Challenges in Achieving Explainability & Control with Supply Chain Forecasts

Rishab Guha
Phillip Jang
Andrea Tambalotti
rishguha@amazon.com
philjang@amazon.com
atamba@amazon.com
Amazon, SCOT
New York, NY, USA

ABSTRACT

[Industry Challenge Submission] AI and Deep Learning methods have revolutionized many forecasting applications but have not achieved widespread adoption in industry for aggregate forecasting. This paper challenges the AI research community by identifying three critical capabilities that current AI approaches lack: (1) multivariate consistency at scale, (2) explainable and controllable long-run assumptions, and (3) flexible incorporation of forward-looking external inputs. We describe a Bayesian state-space framework that is used in production to address these requirements at a major e-commerce retailer, where our forecasts influence billions of dollars in spending decisions. By detailing how traditional time series methods solve these challenges today, we identify concrete opportunities for AI researchers to develop hybrid approaches that combine the accuracy advantages of modern AI with the explainability and control benefits of traditional methods.

ACM Reference Format:
Rishab Guha, Phillip Jang, and Andrea Tambalotti. 2025. Challenges in Achieving Explainability & Control with Supply Chain Forecasts. In *Proceedings of the 1st Workshop on "AI for Supply Chain: Today and Future" @ 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

AI methods have seen wide adoption across a variety of supply chain forecasting applications, driven by improved accuracy compared to traditional forecasting methodologies. However, adoption of AI forecasting techniques has been relatively limited in the aggregate forecasting space, e.g., forecasts of total daily volume across an entire supply chain. Aggregate forecasting (forecasting total demand across an entire system rather than individual products or locations) poses unique technical challenges that are currently better addressed by traditional methodologies than by AI. This paper describes these challenges from the perspective of a team that produces aggregate demand forecasts used for supply chain planning at a large e-commerce retailer. Our forecasts, which are currently produced using traditional time series methods, are used by supply chain planners to determine the optimal amount of capacity to build, inventory to buy, and labor to hire, accounting for billions of dollars of expenditure across multiple countries. We have tried to adopt AI / deep learning forecasting methods in the past, but have found that while current frontier methodologies are highly accurate, they cannot achieve the explainability and controllability that we require in practice. For instance, when a DL model predicts an unexpected demand spike in six months, supply chain planners need to understand the underlying drivers, and have the tools to adjust the forecast if necessary, before making multi-billion dollar capacity decisions.

We identify three capabilities, beyond accuracy, required for a forecasting model to achieve business adoption: (1) explainable multivariate coherency, (2) controllability of forecast assumptions, and (3) the ability to flexibly condition a forecast on forward-looking external inputs. We have found that traditional time-series methods are currently better at achieving these capabilities than AI / ML methods. Section 2 presents a short example, in the context of macroeconomic forecasting with public data. Section 3 expands on these requirements in the supply-chain forecasting use-case, and reviews related literature. Section 4 makes the challenges precise by describing our current production framework, which uses Bayesian state-space methods to achieve the required capabilities. Section 5 concludes by identifying concrete opportunities for AI / ML research to build towards "hybrid" forecasting approaches that maintain the explainability / control of traditional methods while leveraging the improved accuracy and scalability of AI.

2 AN EMPIRICAL EXAMPLE

To fix ideas, and show the challenges we face running aggregate forecasting models in practice, we consider the problem of forecasting macroeconomic inflation, which is similar to many aggregate forecasting problems we face in a supply chain context. We use publicly-available macroeconomic data to estimate a joint forecast of monthly inflation across 4 segments: Food / Beverage, Housing, Apparel, and Core CPI. We compare our team's aggregate supply-chain forecasting model, a Bayesian state-space factor model we call Gotham, with Prophet [9], a time-series model commonly used by practitioners, as well as three recently-released transformer-based

KDD '25, August 3, 2025, Toronto, ON, Canada.

ACM ISBN 979-8-4007-1454-2/25/08.
https://doi.org/10.1145/XXXXXX.XXXXXX

foundational time-series forecasting models: Chronos [2], Moirai-MoE [11] [8], and TimesFM [4]. Backtests (described in Appendix A) show that the models are roughly comparable to each other in accuracy, with no model dominant across all series.

However, accuracy is not the primary obstacle AI forecasting models face in achieving adoption for industrial supply chain forecasting applications. To show the importance of explainability and controllability, we examine forecasts made in Q1 2023, at the height of the post-pandemic surge in macroeconomic inflation. Figure 1 shows forecasts for Core CPI year-on-year growth across models made in April 2023. The "traditional" time-series models, Gotham and Prophet, show a linear continuation of recent trends, which is easily explicable given the data, though overly conservative in hindsight. The AI models all predict complex non-linear fluctuations in year-on-year growth (e.g., the sudden uptick in forecasted growth by Moirai at a six month horizon), which are not obvious in the historical data. The AI models do not provide any natural functionality to explain these non-linearities (example, explanations might look like "there is a seasonal increase in YoY growth rates every June," or "the YoY growth rate increases in October '23 because the model views the price level in October '22 as affected by a one-off shock"). In our experience in forecasting aggregate series, human supply chain decision-makers will not accept a forecast with unexplained fluctuations of the type seen in Fig. 1, as these forecasts inform capital investments and strategic planning that require clear justification to leadership. Bayesian time-series models provide the forecaster with the ability to explain a forecast as a linear combination of interpretable components, and to control these components through conditioning or adjusting priors; the AI models we have considered do not support similar functionality.

All of the forecasts in Figure 1 underestimate the speed of the mean-reversion in CPI that was actually observed in 2023. However, professional macroeconomic forecasters were able to correctly capture the pace of this reversion based on their qualitative knowledge of expected Federal Reserve policy. The average respondent in the Q1 2023 Survey of Professional Forecasters (SPF), conducted by the Federal Reserve , expected that in 12 months the core CPI inflation rate would be around 3%. We can use state-space methods to condition our model's forecast on this forward-looking estimate. Figure 2 shows this conditional forecast. We see that the model is able to statistically integrate this information, and better-predict the actual path of inflation. Additionally, because our model is a multivariate factor model, it is able to coherently update its forecast for Apparel CPI, even though the SPF only gathers expectations for Core CPI. The Apparel forecast shows that conditioning on qualitative external information can involve trading off short-run forecast performance (where pure data-driven extrapolation tends to perform best) against the long-run. We have not identified any ways of replicating this functionality with AI models.

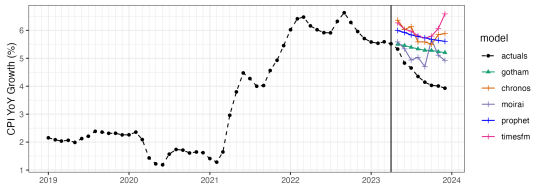

Figure 1: Core CPI forecasts across models, made in April 2023.

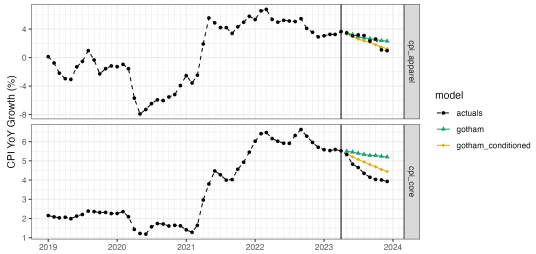

Figure 2: Comparison of CPI forecasts made in Q1 2023 between unconditional Gotham and Gotham conditioned on a prior for long-run CPI growth based on the Survey of Professional Forecast consensus estimate as of Q1 2023.

3 OUR BUSINESS PROBLEM, AND TECHNICAL CHALLENGES

Our team produces aggregate demand forecasts used for planning at a large e-commerce platform. Let $y_{i,t}$ be the realized demand for a segment i on calendar date t. In our application, i segments demand by product group (e.g., electronics vs. toilet paper), measurement variable (e.g., dollars of revenue ordered vs. number of packages shipped), and delivery channel (e.g., first-party vs. third-party). We forecast a vector that contains daily demand values across multiple segments, for a forecast horizon of multiple years. We face three major technical challenges, similar to those identified in the inflation-forecasting example, that are difficult to address with current AI/ML methods: (1) the need for multivariate consistency (2) a requirement that we be able to explain and control major assumptions around long-run forecast behavior and (3) a requirement that we be able to condition our forecast on forward-looking external inputs.

3.1 Multivariate Consistency

Our forecasts are consumed by multiple agents within the supply chain planning process. For example, financial planning teams care about next quarter's revenue forecast, transportation planning teams care about average daily demand on a short horizon, and capacity-planning teams care about peak daily demand multiple years in the future. To prevent costly misalignments—e.g., budgeting for more capacity than we have labor to staff—we need to ensure that forecasts are jointly coherent across related i. For example, forecasts for total revenue and total shipped packages should reflect a mutually consistent expectation for the underlying growth of the business. This rules out series-by-series univariate forecasting methodologies such as Prophet [9] or Chronos [2], which do not allow for any way to enforce coherence across forecasts. Some AI foundation models, such as Moirai [11], and TimesFM [4], do support native multivariate forecasting, but they do not provide any way to enforce restrictions on forecast covariances (e.g, we

expect revenue growth and units growth to in general be positively correlated), or explain common drivers.

Forecasts must also be flexibly extensible to new levels of disaggregation. For example, given a forecast of demand for electronics we may need to quickly produce forecasts for electronics manufactured in China vs. those manufactured elsewhere. This requirement rules out application-specific neural networks, because it is often not feasible to increase the dimensionality of the model and retrain the network each time we would like to consider a new layer of disaggregation. Sequence-based models (e.g., [11]) can address this concern by "patching" multiple time series together into a single series, but these approaches have difficulty imposing exact restrictions (e.g., that forecasts for electronics-made-in-China and electronics-not-made-in-China must collectively sum to the forecast for electronics), which is essential for decision-making consistency. A key concern is to ensure that high-level assumptions (e.g., about overall trends in the demand for electronics) are sensibly preserved for the disaggregated forecasts.

3.2 Assumption Explainability and Control

Because our aggregate demand forecasts inform irreversible long-term investments, they must be understood and approved by human decision makers. This means that we are required to be able to explain the key economic and business assumptions of our model. For example, we must be able to answer questions like "what is your long-run assumption for the third-party share of total sales?" or "do you assume that recent growth in average sales price is permanent or transitory?" Answering these questions requires producing a mapping between qualitative statements and the quantitative forecast produced by our model. There has been substantial academic research on "explainable AI," but most of this has focused on assessing feature-importance in a cross-sectional classification or prediction setting. We are unaware of any work that can provide human-understandable explanations of time-series forecasts.

We are often required to translate feedback on our assumptions into changes to our forecast. This requires models that are not just explainable, but controllable. In many cases the consumers of our forecast have information about future trends or business initiatives, and want to make sure that our forecast is consistent with that information. In other cases, we are must perform scenario analyses, to test the forecast's sensitivity is to different sets of assumptions. We have found that this requirement prevents us from directly applying most deep-learning forecasting approaches (e.g., [10], [6]), which take a "black-box" forecasting approach that makes it difficult to directly control model assumptions. We note that some academic work suggests that incorporating human judgment into model-based decisions may not systematically improve performance (e.g., [1]). However, in an industry context, there are often specific cases in which a forecast must incorporate judgment in order to be credible.

3.3 External Conditioning

Finally, we need to ensure that our demand forecasts are consistent with investment and operational plans made by other teams across the supply chain. This requires a model that can be conditioned on these plans. For example, given a set of potential paths for investment in improving delivery speeds, we need to be able to produce forecasts of demand conditional on each path. This set of forecasts is an input into the iterative decision process that ends with a single aligned plan, and a demand forecast conditional on that plan. We follow a similar process to produce forecasts for scenarios conditional on macroeconomic outcomes—e.g., a recession, or an increase in inflation—that are produced by "satellite" macroeconomic forecasting models.

Achieving this requirement requires a modeling system that is capable of conditioning forecasts on forward-looking estimates of "drivers" that are related to our core forecasted variables. However, this conditioning must be optional: we need to be able to produce a forecast that does not rely on forward-looking input plans, for cases where input plans are stale or don't exist yet. Additionally, we usually do not have a comprehensive historical record of point-in-time plans that can be used for training. This rules out including forward-looking realizations of drivers as exogenous predictors in a machine-learning model.

4 ACHIEVING BUSINESS REQUIREMENTS WITH TRADITIONAL TIME-SERIES METHODS

To make the challenges we face precise, this section describes a modeling framework that meets the business requirements described above using traditional Bayesian state-space methods. This framework substantially extends the model described in [7], with a particular focus on improving forecast scalability and controllability using state-space modeling tools. We have used this framework in production since 2020. We hope to inspire AI / ML research that can maintain these capabilities, while further improving accuracy and scalability.

The basic framework

Let $y_{i,t}$ be a log-level variable i (e.g., the log of dollars of revenue from electronics), observed at a monthly frequency. Let $\mathcal{I}$ collect a set of variables that we forecast jointly. We decompose the year-on-year log-growth, $y_{i,t} - y_{i,t-12}$ into three parts: a "trend" component, $\bar{g}$, which controls the average rate of demand growth, a "cyclical" component, $\tilde{g}$, which controls transitory fluctuations around this trend, and level-shock components $\psi_{i,t}, \psi_{i,t-12}$ which control idiosyncratic shocks and time-varying seasonality:

$$y_{i,t} - y_{i,t-12} = \bar{g}_{i,t} + \tilde{g}_{i,t} + (\psi_{i,t} - \psi_{i,t-12}) \tag{1}$$

The trend component, $\bar{g}$, follows a factor structure, in which each variable i has factor loadings $\lambda_{i,k}$, and each unobserved factor $f_{k,t}$ follows a random walk.

$$\bar{g}_{i,t} = \sum_k \lambda_{i,k} \cdot f_{k,t} \tag{2}$$

$$f_{k,t} = f_{k,t-1} + \eta_t \tag{3}$$

The factor structure imposes long-term coherence across related variables i through dependence on common factors k. In most of our applications we manually specify $\lambda_{i,k} \in \{-1, 0, 1\}$ (where 1 and -1 indicate the variable is affected by the factor, and 0 indicates it is not), to impose judgment about how variables are related. For example, we might identify a specific f_k as representing secular trends in the

demand for clothing, by setting $\lambda_{i,k} = 1$ for all i related to clothing (potentially including i that correspond to macroeconomic demand for clothing) and $\lambda_{i,k} = 0$ otherwise.

The cyclical component, $\tilde{g}$, evolves according to a vector autoregression (VAR) specification, in which each variable is modeled as an autoregressive function of the lags of other variables.

$$\tilde{g}_{i,t} = \left(\sum_{i} \sum_{l=1}^{p} \beta_{i,l} \cdot \tilde{g}_{i,t-l} \right) + \epsilon_{i,t} \tag{4}$$

This allows our model to flexibly capture lead/lag dynamics such as the tendency for improvements in supply chain metrics to improve demand in subsequent months.

The level-shock component, ψ, accounts for time-varying seasonality and idiosyncratic shocks that affect the observed level of demand but not underlying growth rates. For example during leap years February will have an additional day, mechanically increasing sales by $\sim 3\%$. ψ can also account for temporary shocks that impact demand in a specific month but should not affect the forward-looking forecast e.g., a demand drop due to an internet outage.

We estimate and forecast the unobserved states in (1) using Bayesian state-space methods to sample draws from the posterior of all unobserved states from their distribution conditional on the input data. The details of the estimation equation for each state, and the estimation algorithm we use, are included in Appendix B.

4.1 Achieving multivariate consistency

We use the factor-based trend component $\bar{g}$ to enforce multivariate consistency across all $i \in \mathcal{I}$. For example, assume that $y_{1,t}$ measures aggregate national GDP. If we impose the loadings $\lambda_{1,1} = 1$ and $\lambda_{1,i} = 0$ for all $i > 1$, then the factor $f_{1,t}$ will represent trends in GDP growth. For any other variable j (e.g., total sales for our e-commerce platform), we can impose the assumption that j depends on economic growth by specifying (or estimating) a positive value for $\lambda_{j,1}$. All variables with $\lambda_{j,1} > 0$ will share a common exposure to this macroeconomic growth factor, and all forecasts will be aligned with a single forecast for GDP growth. This helps enforce coherency in revisions: following a positive forecast error in $y_{1,t}$, the model will revise its estimate for $f_{1,t}$ upwards, which will simultaneously increase the forecast for all related j. This consistency improves model explainability and is difficult to achieve with modern multivariate AI approaches, in which the response of the forecast of j to a positive surprise in i is usually a complex non-linear function that is difficult to constrain.

Though equation (1) can model multiple variables, the number of parameters required to estimate the VAR component $\tilde{g}$ grows quadratically with the number of variables, creating computational challenges when scaling to thousands of series. In order to scalably expand the number of series we forecast, while maintaining consistency, we employ a hierarchical approach. Let $\{x_j\}$ with $j \in \mathcal{J}$ be a set of variables which segment a "top-level" variable y_i: for example, if y_i is worldwide log electronics revenue, x_j may be log electronics revenue for products manufactured in country j. We use (1) to model y_i, and use an analogous approach to recursively model $x_{j,t} - y_{i,t}$:

$$(x_{j,t} - y_{i,t}) - (x_{j,t-12} - y_{i,t-12}) = \bar{g}^x_{j,t} + \tilde{g}^x_{j,t} + (\psi^x_{j,t} - \psi^x_{j,t-12}) \tag{5}$$

Substituting (1) shows that this two-step formulation is equivalent to the forecasting equation

$$x_{j,t} - x_{j,t-12} = \bar{g}_{i,t} + \bar{g}^x_{j,t} + \tilde{g}_{i,t} + \tilde{g}^x_{j,t} + (\psi_{i,t} - \psi_{i,t-12}) + (\psi^x_{j,t} - \psi^x_{j,t-12}) \tag{6}$$

which shows that conditioning on $y_{i,t}$ enforces consistency between the factor structure driving $\mathcal{I}$ and forecasts of variables in $\mathcal{J}$. This lets us express the forecast of x_j as coherent with all the forecasts for all variables in $\mathcal{I}$, through its dependence on y_i. However, the top-down nature of this extension implies that the forecast of x_j can never influence the forecast of y_i. We believe that AI technologies could help automatically identify cases in which we would like to relax this restriction (e.g., when shocks to $x_{j,t}$ are fast-growing enough to be material to $y_{i,t}$ in the future, but not large enough to show up in the direct forecast of $y_{i,t}$), and estimate optimal forecast adjustments.

4.2 Achieving assumption-level explainability / control

A key advantage of our framework is that it provides an intuitive decomposition of growth rates into a permanent component $\bar{g}_{i,t}$ and a transitory component $\tilde{g}$. We use the state-space setting of our model to expose these states as observable variables. This lets us use the smoothed estimates of states at each point in time to explain the model's forecasts by attributing fluctuations in growth to permanent vs. transitory components.

We can use state-space conditioning to modify this attribution in response to human feedback. For example, assume that we see growth for series i increase rapidly from a steady-state of 10% to 50% at time t. The model will usually attribute most of the acceleration to $\tilde{g}_{i,t}$ (because sudden large changes in growth rates typically don't persist in historical data) and project that growth rates will mean-revert back to a long-run average of 10%. However, in some cases we may know that the increase in growth was driven by a policy change, or a new initiative, which is expected to be persistent. We can view this as applying a prior that the permanent trend component of growth at t should be close to 50%. We add a synthetic observation to the state space of the form $\bar{g}_{i,t} = \log(1.50) + \mathcal{N}(0, \sigma^2_{i,t})$ where σ^2 controls the informativeness of our prior. The model will flexibly balance this prior against the trends estimated in the data to arrive at a balanced posterior. The state-space smoother will update its forecast of the underlying factors driving $\bar{g}_{i,t}$ based on this prior, and so will consistently adjust the forecast for all $i \in \mathcal{I}$.

We also use priors to flexibly impose long-run restrictions or dynamics on our forecasts. For example, we may believe that the new policy will boost growth for 1-2 years, but that on a 5-year horizon growth rates will converge back to 10%. In this case we could add another synthetic observation, of the form $\bar{g}_{i,t+60} = \log(1.10) + \mathcal{N}(0, \sigma^2_{i,t+60})$ to impose this prior—this is similar to the approach we took with SPF expectations in the inflation-forecasting example. We have found that this approach works well in practice, but requires substantial manual work and subject-matter expertise. We believe there is an opportunity for research on how AI / LLM methods can automatically transform qualitative feedback from non-technical stakeholders into quantitative priors.

4.3 Achieving external conditioning

We can also use synthetic observations within the state space to flexibly condition our state space on forecasts produced by other models. To fix ideas, assume that y_t is a vector of demand variables (e.g., revenue across multiple product types) and that z_t is a "driver" variable, e.g., the average number of days it takes for a customer to receive their order. In general, demand will increase as z_t improves (e.g., shorter delivery times typically lead to higher conversion rates), and the model will estimate this relationship. However z_t is a complex function of supply chain investments, labor dynamics, network topology, etc., and is likely to be modeled by dedicated teams to determine an investment plan that corresponds to a projected trajectory (or "glidepath") $\tilde{z}_t$. Our challenge is to produce a "baseline" forecast for z_t, and then update this baseline to produce a forecast consistent with the glidepath.

We address this challenge with state-space conditioning. Once we include z_t in $\vec{I}$, we can generate synthetic observations that set $z_t = \tilde{z}_t$ for all available future dates. We can compare the forecast of y_t conditional on $\tilde{z}_t$ to the unconditional forecast of y_t in order to estimate the fraction of demand impact attributable to the glidepath. We also use the forecast of z_t from the unconditional run as an "implied forecast," to evaluate the reasonability of the provided glidepath.

5 OPPORTUNITIES FOR AI RESEARCH

Based on our experience producing aggregate forecasts in industry, we believe there are several opportunities to use AI methods to improve forecast accuracy, and enable "hands-off-the-wheel" scalable forecasting. However, we believe that it is unlikely AI methods will be able to entirely replace traditional approaches, given the difficulty of addressing explainability and controllability requirements discussed above. We believe that AI methods can complement current methodologies, as part of a "hybrid" forecasting approach that integrates AI components within a traditional state-space framework.

First, we believe there is an opportunity to use unsupervised AI methods to discover the factor structure that relates variables to each other. In order to achieve multivariate consistency, we need to reduce the dimensionality of the trend-forecasting component of our model, from an arbitrarily-large number of forecasted variables to a small number of dominant factors. In order to make our factors controllable, we set factor loadings $\lambda_{i,k} \in \{-1, 0, 1\}$ using human judgment. For example, we may define a "clothing" factor by setting loadings to 1 for all time series related to demand for clothing. We have found that automated dimension-reduction methods (e.g., principal-components analysis) produce factors that do not have an explainable interpretation. We also need to use human judgment to define the hierarchy we use for the recursive hierarchical forecasting approach described in 4.1. There are opportunities to use LLM / AI methods to automate this process, by using semantic / contextual information about what each series index i actually represents to generate hierarchies and factor loadings. For example, an agentic LLM system could use qualitative business documents to learn that clothing, electronics, and food, are the most important product segments in a given country, identify factors for each segment, and assign each modeled time series a loading on these factors.

LLM-based methods can also help explain and control forecast assumptions. While our statistical framework allows the modeler to construct sophisticated mappings between qualitative business assumptions and quantities within the state space, analyzing and manipulating these mappings requires substantial subject-matter expertise. An SME needs know how to translate business questions (e.g., "what is your current assumption for the long-run inflation rate") into queries about specific states within the state space. Running scenarios to evaluate the impact of perturbing assumptions (e.g., "what would our forecast look like if inflation goes back to 6%?") requires configuring the priors that encode these assumptions in the model. We believe there is an opportunity for LLM-driven agentic systems to learn how to translate natural-language questions from non-technical business users into API calls that can be used to manipulate the statistical forecasting system. This would reduce dependence on statisticians/modelers and improve hands-off-the-wheel automation.

Finally, we see an opportunity to use AI / deep-learning methods to account for seasonality with a more sophisticated approach than traditional statistical methods such as modeling year-on-year growth rates, or using Fourier basis terms. Many of the series we forecast have similar seasonal patterns, which can be determined from qualitative context: for example, we might expect computers, stationary, and clothing, to all exhibit similar back-to-school seasonality patterns. We believe that AI methods can help us determine these seasonal commonalities at scale, by collapsing qualitative text-based context about a time series into an embedding vector that can be used to determine seasonal similarity between time series. This embedding vector can then be used as input into a state-space model.

In conclusion, we see many opportunities for AI methods to improve the automation and customization of production supply chain forecasts. This will require close integration with legacy statistical forecasting systems, in order to maintain forecast explainability, coherency, and control. We believe that AI can help scale sophisticated forecasting models while reducing reliance on subject-matter experts, and look forward to seeing the research community address this challenge.

REFERENCES

[1] Nikhil Agarwal, Alex Moehring, Pranav Rajpurkar, and Tobias Salz. 2023. *Combining Human Expertise with Artificial Intelligence: Experimental Evidence from Radiology.* Working Paper 31422. National Bureau of Economic Research. https://doi.org/10.3386/w31422

[2] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. 2024. Chronos: Learning the Language of Time Series. arXiv:2403.07815 [cs.LG] https://arxiv.org/abs/2403.07815

[3] Andrea Carriero, Davide Pettenuzzo, and Shubhranshu Shekhar. 2025. Macroeconomic Forecasting with Large Language Models. arXiv:2407.00890 [econ.EM] https://arxiv.org/abs/2407.00890

[4] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. arXiv:2310.10688 [cs.CL] https://arxiv.org/abs/2310.10688

[5] J. Durbin and S. J. Koopman. 2012. *Time Series Analysis by State Space Methods* (2 ed.). Oxford University Press.

[6] Carson Eisenach, Yagna Patel, and Dhruv Madeka. 2022. MQTransformer: Multi-Horizon Forecasts with Context Dependent and Feedback-Aware Attention. arXiv:2009.14799 [cs.LG] https://arxiv.org/abs/2009.14799

[7] Rishab Guha, Eric Aldrich, Bertram Ieong, Shanshan Li, Domenico Giannone, and Juan Huerta. 2021. Towards Robust, Scalable and Interpretable Time Series Forecasting using Bayesian Vector Auto-Regression. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139).* https://roseyu.com/time-series-workshop/submissions/2021/TSW-ICML2021_paper_23.pdf

[8] Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. 2024. Moirai-MoE: Empowering Time Series Foundation Models with Sparse Mixture of Experts. arXiv:2410.10469 [cs.LG] https://arxiv.org/abs/2410.10469

[9] Sean J. Taylor and Benjamin Letham. 2017. *Forecasting at scale.* Technical Report e3190v2. PeerJ Preprints. https://doi.org/10.7287/peerj.preprints.3190v2

[10] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. 2018. A Multi-Horizon Quantile Recurrent Forecaster. arXiv:1711.11053 [stat.ML] https://arxiv.org/abs/1711.11053

[11] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. Unified Training of Universal Time Series Forecasting Transformers. arXiv:2402.02592 [cs.LG] https://arxiv.org/abs/2402.02592

	Gotham	Prophet	Chronos	Moirai-MoE	TimesFM
Core	0.310	0.319	0.395	0.315	0.379
Apparel	1.265	1.920	1.396	1.323	1.265
Food & Beverage	0.723	0.911	0.772	0.491	0.540
Housing	0.372	0.360	0.463	0.340	0.467

Table 1: Mean absolute percentage error across forecasts of monthly CPI components at a forecast horizon of ≤ 6 months, from 2012-2024.

A BACKTEST DETAILS

To support the empirical example in Section 2, we ran a horse-race back-test in which we compared the modeling framework described in Section 4 (named Gotham) with Prophet [9], a Bayesian time-series model commonly used by practitioners, as well as three recently-released transformer-based foundational time-series forecasting models: Chronos [2], Moirai-MoE [8] [11], and TimesFM [4].

We focus on forecasting non-seasonally-adjusted Consumer Price Index (CPI) values across four segments: Apparel[1], Food & Beverage[2], Housing[3], and Core CPI[4]. We retrieved data from FRED, which is maintained by the Federal Reserve Bank of St. Louis.

For all considered models, we use base / default settings, with no application-specific fine-tuning. Gotham was developed for supply-chain forecasting using aggregate data from our e-commerce platform, and is not specifically tuned or designed for inflation forecasting. Because the inflation data is publicly-available macroeconomic data, it is likely included in the training corpus of the foundational models, so we view back-test results for those models as a lower bound for true out-of-sample accuracy.

We run rolling back-tests for CPI forecasts from January 2011 through October 2024, starting on the first month of each quarter. We evaluate accuracy series-by-series pooled across all forecast horizons ≤ 6 months. We use data starting January 1990 as context for all models. We run multivariate forecasts for Gotham, Moirai-Moe, and TimesFM, and univariate forecasts for Chronos and Prophet (which do not support multivariate forecasting).

Table 1 shows our results. We see that Gotham and Moirai-MoE are the best-performing models, but that no model dominates the others across all forecasted series. This is not surprising—as [3] show, foundational time-series models perform well at low-frequency macroeconomic forecasting tasks, but do not consistently outperform traditional time-series models such as Gotham or Prophet.

[1] U.S. Bureau of Labor Statistics, Consumer Price Index for All Urban Consumers: Apparel in U.S. City Average [CPIAPPNS], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/CPIAPPNS, May 25, 2025.

[2] U.S. Bureau of Labor Statistics, Consumer Price Index for All Urban Consumers: Food and Beverages in U.S. City Average [CPIFABNS], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/CPIFABNS, May 25, 2025.

[3] U.S. Bureau of Labor Statistics, Consumer Price Index for All Urban Consumers: Housing in U.S. City Average [CPIHOSNS], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/CPIHOSNS, May 25, 2025.

[4] U.S. Bureau of Labor Statistics, Consumer Price Index for All Urban Consumers: All Items Less Food and Energy in U.S. City Average [CPILFENS], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/CPILFENS, May 25, 2025.

B MODEL ESTIMATION DETAILS

Define growth rates $g_t = y_t - y_{t-12} \in \mathbb{R}^n$ and shock component $\psi_t = S_t \xi$ where $S_t \in \mathbb{R}^{n \times J}$ are user-specified covariates such as seasonal dummies, jump indicators, and step functions whose contributions $\xi \in \mathbb{R}^J$ are unknown.

Let $\mathcal{S}_t = S_t - S_{t-12}$. We can rewrite the estimation equations in state space form. In Durbin and Koopman [5] notation,

$$\begin{bmatrix} g_t \\ \bar{g}_t \end{bmatrix} = \underbrace{\left[\begin{array}{c|ccccc|c} \Lambda & I_n & \mathbf{0} & \cdots & \mathbf{0} & \mathcal{S}_t \\ \Lambda & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{array}\right]}_{Z_t} \underbrace{\begin{bmatrix} f_t \\ \hline \tilde{g}_t \\ \vdots \\ \tilde{g}_{t-p+1} \\ \hline \xi \end{bmatrix}}_{\alpha_t} + \mathcal{N}_{2n}\left(\mathbf{0}, H_t = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma_t^{\bar{g}} \end{bmatrix}\right), \tag{7}$$

$$\underbrace{\begin{bmatrix} f_{t+1} \\ \hline \tilde{g}_{t+1} \\ \vdots \\ \tilde{g}_{t-p+3} \\ \tilde{g}_{t-p+2} \\ \hline \xi \end{bmatrix}}_{\alpha_{t+1}} = \underbrace{\begin{bmatrix} I_m & & & & \\ & B_1 & \cdots & B_{p-1} & B_p \\ & I_n & \cdots & \mathbf{0} & \mathbf{0} \\ & \vdots & \ddots & \vdots & \vdots \\ & \mathbf{0} & \cdots & I_n & \mathbf{0} \\ & & & & I_J \end{bmatrix}}_{T_t} \underbrace{\begin{bmatrix} f_t \\ \hline \tilde{g}_t \\ \vdots \\ \tilde{g}_{t-p+2} \\ \tilde{g}_{t-p+1} \\ \hline \xi \end{bmatrix}}_{\alpha_t} + \mathcal{N}_{m+pn+J}\left(\mathbf{0}, Q_t = \begin{bmatrix} \Sigma^f & & & & \\ & \Sigma^c & \mathbf{0} & \cdots & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ & \vdots & \vdots & \ddots & \vdots \\ & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ & & & & \mathbf{0} \end{bmatrix}\right), \tag{8}$$

$$\alpha_0 \sim \mathcal{N}_{m+pn+J}\left(a_0 = \left[\begin{array}{c|c|c} \bar{f}_0 & g_{-1:-p} & \mathbf{0} \end{array}\right], \quad P_0 = \begin{bmatrix} \Sigma^{f_0} & & \\ & \Sigma^{c_\infty} & \\ & & \Sigma^{\xi} \end{bmatrix}\right). \tag{9}$$

The primary controls involve setting priors on the unobserved trend $\bar{g}_t$. By augmenting the observation equation 7 with a second set of rows for the latent trend $\bar{g}_t$, we gain the ability to set priors with synthetic noisy observations. If we have priors $\bar{g}_{i,t} \sim \mathcal{N}(\mu_{i,t}, \sigma^2_{i,t})$ for the following collection of targeted series and dates $\mathcal{T} = \{(i_k, t_k)\}$, then we set

$$\begin{cases} \bar{g}_{i,t} = \mu_{i,t} \text{ and } \Sigma_t^{\bar{g}}(i,i) = \sigma^2_{i,t} & \text{if } (i,t) \in \mathcal{T}, \\ \bar{g}_{i,t} = \text{nan and } \Sigma_t^{\bar{g}}(i,i) = 0 & \text{otherwise.} \end{cases}$$

More generally, we can apply priors to arbitrary linear combinations of the state vector $C\alpha_t$ with a similar augmentation to the observation equation. We can also asymptotically recover the unobserved case with uninformative priors by taking $\sigma^2_{i,t} \to \infty$.

The model is estimated using standard Bayesian state space methods, involving alternating Gibbs steps between sampling full conditionals of parameters $\theta|\alpha, y$, and sampling states from the Kalman simulation smoother $\alpha|\theta, y$.