Probabilistic Transformer for Time Series Analysis

Binh Tang
Department of Statistics and Data Science
Cornell University
Ithaca, NY 14850
bvt5@cornell.edu

David S. Matteson
Department of Statistics and Data Science
Cornell University
Ithaca, NY 14850
matteson@cornell.edu

Abstract

Generative modeling of multivariate time series has remained challenging partly due to the complex, non-deterministic dynamics across long-distance time steps. In this paper, we propose deep probabilistic methods that combine state-space models (SSMs) with transformer architectures. In contrast to previously proposed SSMs, our approaches use attention mechanism to model non-Markovian dynamics in the latent space and avoid recurrent neural networks entirely. We also extend our models to include several layers of stochastic variables organized in a hierarchy for further expressiveness. Compared to transformer models, ours are probabilistic, non-autoregressive, and capable of generating diverse long-term forecasts with accounted uncertainty. Extensive experiments show that our models consistently outperform competitive baselines on various tasks and datasets, including time series forecasting and human motion prediction.

1 Introduction

Generative modeling of multivariate time series is a challenging problem with wide-ranging applications in demand forecasting [15, 76], autonomous driving [2, 16], robotics [29, 67], and health care [20, 21, 59]. Despite remarkable progress in recent years, models that predict high-dimensional future observations from a few past examples have remained intractable, partly due to the complex, non-deterministic temporal dynamics across long-distance time steps. Given a sequence of human poses, for example, such models must internally figure out the involved dynamics of various body components across space and time while maintaining the inherent uncertainty of multiple plausible futures, even though only one such future is observed.

Among proposed probabilistic approaches, state space models (SSMs) provide a principled framework for learning and drawing inference from sequential inputs [27, 66]. While autoregressive models feed its predictions back into the dynamics model without any compressed representation of data, SSMs model stochastic transitions between abstract states using latent variables, allowing for efficient state-to-state sampling without the need to render high-dimensional observations. Gaussian linear dynamical systems (LDSs), one of the best known SSMs [92], for example, postulate linear state transitions and enjoy exact inference via the celebrated Kalman filter algorithm.

While early extensions of LDSs focus on linearization [46] and unscented transform [88], recent work that marry state space models with deep neural networks offers much more flexibility to model complex dependencies across different time steps. Some approaches retain the Markovian dynamics of LDSs and only replace their linear observation models with feed-forward networks [23, 31, 47, 71], whereas others favor nonlinear state transitions and parametrize such dependencies via recurrent neural networks (RNNs) [22, 23, 30, 39, 51, 75]. Despite differences, both Markovian transitions and RNNs are often not capable of capturing long-range dependencies in highly structured sequential inputs [36, 100], limiting the capacity of the corresponding SSMs.

In this work, we propose to combine the complementary strengths of SSMs and transformer architectures [85], a powerful mechanism for modeling long-term interactions that enjoys success across a variety of sequence modeling tasks [26, 48, 99]. In contrast to most SSMs, our models make extensive use of attention mechanism [5, 85] between latent variables to model non-Markovian dynamics (see Figure 1). Compared to transformer-based methods, our models are probabilistic, non-autoregressive in a similar fashion to LDSs, and capable of generating diverse long-term forecasts with uncertainty estimates.

Our main contributions are threefold. First, we propose novel SSMs based on transformer architectures for multivariate time series, which include generative models and inference procedures based on variational inference [49, 74]. Second, we extend our models to include several layers of stochastic latent variables organized in a hierarchy for further expressiveness. Third, we conduct extensive experiments on time series forecasting and human motion prediction and demonstrate that our Probabilistic Transformer (ProTran) performs remarkably well compared to various state-of-the-art baselines.

2 Preliminaries

2.1 Variational State Space Models

Let \( \{x_{i:T}^{(i)}\}_{i=1}^N \) consist of \( N \) univariate time series where \( x_{1:T}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_T^{(i)}) \) and \( x_t^{(i)} \) denotes the value of the \( i \)-th time series at time \( t \). We consider the multivariate form \( x_{1:T} = (x_1, x_2, \ldots, x_T) \) where \( x_t = (x_1^{(1)}, \ldots, x_t^{(N)}) \in \mathbb{R}^N \). Conditioning on observed values up to time \( C \), we aim to produce distributional forecasts into the future \( p(x_{C+1:T} \mid x_{1:C}) \). For clarity, we refer to \( x_{1:C} \) and \( x_{C+1:T} \) as contexts and targets, respectively.

We are interested in probabilistic models parametrized by \( \theta \) of the form

\[
p_\theta(x_{1:T} \mid x_{1:C}) = \int p_\theta(x_{1:T} \mid z_{1:T})p_\theta(z_{1:T} \mid x_{1:C})d\mathbf{z}_{1:T} \tag{1}
\]

where \( z_{1:T} = (z_1, z_2, \ldots, z_T) \) denotes the corresponding sequence of latent variables, sometimes referred to as states. In other words, we assume a generative model that can be decomposed into a transition model \( p_\theta(z_{1:T} \mid x_{1:C}) \) between the latent variables conditioned on the contexts, and an emission model \( p_\theta(x_{1:T} \mid z_{1:T}) \) from the latent variables to observable outputs. In particular, we
further impose several assumptions on both models: \(^1\)

\[
p_\theta(z_{1:T} \mid x_{1:C}) = \prod_{t=1}^T p_\theta(z_t \mid z_{1:t-1}, x_{1:C}), \quad p_\theta(x_{1:T} \mid z_{1:T}) = \prod_{t=1}^T p_\theta(x_t \mid z_t).
\]

As demonstrated in Figure 1(b), the latent variable \(z_{t+1}\) depends not only on \(z_t\) but also on all of its preceding latent variables, including \(z_{t-1}\), in contrast to linear dynamical systems (LDSs). In addition, the transition and emission models allow for non-linearity via neural network parametrizations. These assumptions aim to maximize model capacity for real-world applications with complex emissions or temporal dependencies.

However, neither \(x_{1:t-1}\) nor \(z_{1:t-1}\) are included in the emission model \(p(x_t \mid z_{1:T}, x_{1:C})\). Such assumptions are important, as it has been argued previously that a leakage of information from the latent space in autoregressive models can hinder long-term predictions [23, 47]. While all ground truth observations are available during training, the entire sequence has to be generated sequentially at test time, making the dependencies on \(x_{1:t-1}\) prone to accumulated errors over multiple time steps. By letting the latent variable \(z_t\) capture all information needed to render \(x_t\), we also avoid the computational costs associated with repeatedly decoding and encoding \(x_t\) in multi-step predictions.

The inclusion of nonlinear state transitions and observation models necessarily requires approximate inference. We follow the stochastic variational inference framework [49, 74] and assume that the variational posterior parametrized by \(\phi\) can be decomposed auto-regressively as \(q_\phi(z_{1:T} \mid x_{1:T}) = \prod_t q_\phi(z_t \mid z_{1:t-1}, x_{1:T})\), which leads to a lower bound on the log likelihood:

\[
\log p_\theta(x_{1:T} \mid x_{1:C}) \geq \sum_{t=1}^T \left( \mathbb{E}_q \left[ \log p_\theta(x_t \mid z_t) \right] - \text{KL}(q_\phi(z_t \mid z_{1:t-1}, x_{1:T}) \mid \mid p_\theta(z_t \mid z_{1:t-1}, x_{1:C})) \right),
\]

where KL is the Kullback-Leibler divergence.

For computational stability, we assume homoscedasticity and choose Laplace distribution with scale parameter \(\beta\) as a parametric form for \(p_\theta(x_t \mid z_t)\), i.e. we optimize for \(L_1\) reconstruction loss with a cross-validated factor \(\beta\) for the KL term, following similar variational autoencoder (VAE) work [24, 41, 86]. Such an assumption does not necessarily limit the capacity of our models, as powerful stochastic transitions and flexible emission models can theoretically characterize arbitrary noise covariance [66]. Incorporating structured probabilistic outputs such as Gaussian copulas [75] or normalizing flows [23] can potentially further improve our model performance.

### 2.2 Transformer Architectures

Central to our models and other transformer-based approaches [48, 85] is the notion of attention [5], which allows the models to focus on important parts within a context. Multi-head attention, for example, maps a sequence of queries \(Q \in \mathbb{R}^{\ell_q \times d}\) of length \(\ell_q\) to a sequence of outputs \(O = [O_1, \ldots, O_H] \in \mathbb{R}^{\ell_q \times d}\) of the same size by attending over given \(\ell_h\) key-value pairs \(K \in \mathbb{R}^{\ell_k \times d}\), \(V \in \mathbb{R}^{\ell_v \times d}\):

\[
O_h = \text{Attention}(Q_h, K_h, V_h) = \text{Softmax} \left( \frac{Q_h K_h^T}{\sqrt{d}} \right) V_h,
\]

where \(Q_h = Q W_h^Q, K_h = K W_h^K, V_h = V W_h^V\) are projected queries, keys, and values corresponding to head \(h \in [1, H]\) with learning parameters \(W_h^Q, W_h^K, W_h^V\), respectively. In case \(Q = K = V\), we refer to such an attention mechanism as self-attention.

Given fully observed sequences of inputs, the mapping can be computed efficiently without any imposed sequential order often seen in recurrent neural networks [19, 42]. More importantly, the direct connections between long-distance time steps are baked into the mechanism as information from previous time steps is easily accessible without being compressed into a fixed representation, easing optimization and learning of long-term dependencies [5, 85].

---

\(^1\)For notational simplicity, we assume \(x_0 = z_{1:0} = \emptyset\) and \(p(z \mid x_0) = p(z)\).
Without recurrence, Transformer [85] encodes information about each time step \( t \) with predefined sinusoidal positional embeddings \( \text{Position}(t) = [p_i(1), \ldots, p_i(d)] \in \mathbb{R}^d \) where the \( i \)-th embedding is given by \( p_i(i) = \sin(t \cdot c^i/d) \) for even \( i \) and \( p_i(i) = \cos(t \cdot c^i/d) \) for odd \( i \) and \( c \) is some large constant. Empirical results show that such positional embeddings are also important to our models.

3 Probabilistic Transformer

In this section, we first present our single-layered model and subsequently its multi-layered extension for a hierarchy of stochastic latent variables. As alluded earlier, our model consists of a generative model and an inference model that share information and parameters extensively.

3.1 Single-Layered Probabilistic Transformer

**Generative Model.** Given some contexts \( x_{1:C} \), we first apply a linear projection and combine it with a positional embedding to obtain \( h_{1:C} \in \mathbb{R}^d \), i.e.

\[
 h_t = \text{LayerNorm}(\text{MLP}(x_t) + \text{Position}(t)),
\]

where LayerNorm and MLP denote layer normalizations [4] and multi-layer perceptrons, respectively. While a traditional transformer model often dedicates an entire encoder for the same purpose [55, 72], we find such a simple mapping works sufficiently well in conjunction with the context-attention module of the corresponding decoder.

As implied in Equation (2), our latent dynamics decomposes auto-regressively. At each time step, we parametrize the distribution \( p_0(z_t | z_{1:t-1}, x_{1:C}) \) by a Gaussian with parameters resulting from two sequential steps of attention: a self-attention over the previously inferred states \( z_{1:t-1} \) and another attention over the projected contexts \( h_{1:C} \). These two operations mirror those found in the decoder of Transformer [85], with the stochastic latent variables replacing its decoder inputs.

Unfortunately, using stochastic samples of \( z_t \) as attention queries is problematic, as purely stochastic transitions make it difficult for the model to reliably retain information across multiple time steps [17, 30, 39]. We therefore encapsulate the latent variables in hidden representations \( w_t \) that also has a deterministic component. Combined with the attention steps, such representations help model long-range temporal dependencies while accounting for the stochasticity of future observations.

Starting with a learnable, context-agnostic representation \( w_0 \), we recursively update \( w_t \) using a stochastic sample from \( p_0(z_t | z_{1:t-1}, x_{1:C}) \) and the positional embedding for the current time step \( t \). The generating process for the time step \( t \) can be summarized by the following pseudocode:

\[
 \hat{w}_t = \text{LayerNorm}(w_{t-1} + \text{Attention}(w_{t-1}, w_{1:t-1}, w_{1:t-1})))
\]

\[
 w_t = \text{LayerNorm}(\hat{w}_t + \text{Attention}(\hat{w}_t, h_{1:C}, h_{1:C}))
\]

\[
 z_t = \text{Sample}(\mathcal{N}(z_t; \text{MLP}(\hat{w}_t), \text{Softplus}(\text{MLP}(w_t))))
\]

\[
 w_t = \text{LayerNorm}(\hat{w}_t + \text{MLP}(z_t) + \text{Position}(t)),
\]

where \( \text{Sample} \) and \( \text{Softplus} \) are the Gaussian sampling and approximating rectifier operators.

Each stochastic sample of \( w_{1:T} \) is then mapped to a sequence of \( x_{1:T} \) via a multi-layer perceptron. We emphasize that our generation procedure in the latent space is more efficient than others in the observation space, which requires encoding and decoding high-dimensional inputs repeatedly.

**Inference Model.** We parametrize the approximate posterior \( q_\phi(z_t | z_{1:t-1}, x_{1:T}) \) at time step \( t \) in a similar fashion to the prior \( p_0(z_t | z_{1:t-1}, x_{1:C}) \). Indeed, these parametrizations share most parameters and are done simultaneously in the same recursive loop, following the exact same steps in Equation (6) and Equation (7) (see Figure 1). We note that similar sharing techniques between the generative and inference processes have emerged as a common theme among recent successful VAE models [17, 62, 83].

While the prior only has access to the conditioning observations \( x_{1:C} \), the approximate posterior should take into account all observations during training, including the targets \( x_{C+1:T} \). Due to the inherent unidirectional aspect of RNNs, previous work that uses RNNs to parametrize the approximate posterior often disregards such a property [22, 30, 51] and often resorts to a filtering routine.
\[ p(z_t | z_{1:t-1}, x_{1:t}) \]. In contrast, our inference procedure resembles more of the smoothing process of LDSs, factoring in both past and future observations via another application of self-attention:

\[
\begin{align*}
  k_t &= \text{Attention}(h_{1:T}, h_{1:T}, h_{1:T}) \\
  z_t &= \text{Sample}(\mathcal{N}(z_t; \text{MLP}(\tilde{w}_t), \text{Softplus}(\text{MLP}(\tilde{w}_t, k_t)))).
\end{align*}
\]

Equation (10)

Here, we replace Equation (8) in the generative model with Equation (11), where the hidden representation \( k_t \) summarizing all information relevant to the current time step \( t \) has been concatenated to the latent-and-context-aware representation \( \tilde{w}_t \) preceding the Gaussian parametrization.

The generative model and the inference model are trained end-to-end with a single stochastic variational inference objective stated in Equation (3). Such a variational bound includes the reconstruction loss for \( x_{1:C} \) and the KL term for \( z_{1:C} \). Alternatively, we can exclude these terms from the objective, which is equivalent to starting the inference process from \( t = C + 1 \) instead of \( t = 1 \).

Our models incur a time complexity of \( O(T^2d) \) and a memory cost of \( O(T^2d) \), where \( T \) is the total sequence length and \( d \) is the dimensionality of the latent space. The recursive latent dynamics also does not allow the take full advantage of parallelizable attentions. However, we find that our models are still efficient in practice, especially for reasonably small values of \( T \).

### 3.2 Multi-Layered Extension for Probabilistic Transformer

Inspired by recent work on hierarchical VAEs for non-sequential inputs \([17, 80, 83, 101]\), we extend our proposed model to include several layers of latent variables, aiming to further increase its flexibility for modelling sequential data.

We represent each time step \( t \) with a Markov chain of \( L \) latent variables \( z_{t:1}^{(1:L)} = (z_t^{(1)}, \ldots, z_t^{(L)}) \) for simplicity (see Figure 1). The generative and inference model also decompose auto-regressively across different time steps and may exhibit non-Markovian dynamics:

\[
\begin{align*}
  p_\phi(x_{1:T}, z_{1:T}^{(1:L)} | x_{1:C}) &= \prod_{t=1}^T \prod_{\ell=1}^L p_\phi(x_t | z_t^{(\ell)}) \\
  q_\phi(z_{1:T}^{(1:L)} | x_{1:T}) &= \prod_{t=1}^T \prod_{\ell=1}^L q_\phi(z_t^{(\ell)} | z_{1:t-1}^{(\ell)}, z_{1:T}^{(\ell-1)}, x_{1:T}).
\end{align*}
\]

Equation (12)

Intuitively, we generate samples \( x_{1:T} \) conditioning on \( x_{1:C} \) by following the latent dynamics from the bottom up and using the generative process described earlier within each layer. Analogously, inference proceeds in the same order, resulting in a variational bound similar to Equation (3):

\[
\log p_\phi(x_{1:T} | x_{1:C}) \geq \sum_{t=1}^T \mathbb{E}_q \left[ \log p_\phi(x_t^{(L)} | z_t) \right] - \sum_{\ell=1}^L \text{KL}(q_\phi(z_t^{(\ell)} | z_{1:t-1}^{(\ell)}, z_{1:T}^{(\ell)}, x_{1:T}) \| p_\phi(z_t^{(\ell)} | z_{1:t-1}^{(\ell)}, z_{1:T}^{(\ell)}, x_{1:C})).
\]

Equation (14)

As before, we parametrize the prior \( p_\phi(z_t^{(\ell)} | z_{1:t-1}^{(\ell)}, z_{1:T}^{(\ell)}, x_{1:C}) \) using self-attention over the inferred latent variables from previous time steps \( w_{t-1}^{(\ell)} \) on the same layer and another attention over context \( h_{1:C} \). In this case, however, we include an additional self-attention over all latent variables from the layer immediately below it (see Equation (16)):

\[
\begin{align*}
  \tilde{w}_t^{(\ell)} &= \text{LayerNorm}(w_{t-1}^{(\ell)} + \text{Attention}(w_{t-1}^{(\ell)}, w_{1:T}^{(\ell-1)})) \\
  w_t^{(\ell)} &= \text{LayerNorm}(\tilde{w}_t^{(\ell)} + \text{Attention}(\tilde{w}_t^{(\ell)}, w_{1:t-1}^{(\ell)})) \\
  \tilde{w}_t^{(\ell)} &= \text{LayerNorm}(w_t^{(\ell)} + \text{Attention}(w_t^{(\ell)}, h_{1:C}, h_{1:C})) \\
  z_t^{(\ell)} &= \text{Sample}(\mathcal{N}(z_t^{(\ell)}; \text{MLP}(\hat{w}_t^{(\ell)}), \text{Softplus}(\text{MLP}(\hat{w}_t^{(\ell)})))) \\
  w_t^{(\ell)} &= \text{LayerNorm}(w_t^{(\ell)} + \text{MLP}(z_t^{(\ell)})),
\end{align*}
\]

Equation (16)
Stacking multiple layers of latent variables increases model expressiveness, but it also result in a linear increase in running time and the number of parameters. The time complexity for the $L$-layers transformer is $O(LT^2d)$, while the space complexity remains $D(T^2d)$ due to the Markovian structure of the chain $z_t^{(LL)}$ at each time step $t$. In our experiments, we restrict the number of layers of our hierarchical models to two or three.

4 Related Work

Deep State Space Models. Deep neural networks have been extensively combined with state space models, resulting in flexible, yet principledly motivated latent variable approaches. While some work keep the linear state transition intact to leverage the efficient Kalman filter algorithms [23, 31, 47, 71], more expressive, nonlinear latent dynamics parametrized by neural networks have been proposed [51, 52]. All such models are limited to the Markovian dynamics of LDSs, which hinders learning of long-range dependencies. The limitation is often alleviated by combining the stochastic transitions with a deterministic RNN that enables access to all past states [3, 8, 22, 30, 39, 77]. Our models are similarly non-Markovian, but the dependencies on the past states are done via attention, which allows for easy connections between long-distance time steps. In addition, while most existing deep SSMs represent each time step with a single latent variable, our models include several layers of hierarchical latent variables with tractable inference mechanism.

Attentive Recurrent Networks. Attention mechanism has also been widely adopted in recent time series work using sequence-to-sequence models [1, 28] or transformer architectures [14, 55, 57, 72, 81, 94]. While our models are equipped with latent variables, these transformer approaches [55, 72] lack inference mechanism and are susceptible to feeding back observation noise into the dynamics model at test time. Our work, however, can be considered as an extension of the attentive state space model proposed in [1], with discrete latent states replaced by their continuous analogs. Recent developments in natural language processing [58, 60, 90] also combine transformer and VAE; however, these approaches often use a time-agnostic latent variable, in contrast to our SSM formulation.

Time Series Forecasting. Traditional univariate time series models, such as Box-Jenkins methods [12] and exponential smoothing [43], often assume independence between any collection of time series [76]. While multivariate extensions of the classical approaches, including vector autoregression [82] and multivariate GARCH [7], do not require such a strong assumption, they come with many others such as stationarity and homocedasticity, demand manual selection of covariates and models, and do not scale well to even a moderate number of time series [40, 69].

Deep learning methods for time series forecasting have recently emerged as an expressive, scalable framework for industrial applications [10, 68, 79, 91]. While early work focus on point forecasts [53, 70, 96], recent approaches often employ recurrent neural networks with probabilistic forecasts parametrized directly [76], using quantile functions [33], Gaussian copulas [75], normalizing flows [23], or diffusion models [73]. In contrast, our models are entirely devoid of such recurrent architectures and rely on latent variables to output distributional forecasts.

Human Motion Prediction. Despite being almost identical in formulation, human motion prediction has often been studied independently from time series forecasting. While some work deterministically generate future motions or video frames [13, 32, 34, 56], stochastic prediction has also been proposed with deep neural networks often outperforming traditional methods such as hidden Markov models [93] or Gaussian processes [89] on complex motion datasets [13, 32, 45, 54, 63]. In contrast to earlier work [95, 97] that employ a global latent variable across different time steps via conditional VAE [49], we leverage the principled framework of state space models for learning and inference of hierarchical, time-dependent latent variables.

5 Experiments

We present our experiment results on two tasks, namely, time series forecasting and human motion prediction. These tasks are often studied independently, despite being almost identical as conditional prediction problems.
Our probabilistic transformers for $S_F$ distribution function probability score (and training processes. To those of the state-of-the-art approaches. See Appendix D for more details about hyper-parameters.

We re-port the numbers of parameters of our models in Table 4 in Appendix C, which are all comparable to those of the state-of-the-art approaches. See Appendix D for more details about hyper-parameters and training processes.

### 5.1 Time-series Forecasting

**Datasets & Covariates.** Following the experiment setup in [72, 73, 75], we evaluate our models and multiple competitive baselines on five popular public datasets: SOLAR, ELECTRICITY, TRAFFIC, TAXI, and WIKIPEDIA. The data is recorded with hourly or daily frequency and shows seasonal patterns of different frequencies (see Appendix A for more dataset details). As in [72, 73], the covariates include lagged inputs, fixed time embeddings (e.g. day of week, hour of day), and learnable time-series embeddings. The inputs are scaled using the conditioning examples before being fed into the model, and the predictions are rescaled appropriately afterward.

**Metrics.** Following [23, 72, 75], we evaluate our model and all baselines using continuous ranked probability score (CRPS) [65] summed across time series, denoted by $\text{CRPS}_{\text{sum}}$. Given a univariate distribution function $F$ and an observation $x$, CRPS is defined as

$$\text{CRPS}(F, x) = \int_{\mathbb{R}} (F(z) - 1_{\{x \leq z\}})^2 \, dz,$$

where $1_{\{x \leq z\}}$ is the indicator function. As argued in de Bézenac et al. [23], $\text{CRPS}_{\text{sum}}$ is a proper scoring rule [35] and can be computed without analytical forecast distributions. We compute the metrics in a rolling fashion and use 100 samples for the distributional forecasts, similar to the aforementioned work.

**Baselines.** We benchmark our models against various baselines, including (1) VES [43], an innovation state space model; (2) VAR-Lasso and VAR [61], two multivariate linear autoregressive models with and without Lasso regularization; (3) GARCH [84], a multivariate conditional heteroskedastic model; (4) DeepAR [76], an autoregressive recurrent neural network; LSTM-Copula and GP-Copula [75], two RNN-based models that use Gaussian copula to model nonlinearity; (5) KV AE [51], a variational approach based on linear dynamics; (6) NKF [23], a normalizing-flow model coupled with Kalman filters; (7) Transformer [72], a transformer-based model based on masked autoregressive flow; and (8) TimeGrad [73], a recent autoregressive approach that uses a diffusion model.

**Implementations.** We use 8-head attentions and 2-layers MLPs to parametrize the generative and inference models. The stochastic latent variables $z_t$ are 16-dimensional while the hidden representations $w_t$ are in $\mathbb{R}^{128}$. Our probabilistic transformers for SOLAR and ELECTRICITY have one stochastic layer while those for the other datasets of higher dimensional observations employ two layers. We report the numbers of parameters of our models in Table 4 in Appendix C, which are all comparable to those of the state-of-the-art approaches. See Appendix D for more details about hyper-parameters and training processes.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SOLAR</th>
<th>ELECTRICITY</th>
<th>TRAFFIC</th>
<th>TAXI</th>
<th>WIKIPEDIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>VES [43]</td>
<td>0.900 ± 0.003</td>
<td>0.880 ± 0.004</td>
<td>0.350 ± 0.002</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VAR [61]</td>
<td>0.830 ± 0.006</td>
<td>0.039 ± 0.001</td>
<td>0.290 ± 0.001</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VAR-Lasso [61]</td>
<td>0.510 ± 0.006</td>
<td>0.025 ± 0.000</td>
<td>0.150 ± 0.002</td>
<td>-</td>
<td>3.100 ± 0.004</td>
</tr>
<tr>
<td>GARCH [84]</td>
<td>0.880 ± 0.002</td>
<td>0.190 ± 0.001</td>
<td>0.370 ± 0.001</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DeepAR [76]</td>
<td>0.336 ± 0.014</td>
<td>0.023 ± 0.001</td>
<td>0.555 ± 0.003</td>
<td>-</td>
<td>0.127 ± 0.042</td>
</tr>
<tr>
<td>LSTM-Copula [75]</td>
<td>0.319 ± 0.011</td>
<td>0.064 ± 0.008</td>
<td>0.103 ± 0.006</td>
<td>0.326 ± 0.007</td>
<td>0.241 ± 0.003</td>
</tr>
<tr>
<td>GP-Copula [75]</td>
<td>0.337 ± 0.024</td>
<td>0.024 ± 0.002</td>
<td>0.078 ± 0.002</td>
<td>0.208 ± 0.183</td>
<td>0.086 ± 0.004</td>
</tr>
<tr>
<td>KV AE [51]</td>
<td>0.340 ± 0.025</td>
<td>0.051 ± 0.019</td>
<td>0.100 ± 0.005</td>
<td>-</td>
<td>0.095 ± 0.012</td>
</tr>
<tr>
<td>NKF [23]</td>
<td>0.320 ± 0.020</td>
<td><strong>0.016 ± 0.001</strong></td>
<td>0.100 ± 0.002</td>
<td>-</td>
<td>0.071 ± 0.002</td>
</tr>
<tr>
<td>Transformer-MAF [72]</td>
<td>0.301 ± 0.014</td>
<td>0.021 ± 0.000</td>
<td>0.056 ± 0.001</td>
<td>0.179 ± 0.002</td>
<td>0.063 ± 0.003</td>
</tr>
<tr>
<td>TimeGrad [73]</td>
<td>0.287 ± 0.020</td>
<td>0.021 ± 0.001</td>
<td>0.044 ± 0.006</td>
<td>0.114 ± 0.020</td>
<td><strong>0.049 ± 0.002</strong></td>
</tr>
<tr>
<td><strong>ProTran (Ours)</strong></td>
<td><strong>0.194 ± 0.030</strong></td>
<td><strong>0.016 ± 0.001</strong></td>
<td><strong>0.028 ± 0.001</strong></td>
<td><strong>0.084 ± 0.003</strong></td>
<td><strong>0.047 ± 0.004</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Table 2:</strong> Ablation study on TRAFFIC.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two Layers</td>
</tr>
<tr>
<td>One Layer</td>
</tr>
<tr>
<td>Context Attention</td>
</tr>
<tr>
<td>Deterministic</td>
</tr>
<tr>
<td>CRPS$_{\text{sum}}$</td>
</tr>
</tbody>
</table>
Figure 2: Prediction intervals and test set ground-truth from ProTran (our model) for the TRAFFIC dataset of the first 16 of 963 time series.

**Accuracy Comparison.** Table 1 shows that our models perform competitively across all five high-dimensional time series datasets, achieving $\text{CRPS}_{\text{sum}}$ comparable to the best methods on ELECTRICITY and WIKIPEDIA while outperforming all baselines, including a non-SSM transformer-based approach [72], by significant margins on SOLAR, TRAFFIC and TAXI. Further analyses with other metrics, including CRPS and MSE, in Appendix B also help confirm our findings.

**Qualitative Results.** Figure 2 shows that the distribution forecasts generated by our model follow closely the ground truths, which is consistent with our accuracy results. In addition, the model appears to capture the uncertainty of future forecasts to some extent; observations of large magnitudes and far into the future seem to correctly have higher variance estimates.

**Ablation Study.** We include a small scale ablation study on the TRAFFIC dataset to investigate which components of our models are essential. Table 2 suggests that removing the stochasticity from $w_t$ has most impacts on model performance, implying that incorporating latent variables into a transformer is indeed useful. Other aspects such as context attention or multiple layers of stochastic variables do not show dramatic effects in this study; however, they do contribute performance gains.

### 5.2 Human Motion Prediction

**Datasets.** Following the experiment setup in [97], we evaluate our models on two public motion capture datasets: Human3.6M [44] and HumanEva-I [78]. While Human3.6 is a large-scale dataset with 3.6 million video frames recorded at 50Hz, HumanEva-I is smaller with only 3 subjects and recorded at 60Hz. We follow the preprocessing steps of previous work [64, 97] and obtain a 17-joint skeleton for Human3.6 and a 15-joint skeleton for HumanEva-I. As in [97], we predict future motion for 2 seconds conditioning on observed motion of 0.5 seconds and 1 second conditioning on 0.25 seconds for Human3.6 and HumanEva-I, respectively.
Figure 3: Ground-truth pose sequences (first row) and corresponding predictions by ProTran (second row). Solid colors indicate later time-steps and faded ones are older. The body-part movements in the predicted and ground-truth poses resemble similar patterns, while certain variations are retained.

Table 3: Human motion prediction results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Human3.6M ADE ↓</th>
<th>Human3.6M FDE ↓</th>
<th>HumanEva-I ADE ↓</th>
<th>HumanEva-I FDE ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERD [32]</td>
<td>0.722</td>
<td>0.969</td>
<td>0.382</td>
<td>0.461</td>
</tr>
<tr>
<td>aclLSTM [56]</td>
<td>0.789</td>
<td>1.126</td>
<td>0.429</td>
<td>0.541</td>
</tr>
<tr>
<td>MT-VAE [95]</td>
<td>0.457</td>
<td>0.595</td>
<td>0.345</td>
<td>0.403</td>
</tr>
<tr>
<td>Pose-Knows [87]</td>
<td>0.461</td>
<td>0.560</td>
<td>0.269</td>
<td>0.296</td>
</tr>
<tr>
<td>HP-GAN [6]</td>
<td>0.858</td>
<td>0.867</td>
<td>0.772</td>
<td>0.749</td>
</tr>
<tr>
<td>Best-Many [11]</td>
<td>0.448</td>
<td>0.533</td>
<td>0.271</td>
<td>0.279</td>
</tr>
<tr>
<td>GMVAE [25]</td>
<td>0.461</td>
<td>0.555</td>
<td>0.305</td>
<td>0.345</td>
</tr>
<tr>
<td>DeliGAN [38]</td>
<td>0.483</td>
<td>0.534</td>
<td>0.306</td>
<td>0.322</td>
</tr>
<tr>
<td>DSP [98]</td>
<td>0.493</td>
<td>0.592</td>
<td>0.273</td>
<td>0.290</td>
</tr>
<tr>
<td>DLow [97]</td>
<td>0.425</td>
<td>0.518</td>
<td>0.251</td>
<td>0.268</td>
</tr>
<tr>
<td><strong>ProTran (Ours)</strong></td>
<td><strong>0.381</strong></td>
<td><strong>0.491</strong></td>
<td><strong>0.258</strong></td>
<td><strong>0.255</strong></td>
</tr>
</tbody>
</table>

**Metrics.** Following previous work on trajectory forecasting [2, 37], we adopt two popular metrics, namely, average displacement error (ADE) and final displacement error (FDE). ADE measures the average $L_2$ distance over all time steps between the ground truth motion and the closest sample, while FDE only considers such distance for the final pose.

**Baselines.** We compare our models against 9 models, including ERD [32] and aclLSTM [56], two deterministic RNN-based approaches; MT-VAE [95] and Pose-Knows [87], two conditional VAE models; HP-GAN [6], a conditional GAN; Best-Many [11], GMVAE [25], DeliGAN [38], and DSP [98], four approaches optimizing for diversity objectives. The results for these baselines are reported as in [97].

**Implementations.** Similar to the previous experiments, we use 8-head attentions and 2-layers MLPs. Since Human3.6M is significantly more complex and multi-modal than the time series forecasting datasets, we make use of 3 stochastic layers, as opposed to 2 layers for HumanEva-I. For Human3.6M, the context and target observations are significantly longer and set up for long-term predictions, so we only infer latent variables for target observations. Appendix C also contains further details about our models and their number of parameters.

**Quantitative Results.** Table 3 shows that our models convincingly outperform all baselines based on both metrics ADE and FDE, with the gains significantly higher for the larger dataset Human3.6M. We emphasize that our favorable performance is evaluated using random samples, while the closest competitor, DLow [97], relies on a separate model for selecting samples to promote diversity, which can potentially be combined with our probabilistic transformer for further improvements.

**Qualitative Results.** We show in Figure 3 human pose predictions made by our model that are most similar to the corresponding ground truths among a collection of such stochastic predictions. The similarities between the body-part movements in both sequences suggest that our model has been able to capture the temporal dynamics quite well.
6 Conclusion & Discussion

In this work, we have introduced generative models for multivariate time series that combines strengths of state space models and transformer architectures. In contrast to previous work, our models do not rely on recurrent neural networks but make extensive use of attention mechanism. We also extend our models to include hierarchical latent variables, inspired by recent developments of VAEs for non-sequential data [17, 83]. Empirical experiments show that our models perform remarkably well on time series forecasting and human motion prediction.

Our models do not come without limitations, however. As in other transformer-based approaches, the reliance on attention incurs a quadratic time and memory complexity. While we do not find it problematic in our experiments, the limitation necessarily hinders applications of our models in tasks characterized by long-term dependencies such as language modelling or music generation [36]. Fortunately, recent work on sparse transformer [9, 18, 50, 55] can potentially address the issue, and we leave such an investigation for future work.

Probabilistic time series forecasting is a fundamental research problem with wide-ranging applications in society. Although we have not explored healthcare applications of our work, previously proposed methods with similar formulations have demonstrated potentials of forecasting techniques [1, 81] in diagnoses or disease control.

7 Acknowledgment

Financial support is gratefully acknowledged from a Xerox PARC Faculty Research Award, National Science Foundation Awards 1455172, 1934985, 1940124, and 1940276, USAID, and Cornell University Atkinson Center for a Sustainable Future.
References


1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes] See Section 1 and 5.
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix D in supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix D in supplemental material.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.
   (b) Did you mention the license of the assets? [Yes] See Appendix A in supplemental material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See supplemental material. We are also committed to open source our code upon publication.
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [Yes] See Appendix A in supplemental material.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Appendix A in supplemental material.

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]