row) express high multimodality and concentrated mass, with three visible high-level actions: stopping, crossing straight, or crossing while skirting around a car. DRF-NET also exhibits strong map interactions, avoiding parked vehicles. However, MDNs predict highly entropic, unimodal distributions, and the ConvLSTM places substantial spurious mass on parked vehicles. Across other test scenes, we observe that DRF-NET constructs low-entropy yet multimodal predictions with similarly strong map and actor interactions. In Figure 7-d, DRF-NET is the only model to correctly predict a crosswalk approach. Still, in failure cases, all models predict crossings too early, possibly due to unknown traffic light state. This could lead to more conservative self-driving vehicle plans if the pedestrians were nearby. Nonetheless, these pedestrians and lights are distant.

5 Conclusion

In this paper, we develop a probabilistic modeling technique applied to pedestrian behavior prediction, called Discrete Residual Flow. We encode multi-actor behaviors into a bird's eye view rasterization aligned with a detailed semantic map. Based on deep convolutional neural networks, a probabilistic model is designed to sequentially update marginal distributions over future actor states from the rasterization. We empirically verify the effectiveness of our model on a large scale, real-world urban dataset. Extensive experiments show that our model outperforms several strong baselines, expressing high likelihoods, low error, low entropy and high multimodality. The strong performance of DRF-NET's discrete predictions is very promising for cost-based and constrained robotic planning.

Acknowledgments

We would like to thank Abbas Sadat for useful discussions during the development of this research.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016.

[2] S. Becker, R. Hug, W. Hübner, and M. Arens. An Evaluation of Trajectory Prediction Approaches and Notes on the TrajNet Benchmark. *CoRR*, abs/1805.07663, 2018.

[3] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] W. Luo, B. Yang, and R. Urtasun. Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting With a Single Convolutional Net. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] S. Casas, W. Luo, and R. Urtasun. IntentNet: Learning to Predict Intention from Raw Sensor Data. In *Conference on Robotic Learning (CoRL)*, 2018.

[6] M. Bansal, A. Krizhevsky, and A. S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *CoRR*, abs/1812.03079, 2018.

[7] C. M. Bishop. Mixture density networks. Technical report, Citeseer, 1994.

[8] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1809.10732*, 2018.

[9] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019.

[10] D. Ridel, E. Rehder, M. Lauer, C. Stiller, and D. Wolf. A literature review on the prediction of pedestrian behavior in urban scenarios. In *Proceedings of the International Conference on Intelligent Transportation Systems*, November 2018.

[11] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras. Human motion trajectory prediction: A survey. *ArXiv*, abs/1905.06113, 2019.

[12] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4636–4644. IEEE, 2017.

[13] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3931–3936. IEEE, 2009.

[14] J. Wu, J. Ruenz, and M. Althoff. Probabilistic map-based pedestrian motion prediction taking traffic participants into consideration. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1285–1292. IEEE, 2018.

[15] E. Rehder, F. Wirth, M. Lauer, and C. Stiller. Pedestrian prediction by planning using deep neural networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–5. IEEE, 2018.

[16] E. Rehder and H. Kloeden. Goal-directed pedestrian prediction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 50–58, 2015. URL `https://www.mrt.kit.edu/z/publ/download/2015/rehder_iccv15.pdf`.

[17] J. F. Fisac, A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, S. Wang, C. J. Tomlin, and A. D. Dragan. Probabilistically Safe Robot Planning with Confidence-Based Human Predictions. In *Robotics: Science and Systems (RSS)*, 2018.

[18] A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, J. F. Fisac, S. Deglurkar, A. D. Dragan, and C. J. Tomlin. A scalable framework for real-time multi-robot, multi-human collision avoidance. *CoRR*, abs/1811.05929, 2018.

[19] B. Yang, M. Liang, and R. Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155, 2018.

[20] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, and J. Schneider. Motion prediction of traffic actors for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1808.05819*, 2018.

[21] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[22] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 1510–1519. IEEE, 2017.

[23] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

[24] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018.

[25] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000.

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[28] K. Saleh, M. Hossny, and S. Nahavandi. Long-term recurrent predictive model for intent prediction of pedestrians via inverse reinforcement learning. In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 2018.

[29] A. Zyner, S. Worrall, and E. Nebot. Naturalistic driver intention and path prediction using recurrent neural networks. *arXiv preprint arXiv:1807.09995*, 2018.

[30] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9628–9639, 2018.

[31] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1321–1330. JMLR.org, 2017. URL `http://dl.acm.org/citation.cfm?id=3305381.3305518`.

[32] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, abs/1412.6980, 2015.

6 Appendix

In this appendix, we provide additional implementation (Section 6.1-6.2), training (Section 6.3) and evaluation (Section 6.4) details for our proposed DRF-NET and baseline architectures. We also provide a derivation of the DRF update equation (Section 6.5).

6.1 Backbone network

In Section 3 of the paper, we described a deep convolutional neural network architecture that represents our spatio-temporal scene rasterization Ω as a global feature $\mathcal{F}$. This CNN architecture forms the initial layers of the proposed model and baselines, though each network is trained end-to-end (backbone parameters are not shared across models). The backbone architecture is detailed in Figure 8, below. The proposed DRF-NET further projects the $N \times 256 \times 144 \times 104$ feature $\mathcal{F}$ into a 128 channel space with a learned 1×1 convolutional filter for memory efficiency.

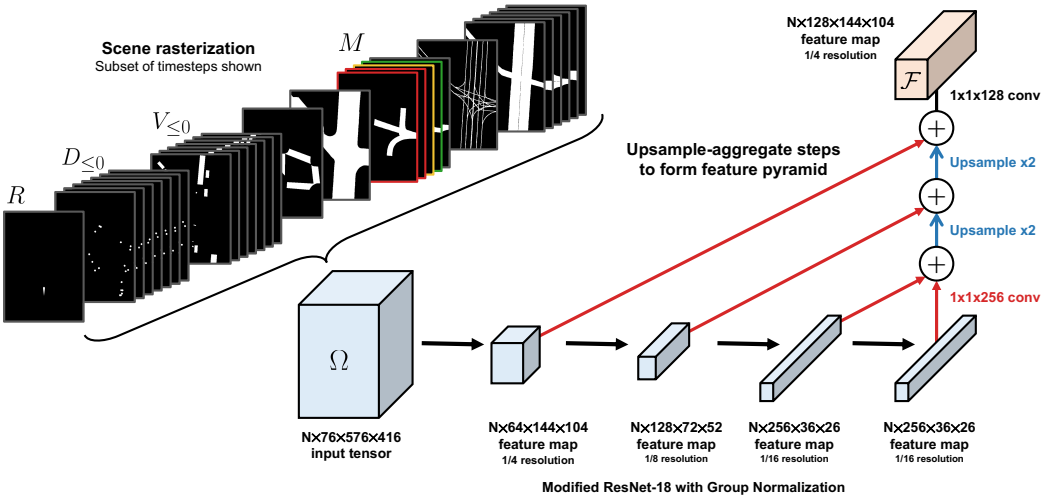

Figure 8: Backbone feature pyramid network (FPN). N denotes the batch size, *e.g.* the number of pedestrians of interest for inference or number of scenarios per batch for training.

6.2 Rasterization

Rasterization dimensions The input bird's eye view (BEV) region is rotated for a fixed pedestrian of interest heading at the current time and spans 52 meters perpendicularly and 72 meters longitudinally, 50 ahead and 22 behind the last observed pose of the pedestrian. We set the input resolution to 0.125 meters per pixel and the output resolution of our spatial distribution to 0.5 meters per pixel. At the input resolution, our BEV rasterization channels are each 576 px by 416 px.

Encoding observed actor behavior We use the object detector proposed in Liang et al. [24], which exploits LiDAR point clouds as well as cameras in an end-to-end fashion in order to obtain reliable bounding boxes of dynamic agents. Further, we associate the object detections using a matching algorithm and refine the trajectories using an Unscented Kalman Filter [25]. These detections are rasterized for $T_p = 30$ past timesteps, with 200 ms elapsing between timesteps. At any past time t, DRF-NET renders a binary image D_t for pedestrian occupancy where pixel $D_{t,i,j} = 1$ when pixel i, j lies within a convex, bounding octagon of a pedestrian's centroid. Other cells are encoded as 0. Bounding polygons of vehicles, bicycles, buses and other non-pedestrian actors are also rendered in a binary image V_t. In Figure 3-c and Figure 8, we show how temporal information is encoded in the channel dimension of tensors D and V.

To discriminate the pedestrian of interest (PoI) from other actors, a grayscale image R masks the tracklet of the pedestrian to be predicted. As a convention, let the current timestep be $t = 0$. If a pixel i, j is contained within the bounding polygon of the PoI at timestep $t \leq 0$, then $R_{i,j} = 1 + \gamma t$, $\gamma \in (0, T_p^{-1})$. By doing so, the whole PoI tracklet is encoded in a single channel with decaying intensity

for older detections. This encoding allows for variable track lengths. All rasterization channels are rotated for fixed PoI orientation at $t = 0$. We compute orientation with the difference of the last two observed locations.

Encoding semantic map To represent the scene context of the pedestrian, DRF-NET renders map polygons into 15 semantic map channels, collectively denoted as M, where each channel corresponds to a finely differentiated urban surface label. Crosswalks and drivable surfaces (roadways and intersections) are rasterized into separate channels. While sidewalks are not explicitly encoded, non-drivable surfaces are implied by the road map. Three channels indicate traffic light state, classified from the on-vehicle camera with a known traffic light position: the green, red, and yellow light channels each fill the lanes passing through intersections controlled by the corresponding light state. Similarly, lanes leading to yield and stop signs are encoded into channels. Finally, we encode other detailed lanes, such as turn, bike, and bus lanes, and a combined channel for all lane markers. In detail, the 15 channels are as follows:

1. Aggregated road mask, masking all drivable surfaces
2. Masked crosswalks
3. Masked intersections
4. Masked bus lanes
5. Masked bike lanes
6. All lane markers / dividers
7. Masked lanes leading to stop sign
8. Masked lanes leading to yield sign
9. Lanes controlled by red stop light
10. Lanes controlled by yellow light
11. Lanes controlled by green light
12. Lanes without a turn
13. Right-turn lanes
14. Protected left-turn lanes
15. Unprotected left-turn lanes

This information is annotated in a semi-automated fashion in cities where the self-driving vehicle may operate (Section 4.1), and only polygons and polylines are stored.

6.3 Training

Computing negative log likelihood For density visualization in Figure 7, and for computing discrete negative log likelihood metrics in Table 1, the MDN predicted mixture is numerically integrated by a centered approximation with 9 sampling points for each output grid cell of size 0.5×0.5 squared meters. Discretizing the MDN allows an NLL metric to be compared between continuous predictions and discrete predictions.

Optimization In our experiments with manually annotated trajectories, we train our models from scratch using the Adam optimizer [32] with a learning rate of 10^{-5}. When using trajectories from a real perception system, we fine-tune the models learned using the ground truth data to better deal with missing pedestrians and detector/sensor noise. Each training batch includes 2 pedestrian trajectories. All experiments are performed with distributed training on 16 GPUs.

6.4 Metrics

Measuring modality To compute the number of modes (local maxima) in a distribution, we proposed the ModePool$_k, \epsilon$ operator. Our proposed operator in fact overestimates modality for MDNs, especially for the Density Network, at short timescales due to quantization error and the fixed window size. To compute modality of a continuous distribution, we discretize the distribution. When the distributions are very long and narrow, as in Density Network short term predictions, multiple modes can be registered. Despite this overestimation, models with the proposed discrete prediction space (ConvLSTM, DRF-NET) expressed higher multimodality than the MDNs.

6.5 Derivation of Discrete Residual Flow

We derive Equation (3), the discrete residual flow update equation, as an approximation for explicit marginalization of a joint state distribution. According to the law of total probability,

$$p_{\mathbf{x}_t}(x_t \mid \Omega) = \sum_{x_{t-1}} p_{\mathbf{x}_t, \mathbf{x}_{t-1}}(x_t, x_{t-1} \mid \Omega) \tag{10}$$

$$= \sum_{x_{t-1}} p_{\mathbf{x}_t \mid \mathbf{x}_{t-1}}(x_t \mid x_{t-1}, \Omega)\, p_{\mathbf{x}_{t-1}}(x_{t-1} \mid \Omega) \tag{11}$$

Equation (11) can be seen as a recursive update to the previous timestep's state marginal. Recall that $\mathbf{x}_t$ is a categorical random variable over K bins. Instead of representing the pairwise conditional distribution $p(x_t \mid x_{t-1}, \Omega)$ and conducting the summation once per output bin at $O(K^2)$ cost per timestep, we approximate (11) with a pointwise update,

$$p_{\mathbf{x}_t}(x_t \mid \Omega) = \sum_{x_{t-1}} p_{\mathbf{x}_t \mid \mathbf{x}_{t-1}}(x_t \mid x_{t-1}, \Omega)\, p_{\mathbf{x}_{t-1}}(x_{t-1} | \Omega) \tag{11}$$

$$= \left[\sum_{x_{t-1}} \frac{p_{\mathbf{x}_t \mid \mathbf{x}_{t-1}}(x_t \mid x_{t-1}, \Omega)\, p_{\mathbf{x}_{t-1}}(x_{t-1} \mid \Omega)}{p_{\mathbf{x}_{t-1}}(x_t \mid \Omega)} \right] p_{\mathbf{x}_{t-1}}(x_t \mid \Omega) \tag{12}$$

$$\approx \frac{1}{Z_t} \underbrace{\psi_{t;\theta_t}\left(x_t, p_{\mathbf{x}_{t-1}}(\,\cdot \mid \Omega), \Omega\right)}_{\text{Exponentiated residual}} p_{\mathbf{x}_{t-1}}(x_t \mid \Omega) \tag{13}$$

where Z_t is a normalization constant, and $\psi_{t;\theta_t}$ is a parametric approximator for the summation that we refer to as the *residual predictor*. In principle, a sufficiently expressive residual predictor can model the summation exactly. While the residual is applied as a scaling factor in Equation (13), the residual becomes more natural to understand when the recursive definition is expressed in log domain, completing the derivation,

$$\log p_{\mathbf{x}_t}(x_t \mid \Omega) = \log p_{\mathbf{x}_{t-1}}(x_t \mid \Omega) + \log \psi_{t;\theta_t}\left(x_t, p_{\mathbf{x}_{t-1}}(\,\cdot \mid \Omega), \Omega\right) - \log Z_t \tag{14}$$

We construct $\log \psi_{t;\theta_t}$ such that it can be computed in parallel across all locations x_t, and such that the update to $\log p_{\mathbf{x}_t}(\,\cdot \mid \Omega)$ is an elementwise sum followed by normalization. In DRF-NET, $\log \psi_{t;\theta_t}$ is instantiated with a neural network that outputs a 2D image indexable at these locations (Figure 4). Then, the update (14) incurs $O(K)$ cost per timestep.

With this lens, the baseline fully convolutional predictor and the mixture density networks, which assume conditional independence $\mathbf{x}_t \perp \mathbf{x}_{t-1} \mid \Omega$, directly approximate the marginal:

$$\log p_{\mathbf{x}_t}(x_t \mid \Omega) = \log \psi_{t;\theta_t}\left(x_t, \Omega\right) - \log Z_t \tag{15}$$

The baseline ConvLSTM propagates a cell and hidden state between steps and shares parameters of the predictor, without sampling from intermediate marginals:

$$\log p_{\mathbf{x}_t}(x_t \mid \Omega) = \log f_\phi(h_t) - \log Z_t \tag{16}$$

$$h_t, c_t = \psi_\theta\left(p_{\mathbf{x}_{t-1}}(\,\cdot \mid \Omega), h_{t-1}, c_{t-1}\right)$$

$$h_0 = \mathcal{F}(\Omega), c_0 \text{ is a parameter}$$

Discrete residual flow retains most of the benefits of the independence assumption, *i.e.* tractable marginal distribution estimation and parallelizability, with update more closely resembling a Markov chain. However, there is no sampling between timesteps. As we established in our ablation study (Table 3), applying DRF Eq. (14) outperforms the baseline fully convolutional predictor according to Eq. (15) and the ConvLSTM update, Eq. (16).