Volatility Based Kernels and Moving Average Means for Accurate Forecasting with Gaussian Processes

Gregory Benton [*1] **Wesley J. Maddox** [*1] **Andrew Gordon Wilson** [1]

Abstract

A broad class of stochastic volatility models are defined by systems of stochastic differential equations. While these models have seen widespread success in domains such as finance and statistical climatology, they typically lack an ability to condition on historical data to produce a true posterior distribution. To address this fundamental limitation, we show how to re-cast a class of stochastic volatility models as a hierarchical Gaussian process (GP) model with specialized covariance functions. This GP model retains the inductive biases of the stochastic volatility model while providing the posterior predictive distribution given by GP inference. Within this framework, we take inspiration from well studied domains to introduce a new class of models, *Volt* and *Magpie*, that significantly outperform baselines in stock and wind speed forecasting, and naturally extend to the multitask setting.

1. Introduction

Gaussian processes (GP) have had significant success in time series modeling, making them strong candidates for the challenging tasks of modeling and forecasting time dependent financial and climatological data. However, building GPs has historically relied on selecting out-of-the-box kernels and mean functions that make assumptions that do not hold for all cases; for example, Matérn and RBF kernels assume stationarity of the data, and constant or linear means assume that the underlying trends in the data are not time-varying. While the quest to build more flexible GPs has led to a significant amount of research into kernel functions such as spectral mixture and deep kernels, these kernels have been primarily developed for general problems, rather than constructing a kernel from prior knowledge about the task at hand. In this work, we approach GP modeling by building on domain knowledge to construct a novel set of kernel and mean functions with inductive biases well suited for forecasting in domains such as finance and climatology, where the data evolves stochastically through time.

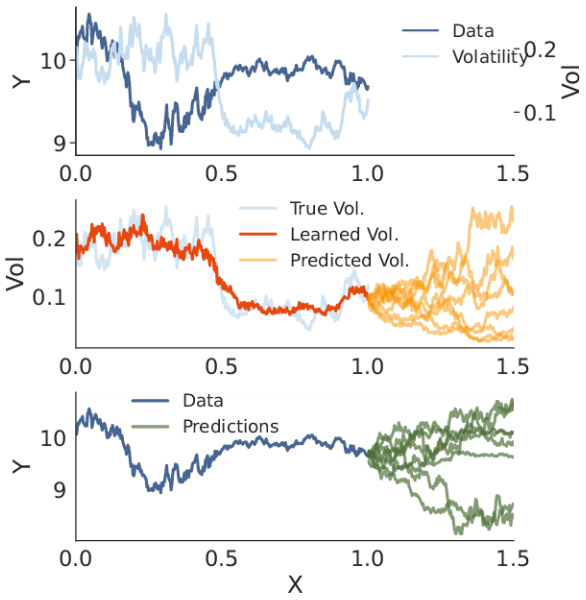


Figure 1. An overview of stochastic volatility, Volt, and forecasting. **Top:** the observed data over and the corresponding volatility path. **Middle:** the learned volatility from the data, and volatility forecasts. **Bottom:** the data over 1 year and forecasts, with each sample path corresponding to a distinct sample from the volatility forecast.

Both financial and climatological time series are nonstationary, and are characterized by having time-varying and stochastic *volatilities*, or degrees of variation. Motivated by stochastic volatility (SV) models, we develop a hierarchical GP model with specialized kernels, terming the model *Volt* Bollerslev (1986); Wilson and Ghahramani (2010); Hagan et al. (2002). Volt uses forecasts of volatility to specify the covariance structure over future data observations. By considering not only a single volatility forecast, but a distribution of volatility forecasts, we induce a distribution over covariance functions in the data domain. Accounting for un-

[*]Equal contribution [1]New York University. Correspondence to: Greg Benton <gwb260@nyu.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

certainty in volatility and propagating it to our data forecasts yields projected distributions that are well calibrated to the data, providing critical tools for understanding risk levels and simulating potential outcomes. For further information on stochastic volatility see Appendix A.1.

Figure 1 provides a graphical representation of the hierarchical GP model described by Volt. Given a set of observations (top row) we infer a volatility path over those returns (middle row), and form a hierarchical GP model where the first GP models volatility, and the second GP is used to forecast distributions over the data given samples from the volatility GP (bottom row).

The covariance structure described by Volt provides a faithful representation of the uncertainty in forecasts, but overlooks the mean function of the data space GP, which is a powerful tool for capturing trends in data. To that end, we jointly introduce Moving Average Gaussian Processes, or *Magpie*, in which we replace the standard parametric mean function in GPs with a moving average. Moving averages are a widely used technique in domains such as climatology and finance (Nau, 2014). By joining the trend fitting capabilities of moving averages with the probabilistic framework of GPs we can produce forecasts that are both accurate and have calibrated uncertainties.

While Volt and Magpie can be used separately, we present them as a single work because it is specifically their *combination* that solves challenging forecasting problems. In time evolving domains like stock prices or wind speeds, the inherent randomness of the processes prevents us from producing accurate point estimates far out into the future, and we need just accuracy, but uncertainty that is faithful to the stochasticity of the data. For this reason one needs both the accurate trend capture provided by Magpie, and the accurate uncertainty representation provided by Volt

Our key contributions are as follows:

- Deriving a hierarchical GP model, Volt, inspired by stochastic volatility models that produces calibrated forecasts of stochastic time series (Section 3).
- Describing a simple but powerful mean function, Magpie, that enables Gaussian process models to accurately forecast trends (Section 3).
- Using Volt and Magpie to produce highly calibrated forecasts in financial and climatological domains (Section 4).
- Extending our procedure to multitask problems by accounting for correlations in both volatility and price across different financial assets and different spatial locations (Section 5).
- We make code available here.

2. Related Work

Early autoregressive approaches to modeling the volatility of time series returns such as GARCH have seen widespread success (Bollerslev, 1986). These approaches typically view the volatility process as a time-evolving series, and are effective for inferring and forecasting volatility, but do not typically interface directly with a model over data as we have with Volt.

Volatility models have been extended to use both neural networks or Gaussian processes as their base components. For example, Cao et al. (2020) use a multi-layer perceptron to estimate volatility surfaces while Luo et al. (2018) use RNNs with rollouts to forecast volatility into the future but only considered one-step lookahead price forecasts. Wilson and Ghahramani (2010) use Gaussian processes to parameterize the volatility using Laplace approximations and MCMC sampling introducing the Gaussian process copula volatility model (GPCV), while Wu et al. (2014) used GP state space models and particle filters to estimate volatility. Similar to Wilson and Ghahramani (2010), Lázaro-Gredilla and Titsias (2011) used Gaussian processes to parameterize volatility models with an exponential link, but used a highly structured variational approximation for inference. Liu et al. (2020) used multi-task Gaussian processes to forecast volatility into the future, applying their models to foreign exchange currency returns, again with one-step lookahead forecasts in price. Crucially, predicting volatility alone does yield a straightforward path to forecast data, which is our central aim with Volt. Furthermore, Volt builds off of the GPCV, but other volatility estimation methods such as the ones described here could also be used.

Stochastic volatility models such as the Heston model (Heston, 1993) and SABR (Hagan et al., 2002), treat the evolution of the price of a security and the associated volatility as a coupled system of SDEs. Such SDEs are commonly used as methods for pricing financial derivatives. Differing from our viewpoint, these models are typically used to price stock options under risk-neutral measures, with Volt and Magpie we are focused on performing predictive inference by conditioning on observations.

The connection between Gaussian processes and SDEs has been extensively studied by Särkkä and Solin (2019) who suggest Kalman filtering based approaches for estimating GP hyperparameters in SDE-inspired GP models, which we do not consider here, preferring simply marginal likelihood based estimation. Systems of linear differential equations have been integrated into GP models previously via latent force models both for ordinary differential equations (Alvarez et al., 2009) and partial differential equations (Särkkä, 2011). To perform inference, Alvarez et al. (2009) derive covariances corresponding to the linear projection of the differential operator onto a specific kernel, while Särkkä

(2011); Särkkä and Solin (2019) use the projection operator explicitly to develop kernel functions to emulate systems of SDEs. Similarly, Zhu and Dunson (2013) use SDEs to derive a nested GP, but their approach produces a standard GP with a non-deep, but structured, covariance function. Autoregressive mean functions for GPs have been explored in Gonzalvez et al. (2019), however in their approach they use autoregressive features as inputs to a GP model, rather than as a way to specify the prior functions.

While many of the references above are focused specifically on finance, Volt and Magpie are applicable to a broad set of domains including climate modeling. Autoregressive and volatility models have successfully been applied to domains such as wind and precipitation forecasting as in Mehdizadeh et al. (2020); Liu et al. (2011) and Tian et al. (2018).

The Gaussian process autoregressive model (Requeima et al., 2019) stacks Gaussian processes of different tasks, using the GP for one task as the mean function for the next. It thus bears only slight resemblance to our moving average or multi-task approaches. Furthermore, many well-studied autoregressive models, e.g. the AR(p) family, can be written as Gaussian processes (Williams, 2010). As an alternative to developing domain specific kernel functions, one could alternatively construct manual combinations of generic kernels, which either requires significant amounts of hand-tuning as in Rasmussen and Williams (2008, Ch 5.4,) or solving discrete optimization problems (Lloyd et al., 2014; Sun et al., 2018). As we wish to develop our models efficiently and succinctly, we also do not consider these models.

3. Methods

We first begin with a brief overview of Gaussian process regression models, before deriving the Volt kernel and Magpie mean functions in Section 3.1. After deriving the Volt kernel and Magpie mean, we explain the inference procedure in Section 3.2 and how we perform forecasting in Section 3.3.

Gaussian Processes Please see Rasmussen and Williams (2008) for a more detailed introduction to Gaussian processes (GPs). We assume noisy observations $y(t) \sim \mathcal{N}(f(t), \sigma^2)$, where $f \sim \mathcal{GP}(\mu(t), k(t,t'))$, so that σ is the observation noise and f is drawn from a GP with mean function $\mu(t)$ and $k(t,t')$ as the covariance function. When using GPs, we can compute the posterior predictive distribution, $p(f(\mathbf{t}^*)|\mathcal{D})$, $\mathcal{D} := \{\mathbf{t}, \mathbf{y}\}$, over new data points $\mathbf{t}^*$ is given by $p(f(\mathbf{t}^*)|\mathcal{D}, \theta) = \mathcal{N}(\mu^*_{f|\mathcal{D}}, \Sigma^*_{f|\mathcal{D}})$ where $\mu^*_{f|\mathcal{D}} = K_{\mathbf{t}^*\mathbf{t}}(K_{\mathbf{tt}}+\sigma^2 I)^{-1}(\mathbf{y}-\mu(\mathbf{t}))+\mu(\mathbf{t}^*)$ and $\Sigma^*_{f|\mathcal{D}} = K_{\mathbf{t}^*\mathbf{t}^*} - K_{\mathbf{t}^*\mathbf{t}}(K_{\mathbf{tt}} + \sigma^2 I)^{-1}K_{\mathbf{tt}^*}$ with $K_{A,B} := k(A,B)$.

3.1. Volt and Magpie

We make the common assumption that both the data $S(t)$, and volatility $V(t)$, have paths with log-normal marginal distributions. We therefore place the following joint SDE structure over $s(t) = \log S(t)$ and $v(t) = \log V(t)$,

$$\begin{aligned} ds(t) &= \mu_s dt + V(t) dW(t) \\ dv(t) &= -\frac{\sigma^2}{2} dt + \sigma dZ(t). \end{aligned} \tag{1}$$

The drift term in Equation (1), $-\frac{\sigma^2}{2}dt$, arises from the log-transformation of the volatility, and ensures that forecast distributions over volatility have a constant mean (for further details see Appendix B.2). Furthermore, this structure allows us to derive closed form expressions for and auto-covariance functions associated with both log-data and log-volatility, allowing us to define the Volt model.

Equation (1) gives a relationship between the log-price and log-volatility that is mirrored by many stochastic volatility models, including GARCH and SABR, where the volatility of the price is itself governed by an SDE (Bollerslev, 1986; Hagan et al., 2002). By recasting Equation (1) as a system of GPs we can move from an SDE sampling approach to a proper forecasting system based on historical observations.

A Gaussian Process Perspective Since for any finite collection of time points, $\mathbf{t} = \{t_i\}_{i=1}^N$, the observations $v = v(\mathbf{t})$ and $s = s(\mathbf{t})$ each have a multivariate normal distribution, v and s now correspond to Gaussian processes. Therefore we only need to derive the mean and covariance functions of the two processes to fully cast our problem as one of forming predictive distributions from GPs.

As $v(t)$ is a scaled Wiener process with constant drift term, the autocovariance function is

$$K_v(t,t') = \sigma^2 \min\{t, t'\} \tag{2}$$

and the mean is $\mu_v(t) = -t\frac{\sigma^2}{2}$ so that, $v(t) \sim \mathcal{GP}(\mu_v(t), K_v(t,t'))$. Conditional on a realization of $V(t) = \exp v(t)$, $s(t)$ is also described by a Gaussian process with $\mathbb{E}[s(t)] = \int_0^t \mu_s dt = t\mu_s$ and ,

$$Cov(s(t), s(t')) = \int_0^{\min\{t,t'\}} V(t)^2 dt = K_s(t,t'; V(t)), \tag{3}$$

producing our model over log-data:

$$s(t) \sim \mathcal{GP}\left(t\mu_s + s(0), K_s(t,t'; V(t))\right). \tag{4}$$

The final Volt model is then a hierarchical composition of

Gaussian processes:

$$\begin{aligned} v(t) &\sim \mathcal{GP}(m_v(t), K_v(t,t')) \\ V(t) &= \exp\left(v(t)\right) \\ s(t) &\sim \mathcal{GP}(m_s(t), K_s(t,t'; V(t))) \\ S(t) &= \exp\left(s(t)\right), \end{aligned} \tag{5}$$

The log-volatility is distributed as a Gaussian process dependent on the the time inputs, the mean m_v, and the *volvol* hyperparameter σ and has a Brownian motion covariance (Eq. 2). Given a realization of a volatility path over time and the parameters of the log-linear mean, the log-price is also distributed as a Gaussian process with covariance given by Eq. 3. To generate predictions using the log-volatility and log-price GPs we first must infer both a volatility path from the observed time series, $S = S(\mathbf{t})$, and the hyperparameters of both the data and volatility models. A complete derivation of the GPs in Equation (5) is in Appendix B.2.

Magpie For the sake of deriving the covariance functions associated with the log-data and log-volatility processes, we have left the mean functions of the data GP in Equation (5) as a simple linear function. While we may believe that there are nontrivial trends in the data over time, we also believe that these trends may be more complex than simple polynomial or periodic functions, and in the context of applications like finance and climatology are likely to change over time with evolving market or climatological conditions.

To address these deficiencies in using simple mean functions in modeling nonstationary signals we replace the simple mean functions typically found in GP models with exponential moving averages (EMA) (Nau, 2014). We use the EMA with a limited number of terms, defined as

$$\begin{aligned} EMA(\mathbf{s})_{i+1} =& \alpha[s_i + (1-\alpha)s_{i-1} + (1-\alpha)^2 s_{i-2} \\ &+ \cdots + (1-\alpha)^{k-1} s_{i-(k-1)}] \end{aligned} \tag{6}$$

where $\alpha = 2/(k+1)$ is a hyperparameter governing the smoothing of the moving average. A smaller value of k uses only more recent observations, enabling a closer match of the data, whereas a larger value of k uses more data and smooths the data more.

While we focus on the EMA in Equation (6), Magpie naturally extends to alternate moving averages, such as lag-corrected moving averages. We provide comparisons of these alternate moving averages, as well as the effect of the k hyperparamter in Appendix Figure 8 and in the extended results of Section 4. With Equation (6) we can define the Magpie mean function as $m_{EMA}(t_{i+1}, \mathbf{s}) = EMA(\mathbf{s})_{i+1}$.

We close this section by noting that moving from a linear to a exponential moving average mean for the GPs breaks the connection with the SDEs described in Section 3.1, making the combination of Volt and Magpie necessarily a practical approach, rather than an entirely theoretically motivated approach.

3.2. Inference

Here we outline the procedure for using a series of price observations to train the hyperparameters of the GPs in Equation 5, and form the associated posterior predictive distributions. In general the training procedure can be thought of as a three step process: a) use a Gaussian Process Copula Volatility (GPCV) model to infer a volatility path, V, given a sequence of observations S, b) learn the hyperparameters of the GP in log-volatility space by maximizing the Marginal Log-Likelihood (MLL) with respect to the GPCV inferred volatility, c) learn the hyperparameters of the GP in log-data space by maximizing the MLL with respect to the observed prices, using the kernel generated by the GPCV inferred volatility path. Note that our use of the GPCV to estimate volatility is a modelling choice and we could alternatively have used any other volatility estimation model such as GARCH.

Inferring Volatility from Training Data One challenge in formulating the model outlined in Equation (5) is the need to have both data and volatility observations for some range of training observations. To estimate the volatility, we use a variant of Gaussian copula process volatility (GPCV) model first proposed by Wilson and Ghahramani (2010). Our GPCV model uses a warped Gaussian process to model the variability of the responses, $w(t)$, according to:

$$\begin{aligned} f(t) &\sim \mathcal{GP}(c, K_v(t,t')) \\ \gamma(f(t)) &= \exp\{f(t)\} \\ w(t) &\sim \mathcal{N}(0, \gamma^2(f(t))). \end{aligned} \tag{7}$$

We use the kernel derived from log-volatility SDE in Equation (5) to infer the latent function $f(t)$, and use variational inference (Hensman et al., 2013; 2015) to train the model. See Appendix B.3 for further details.

Following Wilson and Ghahramani (2010), we consider the responses as the log-returns of the data, that is: $w(t_i) = \log S(t_i) - \log S(t_{i-1})$. We construct a volatility prediction over times $0, \cdots, t-1$ by drawing posterior samples from $f(t)$ and passing them through the warping function $\sigma(\cdot)$; so our estimate for $V(t)$ is

$$\hat{V}(t) := \frac{1}{J}\sum_{j=1}^{J} \gamma(f_j(t)), \;\; f_j(t) \sim q(f(t)|w(t), v, \theta), \tag{8}$$

where $q(f(t)|w(t), v, \theta)$ is our approximate posterior distribution over the latent function $f(t)$. We demonstrate that our approach is able to correctly estimate the true volatility in Figure 1, where the volatility and price are drawn from a SABR volatility model (Hagan et al., 2002).

Training the Gaussian Processes Given the volatility path associated with the training data learned using a GPCV, we assume a Gaussian process priors over the log-volatility and log-data according to Equation (5). Given the volatility over the training data, the single hyperparameter of the log-volatility model is the σ^2 term describing the *volvol*. The hyperparameters of the log-data model are just the parameters of the mean in Equation (5), of which there are none if we are using a non-parametric mean like Magpie. To train we maximize the MLL of the models with respect to their hyperparameters using gradient based optimization (Rasmussen and Williams, 2008, Chapter 5). The total computational cost for inference in Volt, regardless of the use of a Magpie mean, is just the cost of training *one* variational GP and two standard GP models on evenly spaced data, which can be done efficiently via exploiting the (Toeplitz) structure of the data (Wilson and Nickisch, 2015).

3.3. Predictions

In Volt, we condition the log-volatility GP on a log-path inferred by GPCV and the log-data GP on historical observations of log-price and draw samples from the *posterior* distributions, producing a mixture of log-normal distributions over data. Sampling the posterior requires sample N_v log-volatility paths, v^*, over the test inputs and for each of these we generate a kernel $K_s(t, t', V^*)$, and sample N_s data paths, $S^* = \exp(s^*)$, producing $N_v \times N_s$ samples.

In the Gaussian process viewpoint of Section 3.1, standard Monte Carlo simulation of an SDE procedure is equivalent to sampling log-volatility paths, $v^* = v(\mathbf{t}^*)$, from the *prior* distributions of Equation (5) up to time T rather than the *posterior* distribution (Sauer, 2012). With the prior samples of $S_T^* = \exp(s_T^*)$, we can form a Monte Carlo estimate of future distributions over price. However, the distinction between this type of approach and our approach for sampling with Volt is that Volt samples from the *posterior* distributions over volatility and data conditional on observations, while the SDE based approaches sample from the prior distribution over volatility.

Rollout Predictions The Magpie mean only allows for predictions one step ahead, so we do our forecasting in a *rollout* fashion. That is, we use observations $s_0, \ldots, s_t$ to sample $\hat{s}_{t+1}$ from the GP posterior $p(s_{t+1}|s_0, \ldots, s_t)$, then condition our GP (and Magpie mean) on $\hat{s}_{t+1}$ in order to sample $\hat{s}_{t+2}$ from the updated GP posterior $p(s_{t+2}|s_0, \ldots, s_t, \hat{s}_{t+1})$, and so on. These rollout forecasts are critical to the Magpie framework. By sequentially sampling the price forecasts and updating the GP with each observation we allow for trend reversals in the moving average mean in a way that is not possible with other GPs. Rollouts are unnecessary for traditional means because the conditional means over each time step factorize into a single multivariate Gaussian distribution.

4. Forecasting

In both financial and climatological applications we are considering the data as stochastically evolving and are thus interested in forecasting *distributions* over outcomes, rather than point estimates. For this reason we use *calibration* and negative log likelihood as our primary measures of interest, rather than an accuracy metric like mean squared error.

We compute the calibration at percentile p by computing the frequency with which the true observation is less than the empirically computed p^{th} quantile of the forecast distribution. More specifically, for a forecast of the price stock S at time T and percentile p we compute the empirical quantile of the forecast q_T where $\hat{\mathbb{P}}(S_T < q_T) = p$. We can then compute the calibration at p as the empirically observed frequency of the event $S_T < q_T$ by calculating $C_p = \frac{1}{K}\sum_k \mathbb{I}_{\{S_{T_k} < q_{T_k}\}}$ as the average frequency of $S_T < q_T$ over K different forecasts. If our forecasts are well calibrated then this empirical frequency will be close to p for each value of p; therefore by computing the calibration of our forecasts at a range of percentiles, p, we can determine the overall calibration of the forecast distribution. Such a calibration metric is similar to those explored for regression in Kuleshov et al. (2018).

Note that for accurate calibration to occur in this setting our forecast distribution must match the empirical observations at all quantiles. We could not, for example, just forecast that a price increases some fixed percentage of the time that matches the observed frequency of the price increased and expect to achieve accurate calibration.

4.1. Stock Price and Foreign Exchange Rate Forecasting

As Volt and Magpie are primarily inspired by financial time series models, forecasting distributions over stock prices is a core application of our approach. We compare Volt and Magpie to baseline models of GPs with standard kernel and mean functions. Along with these GP models, we include probabilistic LSTMs where we optimize a predicted mean and variance at each time step with respect to the negative log-likelihood (NLL) which have been previously used in a quantitative finance setting (Chauhan et al., 2020). All models assume the marginal distributions of the observations are normally distributed, thus we model the log-price of stocks in each case.

Figure 2 provides a representative comparison of forecasts generated by GPs both with and without Volt and Magpie, and the probabilistic LSTMs used here. Simpler probabilistic models like standard GPs and probabilistic neural networks generally provide overconfident forecasts, and more

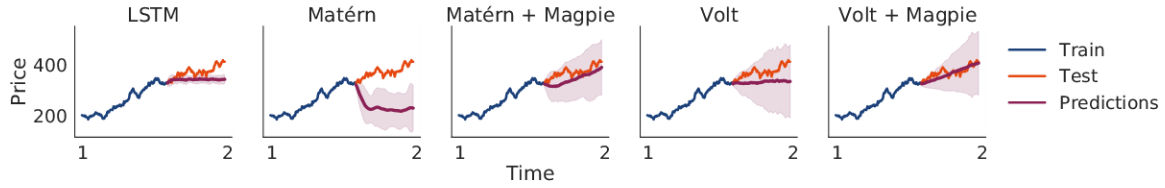


Figure 2. Simulations and forecasts showing the mean and 95% confidence region for various model choices. Probabilistic LSTMs perform well on the training data, but do not extrapolate far from observed data well. Matérn forecasts quickly revert to constant level of uncertainty which leads to overconfidence far away from observations, whereas Volt's increase in uncertainty as we move away from training data produces well calibrated to the data. The constant mean forecasts in both Matérn and Volt fail to pick up the long term trend in the data, which Magpie means accurately capture. The combination of Volt and Magpie, with correct inductive biases in both the kernel *and* mean functions produces forecasts consistent with trends in the data and with well calibrated uncertainty.

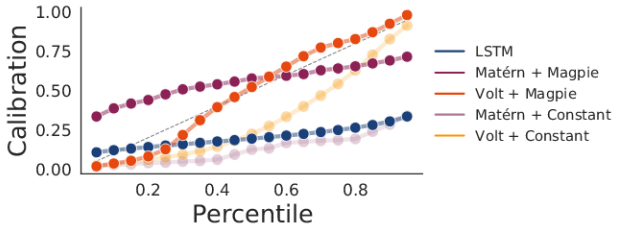


Figure 3. Calibration of various approaches on the 2 years of data from the NASDAQ 100. The forecasts generated by standard kernels and probabilistic LSTMs are significantly overconfident, leading to very poor calibration.

	Stock Prices	Wind Speeds
Volt + Magpie	5.88 ± 0.02	4.28 ± 0.16
Volt + Con.	$\mathbf{4.69 \pm 0.03}$	$\mathbf{3.38 \pm 0.05}$
Matérn + Magpie	9.80 ± 0.27	12.13 ± 0.81
Matérn + Con.	7.74 ± 0.21	18.03 ± 1.90
SM + Magpie	147.84 ± 1.84	110.07 ± 7.81
SM + Con.	80.43 ± 0.57	70.14 ± 5.03
LSTM	49.95 ± 0.59	45.13 ± 1.82
Volt-VHGP + Con.	4.76 ± 3.05	5.75 ± 0.44
Volt-VHGP + Magpie	6.97 ± 1.24	5.91 ± 0.34
GPCV	5.45 ± 1.51	4.89 ± 0.04

Table 1. Negative log likelihoods (NLLs) per test point with 2 standard deviations for the methods compared on both the stock forecasting and wind speed tasks. By accounting for uncertainty in both the volatility and the data forecasts, Volt provides highly accurate test distributions relative to baseline approaches. Volt-VHGP indicates a Volt model where we use variational heteroscedastic GPs from Lázaro-Gredilla and Titsias (2011) in place of GPCV. We provide expanded results including foreign exchange data in Appendix C. In each case the mean and standard deviation are computed over approximately 2 thousand time series 75 to 100 time steps into the future, yielding tens of thousands of individual forecasts.

traditional mean functions in GP models do not capture the long range trends that are commonly present in financial time-series data.

Figure 3 shows the calibration of the compared methods aggregated over thousands of forecasts. We consider stocks in the NASDAQ 100 collection, and a history of two years of daily observations leading up to January 2022. For 25 evenly spaced days we forecast 1000 paths 100 days into the future and compute the calibration curves of the forecasts for the days 75 to 100 days out.

We see in Figure 3 that Volt is able to remedy a significant overconfidence that is present in alternative methods such as standard GP kernels or probabilistic LSTMs. Furthermore, it is the Magpie mean function that enables the distributions to be centered at the correct values, which is why we see the LSTMs and constant mean GPs showing bias in the calibration plots. Note that for accurate calibration to occur in this setting our forecast distribution must match the empirical observations at *all* quantiles. We could not, for example, just forecast that a price increases some fixed percentage of the time that matches the observed frequency of the price increased and expect to achieve accurate calibration.

Table 1 gives the average test negative log likelihood (NLL) values on the stock forecasting task and on the foreign exchange data from Lai et al. (2018). Both variants of the Volt model outperform competing methods such as LSTMs, and GPs with Matérn and Spectral Mixture (SM) kernels (Wilson and Nickisch, 2015). Volt with a constant mean is slightly better than with a Magpie mean in terms of NLL, the Magpie mean is key to achieving high calibration, as we see in Figure 3. SM kernels are a highly class of flexible kernel, but rely on there being frequency components in the data, with non-stationary data such as those studied here, the lack of regularity in the data leads to weak performance.

4.2. Wind Speed Forecasting

Probabilistic forecasting models play an important role in statistical climatology in providing forecast distributions over quantities of interest, such as rainfall or wind speed, that can be used to generate synthetic data or estimate the risk of extreme events. Stochastic volatility models have a

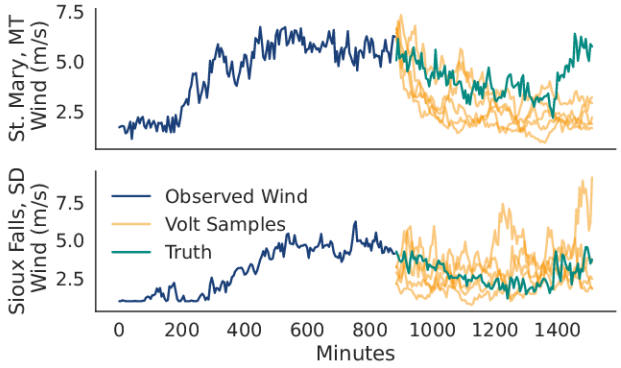


Figure 4. A representative example of observed wind speed and samples of multitask Volt forecasts for two related observation stations. While none of the Volt forecasts perfectly fit the true future observations of wind, each individual roll is a realistic potential realization of wind speed. By generating many plausible outcomes we are able to forecast distributions over wind speed that are highly calibrated to held out test observations.

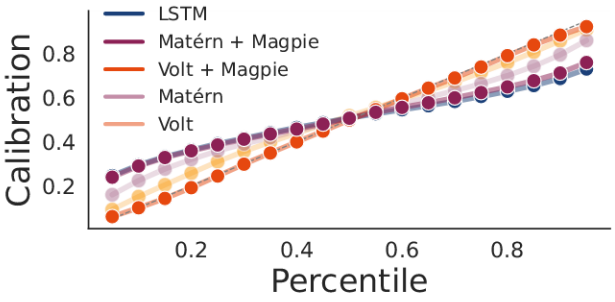


Figure 5. A calibration plot for wind speed, aggregated over hundreds of thousands of distinct forecasts. While probabilistic LSTMs and standard GP models provide competitive baselines, the Volt and Magpie model generates wind speed distribution forecasts that are extremely well calibrated. These calibrated distributions enable us to quickly simulate thousands of scenarios that can be trusted to faithfully represent potential outcomes.

history of use in modeling wind speed, but typically these models have been limited to GARCH based approaches (Liu et al., 2011; Tian et al., 2018), and are thus focused on the *volatility* of wind, rather than forecasting distributions of wind speed itself.

Here we apply Volt and Magpie to the problem of developing a stochastic weather model for wind speed. We source historical wind data from the U.S. Climate Reference Network (USCRN), with observations taken at 5 minute intervals over the 2021 calendar year at 154 spatial locations in the United States (Diamond et al., 2013). Figure 4 provides an example of what the wind speed observations look like, as well as example Volt forecasts in comparison to ground truth held out data. Each forecast path represents a realistic scenario drawn from a distribution over paths from which the true data would, hypothetically, be a representative candidate. By sampling paths from the forecast distribution over wind speeds we can simulate future observations with accurate probability enabling us to estimate statistics of interest, such as expected wind speed or the probability of extreme events.

As with stock price forecasting, we are interested in producing forecast distributions that match the ground truth of the data, rather than attempting to generate point predictions. In Figure 5 we compare the calibration of forecasts against the ground truth wind speeds in the forecast windows. Table 1 provides the NLL values of the various approaches. As with stock forecasting, we see that constant means do provide slightly better NLL values than Magpie means, but Volt models are key to producing accurate forecasts. Distinct from stock price forecasting however, is the bounded nature of wind speed. As we do not expect wind speed to grow indefinitely (as we may see with stocks) we forecast with a small amount of mean reversion applied to the GP models. Experimental details, including a sensitivity to the mean reversion can be found in Appendix C.2.

5. Multi-Task Volatility Modelling

Finally, we extend Volt to model several asset prices at once by using multi-task Gaussian processes with the goal of jointly modelling different time series at once, such as the wind speeds for the continental United States.

First, we extend the GPCV of Wilson and Ghahramani (2010) to several tasks before then placing a multi-task model over volatility in the hierarchical GP formulation. Our approach enables simultaneous estimation of the time series, its volatility, and the relationships between the time series themselves.

5.1. A Multitask GPCV

We extend the Gaussian process copula volatility model (GPCV) described in Section 3.2 to model several jointly related volatilities at once by using multi-task Gaussian processes (Bonilla et al., 2007; Alvarez et al., 2012). We assume that all returns and volatilities are observed at once, with P different responses, so that the covariance between the pth and p'th latent Gaussian process is given by: $k([t,p],[t',p']) = K_v(t,t')K_i(p,p')$ where $K_i(p,p')$ is a lookup table describing the intertask covariance. The intertask covariance is a $P \times P$ matrix; we can regularize it with a LKJ prior (Lewandowski et al., 2009) or to incorporate side information such as geographic coordinates.

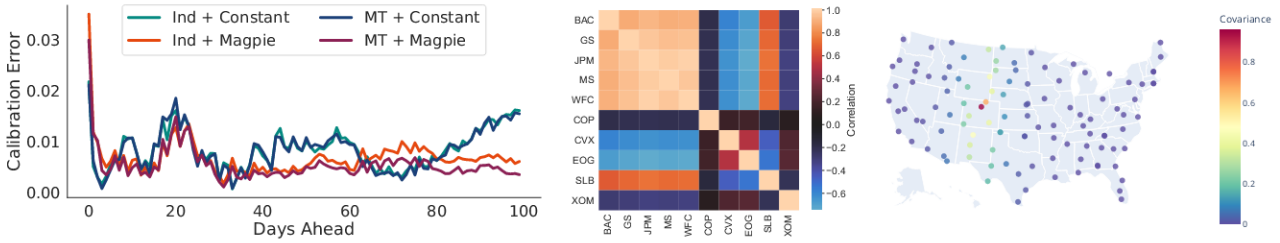


Figure 6. **Left panel:** Calibration error of both Volt and MT Volt at the 95% confidence level as a function of time step lookahead over 30 stocks from an entire sector ETF (XLF). While both are well-calibrated, MT Volt preserves well-calibration to longer time steps. **Center panel:** Estimated correlation matrix of stocks from two different sectors. MT-Volt successfully learns the high volatility correlation amongst the finance sector stocks (first five) with lower correlations between the energy sector stocks (second five). **Right panel:** Estimated volatility covariance with Boulder, CO. Correlations decrease as the stations go further away.

We have the multi-task probabilistic model:

$$f_p(t) \sim \mathcal{GP}(c, K_v(t,t')K_i(p,p')) \\ w_p(t) \sim \mathcal{N}(0, \exp^2(f_p(t)). \tag{9}$$

Again, we use variational inference to infer the latent posterior distribution over each price's latent Gaussian process, see Appendix B.3 for more details. We also use posterior samples from this multi-task GPCV to estimate volatility, $\hat{V}_p(t)$, for each stock price p by following Eq. 8.

In Appendix Figure 11, we simulate price data from a correlated SABR volatility model and use our multi-task GPCV to recover both the volatility as well as the latent correlation structures. This suggests that our inference scheme enables us to accurately recover latent correlations.

5.2. Multi-Task Stock Modeling

After using a multi-task GPCV to estimate the volatility for us, we then use a multi-task Gaussian process model (Bonilla et al., 2007) to estimate volatility, producing the following probabilistic model:

$$v_p(t) \sim \mathcal{GP}(m_v, K_v(t,t')K_p(t,t')) \\ V_p(t) = \exp\left(v_p(t)\right) \\ s_p(t) \sim \mathcal{GP}(m_{s,p}(t), K_{s,p}(t,t'; V_p(t))) \\ S_p(t) = \exp\left(s_p(t)\right). \tag{10}$$

Conditional on the correlated volatility paths, the prices themselves are independent, so we use P independent Gaussian process models to model the prices. Intuitively, this dependency structure makes sense as we expect exogenous shocks (for example, large scale macroeconomic trends) to affect variability in an asset prices, rather than just directly producing an increase or decrease.

5.3. Multitask Stock Price Prediction

In Figure 6 left panel, we consider the calibration of both Volt and MT Volt on predictions across five different groupings of stocks each with between 5 and 30 different stocks in each group, finding that all models are fairly well calibrated in terms of the calibration error, which is the squared error of the average calibration across bins of the empirical observed calibration of the foreast (Kuleshov et al., 2018). The mult-task models tend to improve calibration over independent models, especially when using Magpie means. We display the results for calibration across time steps, mean absolute error (MAE) and negative log likelihood (NLL) in Appendix C.3.

In Figure 6 center, we showcase how the multi-task Volt model of volatility can be used to measure the relationships between assets. We considered 10 stocks, five from the financial sector and five from the energy sector. Volt learns strong correlations amongst the stocks in the financial sector and much weaker cross-correlations with the energy stocks.

5.4. Spatiotemporal Wind Modelling

Finally, we consider multi-task modelling for stochastic weather generation. Here, as we have longitude and latitude coordinates for each of the weather locations, we can incorporate this information into the inter-task covariance matrix by using a geodesic exponential kernel, which is given as $k(x,y) = \exp\{-\arccos(x^\top y)/2\sigma^2\}$ for $x, y \in \mathbb{S}^2$, that is points on the unit sphere (Jayasumana et al., 2013). Note that we have no restrictions on kernel choice and could alternatively consider non-stationary kernels here instead.

We model 110 stations across the United States in the year 2021 again at 5 minute intervals, estimating the relationship between each station using the geodesic exponential kernel described above, and learning the lengthscale. We display the results in Figure 6 right panel with the stations described on a map of the United States. Further experimental results are shown in Appendix C.3.

6. Conclusion

We have proposed Volt and Magpie, which use kernels and mean functions derived from stochastic volatility, in order

to introduce a powerful forecasting method for stochastically generated time series. Volt deviates from the usual assumptions of stochastic differential equation (SDE) models for financial and climatological models, and incorporates historical data through GPs, allowing us to better estimate expectations and forecast distributions. Magpie allows us to replace the often over-simplified mean functions in Gaussian process models with a nonstationary mean leading to forecasts that more closely represent the data.

We have demonstrated that Volt and Magpie can outperform competing methods in generating forecast distributions of stochastically generated processes, with an emphasis on financial and climatological applications. The strong predictive uncertainties allow our method to be used for price forecasting and weather generation in a reliable way that produces trustworthy forecasts that are well calibrated to observations. Finally, we proposed a multi-task extension to Volt that improves on Volt's predictive calibration while additionally allowing for the estimation of the relationships between several assets at once.

The potential applications of our approach are broad, with potential uses in financial domains such as automated trading and strategy development, and climatological research in which Volt and Magpie could serve as a backbone for large spatiotemporal climate models. In the future, it would be useful to extend both the single and multi-task models to use online variational inference (Bui et al., 2017; Maddox et al., 2021b) to enable online deployment of scalable forecasting strategies. We hope our work will catalyze further development of domain based prior kernels for Gaussian processes, and applications of probabilistic machine learning to financial climatological data.

Acknowledgements

We would like to thank Andres Potapczynski for helpful discussions. This research is supported by an Amazon Research Award, Facebook Research, Google Research, Capital One, NSF CAREER IIS-2145492, NSF I-DISRE 193471, NIH R01DA048764-01A1, NSF IIS-1910266, and NSF 1922658 NRT-HDR.

References

Alberg, J. and Lipton, Z. C. (2017). Improving factor-based quantitative investing by forecasting company fundamentals. *arXiv preprint arXiv:1711.04837*.

Alvarez, M., Luengo, D., and Lawrence, N. D. (2009). Latent force models. In *Artificial Intelligence and Statistics*, pages 9–16. PMLR.

Alvarez, M., Rosasco, L., and Lawrence, N. (2012). Kernels for vector-valued functions: a review. *Foundations and Trends® in Machine Learning*, 4(3):195–266.

Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. (2020). Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33.

Benton, G., Maddox, W. J., Salkey, J., Albinati, J., and Wilson, A. G. (2019). Function-space distributions over kernels. *Advances in Neural Information Processing Systems*, 32:14965–14976.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327.

Bonilla, E. V., Chai, K. M., and Williams, C. (2007). Multi-task gaussian process prediction. *Advances in neural information processing systems*, pages 153–160.

Bui, T. D., Nguyen, C. V., and Turner, R. E. (2017). Streaming sparse gaussian process approximations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3301–3309.

Cao, J., Chen, J., and Hull, J. (2020). A neural network approach to understanding implied volatility movements. *Quantitative Finance*, 20(9):1405–1413.

Chandorkar, M., Camporeale, E., and Wing, S. (2017). Probabilistic forecasting of the disturbance storm time index: An autoregressive gaussian process approach. *Space Weather*, 15(8):1004–1019.

Chauhan, L., Alberg, J., and Lipton, Z. (2020). Uncertainty-aware lookahead factor models for quantitative investing. In *International Conference on Machine Learning*, pages 1489–1499. PMLR.

Dai, Z., Álvarez, M. A., and Lawrence, N. D. (2017). Efficient modeling of latent information in supervised learning using gaussian processes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5137–5145.

Diamond, H. J., Karl, T. R., Palecki, M. A., Baker, C. B., Bell, J. E., Leeper, R. D., Easterling, D. R., Lawrimore, J. H., Meyers, T. P., Helfert, M. R., et al. (2013). Us climate reference network after one decade of operations: Status and assessment. *Bulletin of the American Meteorological Society*, 94(4):485–498.

Elliott, R. J. and Kopp, P. E. (2001). Mathematics of financial markets. pages 361–380. Springer.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in Neural Information Processing Systems*, 31:7576–7586.

Gonzalvez, J., Lezmi, E., Roncalli, T., and Xu, J. (2019). Financial applications of gaussian processes and bayesian optimization. *arXiv preprint arXiv:1903.04841*.

Hagan, P. S., Kumar, D., Lesniewski, A. S., and Woodward, D. E. (2002). Managing smile risk. *The Best of Wilmott*, 1:249–296.

Hamilton, J. D. (2020). *Time series analysis*. Princeton university press.

Heaukulani, C. and van der Wilk, M. (2019). Scalable bayesian dynamic covariance modeling with variational wishart and inverse wishart processes. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 4582–4592.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 282–290.

Hensman, J., Matthews, A., and Ghahramani, Z. (2015). Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., and Harandi, M. (2013). Combining multiple manifold-valued descriptors for improved object recognition. In *2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6. IEEE.

Kuleshov, V., Fenner, N., and Ermon, S. (2018). Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pages 2796–2804. PMLR.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. (2018). Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104.

Lázaro-Gredilla, M. and Titsias, M. K. (2011). Variational heteroscedastic gaussian process regression. In *International Conference on Machine Learning*.

Lewandowski, D., Kurowicka, D., and Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of multivariate analysis*, 100(9):1989–2001.

Liu, B., Kiskin, I., and Roberts, S. (2020). An overview of gaussian process regression for volatility forecasting. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 681–686. IEEE.

Liu, H., Erdem, E., and Shi, J. (2011). Comprehensive evaluation of arma–garch (-m) approaches for modeling the mean and volatility of wind speed. *Applied Energy*, 88(3):724–732.

Lloyd, J., Duvenaud, D., Grosse, R., Tenenbaum, J., and Ghahramani, Z. (2014). Automatic construction and natural-language description of nonparametric regression models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Luo, R., Zhang, W., Xu, X., and Wang, J. (2018). A neural stochastic volatility model. In *Thirty-second AAAI conference on artificial intelligence*.

Maddox, W. J., Balandat, M., Wilson, A. G., and Bakshy, E. (2021a). Bayesian optimization with high-dimensional outputs. *Advances in neural information processing systems*.

Maddox, W. J., Stanton, S., and Wilson, A. G. (2021b). Conditioning sparse variational gaussian processes for online decision-making. *Advances in Neural Information Processing Systems*, 34.

Matthews, A. G. d. G. (2017). *Scalable Gaussian process inference using variational methods*. PhD thesis, University of Cambridge.

Mehdizadeh, S., Kozekalani Sales, A., and Safari, M. J. S. (2020). Estimating the short-term and long-term wind speeds: implementing hybrid models through coupling machine learning and linear time series models. *SN Applied Sciences*, 2:1–15.

Nau, R. (2014). Forecasting with moving averages. *Fuqua School of Business, Duke University*, pages 1–3.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Philipov, A. and Glickman, M. E. (2006). Multivariate stochastic volatility via wishart processes. *Journal of Business & Economic Statistics*, 24(3):313–328.

Rakitsch, B., Lippert, C., Borgwardt, K., and Stegle, O. (2013). It is all in the noise: Efficient multi-task gaussian process inference with structured residuals. *Advances in neural information processing systems*, 26:1466–1474.

Rasmussen, C. E. and Williams, C. K. I. (2008). *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 3. print edition.

Requeima, J., Tebbutt, W., Bruinsma, W., and Turner, R. E. (2019). The gaussian process autoregressive regression model (gpar). In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1860–1869. PMLR.

Rosenkrantz, W. A. (2003). Why stock prices have a lognormal distribution. *Department of Mathematics and Statistics, University of Massachusetts at Amhers*.

Rouah, F. D. (2007). The sabr model. https://www.frouah.com/finance%20notes/The%20SABR%20Model.pdf.

Särkkä, S. (2011). Linear operators and stochastic partial differential equations in gaussian process regression. In *International Conference on Artificial Neural Networks*, pages 151–158. Springer.

Särkkä, S. and Solin, A. (2019). *Applied stochastic differential equations*, volume 10. Cambridge University Press.

Sauer, T. (2012). Numerical solution of stochastic differential equations in finance. In *Handbook of computational finance*, pages 529–550. Springer.

Sharpe, W. F. (1994). The sharpe ratio. *Journal of portfolio management*, 21(1):49–58.

Steinruecken, C., Smith, E., Janz, D., Lloyd, J., and Ghahramani, Z. (2019). The automatic statistician. In *Automated Machine Learning*, pages 161–173. Springer, Cham.

Sun, S., Zhang, G., Wang, C., Zeng, W., Li, J., and Grosse, R. (2018). Differentiable compositional kernel learning for gaussian processes. In *International Conference on Machine Learning*, pages 4828–4837. PMLR.

Tian, S., Fu, Y., Ling, P., Wei, S., Liu, S., and Li, K. (2018). Wind power forecasting based on arima-lgarch model. In *2018 International Conference on Power System Technology (POWERCON)*, pages 1285–1289. IEEE.

Williams, C. (2010). Probabilistic Modelling and Reasoning Time Series Modelling: AR, MA, ARMA and All That.

Wilson, A. and Adams, R. (2013). Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*, pages 1067–1075. PMLR.

Wilson, A. and Ghahramani, Z. (2010). Copula Processes. In *Advances in Neural Information Processing Systems*, volume 23, page 9.

Wilson, A. and Nickisch, H. (2015). Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784. PMLR.

Wilson, A. G. and Ghahramani, Z. (2011). Generalised wishart processes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 736–744.

Wu, Y., Hernández-Lobato, J. M., and Ghahramani, Z. (2014). Gaussian process volatility model. *Advances in Neural Information Processing Systems*, 27:1044–1052.

Zhu, B. and Dunson, D. B. (2013). Locally adaptive bayes nonparametric regression via nested gaussian processes. *Journal of the American Statistical Association*, 108(504):1445–1456.

A. Tutorial

This section should serve as a useful reference on much of the more domain-specific language and methodology used throughout the paper.

A.1. Volatility

In the context of a time series S_t, we use *volatility*, denoted V_t, to refer to the standard deviation of the variability in price over some time period. In financial applications we consider stock prices on the daily time scale, and as is standard report volatility as *annualized volatility*, which corresponds to the volatility of a stock over the course of a year.

More specifically, we assume that the *log returns* in observations, $\log\left(\frac{S_{t+1}}{S_t}\right)$, are normally distributed with standard deviation V_t. In this paper we make the common assumption that the volatility itself is a time varying stochastic process, meaning we expect the magnitude of the daily returns to vary over time.

Figure 7 provides an example of the connection between price, log returns, and volatility. On the left we have a simulated set of price observations over one year, and in the center we have the associated log returns. Finally, on the right, we have the volatility path overlaid on the returns. We can see that where volatility is high we have larger returns (both positive and negative), and where volatility is low the returns tend to be small. Naturally if we wish to understand how the price will evolve in the future we need to also understand how volatility will evolve.

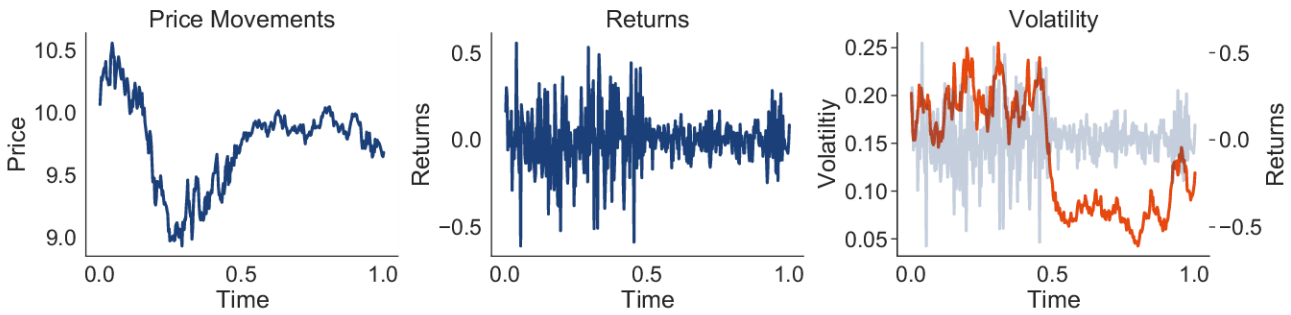


Figure 7. **Left:** Price movements over time. Movements tend to be larger up to time $t = 0.5$ **Center:** Returns over time, as calculated by S_{t+1}/S_t. Here, returns are clearly larger in the first half of the time series. **Right:** Volatility overlaid with returns for the same price. Volatility is clearly higher when the returns have a larger absolute magnitude, whether positive or negative.

B. Extended Methods

B.1. Moving Average Gaussian Processes

Figure 8 gives an example of how various moving averages (or Magpie prior means) appear given a series of price observations for a stock. On the left, the standard EMA formulation displays a clear lag effect, that is ameliorated by using either Double or Triple moving averages (DEMA and TEMA). On the right, we see how the DEMA moving average varies for different smoothing parameters k; for larger values the moving average is less sensitive to fluctuations in the data, but exhibits more bias, similarly smaller values of k produce moving averages that more closely match the data, but are susceptible to outliers.

B.2. Proofs from Derivations

Log-Volatility Kernel Function Recall the SDE governing movements in the log-volatility:

$$dv(t) = -\frac{\sigma^2}{2}dt + \sigma dZ(t). \tag{11}$$

We now derive the covariance function $\text{Cov}(v(t), v(t'))$, assuming without loss of generality, that $t < t'$. For ease of notation, and as the mean does not affect the covariance structure, let $\tilde{v}(t)$ be the same process as $v(t)$ with the mean trend removed.

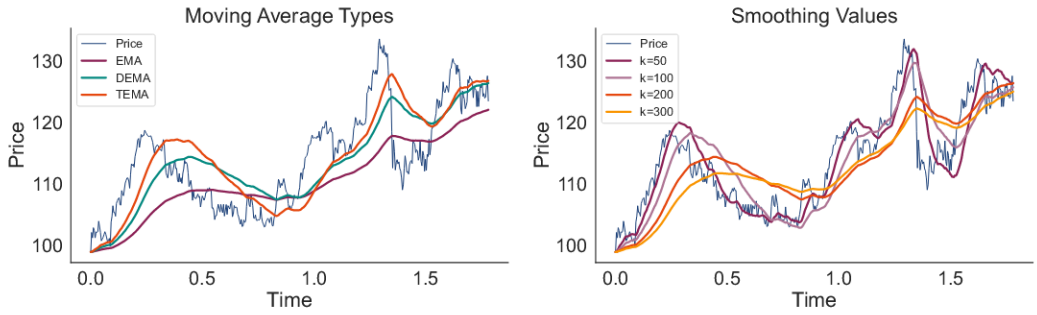


Figure 8. **Left:** a comparison of EMA, DEMA, and TEMA methods for producing moving averages for $k = 200$. Note that for a fixed value of k the DEMA and TEMA curves resolve a portion of the lag issue seen in the EMA curve. **Right:** DEMA curves for various values of k. Increasing k averages over more historical data.

Using independence of increments of the SDE we can determine the covariance as follows:

$$\begin{aligned}
\text{Cov}(v(t), v(t')) &= \text{Cov}(v(t) - E[v(t)], v(t') - E[v(t')]) \\
&= \text{Cov}(\tilde{v}(t), \tilde{v}(t')) \\
&= E[\tilde{v}(t)\tilde{v}(t')] - E[\tilde{v}(t)]E[\tilde{v}(t')] \\
&= E[\tilde{v}(t)\tilde{v}(t')] \\
&= E[\tilde{v}(t)(\tilde{v}(t') - \tilde{v}(t))] + E[\tilde{v}(t')^2] \\
&= E[\tilde{v}(t')^2] = t'\sigma^2 = \min\{t, t'\}\sigma^2.
\end{aligned}$$

So finally we have $\text{Cov}(v(t), v(t')) = K_v(t, t') = \min\{t, t'\}\sigma^2$.

Log-Price Kernel Function

$$ds(t) = \mu_s dt + V(t)dW(t) \tag{12}$$

The covariance function of $s(t)$ can be derived using the fact that the $s(t)$ diffusion has independent increments; first assume that $t < t'$ and that $\tilde{s}(t)$ is the same process as $s(t)$ with the mean trend removed. Therefore,

$$\begin{aligned}
\text{Cov}(s(t), s(t')) &= \text{Cov}(s(t) - E[s(t)], s(t') - E[s(t')]) \\
&= \text{Cov}(\tilde{s}(t), \tilde{s}(t')) \\
&= E[\tilde{s}(t)\tilde{s}(t')] - E[\tilde{s}(t)]E[\tilde{s}(t')] \\
&= E[\tilde{s}(t)\tilde{s}(t')] \\
&= E[\tilde{s}(t)(\tilde{s}(t') - \tilde{s}(t))] + E[\tilde{s}(t')^2] \\
&= E[\tilde{s}(t')^2],
\end{aligned}$$

now since $E[\tilde{s}(t)] = 0$, $E[\tilde{s}(t)^2] = \text{Var}(\tilde{s}(t)) = \text{Var}(s(t))$ which is just the integral of the variance of the diffusion in Equation (1), leaving us with

$$Cov(s(t), s(t')) = \int_0^{\min\{t,t'\}} V(t)^2 dt = K_s(t, t'; V(t)).$$

B.3. GPCV Training

GPCV Likelihood As described in the main text, we model the log returns, $w(t)$, at time t as independently distributed following the construction of Wilson and Ghahramani (2010). That is, $w(t) \sim \mathcal{N}(0, \gamma^2(t))$, where $\gamma(t)$ is the latent standard deviation. We choose $\gamma(t) = \exp\{f(t)\}$, which is equivalent to the parameterization used in Lázaro-Gredilla and Titsias (2011). The exponential parameterization has the nice property that we are also modelling the log prices in the SDE

formulation described in the rest of the paper, unlike Wilson and Ghahramani (2010)'s softplus transformation of the latent process. Wilson and Ghahramani (2010) also study the exponential parameterization for a few experiments.

We note that $\gamma(t)$ is a daily volatility and to convert to an annualized volatilty like in the rest of the paper, we need to rescale it by a factor of $1/\sqrt{t}$, so that $\hat{\gamma}(t) = \gamma(t)/\sqrt{t}$.

Inference Scheme Following Hensman et al. (2013; 2015), we want to compute the ELBO as

$$\log p(y) \geq \mathbb{E}_{q(f)}(\log p(y|f)) - \text{KL}(q(u)||p(u)), \tag{13}$$

where $p(y|f)$ is the GPCV volatility likelihood and $\text{KL}(q(u)||p(u))$ is the Kullback-Leibler divergence between the the variational distribution $q(u) = \mathcal{N}(m, S)$ and the prior $p(u)$. We need to optimize $q(u)$, our free form variational distribution and estimate $\mathbb{E}_{q(f)}(\log p(y|f))$ using Bayesian quadrature as in Hensman et al. (2015).

As T is generally pretty small, we set the inducing points, u, to be the training data points, e.g. $\{t_i\}_{i=1}^T$. We initialize the variational mean m to be the logarithm of the running standard deviation of the log returns, and the variational covariance to be $K_{uu}(K_{uu} + K_{uu}\Sigma_y K_{uu})^{-1}K_{uu}$ where Σ_y is the negative Hessian at the initial value of m.

Computational and memory costs then run at about $\mathcal{O}(T^3)$ time. In the future, we hope to use sliding windows for the inducing points, enabling mini-batching, reducing the cost to $\mathcal{O}(T^3_{\text{window}})$ time (Hensman et al., 2015). Finally, our inference scheme is simply a more flexible version of the fixed-form heteroscedastic scheme used in Lázaro-Gredilla and Titsias (2011), which we found to be too inflexible to fit rougher volatility paths well.

Multi-task Parameterization We follow the ICM-like model parameterization of Dai et al. (2017) by parameterizing $q(u) = \mathcal{N}(m, S_x \otimes S_T)$ and assume that $p(u) = \mathcal{N}(\mu(u), K_{uu} \otimes K_{TT})$. Then we need to compute $q(f)$ which can be done for single-task models as $q(f) = \mathcal{N}(K_{fu}K_{uu}^{-1}m, K_{ff} + K_{fu}K_{uu}^{-1}(S - K_{uu})K_{uu}^{-1}K_{uf})$. In the multi-task setting, this is algebraically written as:

$$\begin{aligned}
q(f) &= \mathcal{N}((K_{fu} \otimes K_{TT})(K_{uu} \otimes K_{TT})^{-1}m, \\
&\quad (K_{ff} \otimes K_{TT}) + (K_{fu} \otimes K_{TT})(K_{uu} \otimes K_{TT})^{-1}(S_x \otimes S_t - (K_{uu} \otimes K_{TT}))(K_{uu} \otimes K_{TT})^{-1}(K_{fu} \otimes K_{TT})^\top) \\
&= \mathcal{N}\left((K_{fu}K_{uu}^{-1} \otimes I)m, (K_{ff} \otimes K_{TT}) + (K_{fu}K_{uu}^{-1} \otimes I)(S_x \otimes S_t - (K_{uu} \otimes K_{TT}))(K_{fu}K_{uu}^{-1} \otimes I)^\top\right) \\
&= \mathcal{N}\left((K_{fu}K_{uu}^{-1} \otimes I)m, (K_{ff} - K_{fu}K_{uu}^{-1}K_{uf} \otimes K_{TT}) + (K_{fu}K_{uu}^{-1}S_xK_{uu}^{-1}K_{uf} \otimes S_t)\right)
\end{aligned} \tag{14}$$

Note that the variational mean term is a batch matrix vector multiplication, while the variational covariance form is a sum of two Kronecker products. Together we can sample from the posterior distribution in $\mathcal{O}(T^3 + P^3)$ time by using Kronecker identities as described in Rakitsch et al. (2013).

In the multi-task setting, we also initialize the variational covariance to be the average initial covariance across tasks and the variational intertask covariance to be the covariance of m across tasks. The intertask covariance is a $P \times P$ matrix parameterized as rank one plus diagonal; we regularize it with a LKJ prior with $\eta = 5.0$ (Lewandowski et al., 2009).

Additionally, we exploit Kronecker identities to efficiently compute the KL divergence in the variational distribution so that training stays at $\mathcal{O}(T^3 + P^3)$ time by broadly following the approach of Dai et al. (2017).

B.4. Model Training

All models were trained in GPyTorch (Gardner et al., 2018) and PyTorch (Paszke et al., 2019) on either a single 24GB GPU or a single 12GB GPU; the multi-task wind experiment used a 48GB Titan RTX GPU. Training time was negligible, with models typically taking less than 1 minute to train. For training, we use 500 steps of Adam with learning rate 0.1 and optimize through the log marginal likelihood.

Multitask GPs We use the ICM model of Bonilla et al. (2007). Like in the GPCV setting, we use a rank one plus diagonal intertask covariance, regularized with a LKJ prior (Lewandowski et al., 2009). By structure exploitation, these models cost $\mathcal{O}(P^3 + T^3)$ for fitting and $\mathcal{O}(P^3 + T^3)$ for posterior sampling when using Matheron's rule (Maddox et al., 2021a).

Data Space GPs We use a standard Gaussian likelihood for these responses on the log transformed data and optimize both the scale of the volatility as well as the noise term, initializing the noise to be 10^{-4}. As these models reduce to a standard exact GP conditional on volatility, computational and memory costs then run at $\mathcal{O}(T^3)$ time.

C. Experimental Details

C.1. Details from Section 4.1

We source daily closing prices for stocks in the Nasdaq 100 for 2 years prior to January 2022. Volt models are trained according to the outline in Section 3.2, and standard GPs are implemented and trained via GPyTorch and BoTorch (Gardner et al., 2018; Balandat et al., 2020). The LSTM model is implemented with 2 hidden layers each with 128 units and takes the form

$$f(s_t, s_{t-1}, s_{t-2}, s_{t-3}, s_{t-4}) = \{\hat{\mu}_{t+1}\hat{\sigma}_{t+1}\}$$

where $\hat{\mu}_{t+1}$ is the predicted mean at time $t+1$, and $\hat{\sigma}_{t+1}$ is the predicted standard deviation at time $t+1$.

For each stock in our universe we select 25 cutoff times at which we generate forecasts, using the preceding 400 observations as training data. At each cutoff time we forecast the log closing price 100 days into the future, and compute the calibration and negative log likelihood of the forecasts 75 to 100 days out. We specifically focus on longer horizon forecasts, as it is generally a harder task for which out of the box methods are ill-suited.

	Stock Prices	Wind Speeds	FX
Volt + Magpie	5.88 ± 0.02	4.28 ± 0.16	-1.69 ± 0.02
Volt + Con.	$\mathbf{4.69 \pm 0.03}$	$\mathbf{3.38 \pm 0.05}$	-1.60 ± 0.02
Matérn + Magpie	9.80 ± 0.27	12.13 ± 0.81	4.23 ± 0.30
Matérn + Con.	7.74 ± 0.21	18.03 ± 1.90	-0.36 ± 0.04
SM + Magpie	147.84 ± 1.84	110.07 ± 7.81	562.67 ± 15.71
SM + Con.	80.43 ± 0.57	70.14 ± 5.03	356.55 ± 11.98
LSTM	49.95 ± 0.59	45.13 ± 1.82	10.66 ± 0.44
Volt-VHGP + Con.	4.76 ± 3.05	5.75 ± 0.44	-1.58 ± 0.03
Volt-VHGP + Magpie	6.97 ± 1.24	5.91 ± 0.34	-1.66 ± 0.02
GPCV	5.45 ± 1.51	4.89 ± 0.04	$\mathbf{-1.79 \pm 0.02}$

Table 2. Negative log likelihoods (NLLs) per test point for the methods compared on both the stock forecasting and wind speed tasks, averaged of tens of thousands of forecasts. While there is a slight improvement in NLL from using a constant mean, the inclusion of Magpie is central to achieving high calibration.

C.2. Details from Section 4.2

We source data from Diamond et al. (2013) for the 2021 calendar year. Wind measurements are taken at 15 minute intervals for all 154 stations in the observation network. In order to treat the observed wind speed as log-normally distributed we add 1 to each observation (to shift the 0 m/s observations to a value of 1), and then model the log of the resulting time series.

Figure 9 compares the performance of Volt alone and Volt with Magpie mean functions with various smoothing parameters. Magpie means aid in calibration, although the effect is less pronounced as we see with stock forecasting in Figure 3.

A key distinction between wind speed forecasting and stock price forecasting is that wind speeds tend to revert to a consistent level, whereas stock prices may increase by thousands then stabilize at a new level. For this reason we explore the use of mean reversion in our rollout forecasts. To add mean reversion to the rollouts we simply adjust the posterior mean of the GP towards the mean of the training data by a factor of θ. That is, rather than sampling from the GP posterior $s_{t^*} \sim \mathcal{N}(\mu^*_{f|\mathcal{D}}|\Sigma^*_{f|\mathcal{D}})$ we sample from $s_{t^*} \sim \mathcal{N}(\mu^*_{f|\mathcal{D}} - \theta(\mu^*_{f|\mathcal{D}} - \frac{1}{N}\sum_i s_i)|\Sigma^*_{f|\mathcal{D}})$.

In this mean reversion setting, θ controls the speed at which rollouts tend to revert towards the mean. At $\theta = 0$ we are in the standard GP prediction case, at $\theta = 1$, we only ever sample from a distribution centered around the mean of the training observations. Figure 10 provides a comparison of the calibration under differing levels of mean reversion for Volt. The standard Volt rollouts are in general well calibrated for this problem, but we see that just a small amount of mean reversion can increase the overall calibration notably.

C.3. Details from Section 5

In Figure 11, we construct a multi-task SABR volatility model with correlations given by the farthest right panel and volatility processes given as the blue lines in the left three panels. We then use our multi-task GPCV model to estimate and predict the true volatilities for each task in the left three panels, while also estimating the true relationships between

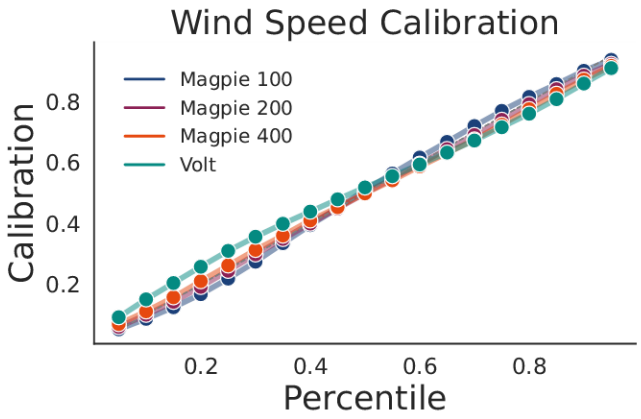


Figure 9. A comparison of Volt with a constant mean, and Volt with various Magpie means in terms of calibration in wind forecasts. While Volt with a constant mean is well calibrated, it is aided by the inclusion of a Magpie mean with large smoothing parameter.

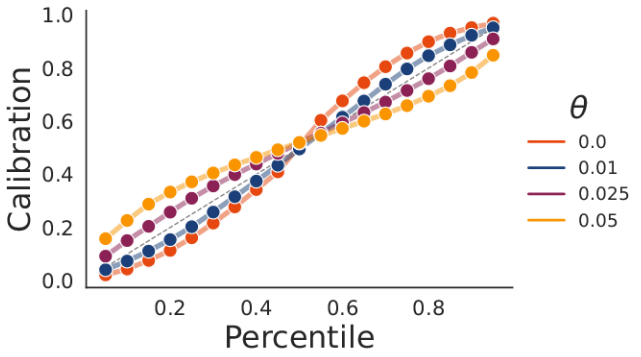


Figure 10. Calibration of different mean reversion θ values across stock prices.

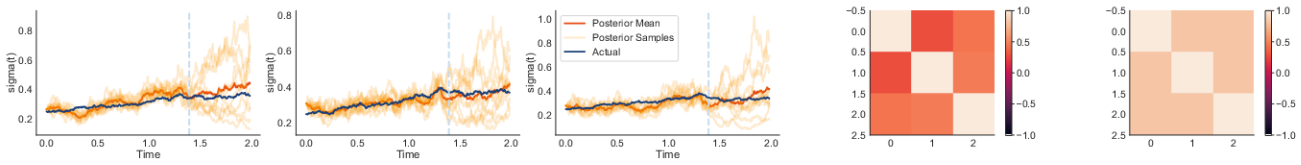


Figure 11. **Left three panels:** Predicted correlated volatility models. **Fourth panel:** Estimated correlation matrix between volatility models. **Fifth panel:** True volatility correlation. Multitask GPCVs are able to both predict volatility while also estimating the true correlation between volatility.

each volatility. The estimated relationships are shown in the fourth panel from left, which is pleasingly similar to the true correlation shown at far right.

For Figure 6left and Figures 12, 14a, 14b, we fit stocks comprising of five different exchange traded fund SPDRs[1] collected over 5 years of daily data from 09/2016 to 09/2021. These SPDRS are XLE, XLF, XLK, XLRE, XLY; each had six stocks in it except for XLF which had 30. We fit on 300 days and evaluated 100 days into the future, with 5 rolling testing sets for each prediction.

For Figure 6 center, we used the same training data except used only five stocks from the XLE SPDR and five from the XLF SPDR.

For Figure 6 right and Figures 13, 15 we fit about 100 different wind stations (depending on amount of missing data) at 5 minute intervals across 2021 with 25 indepdendent rolling splits. We fit on 252 increments and tested on 100 increments. Here, on the multi-task ones, we used a larger RTX 8000 GPU.

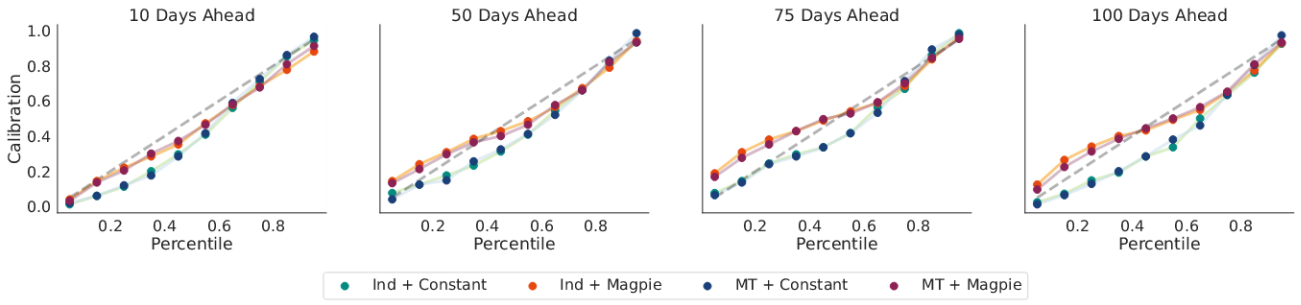


Figure 12. Calibration of multi-task Volt and independent models across time step lookaheads for 5 different SPDRS.

[1] https://en.wikipedia.org/wiki/SPDR

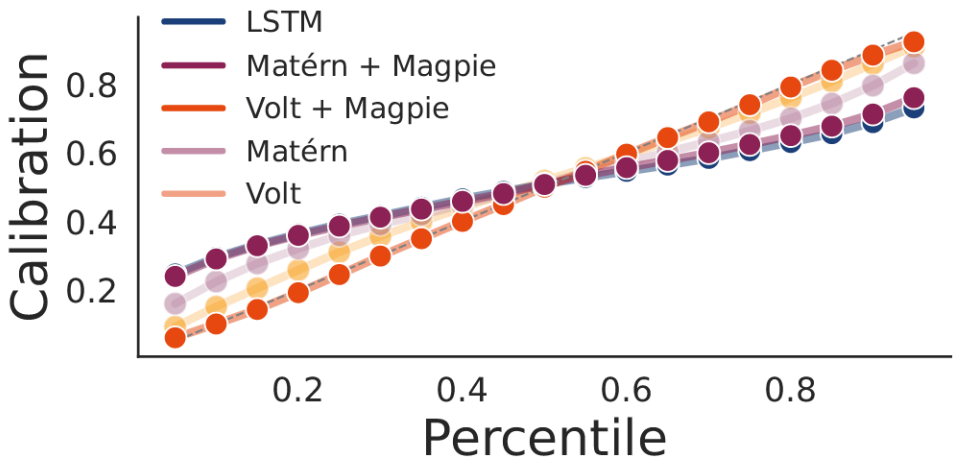
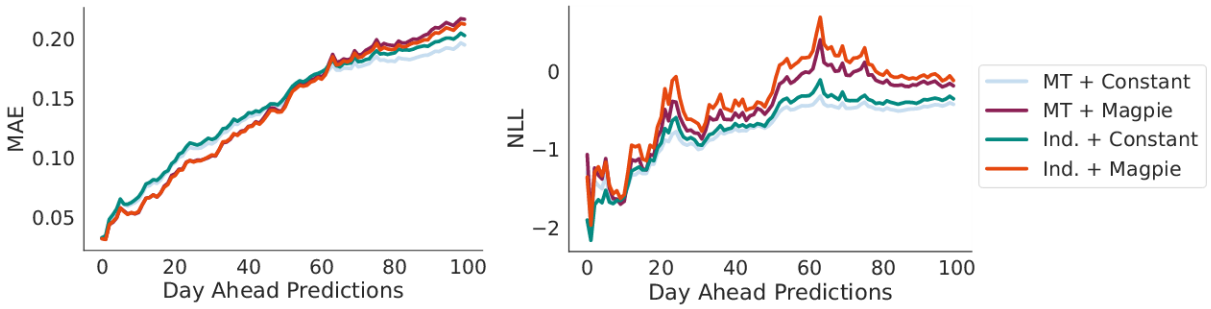


Figure 13. Calibration of multi-task Volt and independent models across time step lookaheads for the wind forecasting datasets.



(a) MAE

(b) NLL

Figure 14. **Left panel:** Mean absolute error of rollouts. **Right panel:** Negative log likelihood of rollouts.

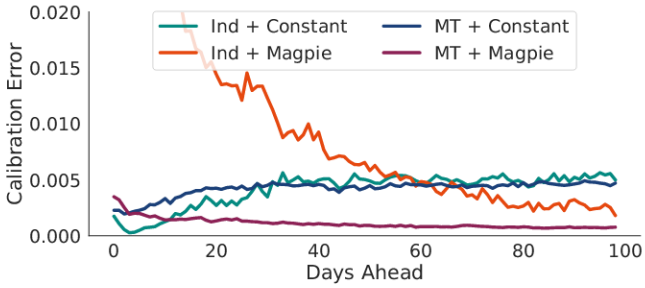


Figure 15. Calibration error of the models across different time step lookaheads for the wind forecasting task.