Haar Wavelet based Block Autoregressive Flows for Trajectories

Apratim Bhattacharyya[1], **Christoph-Nikolas Straehle**[2], **Mario Fritz**[3], and **Bernt Schiele**[1]

[1]Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
[2]Bosch Center for Artificial Intelligence, Renningen, Germany
[3]CISPA Helmholtz Center for Information Security, Saarland Informatics Campus, Saarbrücken, Germany

Abstract

Prediction of trajectories such as that of pedestrians is crucial to the performance of autonomous agents. While previous works have leveraged conditional generative models like GANs and VAEs for learning the likely future trajectories, accurately modeling the dependency structure of these multimodal distributions, particularly over long time horizons remains challenging. Normalizing flow based generative models can model complex distributions admitting exact inference. These include variants with split coupling invertible transformations that are easier to parallelize compared to their autoregressive counterparts. To this end, we introduce a novel Haar wavelet based block autoregressive model leveraging split couplings, conditioned on coarse trajectories obtained from Haar wavelet based transformations at different levels of granularity. This yields an exact inference method that models trajectories at different spatio-temporal resolutions in a hierarchical manner. We illustrate the advantages of our approach for generating diverse and accurate trajectories on two real-world datasets – Stanford Drone and Intersection Drone.

1 Introduction

Anticipation is a key competence for autonomous agents such as self-driving vehicles to operate in the real world. Many such tasks involving anticipation can be cast as trajectory prediction problems, *e.g.*anticipation of pedestrian behaviour in urban driving scenarios. To capture the uncertainty of the real world, it is crucial to model the distribution of likely future trajectories. Therefore recent works [4, 3, 28, 36] have focused on modeling the distribution of likely future trajectories using either generative adversarial networks (GANs, [15]) or variational autoencoders (VAEs, [24]). However, GANs are prone to mode collapse and the performance of VAEs depends on the tightness of the variational lower bound on the data log-likelihood which is hard to control in practice [9, 20]. This makes it difficult to accurately model the distribution of likely future trajectories.

Normalizing flow based exact likelihood models [12, 13, 23] have been considered to overcome these limitations of GANs and VAEs in the context of image synthesis. Building on the success of these methods, recent approaches have extended the flow models for density estimation of sequential data *e.g.*video [26] and audio [21]. Yet, VideoFlow [26] is autoregressive in the temporal dimension which results in the prediction errors accumulating over time [27] and reduced efficiency in sampling. Furthermore, FloWaveNet [21] extends flows to audio sequences with odd-even splits along the temporal dimension, encoding only *local* dependencies [5, 20, 25]. We address these challenges of flow based models for trajectory generation and develop an exact inference framework to accurately model future trajectory sequences by harnessing long-term spatio temporal structure in the underlying trajectory distribution.

Machine Learning for Autonomous Driving Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

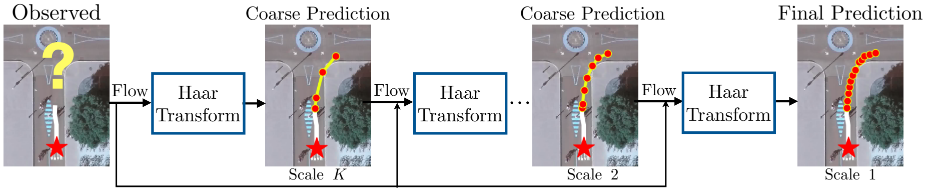


Figure 1: Our normalizing flow based model uses a Haar wavelet based decomposition to block autoregressively model trajectories at K coarse-to-fine scales.

In this work, we propose *HBA-Flow*, an exact inference model with coarse-to-fine block autoregressive structure to encode long term spatio-temporal correlations for multimodal trajectory prediction. The advantage of the proposed framework is that multimodality can be captured over long time horizons by sampling trajectories at coarse-to-fine spatial and temporal scales (Fig. 1). Our contributions are: 1. we introduce a block autoregressive exact inference model using Haar wavelets where flows applied at a certain scale are conditioned on coarse trajectories from previous scale. The trajectories at each level are obtained after the application of Haar wavelet based transformations, thereby modeling long term spatio-temporal correlations. 2. Our HBA-Flow model, by virtue of block autoregressive structure, integrates a multi-scale block autoregressive prior which further improves modeling flexibility by encoding dependencies in the latent space. 3. Furthermore, we show that compared to fully autoregressive approaches [26], our HBA-Flow model is computationally more efficient as the number of sampling steps grows logarithmically in trajectory length. 4. We demonstrate the effectiveness of our approach for trajectory prediction on Stanford Drone and Intersection Drone, with improved accuracy over long time horizons.

2 Related Work

Pedestrian Trajectory Prediction. Work on traffic participant prediction dates back to the Social Forces model [18]. More recent works [1, 18, 39, 34] consider the problem of traffic participant prediction in a social context, by taking into account interactions among traffic participants. Notably, Social LSTM [1] introduces a social pooling layer to aggregate interaction information of nearby traffic participants. An efficient extension of the social pooling operation is developed in [10] and alternate instance and category layers to model interactions in [29]. Weighted interactions are proposed in [7]. In contrast, a multi-agent tensor fusion scheme is proposed in [41] to capture interactions. An attention based model to effectively integrate visual cues in path prediction tasks is proposed in [35]. However, these methods mostly assume a deterministic future and do not directly deal with the challenges of uncertainty and multimodality.

Generative Modeling of Trajectories. To deal with the challenges of uncertainty and multimodality in anticipating future trajectories, recent works employ either conditional VAEs or GANs to capture the distribution of future trajectories. This includes, a conditional VAE based model with a RNN based refinement module [28], a VAE based model [14] that "personalizes" prediction to individual agent behavior, a diversity enhancing "Best of Many" loss [3] to better capture multimodality with VAEs, an expressive normalizing flow based prior for conditional VAEs [4] among others. However, VAE based models only maximize a lower bound on the data likelihood, limiting their ability to effectively model trajectory data. Other works, use GANs [16, 41, 36] to generate socially compliant trajectories. GANs lead to missed modes of the data distribution. Additionally, [33, 11] introduce push-forward policies and motion planning for generative modeling of trajectories. Determinantal point processes are used in [40] to better capture diversity of trajectory distributions. The work of [30] shows that additionally modeling the distribution of trajectory end points can improve accuracy. However, it is unclear if the model of [30] can be used for predictions across variable time horizons. In contrast to these approaches, in this work we directly maximize the exact likelihood of the trajectories, thus better capturing the underlying true trajectory distribution.

Autoregressive Models. Autoregressive exact inference models like PixelCNN [38] have shown promise in generative modeling. Autoregressive models for sequential data includes a convolutional autoregressive model [37] for raw audio and an autoregressive method for video frame prediction [26]. In particular, for sequential data involving trajectories, recent works [31] propose an autoregressive method based on visual sources. The main limitation of autoregressive approaches is that the models are difficult to parallelize. Moreover, for sequential data, errors tend to accumulate over time [27].

Normalizing Flows. Split coupling normalizing flow models with affine transformations [12] offer computationally efficient tractable Jacobians. Recent methods [13, 23] have therefore focused on split coupling flows which are easier to parallelize. Flow models are extended in [13] to multiscale architecture and the modeling capacity of flow models is further improved in [23] by introducing 1×1 convolution. Recently, flow models with more complex invertible components [8, 19] have been leveraged for generative modeling of images. Recent works like FloWaveNet [21] and VideoFlow [21] adapt the multi-scale architecture of Glow [23] with sequential latent spaces to model sequential data, for raw audio and video frames respectively. However, these models still suffer from the limited modeling flexibility of the split coupling flows. The "squeeze" spatial pooling operation in [23] is replaced with a Haar wavelet based downsampling scheme in [2] along the spatial dimensions. Although this leads to improved results on image data, this operation is not particularly effective in case of sequential data as it does not influence temporal receptive fields for trajectories – crucial for modeling long-term temporal dependencies. Therefore, Haar wavelet downsampling of [2] does not lead to significant improvement in performance on sequential data (also observed empirically). In this work, instead of employing Haar wavelets as a downsampling operation for reducing spatial resolution [2] in split coupling flows, we formulate a coarse-to-fine block autoregressive model where Haar wavelets produce trajectories at different spatio-temporal resolutions.

3 Block Autoregressive Modeling of Trajectories

In this work, we propose a coarse-to-fine block autoregressive exact inference model, *HBA-Flow*, for trajectory sequences. We first provide an overview of conditional normalizing flows which form the backbone of our HBA-Flow model. To extend normalizing flows for trajectory prediction, we introduce an invertible transformation based on Haar wavelets which decomposes trajectories into K coarse-to-fine scales (Fig. 1). This is beneficial for expressing long-range spatio-temporal correlations as coarse trajectories provide global context for the subsequent finer scales. Our proposed HBA-Flow framework integrates the coarse-to-fine transformations with invertible split coupling flows where it block autoregressively models the transformed trajectories at K scales.

3.1 Conditional Normalizing Flows for Sequential Data

We base our HBA-Flow model on normalizing flows [12] which are a type of exact inference model. In particular, we consider the transformation of the conditional distribution $p(\mathbf{y}|\mathbf{x})$ of trajectories $\mathbf{y}$ to a distribution $p(\mathbf{z}|\mathbf{x})$ over $\mathbf{z}$ with conditional normalizing flows [2, 4] using a sequence of n transformations $g_i : \mathbf{h}_{i-1} \mapsto \mathbf{h}_i$, with $\mathbf{h}_0 = \mathbf{y}$ and parameters θ_i,

$$\mathbf{y} \overset{g_1}{\longleftrightarrow} \mathbf{h}_1 \overset{g_2}{\longleftrightarrow} \mathbf{h}_2 \cdots \overset{g_n}{\longleftrightarrow} \mathbf{z}. \tag{1}$$

Given the Jacobians $\mathbf{J}_{\theta_i} = {}^{\partial \mathbf{h}_i}/_{\partial \mathbf{h}_{i-1}}$ of the transformations g_i, the exact likelihoods can be computed with the change of variables formula,

$$\log p_\theta(\mathbf{y}|\mathbf{x}) = \log p(\mathbf{z}|\mathbf{x}) + \sum_{i=1}^{n} \log |\det \mathbf{J}_{\theta_i}|, \tag{2}$$

Given that the density $p(\mathbf{z}|\mathbf{x})$ is known, the likelihood over $\mathbf{y}$ can be computed exactly. Recent works [12, 13, 23] consider invertible split coupling transformations g_i as they provide a good balance between efficiency and modeling flexibility. In (conditional) split coupling transformations, the input $\mathbf{h}_i$ is split into two halves $\mathbf{l}_i,\ \mathbf{r}_i$, and g_i applies an invertible transformation only on $\mathbf{l}_i$ leaving $\mathbf{r}_i$ unchanged. The transformation parameters of $\mathbf{l}_i$ are dependent on $\mathbf{r}_i$ and $\mathbf{x}$, thus $\mathbf{h}_{i+1} = [g_{i+1}(\mathbf{l}_i|\mathbf{r}_i, \mathbf{x}), \mathbf{r}_i]$. The advantage of (conditional) split coupling flows is that both inference and sampling are parallelizable when the transformations g_{i+1} have an efficient closed form expression of the inverse g_{i+1}^{-1}, *e.g.*affine [23] or non-linear squared [42] and unlike residual flows [8].

As most of the prior work, *e.g.*[2, 12, 13, 23], considers split coupling flows g_i that are designed to deal with fixed length data, these models are not directly applicable to data of variable length such as trajectories. Moreover, recall that for variable length sequences, while VideoFlow [26] utilizes split coupling based flows to model the distribution at each time-step, it is still fully autoregressive in the

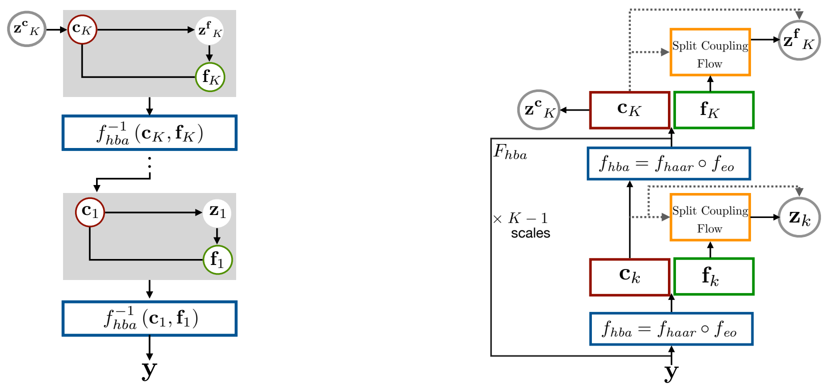

$\mathbf{z}^{\mathbf{c}}{}_K$
$\mathbf{c}_K$
$\mathbf{z}^{\mathbf{f}}{}_K$
$\mathbf{f}_K$
$f_{hba}^{-1}(\mathbf{c}_K, \mathbf{f}_K)$
$\mathbf{c}_1$
$\mathbf{z}_1$
$\mathbf{f}_1$
$f_{hba}^{-1}(\mathbf{c}_1, \mathbf{f}_1)$
$\mathbf{y}$
Split Coupling Flow
$\mathbf{z}^{\mathbf{f}}{}_K$
$\mathbf{z}^{\mathbf{c}}{}_K$
$\mathbf{c}_K$
$\mathbf{f}_K$
F_{hba}
$f_{hba} = f_{haar} \circ f_{eo}$
$\times$ $K-1$ scales
Split Coupling Flow
$\mathbf{z}_k$
$\mathbf{c}_k$
$\mathbf{f}_k$
$f_{hba} = f_{haar} \circ f_{eo}$
$\mathbf{y}$

Figure 2: Left: *HBA-Flow* generative model with the Haar wavelet [17] based representation F_{hba}. Right: Our multi-scale *HBA-Flow* model with K scales of Haar based transformation.

temporal dimension, thus offering limited computational efficiency. FloWaveNets [21] split $\mathbf{l}_i$ and $\mathbf{r}_i$ along even-odd time-steps for audio synthesis. This even-odd formulation of the split operation along with the inductive bias [25, 20, 5] of split coupling based flow models is limited when expressing local and global dependencies which are crucial for capturing multimodality of the trajectories over long time horizons. Next, we introduce our invertible transformation based on Haar wavelets to model trajectories at various coarse-to-fine levels to address the shortcomings of prior flow based methods [26, 21] for sequential data.

3.2 Haar Wavelet based Invertible Transform

Haar wavelet transform allows for a simple and easy to compute coarse-to-fine frequency decomposed representation with a finite number of components unlike alternatives *e.g.*Fourier transformations [32]. In our HBA-Flow framework, we construct a transformation F_{hba} comprising of mappings f_{hba} recursively applied across K scales. With this transformation, trajectories can be encoded at different levels of granularity along the temporal dimension. We now formalize invertible function f_{hba} and its multi-scale Haar wavelet based composition F_{hba}.

Single Scale Invertible Transformation. Consider the trajectory at scale k as $\mathbf{y}_k = [\mathbf{y}_k^1, \cdots, \mathbf{y}_k^{T_k}]$, where T_k is the number of timesteps of trajectory $\mathbf{y}_k$. Here, at scale $k = 1$, $\mathbf{y_1} = \mathbf{y}$ is the input trajectory. Each element of the trajectory is a vector, $\mathbf{y}_k^j \in \mathbb{R}^d$ encoding spatial information of the traffic participant. Our proposed invertible transformation f_{hba} at any scale k is a composition, $f_{hba} = f_{haar} \circ f_{eo}$. First, f_{eo} transforms the trajectory into even $(\mathbf{e}_k)$ and odd $(\mathbf{o}_k)$ downsampled trajectories,

$$f_{eo}(\mathbf{y}_k) = \mathbf{e}_k, \mathbf{o}_k \text{ where, } \mathbf{e}_k = [\mathbf{y}_k^2, \cdots, \mathbf{y}_k^{T_k}] \text{ and } \mathbf{o}_k = [\mathbf{y}_k^1, \cdots, \mathbf{y}_k^{T_k - 1}]. \tag{3}$$

Next, f_{haar} takes as input the even $(\mathbf{e}_k)$ and odd $(\mathbf{o}_k)$ downsampled trajectories and transforms them into coarse $(\mathbf{c}_k)$ and fine $(\mathbf{f}_k)$ downsampled trajectories using a scalar "mixing" parameter α. In detail,

$$\begin{aligned} f_{haar}(\mathbf{e}_k, \mathbf{o}_k) = \mathbf{f}_k, \mathbf{c}_k \text{ where, } \mathbf{c}_k &= (1-\alpha)\mathbf{e}_k + \alpha\mathbf{o}_k \quad \text{and} \\ \mathbf{f}_k &= \mathbf{o}_k - \mathbf{c}_k = (1-\alpha)\mathbf{o}_k + (\alpha - 1)\mathbf{e}_k \end{aligned} \tag{4}$$

where, the coarse $(\mathbf{c}_k)$ trajectory is the element-wise weighted average of the even $(\mathbf{e}_k)$ and odd $(\mathbf{o}_k)$ downsampled trajectories and the fine $(\mathbf{f}_k)$ trajectory is the element-wise difference to the coarse downsampled trajectory. The coarse trajectories $(\mathbf{c}_k)$ provide global context for finer scales in our block autoregressive approach, while the fine trajectories $(\mathbf{f}_k)$ encode details at multiple scales. We now discuss the invertibilty of this transformation f_{hba} and compute the Jacobian.

Lemma 1. *The generalized Haar transformation $f_{hba} = f_{haar} \circ f_{eo}$ is invertible for $\alpha \in [0, 1)$ and the determinant of the Jacobian of the transformation $f_{hba} = f_{haar} \circ f_{eo}$ for sequence of length T_k with $\mathbf{y}_k^j \in \mathbb{R}^d$ is* $\det \mathbf{J}_{hba} = (1-\alpha)^{(d \cdot T_k)/2}$.

We provide the proof in the appendix. This property allows our HBA-Flow model to exploit f_{hba} for spatio-temporal decomposition of the trajectories $\mathbf{y}$ while remaining invertible with a tractable

Jacobian for exact inference. Next, we use this transformation f_{hba} to build the coarse-to-fine multi-scale Haar wavelet based transformation F_{hba} and discuss its properties.

Multi-scale Haar Wavelet based Transformation. To construct our generalized Haar wavelet based transformation F_{hba}, the mapping f_{hba} is applied recursively at K scales (Fig. 2, left). The transformation f_{hba} at a scale k applies a low and a high pass filter pair on the input trajectory $\mathbf{y}_k$ resulting in the coarse trajectory $\mathbf{c}_k$ and the fine trajectory $\mathbf{f}_k$ with high frequency details. The coarse (spatially and temporally sub-sampled) trajectory ($\mathbf{c}_k$) at scale k is then further decomposed by using it as the input trajectory $\mathbf{y}_{k+1} = \mathbf{c}_k$ to f_{hba} at scale $k+1$. This is repeated at K scales, resulting in the complete Haar wavelet transformation $F_{hba}(\mathbf{y}) = [\mathbf{f}_1, \cdots, \mathbf{f}_K, \mathbf{c}_K]$ which captures details at multiple (K) spatio-temporal scales. The finest scale $\mathbf{f}_1$ models high-frequency spatio-temporal information of the trajectory $\mathbf{y}$. The subsequent scales $\mathbf{f}_k$ represent details at coarser levels, with $\mathbf{c}_K$ being the coarsest transformation which expresses the "high-level" spatio-temporal structure of the trajectory (Fig. 1).

Next, we show that the number of scales K in F_{hba} is upper bounded by the logarithm of the length of the sequence. This implies that F_{hba}, when integrated in the multi-scale block auto-regressive model provides a computationally efficient setup for generating trajectories.

Lemma 2. *The number of scales K of the Haar wavelet based representation F_{hba} is $K \leq \log(T_1)$, for an initial input sequence $\mathbf{y}_1$ of length T_1.*

Proof. The Haar wavelet based transformation f_{hba} halves the length of trajectory $\mathbf{y}_k$ at each level k. Thus, for an initial input sequence $\mathbf{y}_1$ of length T_1, the length of the coarsest level K in $F_{hba}(\mathbf{y})$ is $|\mathbf{c}_K| = {}^{T_1}/_{2^K} \geq 1$. Thus, $K \leq \log(T_1)$. □

3.3 Haar Block Autoregressive Framework

HBA-Flow model. We illustrate our HBA-Flow model in Fig. 2. Our HBA-Flow model first transforms the trajectories $\mathbf{y}$ using F_{hba}, where the invertible transform f_{hba} is recursively applied on the input trajectory $\mathbf{y}$ to obtain $\mathbf{f}_k$ and $\mathbf{c}_k$ at scales $k \in \{1, \cdots, K\}$. Therefore, the log-likelihood of a trajectory $\mathbf{y}$ under our HBA-Flow model can be expressed using the change of variables formula as,

$$\begin{aligned} \log(p_\theta(\mathbf{y}|\mathbf{x})) &= \log(p_\theta(\mathbf{f}_1, \mathbf{c}_1|\mathbf{x})) + \log|\det(\mathbf{J}_{hba})_1| \\ &= \log(p_\theta(\mathbf{f}_1, \cdots, \mathbf{f}_K, \mathbf{c}_K|\mathbf{x})) + \sum_{i=1}^{K} \log|\det(\mathbf{J}_{hba})_i|. \end{aligned} \tag{5}$$

Next, our HBA-Flow model factorizes the distribution of fine trajectories w.l.o.g. such that $\mathbf{f}_k$ at level k is conditionally dependent on the representations at scales $k+1$ to K,

$$\begin{aligned} \log(p_\theta(\mathbf{f}_1, \cdots, \mathbf{f}_K, \mathbf{c}_K|\mathbf{x})) = \log(p_\theta(\mathbf{f}_1|\mathbf{f}_2, \cdots, \mathbf{f}_K, \mathbf{c}_K, \mathbf{x})) + \cdots \\ + \log(p_\theta(\mathbf{f}_K|\mathbf{c}_K, \mathbf{x})) + \log(p_\theta(\mathbf{c}_K|\mathbf{x})). \end{aligned} \tag{6}$$

Finally, note that $[\mathbf{f}_{k+1}, \cdots, \mathbf{f}_K, \mathbf{c}_K]$ is the output of the (bijective) transformation $F_{hba}(\mathbf{c}_k)$ where f_{hba} is recursively applied to $\mathbf{c}_k = \mathbf{y}_{k+1}$ at scales $\{k+1, \cdots, K\}$. Thus HBA-Flow equivalently models $p_\theta(\mathbf{f}_k|\mathbf{f}_{k+1}, \cdots, \mathbf{c}_K, \mathbf{x})$ as $p_\theta(\mathbf{f}_k|\mathbf{c}_k, \mathbf{x})$,

$$\begin{aligned} \log(p_\theta(\mathbf{y}|\mathbf{x})) =& \log(p_\theta(\mathbf{f}_1|\mathbf{c}_1, \mathbf{x})) + \cdots + \log(p_\theta(\mathbf{f}_K|\mathbf{c}_K, \mathbf{x})) \\ &+ \log(p_\theta(\mathbf{c}_K|\mathbf{x})) + \sum_{i=1}^{K} \log|\det(\mathbf{J}_{hba})_i|. \end{aligned} \tag{7}$$

Therefore, as illustrated in Fig. 2 (right), our HBA-Flow models the distribution of each of the fine components $\mathbf{f}_k$ block autoregressively conditioned on the coarse representation $\mathbf{c}_k$ at that level. The distribution $p_\theta(\mathbf{f}_k|\mathbf{c}_k, \mathbf{x})$ at each scale k is modeled using invertible conditional split coupling flows (Fig. 2, right) [21], which transform the input distribution to the distribution over latent "priors" $\mathbf{z}_k$. This enables our framework to model variable length trajectories. The log-likelihood with our HBA-Flow approach can be expressed using the change of variables formula as,

$$\log(p_\theta(\mathbf{f}_k|\mathbf{c}_k, \mathbf{x})) = \log(p_\phi(\mathbf{z}_k|\mathbf{c}_k, \mathbf{x})) + \log|\det(\mathbf{J}_{sc})_k|. \tag{8}$$

$\log|\det(\mathbf{J}_{sc})_k|$ is the log determinant of Jacobian $(\mathbf{J}_{sc})_k$ of the split coupling flow at level k. Thus, the likelihood of a trajectory $\mathbf{y}$ under our HBA-Flow model is expressed exactly, Eqs. (7) and (8).

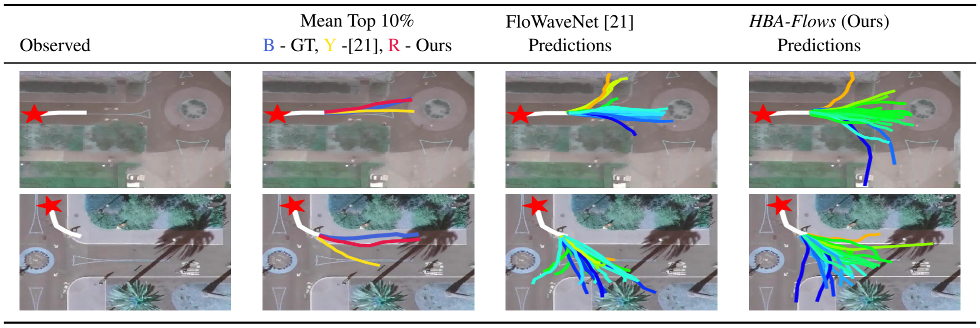


Figure 3: Mean top 10% predictions (Blue - Groudtruth, Yellow - FloWaveNet [21], Red - Our *HBA-Flow* model) and predictive distributions on Stanford Drone dataset. The predictions of HBA-Flow are more diverse and better capture the multimodality the future trajectory distribution.

The key advantage of our approach is that after spatial and temporal downsampling of coarse scales, it is easier to model long-term spatio-temporal dependencies. Moreover, conditioning the flows at each scale on the coarse trajectory provides global context as the downsampled coarse trajectory effectively increases the spatio-temporal receptive field. This enables our HBA-Flows better capture multimodality in the distribution of likely future trajectories.

HBA-Prior. Complex multimodel priors can considerably increase the modeling flexibility of generative models [4, 21, 26]. The block autoregressive structure of our HBA-Flow model allows us introduce a Haar block autoregressive prior (HBA-Prior) over $\mathbf{z} = [\mathbf{z}_1, \cdots, \mathbf{z^f}_K, \mathbf{z^c}_K]$ in Eq. (8), where $\mathbf{z}_k$ is the latent representation for scales $k \in \{1, \cdots, K-1\}$ and $\mathbf{z^f}_K, \mathbf{z^c}_K$ are the latents for the coarse and fine representations scales K. The log-likelihood of the prior factorizes as,

$$\begin{aligned}\log(p_\phi(\mathbf{z}|\mathbf{x})) = \log(p_\phi(\mathbf{z}_1|\mathbf{z}_2, \cdots, \mathbf{z^f}_K, \mathbf{z^c}_K, \mathbf{x})) + \cdots \\ + \log(p_\phi(\mathbf{z^f}_K|\mathbf{z^c}_K, \mathbf{x})) + \log(p_\phi(\mathbf{z^c}_K|\mathbf{x})).\end{aligned} \tag{9}$$

Each coarse level representation $\mathbf{c}_k$ is the output of a bijective transformation of the latent variables $[\mathbf{z}_{k+1}, \cdots, \mathbf{z^f}_K\, \mathbf{z^c}_K]$ through the invertible split coupling flows and the transformations f_{hba} at scales $\{k+1, \cdots, K\}$. Thus, HBA-Prior models $p_\phi(\mathbf{z}_k|\mathbf{z}_{k+1}, \cdots, \mathbf{z^f}_K, \mathbf{z^c}_K, \mathbf{x})$ as $p_\phi(\mathbf{z}_k|\mathbf{c}_k, \mathbf{x})$ at every scale (Fig. 2, left). The log-likelihood of the prior can also be expressed as,

$$\begin{aligned}\log(p_\phi(\mathbf{z}|\mathbf{x})) = \log(p_\phi(\mathbf{z}_1|\mathbf{c}_1, \mathbf{x})) + \cdots + \log(p_\phi(\mathbf{z}_{K-1}|\mathbf{c}_{K-1}, \mathbf{x})) \\ + \log(p_\phi(\mathbf{z^f}_K|\mathbf{c}_K, \mathbf{x})) + \log(p_\phi(\mathbf{z^c}_K|\mathbf{x})).\end{aligned} \tag{10}$$

We model $p_\phi(\mathbf{z}_k|\mathbf{c}_k, \mathbf{x})$ as conditional normal distributions which are multimodal as a result of the block autoregressive structure. In comparison to the fully autoregressive prior in [26], our HBA-Prior is efficient as it requires only $\mathcal{O}(\log(T_1))$ sampling steps.

Analysis of Sampling Time. From Eq. (6) and Fig. 2 (left), our HBA-Flow model autoregressively factorizes across the fine components $\mathbf{f}_k$ at K scales. From Lemma 2, $K \leq \log(T_1)$. At each scale our HBA-Flow samples the fine components $\mathbf{f}_k$ using split coupling flows, which are easy to parallelize. Thus, given enough parallel resources, our HBA-Flow model requires maximum $K \leq \log(T_1)$ *i.e.* $\mathcal{O}(\log(T_1))$ sampling steps and is significantly more efficient compared to fully autoregressive approaches *e.g.*VideoFlow [26], which require $\mathcal{O}(T_1)$ steps.

4 Experiments

We evaluate our approach for trajectory prediction on two challenging real world datasets – Stanford Drone [34] and Intersection Drone [6]. These datasets contain trajectories of traffic participants including pedestrians, bicycles, cars recorded from an aerial platform. The distribution of likely future trajectories is highly multimodal due to the complexity of the traffic scenarios *e.g.*at intersections. We are primarily interested in measuring the match of the learned distribution to the true distribution. Therefore, we follow [4, 3, 28, 31] and use Euclidean error of the top 10% of samples (predictions) and the (negative) conditional log-likelihood (-CLL) metrics. The Euclidean error of the top 10% of samples measures the coverage of all modes of the target distribution and is relatively robust to random guessing as shown in [4]. We provide architecture details in the appendix.

Table 1: Five fold cross validation on the Stanford Drone dataset. Lower is better for all metrics. Visual refers to additional conditioning on the last observed frame. Top: state of the art, Middle: Baselines and ablations, Bottom: Our HBA-Flow.

Method	Visual	Er @ 1sec	Er @ 2sec	Er @ 3sec	Er @ 4sec	-CLL	Speed
"Shotgun" [31]	–	0.7	1.7	3.0	4.5	91.6	–
DESIRE-SI-IT4 [28]	✓	1.2	2.3	3.4	5.3	–	–
STCNN [31]	✓	1.2	2.1	3.3	4.6	–	–
BMS-CVAE [3]	✓	0.8	1.7	3.1	4.6	126.6	58
CF-VAE [4]	–	**0.7**	1.5	2.5	3.6	84.6	47
CF-VAE [4]	✓	**0.7**	1.5	2.4	3.5	84.1	88
Auto-regressive [26]	–	**0.7**	1.5	2.6	3.7	86.8	134
FloWaveNet [21]	–	**0.7**	1.5	2.5	3.6	84.5	**38**
FloWaveNet [21] + HWD [2]	–	**0.7**	1.5	2.5	3.6	84.4	**38**
FloWaveNet [21]	✓	**0.7**	1.5	2.4	3.5	84.1	77
HBA-Flow (Ours)	–	**0.7**	1.5	2.4	3.4	84.1	41
HBA-Flow + Prior (Ours)	–	**0.7**	**1.4**	**2.3**	3.3	83.4	43
HBA-Flow + Prior (Ours)	✓	**0.7**	**1.4**	**2.3**	**3.2**	**83.1**	81

4.1 Stanford Drone

We use the standard five-fold cross validation evaluation protocol [4, 3, 28, 31] and predict the trajectory up to 4 seconds into the future. We use the Euclidean error of the top 10% of predicted trajectories at the standard (1/5) resolution using 50 samples and the CLL metric in Table 1. We additionally report sampling time for a batch of 128 samples in milliseconds. We compare our HBA-Flow model to the following state-of-the-art models: The handcrafted "Shotgun" model [31], the conditional VAE based models of [3, 4, 28] and the autoregressive STCNN model [31]. We additionally include the various exact inference baselines for modeling trajectory sequences: the autoregressive flow model of VideoFlow [26], FloWaveNet [21] (without our Haar wavelet based block autoregressive structure), FloWaveNet [21] with the Haar wavelet downsampling of [2] (FloWaveNet + HWD), our HBA-Flow model with a Gaussian prior (without our HBA-Prior). The FloWaveNet [21] baselines serves as ideal ablations to measure the effectiveness of our block autoregressive HBA-Flow model. For fair comparison, we use two scales (levels) $K = 2$ with eight non-linear squared split coupling flows [42] each, for both our HBA-Flow and FloWaveNet [21] models. Following [4, 31] we additionally experiment with conditioning on the last observed frame using a attention based CNN (indicated by "Visual" in Table 1).

We observe from Table 1 that our HBA-Flow model outperforms both state-of-the-art models and baselines. In particular, our HBA-Flow model outperforms the conditional VAE based models of [4, 3, 28] in terms of Euclidean distance and -CLL. Further, our HBA-Flow exhibits competitive sampling speeds. This shows the advantage of exact inference in the context of generative modeling of trajectories – leading to better match to the groundtruth distribution. Our HBA-Flow model generates accurate trajectories compared to the VideoFlow [26] baseline. This is because unlike VideoFlow, errors do not accumulate in the temporal dimension of HBA-Flow. Our HBA-Flow model outperforms the FloWaveNet model of [21] with comparable sampling speeds demonstrating the effectiveness of the coarse-to-fine block autoregressive structure of our HBA-Flow model in capturing long-range spatio-temporal dependencies. This is reflected in the predictive distributions and the top 10% of predictions of our HBA-Flow model in comparison with FloWaveNet [21] in Fig. 3. The predictions of our HBA-Flow model are more diverse and can more effectively capture the multimodality of the trajectory distributions especially at complex traffic situations *e.g.*intersections and crossings. We provide additional examples in the appendix. We also observe in Table 1 that the addition of Haar wavelet downsampling [2] to FloWaveNets [21] (FloWaveNet + HWD) does not significantly improve performance. This illustrates that Haar wavelet downsampling as used in [2] is not effective in case of sequential trajectory data as it is primarily a spatial pooling operation for image data. Finally, our ablations with Gaussian priors (HBA-Flow) additionally demonstrate the effectiveness of our HBA-Prior (HBA-Flow + Prior) with improvements with respect to accuracy. We

Table 2: Evaluation on the Stanford Drone using the split of [11, 36, 41].

Method	mADE ↓	mFDE ↓
SocialGAN [16]	27.2	41.4
MATF GAN [41]	22.5	33.5
SoPhie [36]	16.2	29.3
Goal Prediction [11]	15.7	28.1
CF-VAE [4]	12.6	22.3
HBA-Flow + Prior (Ours)	**10.8**	**19.8**

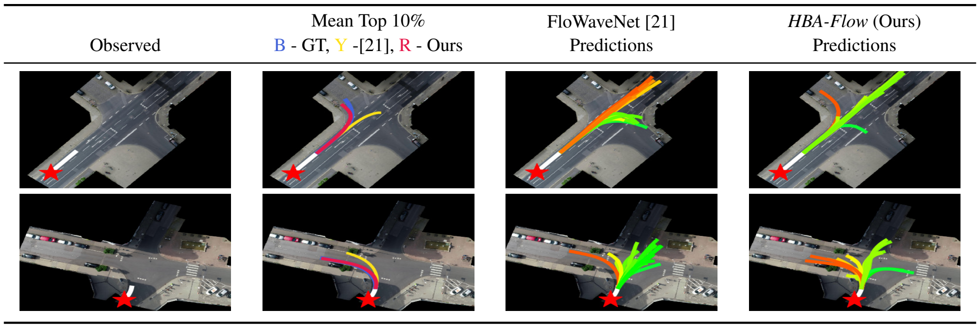


Figure 4: Mean top 10% predictions (Blue - Groudtruth, Yellow - FloWaveNet [21], Red - Our *HBA-Flow* model) and predictive distributions on Intersection Drone dataset. The predictions of our HBA-Flow model are more diverse and better capture the modes of the future trajectory distribution.

Table 3: Five fold cross validation on the Intersection Drone dataset.

Method	Er @ 1sec	Er @ 2sec	Er @ 3sec	Er @ 4sec	Er @ 5sec	-CLL
BMS-CVAE [3]	0.25	0.67	1.14	1.78	2.63	26.7
CF-VAE [4]	0.24	0.55	0.93	1.45	2.21	21.2
FloWaveNet [21]	0.23	0.50	0.85	1.31	1.99	19.8
FloWaveNet [21] + HWD [2]	0.23	0.50	0.84	1.29	1.96	19.5
HBA-Flow + Prior (Ours)	**0.19**	**0.44**	**0.82**	**1.21**	**1.74**	**17.3**

further include a comparison using the evaluation protocol of [34–36, 11] in Table 2. Here, only a single train/test split is used. We follow [4, 11] and use the minimum average displacement error (mADE) and minimum final displacement error (mFDE) as evaluation metrics. Similar to [4, 11] the minimum is calculated over 20 samples. Our HBA-Flow model outperforms the state-of-the-art demonstrating the effectiveness of our approach.

4.2 Intersection Drone

We further include experiments on the Intersection Drone dataset [6]. The dataset consists of trajectories of traffic participants recorded at German intersections. In comparison to the Stanford Drone dataset, the trajectories in this dataset are typically longer. Moreover, unlike the Stanford Drone dataset which is recorded at a University Campus, this dataset covers more "typical" traffic situations. Here, we follow the same evaluation protocol as in Stanford Drone dataset and perform a five-fold cross validation and evaluate up to 5 seconds into the future. We report the results in Table 3. We use the strongest baselines from Table 1 for comparison to our HBA-Flow + Prior model (with our HBA-Prior), with three scales, each having eight non-linear squared split coupling flows [42]. For fair comparison, we compare with a FloWaveNet [21] model with three levels and eight non-linear squared split coupling flows per level. We again observe that our HBA-Flow leads to much better improvement with respect to accuracy over the FloWaveNet [21] model. Furthermore, the performance gap between HBA-Flow and FloWaveNet increases with longer time horizons. This shows that our approach can better encode spatio-temporal correlations. The qualitative examples in Fig. 4 from both models show that our HBA-Flow model generates diverse trajectories and can better capture the modes of the future trajectory distribution, thus demonstrating the advantage of the block autoregressive structure of our HBA-Flow model. We also see that our HBA-Flow model outperforms the CF-VAE model [4], illustrating the advantage of exact inference.

5 Conclusion

In this work, we presented a novel block autoregressive *HBA-Flow* framework taking advantage of the representational power of autoregressive models and the efficiency of invertible split coupling flow models. Our approach can better represent the multimodal trajectory distributions capturing the long range spatio-temporal correlations. Moreover, the block autoregressive structure of our approach provides for efficient $\mathcal{O}(\log(T))$ inference and sampling. We believe that accurate and computationally efficient invertible models that allow exact likelihood computations and efficient sampling present a promising direction of research of anticipation problems in autonomous systems.

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 *CVPR*, 2016.

[2] L. Ardizzone, J. Kruse, C. Lüth, N. Bracher, C. Rother, and U. Köthe. Conditional invertible neural networks for diverse image-to-image translation. 2019.

[3] A. Bhattacharyya, B. Schiele, and M. Fritz. Accurate and diverse sampling of sequences based on a "best of many" sample objective. In *CVPR*, 2018.

[4] A. Bhattacharyya, M. Hanselmann, M. Fritz, B. Schiele, and C.-N. Straehle. Conditional flow variational autoencoders for structured sequence prediction. In *BDL@NeurIPS*, 2019.

[5] A. Bhattacharyya, S. Mahajan, M. Fritz, B. Schiele, and S. Roth. Normalizing flows with multi-scale autoregressive priors. In *CVPR*, 2020.

[6] J. Bock, R. Krajewski, T. Moers, L. Vater, S. Runde, and L. Eckstein. The ind dataset: A drone dataset of naturalistic vehicle trajectories at german intersections. *arXiv preprint arXiv:1911.07602*, 2019.

[7] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *CVPR*, 2019.

[8] T. Q. Chen, J. Behrmann, D. K. Duvenaud, and J.-H. Jacobsen. Residual flows for invertible generative modeling. In *NeurIPS*, 2019.

[9] C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. *ICML*, 2018.

[10] N. Deo and M. M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *CVPR Workshop*, 2018.

[11] N. Deo and M. M. Trivedi. Scene induced multi-modal trajectory forecasting via planning. In *ICRA Workshop*, 2019.

[12] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. In *ICLR*, 2015.

[13] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. In *ICLR*, 2017.

[14] P. Felsen, P. Lucey, and S. Ganguly. Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders. In *ECCV*, 2018.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[16] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018.

[17] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3): 331–371, 1910.

[18] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. In *Physical review E*, 1995.

[19] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *ICML*, 2019.

[20] C.-W. Huang, L. Dinh, and A. Courville. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. *arXiv preprint arXiv:2002.07101*, 2020.

[21] S. Kim, S.-G. Lee, J. Song, J. Kim, and S. Yoon. Flowavenet: A generative flow for raw audio. In *ICML*, 2019.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[23] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.

[24] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[25] P. Kirichenko, P. Izmailov, and A. G. Wilson. Why normalizing flows fail to detect out-of-distribution data. *arXiv preprint arXiv:2006.08545*, 2020.

[26] M. Kumar, M. Babaeizadeh, D. Erhan, C. Finn, S. Levine, L. Dinh, and D. Kingma. Videoflow: A flow-based generative model for video. *ICLR*, 2020.

[27] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[28] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *CVPR*, 2017.

[29] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *AAAI*, 2019.

[30] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. *ECCV*, 2020.

[31] E. Pajouheshgar and C. H. Lampert. Back to square one: probabilistic trajectory forecasting without bells and whistles. In *NeurIPs Workshop*, 2018.

[32] P. Porwik and A. Lisowska. The haar-wavelet transform in digital image processing: its status and achievements. *Machine graphics and vision*, 13(1/2):79–98, 2004.

[33] N. Rhinehart, K. M. Kitani, and P. Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *ECCV*, 2018.

[34] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, 2016.

[35] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese. Car-net: Clairvoyant attentive recurrent network. In *ECCV*, 2018.

[36] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, S. H. Rezatofighi, and S. Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *CVPR*, 2019.

[37] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *ISCA Speech Synthesis Workshop*, 2016.

[38] A. van den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves. Conditional image generation with PixelCNN decoders. In *NIPS*, 2016.

[39] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *CVPR*, 2011.

[40] Y. Yuan and K. Kitani. Diverse trajectory forecasting with determinantal point processes. *ICLR*, 2020.

[41] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *CVPR*, 2019.

[42] Z. M. Ziegler and A. M. Rush. Latent normalizing flows for discrete sequences. In *ICML*, 2019.

Appendix A. Additional Details of Lemma 1

.1 Proof of Lemma 1

Lemma 3. *The generalized Haar transformation $f_{hba} = f_{haar} \circ f_{eo}$ is invertible for $\alpha \in [0, 1)$ and the determinant of the Jacobian of the transformation $f_{hba} = f_{haar} \circ f_{eo}$ for sequence of length T_k with $\mathbf{y}_k^j \in \mathbb{R}^d$ is $\det \mathbf{J}_{hba} = (1 - \alpha)^{(d \cdot T_k)/2}$.*

Proof. To compute the Jacobian of f_{hba}, note that each element of the output fine ($\mathbf{f}_k$) and coarse ($\mathbf{c}_k$) trajectories can be expressed in terms of the elements of the input trajectory $\mathbf{y}_k$. From Eqs. (3) and (4) in the main paper, the coarse ($\mathbf{c}_k$) trajectories at level k can be expressed as,

$$\begin{aligned}\mathbf{c}_k &= (1-\alpha)\mathbf{e}_k + \alpha\mathbf{o}_k \\ &= (1-\alpha)\cdot[\mathbf{y}_k^2, \cdots, \mathbf{y}_k^{T_k}] + \alpha\cdot[\mathbf{y}_k^1, \cdots, \mathbf{y}_k^{T_k-1}] \\ &= [\alpha\mathbf{y}_k^1 + (1-\alpha)\mathbf{y}_k^2, \alpha\mathbf{y}_k^3 + (1-\alpha)\mathbf{y}_k^4, \cdots, \alpha\mathbf{y}_k^{T_k-1} + (1-\alpha)\mathbf{y}_k^{T_k}].\end{aligned} \tag{11}$$

Similarly, the fine ($\mathbf{f}_k$) trajectories at level k can be expressed as,

$$\begin{aligned}\mathbf{f}_k =&(1-\alpha)\mathbf{o}_k + (\alpha-1)\mathbf{e}_k \\ =&(1-\alpha)\cdot[\mathbf{y}_k^1, \cdots, \mathbf{y}_k^{T_k-1}] + (\alpha-1)\cdot[\mathbf{y}_k^2, \cdots, \mathbf{y}_k^{T_k}] \\ =&[(1-\alpha)\mathbf{y}_k^1 + (\alpha-1)\mathbf{y}_k^2, (1-\alpha)\mathbf{y}_k^3 + (\alpha-1)\mathbf{y}_k^4, \cdots, \\ &(1-\alpha)\mathbf{y}_k^{T_k-1} + (\alpha-1)\mathbf{y}_k^{T_k}].\end{aligned} \tag{12}$$

We can now rearrange the elements of the output trajectory f_{hba} by placing elements from $\mathbf{f}_k$ and $\mathbf{c}_k$ in an alternating fashion,

$$\begin{aligned}f_{hba}(\mathbf{y}_k) = \mathbf{f}_k, \mathbf{c}_k = [&(1-\alpha)\mathbf{y}_k^1 + (\alpha-1)\mathbf{y}_k^2,\ \alpha\mathbf{y}_k^1 + (1-\alpha)\mathbf{y}_k^2,\ \cdots\ , \\ &(1-\alpha)\mathbf{y}_k^{T_k-1} + (\alpha-1)\mathbf{y}_k^{T_k},\ \alpha\mathbf{y}_k^{T_k-1} + (1-\alpha)\mathbf{y}_k^{T_k}].\end{aligned} \tag{13}$$

As each element $\mathbf{y}_k^j \in \mathbb{R}^d$, we can further simplify the output trajectory f_{hba} in terms of the individual elements in $\mathbf{y}_k^j$. This results in a block diagonal Jacobian $\mathbf{J}_{hba} \in \mathbb{R}^{d \cdot T_k \times d \cdot T_k}$ of f_{hba} of the form,

$$\mathbf{J}_{hba} = \begin{pmatrix} (1-\alpha) & (\alpha-1) & 0 & 0 & 0 & \cdots & 0 & 0 \\ \alpha & (1-\alpha) & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & (1-\alpha) & (\alpha-1) & 0 & \cdots & 0 & 0 \\ 0 & 0 & \alpha & (1-\alpha) & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & (1-\alpha) & (\alpha-1) \\ 0 & 0 & 0 & 0 & 0 & \cdots & \alpha & (1-\alpha) \end{pmatrix}. \tag{14}$$

The repeating block in $\mathbf{J}_{hba}$ repeats $(d \cdot T_k)/2$ times as the trajectory is of length T_k and each element of the trajectory has d dimensions. Therefore, the determinant of the Jacobian $\mathbf{J}_{hba}$ is $(1 - \alpha)^{(d \cdot T_k)/2}$.

To show that $f_{hba} = f_{haar} \circ f_{eo}$ is invertible, first note that f_{eo} rearranges the elements of the input trajectory as is thus trivially invertible. Now, note that f_{haar} is a linear system. For $\alpha \in [0, 1)$ we see that $\det \mathbf{J}_{hba} > 0$. Thus, the linear system f_{haar} in Eq. (4) in the main paper is non-singular and invertible. Thus, f_{hba} is invertible. □

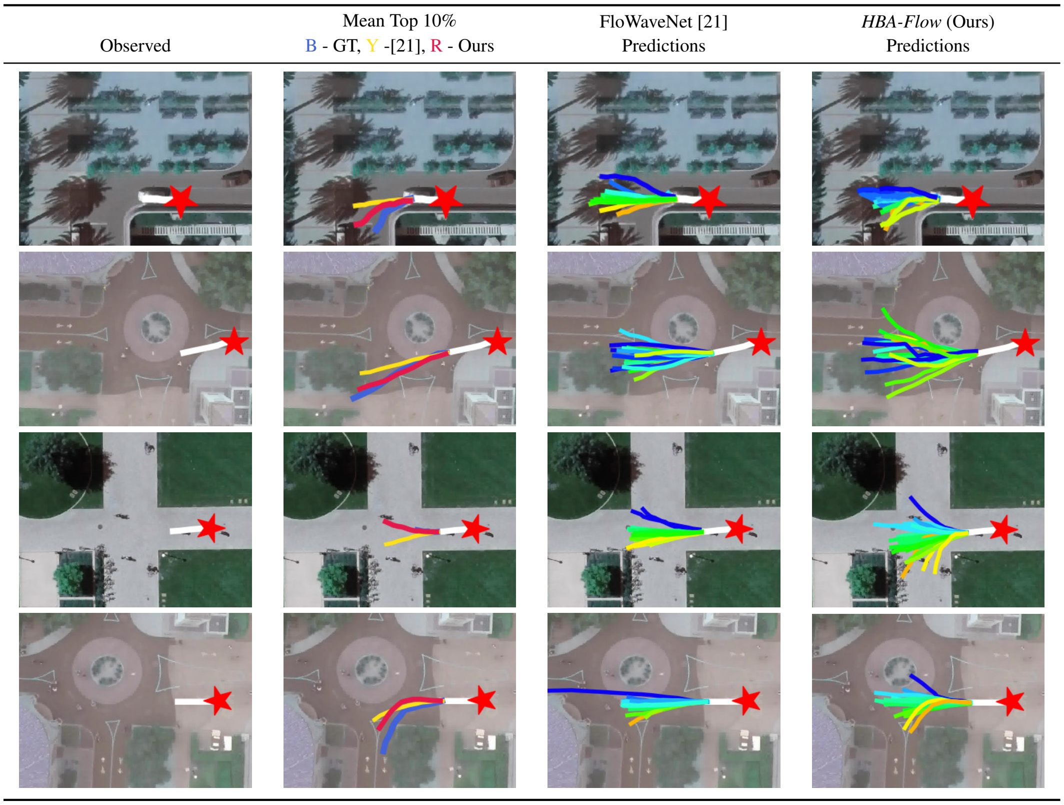
Appendix B. Architecture and Optimization

Here, we provide additional architectural details of our HBA-Flow model in Fig. 2 (right), in particular the split coupling flows. The split coupling flows in our HBA-Flow model are based on those of FloWaveNet [21]. However, as mentioned in the main paper, we employ more powerful non-linear squared flows [42] across baselines versus the affine flows used in [21]. The non-causal wavenets in the split coupling flows are similar to the ones employed in [21] with 4 convolutional layers with 256 filters each. In practice, we do not find it necessary to employ activation normalization layers along with the more powerful non-linear squared flows. We use identical non-causal wavenets to learn the parameters of our HBA-Prior.

Finally, note that we train the full HBA-Flow model along with the prior using the AdaMax [22] optimizer. The "mixing" parameter α in f_{hba} is learnable, although $\alpha = 0.5$ also works well in practice.

Appendix C. Qualitative Results

We provide additional qualitative results on Stanford Drone in Fig. 5 and Intersection Drone in Fig. 6 comparing to FloWaveNet [21]. These results further support the results in Figs. 4 and 5 in the main paper. We again see that the predictions of our HBA-Flow model are more diverse and can more effectively capture the modes of the trajectory distributions at complex traffic situations like intersections and crossings. Again, this is further supported by the top 10% of predictions, which are closer to the groundtruth trajectories.



Observed

Mean Top 10%
B - GT, Y -[21], R - Ours

FloWaveNet [21]
Predictions

HBA-Flow (Ours)
Predictions

Figure 5: Mean top 10% predictions (Blue - Groudtruth, Yellow - FloWaveNet [21], Red - Our *HBA-Flow* model) and predictive distributions on Stanford Drone dataset. The predictions of our HBA-Flow model are more diverse and better capture the modes of the future trajectory distribution.

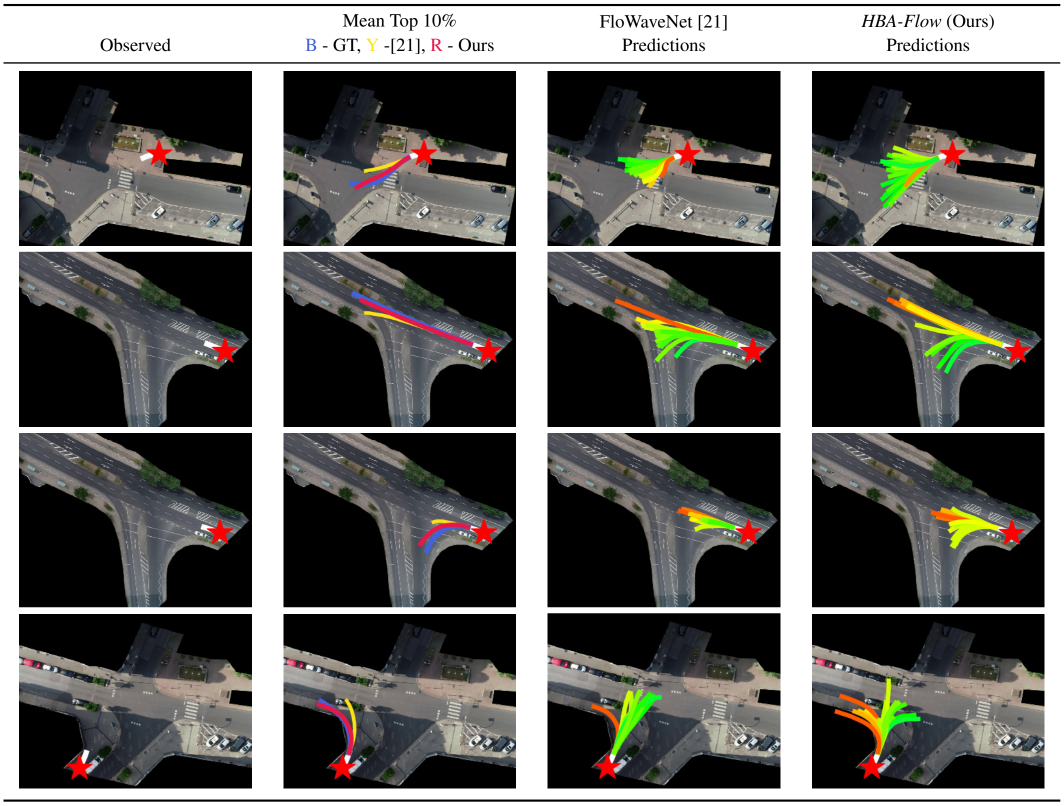


Figure 6: Mean top 10% predictions (Blue - Groudtruth, Yellow - FloWaveNet [21], Red - Our *HBA-Flow* model) and predictive distributions on Intersection Drone dataset. The predictions of our HBA-Flow model are more diverse and better capture the modes of the future trajectory distribution.