

# Using Transformers and Reinforcement Learning as Narrative Filters in Macroeconomics\*

Vegard H. Larsen<sup>†</sup>      Leif Anders Thorsrud<sup>‡</sup>

This version: February 2, 2026

## Abstract

Building on recent advances in Natural Language Processing and modeling of sequences, we study how a multimodal Transformer-based deep learning architecture can be used for measurement and structural narrative attribution in macroeconomics. The framework we propose combines (news) text and (macroeconomic) time series information using cross-attention mechanisms, easily incorporates differences in data frequencies and reporting delays, and can be used together with Reinforcement Learning to produce structurally coherent summaries of high-frequency news flows. Applied and tested on both simulated and real-world data out-of-sample, the results we obtain are encouraging.

**JEL-codes:** C45, C55, E32, E37

**Keywords:** Multimodal Transformer, Structural decomposition, Text analytics, Macroeconomic nowcasting

---

\*Comments from Hilde C. Bjørnland, Peter Winker, Rune Sørensen, and seminar participants at BI Norwegian Business School, SSB, and the 5th Dolomiti Macro Meetings improved the quality of this paper considerably. This work is part of the research activities at the Centre for Applied Macroeconomics and Commodity Prices (CAMP) at the BI Norwegian Business School.

<sup>†</sup>Centre for Applied Macroeconomics and Commodity Prices, BI Norwegian Business School. Email: `vegard.h.larsen@bi.no`

<sup>‡</sup>Corresponding author. Centre for Applied Macroeconomics and Commodity Prices, BI Norwegian Business School, Nydalsveien 37, 0484 Oslo, Norway. Email: `leif.a.thorsrud@bi.no`

# 1 Introduction

*“It would be nice to point to recognizable events, of the type that is reported by newspapers, as the source of economic fluctuations, rather than to residuals from some equations.”* [Cochrane \(1994, p. 298\)](#)

A fundamental problem in macroeconomics is identifying the underlying drivers of economic fluctuations and measuring their broader dynamic implications. Empirical research typically addresses the question using a particular statistical model, with the vector autoregressions, introduced by [Sims \(1980\)](#), and later extensions incorporating factor structures or larger systems ([Stock and Watson, 1989](#); [Bernanke et al., 2005](#); [Bańbura et al., 2010](#)), having become standard frameworks. Within this tradition, however, the fundamental driving forces of economic fluctuations are residuals from the equations defining the assumed statistical model. Although successful, this has motivated developments of methods, starting already with [Friedman and Schwartz \(1963\)](#), that use text and historical sources to add narrative content to the statistical decompositions ([Romer and Romer, 1989](#); [Ramey, 2011](#); [Stock and Watson, 2012](#); [Antolín-Díaz and Rubio-Ramírez, 2018](#)).

This paper takes up the same theme from a different perspective, leveraging Transformer-based neural architectures ([Vaswani et al., 2017](#)). These types of architectures underlie recent advances in Natural Language Processing (NLP) and Large Language Models (LLMs), but are designed for sequence data and increasingly used to analyze time series dynamics as well as multimodal learning using both text and time series variables ([Wen et al., 2023](#); [Xu et al., 2023](#)).

In particular, we explore how a Transformer-based architecture can be combined with a user-specific economic view, or structural prior, to build a model called the *Narrater*, which simultaneously processes text and time series data via joint contextualized representations of the two data modalities. In line with the description above, the purpose of this model is to provide a reliable estimate of economic fluctuations and their underlying structural drivers, coupled with narrative attribution of the associated texts. We then show how the model can be used out-of-sample together with Reinforcement Learning (RL) techniques to produce structurally coherent summaries of a high-frequency flow of textual information. Accordingly, more than training a deep neural network solely to optimize predictive performance, we explore how RL and a multimodal Transformer architecture can be used as a structural narrative filter.

To fix ideas, the problem we study is based on a common and broadly defined hypothesis where text, here economic news, provides timely information about the most important events affecting macroeconomic fluctuations. Specifically, we assume that both the texts and economic time series are (partly) driven by the same underlying causes

and events, and that their joint contextualized representations are informative about the class and sentiment of the associated texts, as well as the historical shock decomposition commonly used to understand economic fluctuations.<sup>1</sup>

These arguments do not imply that the Transformer architecture alone can identify the underlying sources of aggregate fluctuations. Identifying causal relationships from observational, aggregated data requires imposing (strong) assumptions. Thus, what we explore is how a Transformer-based architecture can filter multimodal information in line with a particular structural view. Fundamentally, this structural view must be encoded in the training data, which we simulate. Conceptually, this approach is related to the recent literature on prior-fitted networks (Müller et al., 2021; Hollmann et al., 2022; Nagler, 2023), which studies how neural networks can be trained on simulated data to perform amortized inference under an implicitly specified prior over data-generating processes. In contrast to this literature, which typically considers broad task distributions and emphasizes predictive optimality across heterogeneous environments, we fix a single structural model and use it as a disciplined prior that governs both the joint distribution of observables and the associated structural objects of interest. To mitigate in-sample circularity and assess usefulness, we strictly separate training, validation, and a quasi real-time out-of-sample evaluation on non-synthetic data. This prevents re-use of the same simulated sequences and allows testing whether the supervised narrative filter remains informative when confronted with unseen news and time series data. Importantly, this separation neither provides model-free identification nor validates the structural view; it only ensures that the reported performance is not an artifact of evaluating on training data.

In our benchmark configuration we consider the problem of estimating the business cycle and structural demand, supply, and noise components. The practical usefulness of this configuration is easy to motivate. From, e.g., an inflation-targeting central bank perspective having accurate business cycle estimates is important, but knowing that the business cycle is growing because of higher demand, rather than supply, might be equally important because the policy rate response will easily be very different in these two cases.

Still, the framework we propose is general and not tied to any specific structural implementation. It can, for example, be implemented under assumptions ranging from a single common business cycle shock (e.g., Angeletos et al. (2020)) to much richer descriptions of

---

<sup>1</sup>See, e.g., Veldkamp (2011), Shiller (2017) and Chahrour et al. (2021) for information choice mechanisms where the news-media-economics linkages are described more explicitly within either a behavioral or rational expectations framework. The proposed framework can in principle also be used for narrative attribution of shocks. This would however require construction of training data where the mapping between economic variables and the texts is such that the latter contains unpredictable news. We find it easier to construct training data under the less restrictive assumption of predictability, as captured by the historical shock decomposition.

macroeconomic fluctuations (e.g., [Smets and Wouters \(2007\)](#)). To demonstrate the framework’s flexibility, we also test it in a setting where the goal is to capture global oil market dynamics as identified in [Baumeister and Hamilton \(2019\)](#), and describe two additional, less-structural, model configurations. One identifies news topics relevant to business cycle fluctuations (as first proposed by [Thorsrud \(2018\)](#) and [Larsen and Thorsrud \(2019\)](#)). The other focuses on the model’s ability to handle mixed-frequency data.

The design of the deep neural network we propose builds on the attention mechanism ([Vaswani et al., 2017](#)), and has an encoder-decoder structure using Transformer layers. This facilitates processing multimodal data and generative modeling of the decoder output. To be precise, the model we design combines two encoders, one for time series of text data and one for multivariate economic time series. At a given time point, the language encoder inputs a text sequence, which is processed via a Bidirectional Encoder Representations from Transformers (BERT) architecture ([Devlin et al., 2018](#)), and outputs an embedding representation of the whole sequence. The BERT weights are shared across time points and potential time dependencies across the embedding representations are modeled using a single Transformer block. Similarly, the multivariate time series encoder consist of one Transformer block which outputs embedding representations of the multivariate time series data. The text and time series embeddings are then fused with the embedding representation in the decoder, e.g., the business cycle, using cross-attention layers. The end product is a network that simultaneously processes text and time series data and performs multi-task learning ([Caruana, 1997](#)). I.e., the model autoregressively outputs not only a structural decomposition via the decoder, but also performs narrative attribution by outputting the associated sentiment and class of the underlying news flow.

While the *Narrater* is large compared to traditional econometric models it is deliberately designed to be very parsimonious relative to typical LLMs used for text generation. This, in combination with using pre-trained language encoders and domain specific fine-tuning, facilitates efficient training and cost-effective usage. Besides capturing potentially complex and long-range dependencies within a sequence, the Transformer architecture enables these layers to capture non-linear features of the underlying data via non-linear activation functions and multi-head attention with softmax weighting. Moreover, the usage of the cross-attention layers permits modeling sequences measured at potentially different frequencies and with different lengths - features that are prominent when modeling macroeconomic data.

Based on similar cost and efficiency arguments the model is trained under the assumption of one text per time period. When confronted with multiple news items within the same period, we show that the model can be combined with RL to produce narrative and structurally coherent summaries. In this setup, policy functions are estimated based on

reward functions favoring low predictive loss, while optionally incorporating terms that encourage narrative priors and coherence. A simple special case is greedy selection, which requires no additional estimation or learning.

Empirical results from both simulation and real-world data demonstrate the model’s value. On synthetic validation data, the model achieves near-perfect classification accuracy and outperforms simple baseline methods and LLMs on classification and sentiment prediction. Permutation tests confirm that both modalities contribute to predictive performance, while manual audits on the simulated training data suggest strong alignment between model-generated labels and human judgment.

Ablation studies reveal that multi-task learning improves predictive performance, removing either encoder significantly reduces performance, while experimenting with settings determining the model’s embedding dimensions suggests that larger embeddings encode more relevant information. Fine-tuning the BERT encoder alongside the other model parameters is also essential: performance degrades by up to 80% if the pre-trained parameters are kept fixed.

We test the *Narrater* on real-world Norwegian macroeconomic data and news from Dagens Næringsliv, Norway’s leading business newspaper. In this quasi real-time experiment the model’s business cycle estimates closely match those from traditional methods and the extracted sentiment is highly correlated with the cycle. Moreover, although the model and RL routine has to process thousands of news articles every quarter, the news summaries and classified news flow accord well with conventional (ex-post) human beliefs and narratives, while the model’s nowcasting performance for GDP growth is comparable or slightly superior to that of a standard Dynamic Factor Model. In terms of describing global oil market fluctuations the model produces reasonable narrative attribution and out-of-sample structural decompositions that are well aligned with those identified in-sample in [Baumeister and Hamilton \(2019\)](#). The model’s forecasting performance for the real price of oil is also on-par with a random walk, which is regarded as a competitive benchmark in this setting. Furthermore, we demonstrate that the model performs well in identifying user-defined news topics and efficiently processes mixed-frequency information. Thus, the proposed framework appears to be easily adaptable and capable of addressing diverse user needs.

The remainder of the paper is structured as follows. Section 2 reviews related literature and situates our contribution. Section 3 presents the *Narrater* model, while Section 4 details the simulation-based training data and estimation procedure and Section 5 reports validation results. In Section 6 we describe how to use the model together with RL to generate extractive summaries and present application results as well as model extensions. Section 7 concludes.

## 2 Related literature and contribution

This study intersects with research in both economics and computer science. To our knowledge, it is the first to propose a multimodal Transformer-based architecture for economic analysis, and connects to three main strands of the economics literature.

First, we build on the growing body of work using NLP methods in macroeconomics (Baker et al., 2016; Hansen and McMahon, 2016; Hansen et al., 2018; Larsen et al., 2021; Bybee et al., 2024). More recent studies incorporate Transformer-based models such as BERT for text classification and feature extraction (Liu et al., 2021; Gorodnichenko et al., 2023; Dell, 2024; Gambacorta et al., 2024), with Ash et al. (2026) summarizing the transformative impact of large LLMs on text analysis in economics. Unlike these approaches, which typically treat text as a standalone feature set, we adopt a fully multimodal framework in which text and time series data share a joint representation, estimated simultaneously through a Transformer architecture. This design enables supervised structural decomposition of economic fluctuations alongside sentiment and class predictions in a multi-task setup, rather than treating text as a generic control. Together with RL the model also produces real-time text summaries that tie numerical movements to concrete news items.

Second, our work relates to the macroeconomic literature on signal extraction, business cycle measurement, and nowcasting. This literature often relies on large datasets (Stock and Watson, 1989; Evans, 2005; Giannone et al., 2008; Banbura et al., 2011), handles mixed-frequency inputs (Mariano and Murasawa, 2003; Aruoba et al., 2009), and increasingly incorporates non-linear or text-based features (Marcellino et al., 2016; Thorsrud, 2018; Burri and Kaufmann, 2020; Shapiro et al., 2022). While our model shares these features, and natively handles mixed frequencies and reporting lags through the cross-attention mechanism, it differs from traditional statistical filtering methods by enabling supervised structural decomposition via deep learning.

Third, we connect to the literature on narrative identification, which dates back to Friedman and Schwartz (1963) and uses historical documents to interpret and identify macroeconomic shocks (Romer and Romer, 1989; Ramey, 2011; Stock and Watson, 2012). Although we do not identify shocks per se, but rather propose a framework for narrative attribution consistent with a pre-specified structural view, our contribution builds on this tradition by automating the classification of narrative content and integrating it into a unified estimation framework.

We do not want to argue, however, that the proposed model necessarily is better than existing technologies used in economics. Rather, we view our contribution as an exploration of how recent advances in language modeling and deep learning can provide a scalable framework for interpreting multimodal information through a consistent economic

lens. Going forward, we envision numerous improvements and extensions to this type of model architecture. Beyond our applications, the framework is potentially useful in any setting where high-frequency text (or other unstructured streams) co-evolves with numeric time series.

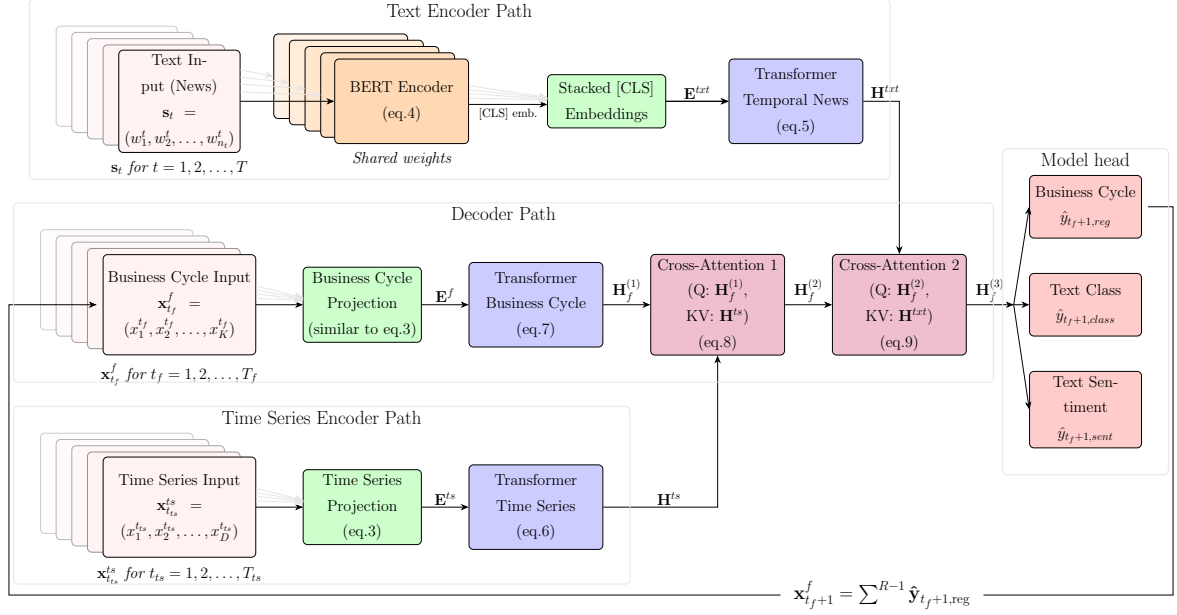
At a more general level our study speaks to large fields, foremost within computer science, studying multimodal neural network architectures, LLMs for time series modeling, and so-called foundational models (FM). In terms of the former, [Xu et al. \(2023\)](#) survey multimodal learning with Transformers, highlighting their intrinsic advantages and scalability in modeling different modalities and tasks with fewer modality specific architectural assumptions than comparable methods. Early summation or concatenation of the embedding representations of the different data modalities is a common design principle. Here, to allow for potentially mixed-frequency data and different sequence lengths across modalities, we instead follow a so-called multi-stream design principle using cross attention. This is similar to in, e.g., [Lu et al. \(2019\)](#) and [Tsai et al. \(2019\)](#), but extend this line of research from vision and language modeling to time series and economics.

A related literature uses LLM architectures to explicitly build and train foundational models for time series analysis ([Das et al., 2024](#); [Woo et al., 2024](#); [Jin et al., 2024](#)). However, for predicting macroeconomic time series the recent study by [Carriero et al. \(2024\)](#) suggest that these types of FMs do no better than existing econometric models. Similarly, studies such as [Gambacorta et al. \(2024\)](#) and [Dell \(2024\)](#) demonstrate that specific fine-tuning of, e.g., BERT, delivers performance on par, or better, than LLMs in a range of (economic) domain specific classification tasks, e.g., classifying the sentiment of monetary policy speeches. This motivates a domain specific approach also for narrating economic fluctuations, which we pursue here in a multimodal setting.

Finally, we speak to a large literature in NLP studying text summarization ([Zhang et al., 2025](#)). Within this tradition exploring how multimodal input, and output, can improve summarization quality is a growing area of interest extending traditional text summarization to include other forms of data such as images, tables, and audio. Although the benchmark *Narrater* is not a summarization model per se, we demonstrate how its multimodal design features in combination with RL facilitate narrative attribution and summarization.

### 3 The Narrater

The *Narrater* is built using standard tools from the deep learning literature and recent advances in modeling sequences using Transformer-based architectures ([Vaswani et al., 2017](#)).



**Figure 1.** The *Narrater* model architecture in the benchmark case of business cycle extraction.

Figure 1 shows a high-level overview of the proposed model architecture in the benchmark case of business cycle extraction. Two encoders - text and time series - map each modality to embeddings of a shared hidden dimension. The time-series decoder integrates these embeddings and autoregressively produces both the model output and the next-period decoder input. The text encoder uses a BERT language model. BERT-like models, introduced by Devlin et al. (2018), are pre-trained on large-scale corpora and can be fine-tuned for a broad range of downstream tasks with minimal architectural modification, often achieving state-of-the-art or near-state-of-the-art results. From the BERT block we extract the sequence embeddings which are then forwarded to an independent Transformer layer to capture the assumption that text and time series are driven by common shocks and to some extent share dynamics - so text in period  $t$  may predict texts in later periods. For the same reason, independent Transformer layers are included in the time series encoder and decoder, and cross-attention blocks fuse the modalities into shared contextualized embeddings.

In sum, these choices aim to produce a network that processes text and time-series data simultaneously and autoregressively outputs estimates of economic fluctuations and their underlying structural drivers, coupled with a narrative decomposition of the textual flow into its sentiment and class. Below we describe each part of the model in greater detail, with Section 3.5 highlighting particular modeling choices and benchmark specifications. Readers unfamiliar with Transformers and BERT might want to consult Appendix D before proceeding.



### 3.1 The text and time series encoders

Let a collection of  $T$  text sequences be denoted by  $\mathbf{s}_t = (w_1^{(t)}, w_2^{(t)}, \dots, w_{n_t}^{(t)})$  for  $t = 1, \dots, T$ , where  $w_i^{(t)}$  represents the  $i$ -th token of the  $t$ -th sequence and  $n_t$  is the length of that sequence. Then write the BERT architecture as a function that performs padding, masking, inserts special tokens, and transforms each input sequence  $\mathbf{s}_t$  into a contextualized representation:

$$\mathbf{e}_t^{txt} = f_{\theta_1}^{BERT}(\mathbf{s}_t) \in \mathbb{R}^{N \times d}, \quad (1)$$

where  $d$  is the embedding dimension and  $\theta_1$  encompass all the embedding matrices (token and position) and weights of the Transformer encoder layers. Here we let  $\mathbf{e}_t^{txt}$  represent the sequence embedding associated with the  $[CLS]$  token. This is a common way of summarizing sequence content when using this type of architecture and serves as an efficient dimension reduction step in our application.<sup>2</sup> Next, let  $\mathbf{E}^{txt} \in \mathbb{R}^{T \times d}$  define the  $\mathbf{e}_t^{txt}$  vectors stacked for all  $t \in \{1, \dots, T\}$ . In this way, the same BERT structure and parameters ( $\theta_1$ ) are reused to model multiple sequences, ensuring that each sequence is encoded into a high-level representation space that is consistent across all  $T$  sequences.

Now, to allow for temporal relationships in the embedded text sequences we feed  $\mathbf{E}^{txt}$  through a separate Transformer block:

$$\mathbf{H}^{txt} = f_{\theta_2}^{TF}(\mathbf{E}^{txt}) \in \mathbb{R}^{T \times d}, \quad (2)$$

where the  $f_{\theta_2}^{TF}$  function is a standard Transformer structure. Similarly, let  $\mathbf{x}_{t_{ts}}^{ts} \in \mathbb{R}^D$  denote a  $D$ -dimensional vector of macroeconomic variables at time  $t_{ts} \in \{1, \dots, T_{ts}\}$ . Each vector is then first projected into a  $d$ -dimensional embedding space via:

$$\mathbf{e}_{t_{ts}}^{ts} = \mathbf{W}_{ts} \mathbf{x}_{t_{ts}}^{ts} + \mathbf{b}_{ts}, \quad (3)$$

where  $\mathbf{W}_{ts} \in \mathbb{R}^{d \times D}$  and  $\mathbf{b}_{ts} \in \mathbb{R}^d$  are learned parameters. The resulting embeddings are stacked into  $\mathbf{E}^{ts} \in \mathbb{R}^{T_{ts} \times d}$  and serve as the input sequence to a Transformer, with positional encodings added in the usual manner:

$$\mathbf{H}^{ts} = f_{\theta_3}^{TF}(\mathbf{E}^{ts}) \in \mathbb{R}^{T_{ts} \times d}. \quad (4)$$

Accordingly, equations (1), (2), and (4) encode the input  $\mathbf{s}_t$  for  $t = 1, \dots, T$  and  $\mathbf{x}_{t_{ts}}^{ts}$  for  $t_{ts} = 1, \dots, T_{ts}$  into the embedding matrices  $\mathbf{H}^{txt}$  and  $\mathbf{H}^{ts}$ .

---

<sup>2</sup>One could alternatively have carried forward embeddings for all the tokens in the sequence. However, even for moderate time series lengths this would have had severe memory implications for subsequent Transformer-layers and is thus not something we have experimented with.

### 3.2 The decoder

To decode this information we build on the sequence-to-sequence type of architecture used for language translation in Vaswani et al. (2017), but extend and adjust this structure to process text and time series data.

For this purpose, let  $\mathbf{x}_{t_f}^f \in \mathbb{R}^K$  be a time series measure(s), e.g., the business cycle, at time  $t_f$ , with  $K \ll D$  in general and  $K = 1$  here. Next, these  $K$ -dimensional observations are transformed into an embedding of dimension  $d$  following the same projections as in (3) for  $t_f = 1, \dots, T_f$ , and further processed as:

$$\mathbf{H}_{(1)}^f = f_{\theta_4}^{TF}(\mathbf{E}^f) \in \mathbb{R}^{T_f \times d}. \quad (5)$$

To accommodate the multimodal information in  $\mathbf{H}^{txt}$  and  $\mathbf{H}^{ts}$  we then apply two subsequent cross-attention layers. These are similar to regular self-attention (see (21) in Appendix D.1), but with the difference that the queries are defined as  $\mathbf{Q}_{(\cdot)}^f = \mathbf{H}_{(\cdot)}^f \mathbf{W}_{Q_{(\cdot)}^f}$ , and the keys and values as  $\mathbf{K}^{txt} = \mathbf{H}^{txt} \mathbf{W}_{K^{txt}}$  and  $\mathbf{V}^{txt} = \mathbf{H}^{txt} \mathbf{W}_{V^{txt}}$ , or  $\mathbf{K}^{ts} = \mathbf{H}^{ts} \mathbf{W}_{K^{ts}}$  and  $\mathbf{V}^{ts} = \mathbf{H}^{ts} \mathbf{W}_{V^{ts}}$ . In particular, we first construct:

$$\mathbf{H}_{(2)}^f = f_{\theta_5}^{CA}(\mathbf{H}_{(1)}^f, \mathbf{H}^{ts}) \in \mathbb{R}^{T_f \times d}, \quad (6)$$

and then:

$$\mathbf{H}_{(3)}^f = f_{\theta_6}^{CA}(\mathbf{H}_{(2)}^f, \mathbf{H}^{txt}) \in \mathbb{R}^{T_f \times d}, \quad (7)$$

where  $f_{\theta}^{CA}$  denotes the cross-attention mechanism and both functions are combined with a residual connection and layer normalization as in the standard attention layer.

Intuitively, since the queries for the cross-attention mechanism are derived from one sequence (the decoder representation), while the keys and values come from another sequence (the encoder representations), the cross-attention layers allow one sequence to attend to another sequence’s latent representations, enabling the attending sequence to select and integrate the most relevant features from the other sequence. Figure 3 and the associated discussion in Section 3.5 provide a concrete example of this mechanism and how it is particular useful in settings with potentially mixed-frequency data or sequences of different length.

### 3.3 The model-heads

$\mathbf{H}_{(3)}^f$  contains the decoded and multimodal information provided by the original inputs  $\mathbf{s}_t$ ,  $\mathbf{x}_{t_{ts}}^{ts}$ , and  $\mathbf{x}_{t_f}^f$ . In our benchmark model specification we use  $\mathbf{H}_{(3)}^f$  in combination with three separate model heads for inferring the state of the business cycle, and for classifying the type and sentiment of the text sequences. For a particular time index, the regression output is given by:

$$\hat{\mathbf{y}}_{t_f+1,reg} = \mathbf{W}_r \mathbf{h}_{(3)}^f + \mathbf{b}_r \in \mathbb{R}^R, \quad (8)$$

where  $\mathbf{h}_{(3)}^f$  is the transpose of the  $t_f$ -th row vector in  $\mathbf{H}_{(3)}^f$  and the elements of  $\hat{\mathbf{y}}_{t_f+1,reg}$  are discussed in subsequent sections.

Likewise, for text classification and sentiment estimation:

$$\hat{\mathbf{y}}_{t_f+1,class} = \text{softmax}(\mathbf{W}_c \mathbf{h}_{(3)}^f + \mathbf{b}_c) \in \mathbb{R}^C, \quad (9)$$

and:

$$\hat{y}_{t_f+1,sent} = \tanh(\mathbf{W}_s \mathbf{h}_{(3)}^f + b_s), \quad (10)$$

where  $\hat{y}_{t_f+1,sent}$  is a scalar and  $\tanh$  is the hyperbolic tangent function, ensuring that the sentiment is bounded between -1 and 1. Thus, our model generates both continuous predictions ( $\hat{\mathbf{y}}_{t_f+1,reg}$  and  $\hat{y}_{t_f+1,sent}$ ) and a discrete probability distribution ( $\hat{\mathbf{y}}_{t_f+1,class}$ ) from the same hidden representation  $\mathbf{h}_{(3)}^f$ .

### 3.4 Generative modeling

The original Transformer-based encoder-decoder architecture introduced by [Vaswani et al. \(2017\)](#) was generative in nature: given an input sequence (e.g., in English), the decoder produced the output sequence (e.g., in French) one token at a time, conditioning each prediction on the previously generated tokens. Decoder-only models follow a similar autoregressive mechanism.

The *Narrator* inherits this generative structure. Given encoder representations  $\mathbf{H}^{\text{txt}}$  and  $\mathbf{H}^{\text{ts}}$ , along with an initial value  $\mathbf{x}_1^f$ , the model autoregressively generates  $\mathbf{x}_2^f$ , which then serves as input for the next time step. This autoregressive dynamic is not explicitly imposed by equation (8), but arises from our modeling assumption:  $\hat{\mathbf{y}}_{t_f+1,reg} = (\hat{y}_{t_f+1,reg}^{(1)}, \hat{y}_{t_f+1,reg}^{(2)}, \dots, \hat{y}_{t_f+1,reg}^{(R)})$ , where, as discussed further below,  $\mathbf{x}_{t_f+1}^f$  is constructed as a sum over selected components of  $\hat{\mathbf{y}}_{t_f+1,reg}$ .

This design supports multi-step forecasting without requiring separate models for different prediction horizons. However, the effective forecasting window is bounded by the decoder’s maximum sequence length, which defines its memory capacity.

### 3.5 Comments and benchmark specifications

Several model features and hyperparameter choices warrant discussion. First, the choices of  $R$  (number of structural components) and  $C$  (number of text classes) define the model’s structural lens. In our benchmark application we set  $R = 3$  (demand, supply, noise), with demand and supply comprising the latent business cycle factor. This assumes the business cycle is driven by fundamentals while observables include measurement noise. Correspondingly,  $C = 3$ , classifying text into the same three categories. These choices are not

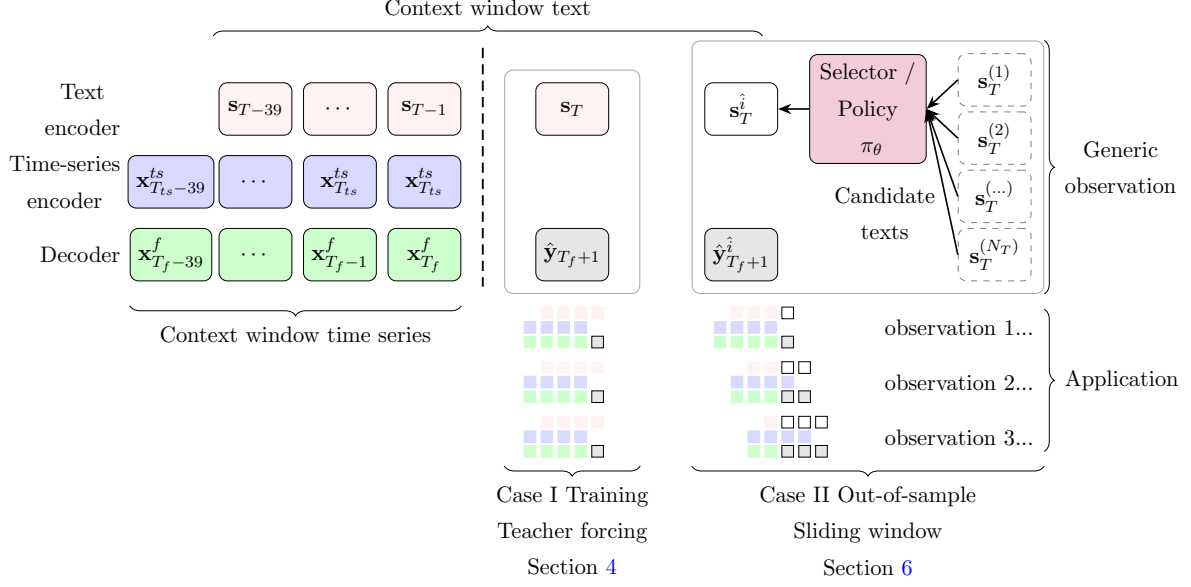
inherent requirements, and the model can easily accommodate alternative specifications - as exemplified in Section 6.4.

Second, as noted in Appendix D.2, BERT comes in various sizes. We use BERT<sub>TINY</sub> (Turc et al., 2019), a compact BERT variant that trades off model size and computation for a slight accuracy reduction, enabling more efficient deployment while maintaining strong performance on many NLP tasks. In particular, our BERT<sub>TINY</sub> has 2 Transformer encoder layers, each with 4 attention heads and a 128-dimensional hidden size (about 4 million parameters).

Third, sequence length limitations in Transformer architectures impact the model’s ability to capture long-term dependencies. For BERT<sub>TINY</sub>, the maximum sequence length is  $N = 512$ , while for our model we set  $T_{ts} = T_f = T = 40$ , equivalent to 10 years of quarterly data. While longer contexts can be used, they incur higher computational costs. Similarly, the embedding dimension  $d$ , which could in principle vary across query, key, and value matrices - is fixed at  $d = 128$ , consistent with BERT<sub>TINY</sub>, using four attention heads. Increasing  $d$  could enhance representational capacity but at a significant computational expense. For this reason, we do not explore larger values, although we do report results for smaller  $d$  values.

Fourth, we use different time indices for different inputs to emphasize the model’s capacity for handling mixed-frequency and asynchronous data via (cross-)attention and the padding and masking operations conventionally used in Transformer-based networks. Indeed, a key motivation for our multi-stream encoder-decoder design using cross-attention is its inherent suitability for handling mixed-frequency data, which is a common data property in macroeconomic contexts (Giannone et al., 2008; Mariano and Murasawa, 2003; Aruoba et al., 2009). In contrast, methods relying on concatenation or simple interactions often require synchronized sequences. While the benchmark specification assumes quarterly data throughout, an assumption relaxed in Section 6.4, we model the typical one-quarter lag in time series availability relative to text. Thus,  $t_{ts} = t_f$  and  $t = t_{ts} + 1$ , reflecting the real-world delay in macroeconomic data releases. Figure 2 illustrates the assumed information structure used for training, and later, for out-of-sample applications and news summation.

Fifth, for the time series decoder and encoder input dimensions, we adopt  $K = 1$  and  $D = 8$  in our benchmark application, where  $D$  is determined by the structure of the training data (Section 4). Given  $K \ll D$ , our encoder-decoder model shares similarities with traditional autoencoders in machine learning (Goodfellow et al., 2016) and factor models in macroeconomics (Stock and Watson, 1989; Forni et al., 2000; Giannone et al., 2008), all of which reduce input to lower-dimensional latent representations. Unlike standard autoencoders, however, our benchmark model provides the decoder with the latent



**Figure 2.** The *Narrater* information structure.

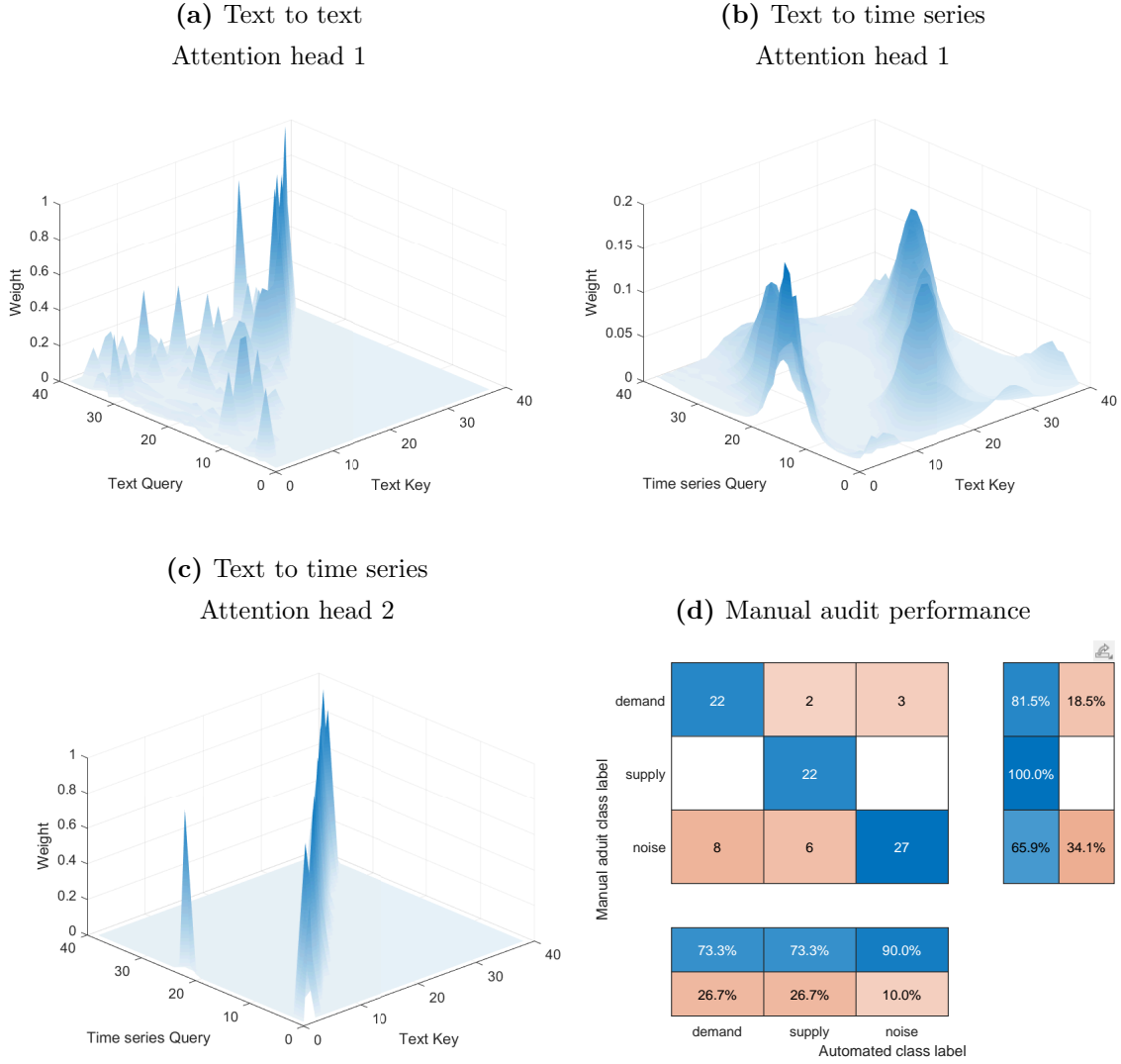
business cycle factor  $x_{t_f}^f$ , making the learning process more supervised. This structure is closer in spirit to adversarial autoencoders (Makhzani et al., 2015), which align the latent code distribution with a prior to ensure meaningful generative outputs.

Finally, attention in the Transformer and cross-attention layers can be either causal or bidirectional. We set all attention layers in (2), (4), (5), and (6) to causal, but allow bidirectional attention in (7). This design lets the model perform smoothing - akin to forward-backward filtering in econometrics - by incorporating future information into current state estimates.

Figure 3 illustrates the cross-attention mechanism and shows estimated attention scores (i.e., the softmax outputs from (21)) for one attention head in (2) and two in (7), based on a validation example. In Figure 3a, the scores reflect a clear autoregressive structure: text at time  $t$  attends only to prior periods. In contrast, the attention heads in (7) capture richer dynamics: Figure 3c emphasizes contemporaneous links between  $H_{(2)}^f$  and  $H^{txt}$ , while Figure 3b captures both leading and lagging effects.

## 4 Training

Multimodal training data are common in many domains but largely unavailable in macroeconomics. To address this constraint, we generate synthetic training data via simulation. While simulation is often used in machine learning simply to augment scarce data, in our setting it plays a more substantive role: it fixes a structural data-generating process that jointly governs observable time-series dynamics, narrative selection, and the associated structural objects of interest. Training on simulated data therefore instills a specific structural inductive bias into the *Narrater*, in the sense that the network learns



**Figure 3.** Figures 3a - 3c report attention scores (equation (21)), for one of the attention heads in (2) and two of the attention heads in (7) when processing one observation in the validation set. Figure 3d reports the manual audit’s confusion matrix.

to interpret multimodal observational realizations through a maintained structural lens. In this respect, the simulated environment functions as an explicit prior over admissible data-causal mappings, allowing the network to act as a supervised narrative filter consistent with the assumed structural model.<sup>3</sup> To avoid over-fitting and circular reasoning, simply assuming the conclusion we aim to test, we clearly split the observable data used as basis for the simulations into a training and validation sample and a sample containing genuinely unseen non-synthetic data.

<sup>3</sup>Inductive bias refers to the set of a priori assumptions embedded in a learning procedure that guide generalization from finite data. In this sense, inductive bias is closer to the use of prior information in Bayesian statistics than to statistical bias in the estimation-theoretic sense; see Caruana (1997); Ruder (2017) for discussions in the context of multi-task learning.

Below we summarize the key aspects of the data simulation process for our benchmark business cycle application and the empirical data sources used to guide it. As the *Narrater* in principle is agnostic to the specific assumptions underpinning the simulations, we provide full methodological details in Appendix B. The subsequent sections outline the estimation procedure, including hyperparameter settings and the loss function.

## 4.1 Simulation

For our main business cycle application we simulate economic time series data from a Dynamic Factor Model (DFM). We acknowledge that the true data-generating process of the economy is undoubtedly more complex and non-linear than any DFM specification. However, the DFM is a well-established and effective framework commonly used in empirical business cycle analysis (Stock and Watson, 2016). DFMs can also be used for structural inference, and the log-linearized solutions of Dynamic Stochastic General Equilibrium (DSGE) models can be expressed as constrained DFMs.

Specifically, we assume that observable macroeconomic variables are driven by two latent factors, an aggregate business cycle factor and an inflation factor, along with idiosyncratic noise. These factors evolve in response to exogenous demand and supply shocks. A demand shock drives prices and output in the same direction, while a supply shock drives them in opposite directions. The factors themselves are identified using the unit identification scheme discussed in, e.g., Bai and Wang (2014), letting in particular GDP growth load with one on the aggregate business cycle factor.

The DFM is estimated using the time series described in Section 4.3. After estimating the parameters, artificial datasets are simulated and split into training and validation sets. Each observation includes simulated macroeconomic variables  $\mathbf{x}_{t_{ts}}^{ts}$ , the latent factor  $\mathbf{x}_{t_f}^f$ , and its historical shock decomposition  $\hat{\mathbf{y}}_{t_f+1,reg}$ , for  $t_{ts} = t_f = 1, \dots, T_{ts} + 1$ .

To link the simulated times series to text, a stratified sampling strategy inspired by Dell (2024) is applied. This aims to define the semantic territory of the category of interest in vector space without injecting unintended tone or framing. For this purpose we first define a set of benchmark key-word-based text sequences representing demand and supply narratives, and good or bad sentiment. In our application we think about demand narratives as associated with general expectations and demand, whereas supply narratives are written about in the context of productivity, efficiency and innovation. Good or bad sentiment is simply defined using positive and negative word sequences. Next, we obtain the embedding representation of these benchmark sequences and real-world text sequences (from the corpus described in Section 4.3) are labeled using their embedding representations and (cosine) similarity to these benchmarks. Noise is defined as texts that do not fit either the demand or supply benchmarks well, using a 90th

percentile threshold as explained in Appendix B.

The class and sentiment of these texts are assumed to be influenced by the same shocks driving the DFM, consistent with studies portraying the media as “information intermediaries” (see, e.g., Nimark and Pitschner, 2019; Chahrour et al., 2021). However, we do not assume a deterministic mapping between shocks and texts, acknowledging the possibility of incomplete or inaccurate reporting. Instead, to construct  $\mathbf{s}_t$ ,  $y_{t_f, class}$ , and  $y_{t_f, sent}$  for training the text class is first sampled from a categorical distribution, where the class probabilities depend on the relative magnitude of the elements in  $m_{t_f}^{(r)} = |y_{t_f, reg}^{(r)}|$  at each time point:  $y_{t_f, class} \equiv c_{t_f} \sim \text{Categorical}(\boldsymbol{\pi}_{t_f})$  with  $\boldsymbol{\pi}_{t_f} = (m_{t_f}^{(c)}) / \sum_{r \in R} (m_{t_f}^{(r)})$   $c \in R$ . Let  $\mathcal{B}_c$  be the set of labeled sequences for class  $c_t$  with known sentiments  $\{s(b) : b \in \mathcal{B}_c\}$ . Then, the class-consistent article whose sentiment is closest (in absolute deviation) to the target is chosen:  $b_{t_f}^* := \arg \min_{b \in \mathcal{B}_c} |s(b) - s_{t_f}^*|$ , with  $s_{t_f}^* = \tanh(y_{t_f, reg}^{(c_{t_f})})$ , and the text-based input at time  $t$  is simply the selected sequence  $\mathbf{s}_t = \psi(b_{t_f+1}^*)$ , where  $\psi$  maps the chosen article to the representation used by the model, with sentiment  $y_{t_f, sent} = s(b_{t_f}^*)$ .

Crucially, the associations learned by the *Narrater* will be shaped by the assumptions embedded in the chosen data-generation process, and alternative assumptions are discussed below. Still, while any of these assumptions may or may not reflect real-world dynamics, overly simplistic or unrealistic structures are likely to result in poor model performance when applied to real data or evaluated through manual audits. To facilitate such evaluations, the DFM is estimated using data from 1986 to 2010, withholding the post-2010 period for out-of-sample evaluation. Similarly, text data from the post-2010 period is excluded from training to allow for independent testing on data never seen during training.

## 4.2 A manual audit and robustness

The simulation strategy described above is fast and simple and facilitates generating a large training data set in an automated manner. To assess whether the automated labeling procedure aligns well with human classification, we perform a manual audit. From the generated training data we randomly select 30 articles of each category and manually label them as demand, supply, or noise.

Figure 3d summarizes the results from the audit. The precision is above 70% for both demand and supply, and 90% for noise. In terms of sensitivity the performance is on average slightly better, with a perfect score for the supply category. On average the automated labeling procedure obtains an accuracy and F1-score of 79%.

For more concrete examples, Table 1 reports the first sentences of five articles and their manually and automatically assigned class. As illustrated, the labeling task is not



**Table 1.** First sentences of five articles used in the automatically classified training set and the associated manual audit. The news texts are translated from Norwegian to English using GPT-4.5.

| Manual | Automated | Text  |
|--------|-----------|---|
| Noise  | Noise     | <i>“A frequently repeated accusation against the government parties is that they have broken their election promises to give Norway a better school system. The accusation would have been correct if the Conservatives, the Liberal Party, and the Christian Democrats had promised during last year’s election campaign...”</i>   |
| Supply | Supply    | <i>“The Norwegian postal monopoly is falling, and new international players are ready to claim their share of the market. Fierce competition, similar to that in the telecommunications market, is expected. The fall of the postal monopoly will lead to lower prices for consumers, a variety of new services, many new small companies that won’t make money, and eventually postal bankruptcies...”</i> |
| Demand | Demand    | <i>“The National Association of Business Economists (NABE) yesterday released a survey of 47 leading analysts, expressing that the American economy is facing a sharp downturn in the first quarter. Unemployment is expected to rise to around nine percent this year, and this recession is likely the worst...”</i>  |
| Noise  | Demand    | <i>“Denmark should not expect special agreements on sensitive political issues in the Maastricht Treaty. ”The EC cannot grant Denmark exemptions within defense cooperation and the economic and monetary union as long as we demand that future member states must adhere to the Maastricht Treaty,”...”</i>   |
| Noise  | Supply    | <i>“By Friday, all production had already stopped at the small and medium-sized businesses along Greåkerveien, which runs parallel to the Glomma River just outside Sarpsborg. From the morning, there was hectic moving activity — saving whatever could be saved. By the afternoon, the business owners could do nothing but watch as the river rose minute by minute...”</i>                             |

always trivial, and different auditors might reach different conclusions.

In Appendix B.2 we perform a robustness experiment contrasting the embedding-based labeling procedure used here with a simpler Boolean key-word-based approach. This alternative labeling procedure does, however, only obtain an accuracy of 46% on the manual audit, suggesting that the embedding-based labeling procedure is reasonable.

To keep the analysis transparent and results easy to interpret we focus on the simulation baseline described above. Still, in Appendix B.3 we explore the effects of small perturbations to the text-time-series mapping. The results suggest that training on data generated under different assumptions - or directly on pooled data - could potentially further enhance model performance and robustness. We leave such explorations for future research.

### 4.3 Data

Consistent with a large literature on business cycle analysis, we utilize a broad set of macroeconomic time series to infer the latent business cycle. In the benchmark case the dataset includes Norwegian quarterly National Account Statistics: real GDP (GDP), real investments (I), and the unemployment rate (U). We also incorporate quarterly measures of consumer price inflation (CPI), house price inflation (HPI), the interest rate spread between short and long maturities (SPREAD), and household credit volumes (CREDIT). These variables, individually or jointly, have been shown to possess strong predictive power for business cycle dynamics (Estrella and Hardouvelis, 1991; Stock and Watson, 2002; Schularick and Taylor, 2012). Given the significance of oil for the Norwegian economy (Bjørnland and Thorsrud, 2016), we also include the oil price (OIL).

Data are transformed to emphasize cyclical components. Specifically, GDP, I, and CREDIT are log-transformed and differenced over eight quarters, following Hamilton (2018). For CPI, HPI, and OIL, we use the four-quarter log difference to approximate annual inflation rates.

The main text data source is a comprehensive corpus of articles from Dagens Næringsliv (DN), Norway’s largest and most widely read business newspaper, and the fourth largest overall. The articles are sourced from Retriever’s ”Atekst” archive, covering all DN publications from the late 1980s through mid 2023.<sup>4</sup> The resulting dataset contains over 500,000 articles. For each, we retain the timestamp, title, and full text, with the article body serving as input to the model.

### 4.4 Estimation

The model is trained using Stochastic Gradient Descent with the Adam optimizer. We employ a batch size of 16 and an initial learning rate of 0.0005, which is halved every second epoch to facilitate convergence.

All model parameters,  $\theta_1$  through  $\theta_6$ , along with weight and bias matrices  $\mathbf{W}_j$  and  $\mathbf{b}_j$  for  $j \in \{ts, r, c, s\}$ , are jointly estimated. The BERT<sub>TINY</sub> encoder ( $\theta_1$ ) is initialized with pre-trained weights (Turc et al., 2019), while the remaining parameters are initialized using the Glorot scheme (Glorot and Bengio, 2010). To prevent excessive updates to the pre-trained encoder, its learning rate is set to one-fifth of the rate used for the other model components.

The benchmark model outputs three predictions, regression, sentiment, and classifica-

---

<sup>4</sup>For certain recent years with incomplete data in the archive, missing articles were manually retrieved from DN’s digital print edition.

tion targets, and is trained to minimize a weighted composite loss function:

$$L_{\text{total}} = \omega_r L_r(\hat{\mathbf{y}}_{reg}, \mathbf{y}_{reg}) + \omega_s L_s(\hat{\mathbf{y}}_{sent}, \mathbf{y}_{sent}) + \omega_c L_c(\hat{\mathbf{y}}_{class}, \mathbf{y}_{class}), \quad (11)$$

where predictions are evaluated for  $t_f = 2, \dots, T_f + 1$ . Mean squared error (MSE) loss is used for both the regression and sentiment tasks:

$$L_r(\hat{\mathbf{y}}_{reg}, \mathbf{y}_{reg}) = \|\hat{\mathbf{y}}_{reg} - \mathbf{y}_{reg}\|_2^2, \quad L_s(\hat{\mathbf{y}}_{sent}, \mathbf{y}_{sent}) = \|\hat{\mathbf{y}}_{sent} - \mathbf{y}_{sent}\|_2^2,$$

and categorical cross-entropy is used for classification:

$$L_c(\hat{\mathbf{y}}_{class}, \mathbf{y}_{class}) = - \sum_{c=1}^C y_{class}^{(c)} \log(\hat{y}_{class}^{(c)}).$$

Loss weights are set to  $\omega_r = 0.5$  and  $\omega_s = \omega_c = 0.25$ , reflecting equal prioritization of the regression and text classification tasks. These weights can be tuned as hyperparameters to optimize predictive performance.

As described in Section 3.5, the model autoregressively constructs output sequences. However, during training, we use teacher forcing: ground-truth values ( $\mathbf{y}_{t_f+1,reg}$ ) are supplied as inputs rather than the model’s own predictions ( $\hat{\mathbf{y}}_{t_f+1,reg}$ ). Similarly, to facilitate efficient training, input data  $\mathbf{x}_{t_{ts}}^{ts}$  is standardized to zero mean and unit variance within each training observation.

We allocate 90% of the dataset for training and reserve 10% for validation. Validation is performed every 200 iterations. To prevent overfitting, early stopping is triggered after five validation rounds without improvement. Additional regularization is applied through L2 penalties on the model head parameters and dropout. Dropout mitigates overfitting by randomly setting a fraction of hidden units to zero during training (Srivastava et al., 2014), and is applied throughout the attention and feed-forward sublayers, promoting reliance on diverse patterns rather than specific features. The composite loss function in (11), combined with hard parameter sharing via (7), is an example of so-called multi-task learning and provides another form for (implicit) regularization (Ruder, 2017). In machine learning this approach has a long history because it typically “improves generalization by leveraging the domain-specific information contained in the training signals of related tasks” (Caruana, 1997).

Model training is performed on two NVIDIA RTX 4090 GPUs with 24 GB of memory each. Training typically completes in approximately 3 hours.

## 5 Validation results and diagnostics

In this section we use the benchmark business cycle case and validation data to assess overall model performance, conduct permutation tests, component ablations, and hyperparameter sensitivity analyses.

**Table 2.** Panel A reports classification and sentiment performance of the *Narrater* relative to a Naive Bayes (NB) classifier and the Lasso sentiment regression, as well as relative performance measures on all tasks compared to three different LLMs. A score below 1 indicates the *Narrater* performs better. Panel B reports the performance of the benchmark model relative to several alternative model specification; without a text encoder (*NoTxt*); without a time series encoder (*NoTS*); for two different embedding dimensions (*TxtComp* and *TsExp*); estimating the model without simultaneously fine-tuning the language encoder (*NoFt*); removing the sentiment and classification model heads (*NoMTL*).

| Panel<br>A | NB / Lasso | LLM (zero-shot) |      |      | Panel<br>B | Ablation |      | Hyperparameter |       | Fine tuning | No MTL |
|------------|------------|-----------------|------|------|------------|----------|------|----------------|-------|-------------|--------|
|            |            | Gemini          | o3   | R1   |            | NoTxt    | NoTS | TxtComp        | TsExp | NoFt        | NoMTL  |
| Class      | 0.95       | 0.42            | 0.34 | 0.32 |            | 0.36     | 1.00 | 1.00           | 1.00  | 0.94        |        |
| Sent.      | 0.29       | 0.05            | 0.04 | 0.07 |            | 0.08     | 0.70 | 0.63           | 0.63  | 0.16        |        |
| BusC.      |            | 0.55            | 0.68 | 0.00 |            | 0.83     | 0.96 | 1.00           | 0.97  | 0.92        | 0.94   |

## 5.1 Model accuracy

On average, the model achieves nearly 100% classification accuracy across the validation sequences and root mean squared errors (RMSE) of approximately 0.02 for sentiment predictions and 0.18 for the business cycle predictions. To contextualize these results, we compare them to three alternative approaches: a Naive Bayes classifier for text classification, a regularized Lasso regression for sentiment prediction, and three state-of-the-art LLMs with multimodal prompts.

Naive Bayes is a simple but often competitive baseline in multi-class classification tasks. Prior studies have found that deep learning models such as BERT typically outperform Naive Bayes by 5–20% (Minaee et al., 2021). For the sentiment task, we estimate a Lasso regression model using 5-fold cross-validation for hyperparameter tuning. For both models the document-term matrix is used for training. In terms of using LLMs directly for time series modeling, Gruver et al. (2023) was among the first to document that LLMs can surprisingly zero-shot extrapolate time series at a level comparable to or exceeding the performance of purpose-built time series models trained on the downstream tasks, while Requeima et al. (2024) demonstrate that LLMs can process numerical data and make probabilistic predictions at arbitrary locations, guided by natural language text describing a user’s prior knowledge. Inspired by these developments we query Google’s Gemini, OpenAI’s o3, and DeepSeek’s R1 with a multimodal prompt (Appendix F), asking the LLM for the next observation prediction as well as the class and sentiment of the text.

Panel A of Table 2 reports the model’s performance relative to these baselines. While the *Narrater*’s near-perfect classification accuracy suggests the task may not be particularly difficult, the model consistently outperforms Naive Bayes by approximately 5 percentage points. In the sentiment regression task, the improvement is more pronounced,

and the proposed model reduces the RMSE by nearly 70% relative to the Lasso baseline. The three LLMs demonstrate substantially lower performance on the text classification and sentiment scoring tasks, with classification accuracy approaching random chance and sentiment RMSEs an order of magnitude larger than the proposed model’s, suggesting that a more refined prompt engineering strategy should be adopted. In particular, with the used prompt, the LLMs do not seem to be able to efficiently utilize the multimodal information. In terms of predicting the business cycle the relative performance of the o3 model is not bad, but still substantially worse than the *Narrater*. One reason for the somewhat weak LLM performance is that they occasionally make extreme predictions, an observation also made by [Carriero et al. \(2024\)](#). Another reason is likely that these LLMs are foundational models not trained specifically to utilize structural multimodal information to address the problem at hand, and studies such as [Gambacorta et al. \(2024\)](#) and [Dell \(2024\)](#) also demonstrate that specific fine-tuning of, e.g., BERT, delivers performance on par, or better, than LLMs in a range of (economic) domain specific classification tasks.

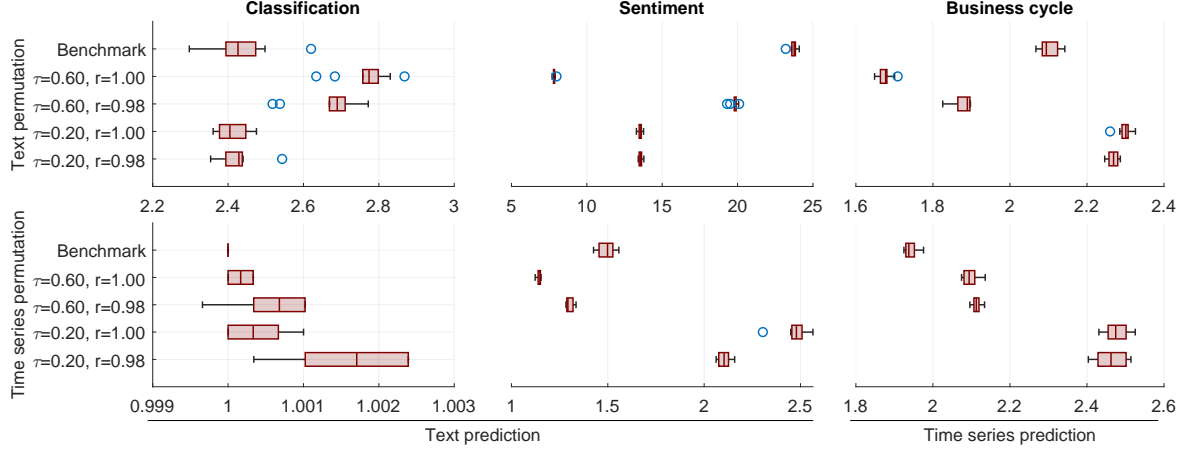
## 5.2 Permutation tests

The model is trained on data with an assumed multimodal structure. To learn about the weights the model assigns to the different data modalities we perform permutation tests.<sup>5</sup> In the tradition of [Breiman \(2001\)](#) and [Fisher et al. \(2019\)](#), an input feature is determined as “important” if shuffling its values increases the model error because in this case the model relied on the feature for the prediction. Here we assess the relative importance of text versus time series input by randomly shuffling the respective encoder input data and then comparing the change in predictive accuracy relative to the original case.

Figure 4 presents results for the last elements in the predicted sequences from 10 repetitions of the permutation test. Results for the benchmark *Narrater* are reported in the first row in each graph, while the remaining rows report robustness. For the classification task, performance drops sharply when the text input is permuted, but not when the time series input is permuted. The absence of a permutation effect on text classification in our baseline reflects that the textual input is sufficiently informative on its own, making additional cues from the time series redundant in this setting. This should not, however, be read as a limitation of the multimodal design itself. In the topic model application described in Appendix C, for example, we observe that expanding the task complexity

---

<sup>5</sup>Rather than applying post-hoc feature attribution methods such as SHAP or LIME, we use permutation tests to evaluate modality-level importance. This approach aligns more closely with our structural design and provides an intuitive assessment of the model’s dependence on each modality. Moreover, for Transformer-based architectures it is still an open research question how to apply feature attribution methods.



**Figure 4.** Traditional box plots of relative performance differences from permutation tests. A value above one indicates that the model performs worse when the particular data input is randomized.

(increasing the number of classes) under the benchmark simulation scheme produces a stronger role for the time series, and thus a clear multimodal interaction.<sup>6</sup> In contrast, both modalities contribute to sentiment and growth predictions. For growth, shuffling either modality roughly doubles the prediction error, indicating joint dependence. For sentiment, however, performance deteriorates far more when the text input is randomized, highlighting its dominant role in that task.

### 5.3 Ablation experiments

To more directly assess the contribution of each model component, we conduct ablation experiments by removing either the text encoder (*NoText*) or the time series encoder (*NoTS*) from the full model. In both cases, the modified models are re-estimated to account for potential reallocation of weights across remaining components.

Panel B of Table 2 summarizes the results. Removing the text encoder leads to a substantial decline in predictive performance across all outputs, especially for sentiment, where accuracy drops markedly. While the time series encoder is less critical for classification, it still plays an important role: sentiment prediction performance declines by approximately 30% in the *NoTS* model compared to the benchmark. These findings closely align with the permutation test results in Figure 4, and demonstrate that the benchmark model effectively integrates both data modalities to improve prediction.

<sup>6</sup>In such cases, borderline class assignments, noise, and greater overlap between textual features make it harder for the model to rely on text alone, increasing the value of complementary information from the time series. Similar patterns are observed for the additional robustness experiments reported in Figure 4. In the interest of brevity, these are discussed in Appendix B.3.

## 5.4 Hyperparameter sensitivity

We also evaluate two reduced-size variants of the benchmark model in which the embedding dimensions of the time series encoder and decoder are reduced by a factor of eight, significantly lowering the number of trainable parameters. To align embedding dimensions in the cross-attention layers, we consider two alternatives: in the first (*TxtComp*), text encoder embeddings are compressed to match the smaller time series embeddings; in the second (*TsExp*), the time series embeddings are expanded to match the original text dimensions. As shown in Table 2, both configurations perform slightly worse than the benchmark, with negligible differences between them, suggesting that the original embedding sizes capture meaningful structure.

The *NoFt* column of Table 2 compares the benchmark to a model in which the BERT language encoder is not fine-tuned, i.e., the pre-trained weights  $\theta_1$  in (1) remain fixed during training. This constraint leads to a substantial drop in sentiment prediction performance and a 6–8% decline in accuracy for the classification and growth tasks. These results underscore the importance of task-specific fine-tuning of the language encoder.<sup>7</sup>

Finally, the last column in Table 2 shows how predictive performance is affected when the benchmark *Narrater* model is estimated without the classification and sentiment regression heads, which previously provided implicit regularization via multi-task learning (Caruana, 1997; Ruder, 2017). In line with this literature we observe a performance drop.

## 6 Reinforcement learning for out-of-sample applications and news summation

To evaluate the benchmark model on non-synthetic data, we conduct a quasi-real-time out-of-sample experiment using DN news and Norwegian macroeconomic time series (Section 4.3). We initialize the model with news texts, macro data, and business cycle values for the sample 1986Q2–1996Q1, then iteratively roll it forward from 1996Q2 to 2023Q2 - noting that data after 2010 were not seen during training or validation.<sup>8</sup>

<sup>7</sup>The BERT<sub>TINY</sub> tokenizer is based on an English-language corpus. We experimented with retraining the encoder using a Norwegian tokenizer and domain-specific corpus, but found that performance declined on a range of language understanding tasks. While perhaps surprising, we suspect that this negative result likely is driven by the (limited) size and quality of the text corpus used for re-training. We also tested the larger, pre-trained Norwegian BERT<sub>BASE</sub> model (Kummervold et al., 2021), without re-estimating the model weights. In line with results reported in Table 2, the Norwegian BERT<sub>BASE</sub> model performed worse than our benchmark. While such models may offer benefits for more complex tasks, they appear sub-optimal for our application.

<sup>8</sup>The experiment is labeled *quasi-real-time* because we do not use real-time vintages of data but still assume an information structure that to a large degree mimics the information structure available in a



Each new quarter contains thousands of news articles, but only one can be mapped into the model state as the representative story. Thus, the real-time application becomes a sequential decision problem: once an article is chosen, it enters the state history and can influence all subsequent predictions. This motivates an RL framing, wherein the *Narrater* environment updates states and rewards and an agent scores articles according to a policy, as illustrated in Figure 2.

To make estimation of the policy function a tractable problem in a high-dimensional setting, while ensuring predictive accuracy, interpretable narrative attribution, robustness and efficient deployment, we build on three strands of work: (i) entropy-regularized control, which motivates the use of soft value functions and Boltzmann policies (Ziebart, 2010; Todorov, 2006; Kappen, 2005; Haarnoja et al., 2018); (ii) Kullback–Leibler (KL) based policy optimization, which provides stable projection methods for updating policies toward value-weighted targets while remaining close to a baseline distribution (Peters et al., 2010; Schulman et al., 2015; Abdolmaleki et al., 2018); and (iii) classical rollout paradigms from stochastic control, which justify usage of short-horizon lookahead to approximate long-horizon value functions (Bertsekas and Tsitsiklis, 1996; Bertsekas, 2019).

Appendix E.1 briefly outlines the mapping between this framework and the classical Bellman recursions. Below we describe our approach in greater detail. A simple special case is greedy selection, which requires no additional estimation or learning.

## 6.1 Reinforcement learning approach

To outline the problem formulation, we start from the maintained assumption that news is available for the current quarter whereas the macroeconomic data is not, i.e.,  $t_{ts} = t_f$  and  $t = t_f + 1$ . Further, for a generic quarterly time segment, let  $i$  denote the  $i$ -th news article in the last quarter  $T$  in the segment and  $N_T$  the total number of articles in this quarter. The state at time  $T$  is then  $s_T = (\mathbf{s}_T, \mathbf{x}_{T_{ts}}^{ts}, \mathbf{x}_{T_f}^f)$ , and the action set  $\mathcal{A}_T = \{a_{T,1}, \dots, a_{T,N_T}\}$  consists of all candidate news articles in quarter  $T$ . Choosing  $a_{T,i}$  corresponds to selecting article  $i$  for narrative attribution as well as model predictions  $\hat{\mathbf{y}}_{T_f+1,reg}^{(i)}$ ,  $\hat{\mathbf{y}}_{T,class}^{(i)}$ , and  $\hat{y}_{T,sent}^{(i)}$ , which in turn feed into the next state  $s_{T+1}$ .

In general, any reward function can be specified. Here, to emphasize accuracy, allow for narrative priors, and favor fundamentals and persistence, we evaluate candidates along four dimensions:

$$\begin{aligned} r_T(s_T, a) = & -\text{RMSE}\left(G\hat{D}P_{1:T_f}^{(i)}, GDP_{1:T_f}\right) + \lambda_N \text{NPRIOR}(a, \tilde{a}_T) \\ & + \lambda_F \text{FUND}(a) + \lambda_P \text{PER}(a, \mathcal{H}_{T-1}), \end{aligned} \quad (12)$$

---

real-world application.



where  $G\hat{D}P_{1:T_f}^{(i)}$  is approximated as the sum of the elements in  $\hat{\mathbf{y}}_{1:T_f,reg}^{(i)}$ , the subscripts associated with  $a$  are dropped for notation simplicity, and the mathematical definition of each term is described in Appendix E. In short, accuracy is captured via  $\text{RMSE}(G\hat{D}P_{1:T_f}^{(i)}, GDP_{1:T_f})$ ;  $\text{NPRIOR}(a, \tilde{a}_T)$  measures similarity between the candidate and a reference narrative  $\tilde{a}_T$ . The usage of reference texts is similar in spirit to the narrative identification approach taken in Antolín-Díaz and Rubio-Ramírez (2018), and helps keep the model aligned with any narrative prior the model user might have for particular time periods;  $\text{FUND}(a)$  scores each candidate article by how likely it is fundamental news (demand or supply) relative to noise and intends to capture the idea that important events are fundamental;  $\text{PER}(a, \mathcal{H}_{T-1})$  measures persistence and rewards overlap with previously chosen narratives, or in other words, provides higher rewards for narratives that prevail. Finally,  $\lambda_N, \lambda_F, \lambda_P$  are hyperparameters governing trade-offs.

Classical optimal control would optimize a policy to maximize cumulative reward, but with thousands of candidate actions per period ( $N_T \gg 2000$ ) this is computationally infeasible, and the resulting policies tend to be brittle with poor exploration. We therefore adopt an entropy-regularized approximation with short rollouts of horizon  $H$  and a parameterized softmax policy (Ziebart, 2010; Todorov, 2006; Kappen, 2005; Haarnoja et al., 2018).

The rollout-based target distribution weights actions by truncated expected returns:

$$q_T(a) \propto \mathbb{E} \left[ \sum_{h=0}^{H-1} \gamma^h r_{T+h}(s_{T+h}, a_{T+h}) \mid a_T = a \right], \quad (13)$$

with discount factor  $\gamma$ . Setting  $H = 1$  recovers a pure contextual bandit formulation, matching the setup used, for example, in news recommendation (Li et al., 2010), while modest  $H$  captures near-term dependencies without exploding complexity.

The policy class is a multinomial logistic regression over features  $\phi(s_T, a)$  extracted from the multimodal representation in (7):

$$\pi_\theta(a \mid s_T) = \frac{\exp\{\beta \phi(s_T, a)^\top \theta\}}{\sum_{a' \in \mathcal{A}_T} \exp\{\beta \phi(s_T, a')^\top \theta\}}, \quad (14)$$

where  $\beta$  is the inverse temperature. This amortizes rollout information into parameters  $\theta$ , so evaluations at deployment do not require new rollouts.

Training projects the policy onto the rollout targets using a KL-regularized objective:

$$\mathcal{L}(\theta) = \sum_{t_s=1}^{T_s} \left[ \text{KL}(q_{t_s} \parallel \pi_\theta(\cdot \mid s_{t_s})) + \lambda_{\text{KL}} \text{KL}(\pi_\theta(\cdot \mid s_{t_s}) \parallel \pi_0(\cdot \mid s_{t_s})) - \lambda_H \mathcal{H}(\pi_\theta(\cdot \mid s_{t_s})) \right], \quad (15)$$

where  $t_s = 1, \dots, T_s$  are the (quarterly) decision points. Further,  $\pi_0$  is a baseline prior (here  $\propto \exp(-\text{RMSE})$ ), entropy  $\mathcal{H}(\cdot)$  promotes exploration, and  $\lambda_{\text{KL}}, \lambda_H$  control conservatism and smoothing. This follows the logic of KL-regularized or trust-region updates

(Peters et al., 2010; Schulman et al., 2015; Abdolmaleki et al., 2018), ensuring stable and interpretable policy improvement even in high-dimensional spaces.

While this general policy learning scheme provides a principled foundation for narrative attribution, in practice it can be useful to employ a simpler, faster rule, more robust to potential out-of-sample distributional shifts. Specifically, if we restrict attention to the immediate predictive reward and collapse  $q_T$  onto the single best candidate, the policy reduces to a greedy bandit:

$$\hat{a}_{T,i} := \arg \min_{a \in \mathcal{A}_T} \text{RMSE} \left( G\hat{D}P_{1:T_f}^{(i)}, GDP_{1:T_f} \right). \quad (16)$$

As such, equation (16) can be seen as a degenerate case of the general RL framework without any rollouts and reward based solely on prediction accuracy.

## 6.2 Application details

In the real-time experiments below, we implement both the greedy baseline of (16) and the RL policy, trained via contextual bandit rollouts and KL-regularized projections. We denote these as the *Narrater-greedy* and *Narrater-RL*, respectively. The RL approach permits integrating additional dimensions of narrative quality - such as coherence with the narrative prior - into a single decision rule. It also balances exploration and exploitation through the softmax temperature and entropy regularization, and delivers probabilistic policy weights that quantify the decisiveness of each choice. The greedy rule cannot accommodate these trade-offs and provides no natural measure of confidence. It does, however, serve as a fast and transparent benchmark.

To explore the effects of different hyperparameter choices, we consider three versions of *Narrater-RL*; the default specification with rewards as in (12); a version emphasizing only RMSE rewards ( $\lambda_F = \lambda_N = \lambda_P = 0$ ); and finally a version with only RMSE and NPRIOR rewards ( $\lambda_F = \lambda_P = 0$ ). At each decision point the *Narrater* delivers an updated “smoothed” history of its output history - analogous to standard forward-backward filtering algorithms in traditional time series analysis. RMSE rewards are computed using this sequence, ensuring consistency with the information structure encountered in real-world applications.

In total 110 quarterly decision points, or time segments, are processed. For the initial segment, covering the sample 1986Q2 to 1996Q1, the history of  $\mathbf{s}_{t-1}$ ,  $\mathbf{x}_{t_s}^{ts}$ , and  $\mathbf{x}_{t_f}^f$  for  $t_f = 1, \dots, T_f$  is constructed by iterating the model forward starting from a random draw of  $\mathbf{s}_1$  and setting  $\mathbf{x}_1^f$  equal to the eight quarter growth in GDP. Implementing extractive summarization and narrative attribution using (16) requires no training. For learning  $\theta$  in (14) only data for decision points prior to 2010 are used. The selected narrative, and thus the extractive summary, maximizes  $\pi_\theta(a \mid s_t)$ , with the optimal selection index denoted  $\hat{i}$ .

The reference texts used in the reward function act as narrative priors: they need not reappear verbatim out of sample, but during training they bias the policy towards articles that align semantically with their content. Thus, the learned policy is effectively anchored in a particular narrative direction. Out of sample this anchoring ensures that new but related articles are more likely to be selected, thereby preserving continuity of the narrative attribution. Although each decision point in the training sample in principle could contain a narrative reference article  $\tilde{a}_T$ , we only specify three related to the collapse of Long-Term Capital Management in 1998Q3, the 9/11 terror attacks in 2001Q3, and the bankruptcy of Bear Stearns in 2008Q1.

Appendix E.2 provides additional training details and hyperparameter settings.

## 6.3 Application results

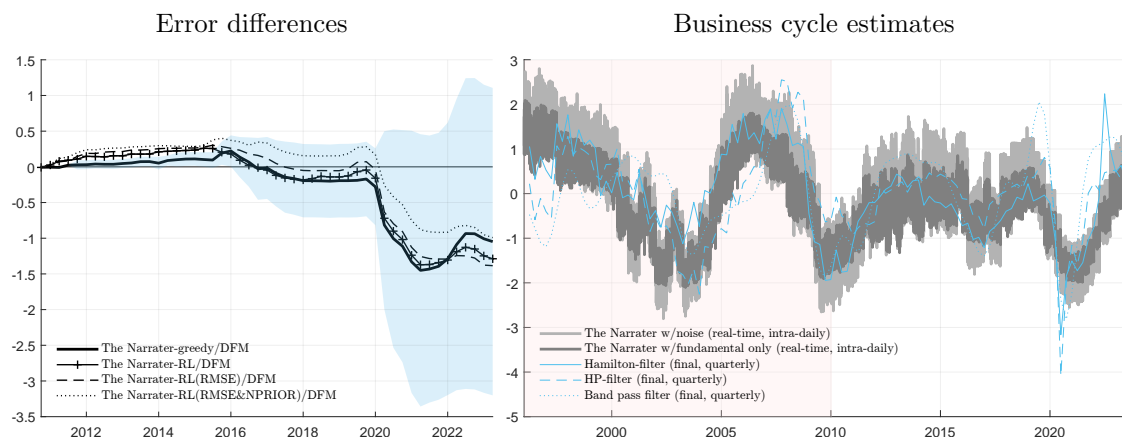
Capturing business cycle fluctuations and narrative attribution are the primary objectives of the *Narrater*. Predictive accuracy of observables, however, provides a more objective evaluation of model performance. For this reason we start by evaluating growth predictions and then in subsequent sections discuss the model’s narrative output.

### 6.3.1 Growth predictions

To evaluate the model’s growth predictions we use the sum of the elements in  $\hat{\mathbf{y}}_{t_f+1,reg}^{(i)}$  as a forecast of  $\Delta_8 GDP_{T_f+1}$ , the observed eight-quarter GDP growth, and compare it against a Dynamic Factor Model (DFM) benchmark. The DFM follows the specification outlined in Section 4.1 and produces recursive one-step-ahead forecasts via the Kalman Filter. To avoid information leakage, we evaluate all models over the 2010–2023 period, which is strictly out-of-sample for the *Narrater*.

Figure 5a shows the cumulative difference in squared prediction errors between the *Narrater* versions and the DFM benchmark. For the sample as a whole, the differences are not statistically significant. However, the *Narrater* models tend to outperform the DFM during the early Covid-19 period, suggesting the model’s capacity to integrate timely textual information during periods of heightened volatility. Furthermore, using a fully RL-based implementation typically yields a small improvement in average forecasting performance, even with a somewhat outdated policy function. Overall, the *Narrater-RL* variant trained with RMSE-only rewards performs best, although the differences are modest.

To examine how predictive accuracy evolves within a quarter as new information becomes available, we focus on the *Narrater-greedy* model and replace the predictions based on the  $\hat{i}$ th article with those obtained as the simple running average of all the  $i$  articles



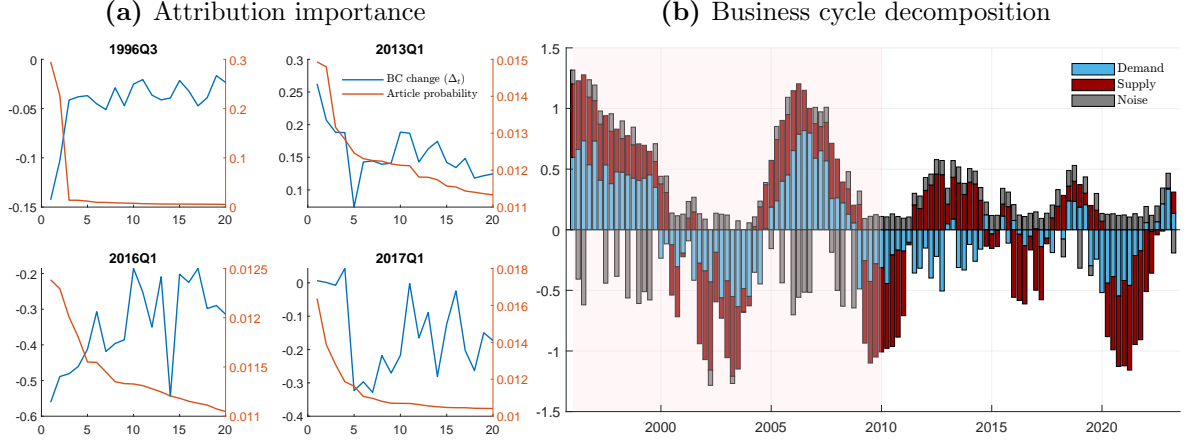
**Figure 5.** Figure 5a reports the cumulative squared prediction error difference between the *Narrater* and the DFM benchmark. A falling value implies a relative *Narrater* improvement. The shaded areas are 95% confidence intervals of differences in predictive performance, computed following Diebold and Mariano (1995) with HAC corrected standard errors. Figure 5b reports the real-time estimates of  $\hat{\mathbf{y}}_{T_f+1,reg}^{(i)}$  implied by all incoming news and aggregated into either the sum of all its components or only the demand and supply components. The white areas are time periods with data not seen during training and validation.

within the quarter. Focusing on daily time steps, Figure A.2a (Appendix A) illustrates this dynamic and echoes findings in the nowcasting literature (Giannone et al., 2008): Compared to the DFM, the *Narrater* shows a relative performance gain of approximately 20% at the start of the quarter, which steadily improves, stabilizing around 27% relative improvement after one month (30 days) of news data has been processed.<sup>9</sup>

### 6.3.2 The business cycle and economic sentiment

Figure 5b presents intra-daily business cycle estimates implied by processing all articles in the application sample. The correlations between the business cycle estimates provided by the *Narrater-greedy* and *Narrater-RL* models are high and above 0.95 (Table A.1 in Appendix A). For clarity we thus use the estimates obtained from rolling the *Narrater-greedy* model forward at each decision point in the sample. Specifically, we display predictions aggregated from the components of  $\hat{\mathbf{y}}_{T_f+1,reg}^{(i)}$ , either as the full sum or restricted to the demand and supply components for each article instance in the sample. For comparison, we also include quarterly business cycle estimates from three standard filtering methods:

<sup>9</sup>The performance scores in this section are a function of training data assumptions, model structure, and the adequacy of the data sources used for prediction. To highlight the role of the latter we perform the predictive experiment using a generic text sequence for all time periods. I.e., we replace the DN news corpus with the sequence; “This is not a real sentence. This is just noise and should not contribute towards lowering the prediction error.”, reflecting prior confidence in the DN corpus. The results reported in Figure A.2b (Appendix A) support this prior and show that the DN-based news helps improve predictive accuracy with a roughly 25% relative improvement.



**Figure 6.** Figure 6a reports, for four randomly chosen dates, the probability score for the most likely articles and their implied business cycle estimate. Figure 6b reports the real-time estimate of the vector  $\hat{\mathbf{y}}_{T_f+1,reg}^{(i)}$ . The white areas are time periods with data not seen during training and validation.

the eight-quarter GDP difference (Hamilton, 2018), the HP filter with  $\lambda = 40000$  (Hodrick and Prescott, 1997), and a band-pass filter targeting cycles between 1.5 and 8 years (Baxter and King, 1999).<sup>10</sup>

The model’s cyclical estimates are well aligned with the inductive bias instilled into the model via the training data, and as Figure 5b illustrates, also well aligned with estimates obtained from the other methods. The usage of the daily news input, however, facilitates more frequent updates using the Transformer-based model than the other filtering methods considered here. It is also clearly visible how adding noise to the demand and supply components of  $\hat{\mathbf{y}}_{t_f+1,reg}^{(i)}$  substantially increases the variation in the predictions.

Turning to the quarterly time frequency Figures A.1a and A.1b (Appendix A), compare the model’s filtered (real-time) and smoothed (retrospective) business cycle estimates. The squares show  $\hat{y}_{T_f+1,reg}^{(i)}$  as predicted at the end of each quarterly segment, while the solid lines reflect the smoothed estimates  $\hat{\mathbf{y}}_{1:T_f+1,reg}^{(i)}$ . Although the filtered estimates are revised slightly as new data becomes available, these adjustments tend to be modest, suggesting robustness in the model’s real-time inference.

A large literature documents a strong link between aggregate sentiment and the business cycle (Blanchard, 1993; Barsky and Sims, 2012; Shapiro et al., 2022). Figure A.3a (Appendix A) shows that *Narrater-greedy* sentiment, averaged across articles and normalized, correlates with the estimated cycle at 0.83 on a quarterly basis. At the article level, sentiment is noisier, has a negative bias, and is more weakly correlated, as illustrated in Figure A.3b (Appendix A). When aggregated to daily frequency, the correlation re-

<sup>10</sup>The HP filter parameter  $\lambda = 40000$  is standard for Norwegian GDP applications. As Norway lacks an official business cycle dating authority (unlike the U.S. NBER), the evaluation of cyclical estimates remains inherently subjective.

mains above 0.6, supporting the model’s assumption that sentiment and macroeconomic conditions are (partly) jointly driven by common shocks.

### 6.3.3 Narrative attribution

Having established that the model produces plausible estimates of the business cycle and reasonable growth predictions, we now turn to its more structural outputs.

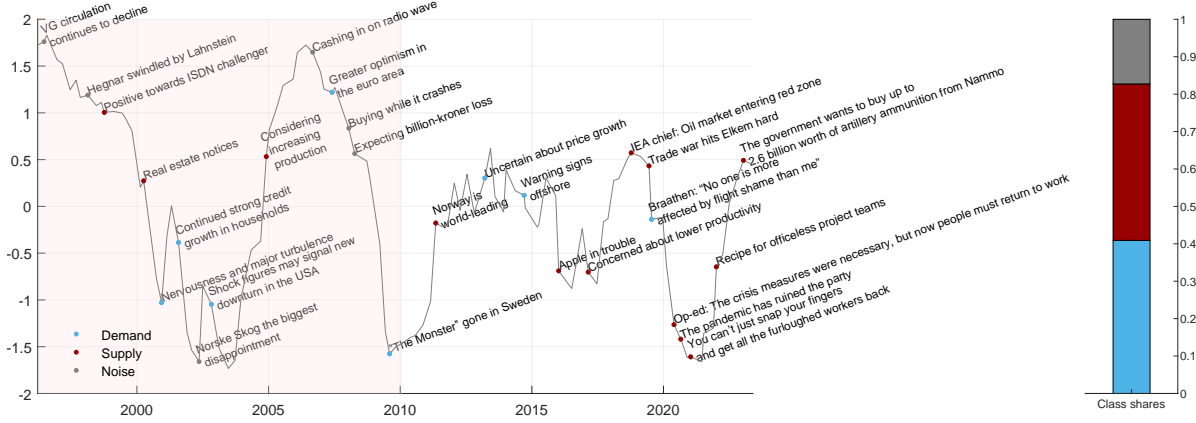
Figure 6a highlights how narrative attribution shapes business cycle estimates. It reports the implied change in the cycle at multiple dates for the 20 articles with the highest selection probabilities using the *Narrater-RL* framework. Selecting a different narrative can materially shift short-term dynamics, underscoring the value of a systematic and principled attribution scheme. As we document below, however, the RL-based and greedy versions considered here produce reasonably similar structural interpretations.

The vector  $\hat{\mathbf{y}}_{T_f+1,\text{reg}}^{(i)}$  delivers a real-time decomposition of the business cycle into demand, supply, and noise. Figure 6b plots demeaned contributions over time. Noticeable business cycle events, such as: the pre–Great Financial Crisis boom is demand-led; the subsequent downturn reflects negative supply shocks; and the Covid-19 recession combines both, with a modest late-sample demand rebound. Noise is present throughout but wanes in recent years. The figure reports output from the *Narrater-greedy* model. The default *Narrater-RL* model yields a similar decomposition; component correlations with the greedy version are 0.93 (demand), 0.93 (supply), and 0.59 (noise).

Although these decompositions cannot be objectively verified, we compare them to the structural decomposition generated by the DFM used to simulate training data. Figure A.4 (Appendix A) shows that 70% (75%) and 82% (88%) of the *Narrater-greedy*’s (*Narrater-RL*) demand and supply decompositions, respectively, fall within the admissible bounds of the DFM’s when this model is estimated using the whole sample. Considering that *Narrater*’s outputs are real-time, unlike the ex-post DFM estimates, this degree of overlap is encouraging.<sup>11</sup>

Figure 7 uses the *Narrater-greedy* model output, and combines quarterly business cycle estimates with extractive summaries of news articles, using the headline and classification of article  $\hat{i}$  for each quarter. To maintain clarity, summaries are shown selectively, while Table 3 provides a list of representative first sentences for interpretability. Additional

<sup>11</sup>Figure A.5 (Appendix A) displays the fraction of news articles classified as demand or supply, aggregated to quarterly frequency, and using the *Narrater-greedy* model version. Three observations stand out. First, only 15–20% of articles are classified as demand- or supply-related, implying most news is categorized as noise. Second, the temporal distribution of classified news aligns with known events, e.g., increased demand coverage in the early 2000s and around the Great Financial Crisis, and increased supply coverage during the early sample period, post-crisis years, and the Covid-19 episode. Third, demand-related coverage shows more variation over time, possibly due to an upward drift in classification rates.



**Figure 7.** Real-time estimate of the vector  $\hat{y}_{T_f+1,reg}^{(i)}$  together with the headline and class of article  $\hat{a}$  in a given quarter. To not clutter the graph extractive news summaries are not illustrated for each quarter. The white areas are time periods with data not seen during training and validation. The news texts are translated from Norwegian to English using GPT-4.5. The stacked bar reports the class share of  $\hat{i}$  articles across the whole sample.

sentiment and class outputs for  $\hat{i}$  articles are shown in Figure A.6 (Appendix A).

Given that each quarter includes over 2000 articles, the quality of these summaries is notable, and suggests meaningful scope for model-driven narrative attribution. For example, headlines such as “Hegnar beaten by Lahnstein”, “The Monster gone in Sweden”, and “Concerned about lower productivity” plausibly reflect noise, demand, and supply stories, respectively. Moreover, the narrative structure aligns well with known macroeconomic developments, for instance, Covid-19-related stories during the 2020 downturn, recession-themed articles during global slowdowns, and oil-related news during the energy crisis in 2014 and 2015.

Evaluated across the whole sample, roughly 40% of the summaries are allocated to the demand and supply classes (Figure 7). To further illustrate how the model output can be used to gauge the narrative strength of this type of attribution we compute two additional metrics for each quarter  $t$ ; the *RMSEMargin* and *ClassSharpness*. The *RMSEMargin* is defined as the relative performance gap between the best-performing article and the average performance of the next  $K = 100$  best articles - all in terms of RMSE loss. Similarly, we define *ClassSharpness* for quarter  $t$  as the proportion of top- $K$  articles whose predicted class matches that of the chosen article.

Figure 8a illustrates how periods with a high *ClassSharpness* and *RMSEMargin* typically are associated with notable global events and business cycle fluctuations. For instance, both the global financial crisis (2009) and the Covid-19 outbreak (2020) display relatively large *RMSEMargins* alongside high class coherence. In contrast, years such as 2002 and 2012 combine low *RMSEMargins* with weaker class coherence, suggesting more random and thus less focused narratives.



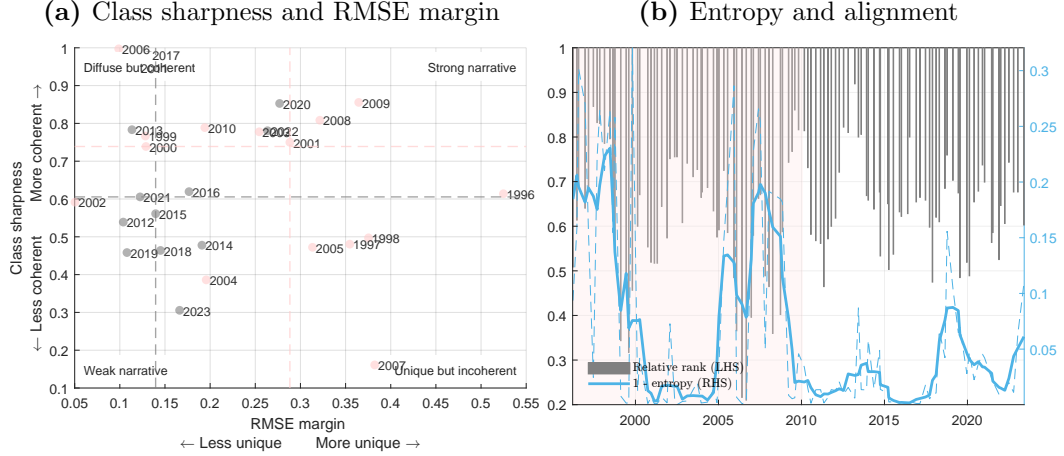
**Table 3.** The headline, first sentences, date, class, and sentiment of selected news stories. The news texts are translated from Norwegian to English using GPT-4.5.

| Date   | Class  | Sent. | Headline and text   |
|--------|--------|-------|---|
| 1998Q1 | Noise  | -0.53 | <i>Hegnar swindled by Lahnstein: “Newspapers with profits exceeding two million kroner lose press support. The Ministry of Culture has enacted a profit cap that means Trygve Hegnar’s Finansavisen and...”</i>   |
| 2000Q4 | Demand | -0.53 | <i>Nervousness and large volatility: “It has been quite a turbulent week for American and European stock investors. The nervous atmosphere in the stock markets was underscored by the significant price movements....”</i>   |
| 2004Q4 | Supply | 0.13  | <i>Considering increasing production: “Saudi Arabia may increase the country’s production capacity to 12.5 million barrels per day to ensure that the oil market is adequately supplied...”</i>   |
| 2009Q3 | Demand | -0.61 | <i>“The Monster” is missing in Sweden: “The economic downturn is hitting Sweden hard, but the country is recovering faster than the government initially feared...”</i>   |
| 2012Q3 | Demand | -0.65 | <i>Interest rates: “A series of disappointing confidence indicators from European businesses sparked new concern about the euro crisis...”</i>  |
| 2017Q1 | Supply | -0.55 | <i>Concerned about falling productivity: “Most of us have gotten less to get by on. But one of the most worrying things is that productivity growth in the industry is falling, warns researcher Ådne Cappelen...”</i>  |
| 2019Q3 | Demand | -0.62 | <i>Braathen: “Nobody is more affected by flight shame than I am”: “Per G. Braathen (58) must lay off one third of the employees at the Swedish airline BRA because passenger numbers are falling short. He believes that the resulting shame will spread to Norway...”</i>  |
| 2020Q2 | Supply | -0.51 | <i>Op-ed: The crisis measures were necessary, but now people must return to work: “Typically enough, there is now, in the start-up phase, criticism of the emergency crisis measures taken during the shutdown phase. Even so, other measures are needed now. Several have criticized the crisis measures for contributing to companies choosing to remain closed...”</i> |

The bar plot in Figure 8b shows that the selected articles from the *Narrater-greedy* summation routine also tend to be among the articles with the highest probability score when applying the default *Narrater-RL* policy function. Indeed, the median of the greedy selections is above the 75th percentile of the top performing articles using the default *Narrater-RL* model, and the two approaches selects the same article class in 65% of the cases. Table A.1 (Appendix A) shows that this statistic increases if a more RMSE oriented reward function is used. Thus, narrative attribution is affected by user preferences, but still fairly consistent across the different hyperparameter specifications considered here.

An advantage with the *Narrater-RL*-based approach is that it naturally delivers prob-





**Figure 8.** Figure 8a reports a scatter plot of the  $RMSEMargin$  and  $ClassSharpness$  measures. Quadrants, illustrated using red and gray broken lines, are defined using the median for the in- and out-of-sample periods, respectively. Values are aggregated to yearly averages for visual clarity. See the text for further details. The black bars in Figure 8b report the relative rank, defined as  $1 - \frac{r_{t_s} - 1}{N_{t_s} - 1}$ , between articles selected by the *Narrater-greedy* and *Narrater-RL* routines, where  $r_{t_s}$  is the rank of the *Narrater-greedy* selection in the probability distribution of the default *Narrater-RL* routine. The blue lines report  $1 - \hat{\mathcal{H}}_{t_s}$ , where the broken line is the raw estimate and the solid line is smoothed with a 4-period moving average for visual clarity. The white areas are time periods with data not seen during training.

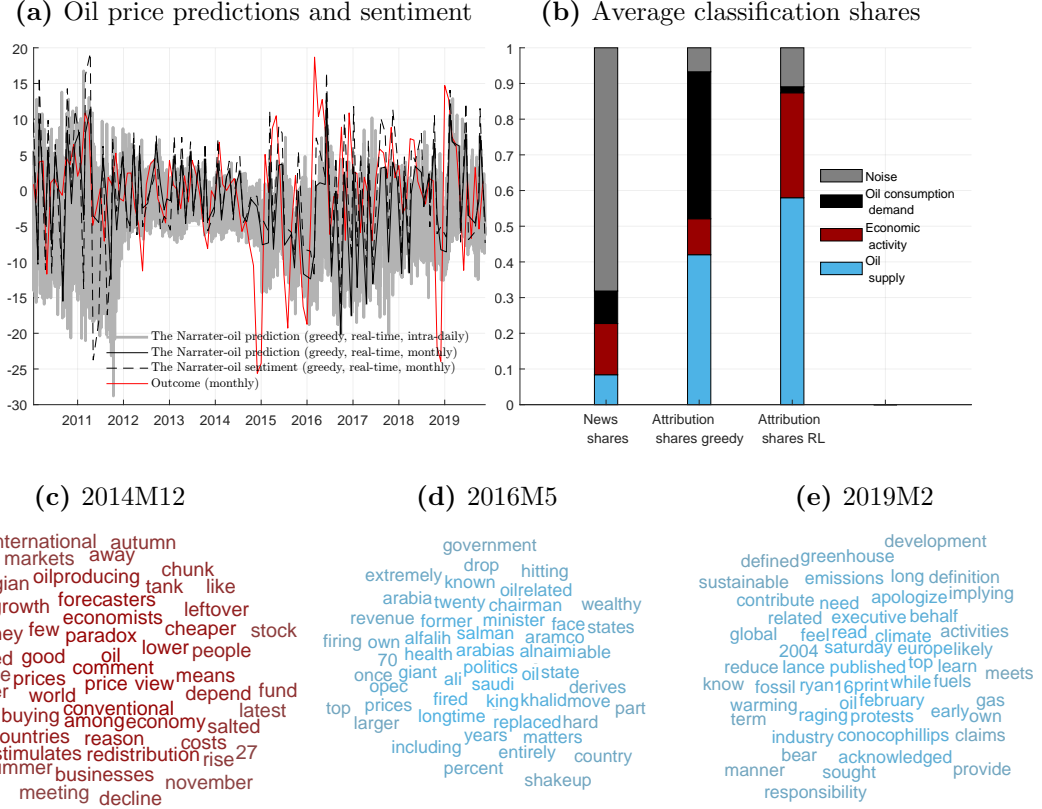
abilistic policy weights that quantify the decisiveness of each choice. The blue line in Figure 8b reports  $1 - \hat{\mathcal{H}}_{t_s}$ , where  $\hat{\mathcal{H}}_{t_s} = H_{t_s} / \log(|\mathcal{A}_{t_s}|)$ , as a measure of attribution decisiveness across time. There is relative high entropy in the early 2000s, following the Great Financial Crisis (GCF) and the oil-price slump in 2014, and also in the period before and after Covid-19.

Another favorable property of the probabilistic approach is that it also enables period-level “abstractive” summaries. Figures 9a–9f provide an illustration where headline words are weighted by an article’s selection probability and rendered as traditional word clouds. We focus on six dates/periods aligned with canonical business cycle episodes. Because entropy varies over time, the sharpness of narrative attribution likewise fluctuates, as evident in the figure, but the relevance of the narrative attribution is striking: The model automatically surfaces the mid-2014 oil-price slump, the Brexit referendum, the China-U.S. trade war, Covid-19 and remote work, and the inflation surge towards the end of the sample.

## 6.4 The oil market and alternative model configurations

To demonstrate the flexibility of the framework, and how it is easily adaptable and capable of addressing diverse user needs, we also test it in a setting where the goal is to capture global oil market dynamics as identified in Baumeister and Hamilton (2019), hereafter





**Figure 10.** Figure 10a reports the real-time estimates of  $\hat{\mathbf{y}}_{T_f+1,reg}^{(i)}$  implied by all incoming news and aggregated into the sum of all its fundamental components. Figure 10b reports the class share of  $i$  and  $\hat{i}$  articles across the out-of-sample evaluation period. Figures 10c–10e report traditional word clouds derived from words in news weighted by the article’s selection probability using the RL-based policy.

in Appendix C.2. As before, no post 2010 data is used for training or simulation, but preserved for the out-of-sample evaluation.

Table C.2 (Appendix C.3) documents that the oil-based model retains the strong classification performance on the evaluation data as previously documented for the benchmark model. Figure 10 summarizes key results from the out-of-sample experiment, where we start by focusing on the results from the greedy policy. First, processing all incoming news stories results in a large outcome space. However, the predictions implied by the selected news are relatively well aligned with the actual outcomes and the correlation between these predictions and (oil) news sentiment is above 0.9. When more formally evaluated the oil price predictions are not significantly different from those obtained from a Random Walk (RW), which is a hard benchmark in this context (Figure A.7a Appendix A). Second, close to 70% of the news articles in this corpus are “irrelevant” for the price of oil, and classified as noise. In contrast, for the articles actually selected by the greedy policy the majority are either supply or demand related. I.e., the proposed filtering framework enables relevant signal extraction even in high noise environments.

In terms of narrative attribution, Figure A.8 (Appendix A) shows that the predicted structural decompositions tend to fall well within the bounds, and in-sample estimates, of the underlying SVAR used by BH. However, even though the time series data likely disciplines the structural decompositions and average classification scores, we observe that the narrative content of the selected articles often is less convincing with too much focus on domestic developments. Although Norway is a major oil exporter, the high-dimensional action space and the fact that DN is a Norwegian focused general business news outlet, are likely explanations.

To improve narrative attribution we thus turn to the full RL-based policy which allows for including narrative prior information. Here this is supplied using the text sequence prototypes used when creating classified training data (Appendix C.2) and allocated across time in line with findings in BH, where oil price fluctuations in the period used for policy estimation are driven equally by oil supply and economic activity disturbances. In addition, we impose a hard relevance gate on the candidate action set, retaining only articles that contain the words *oil* or *international*. This restriction is deliberately minimal and is intended solely to exclude clearly irrelevant domestic business news rather than to inject additional narrative structure.<sup>13</sup>

The full RL-based policy performs on par, or better, than the greedy approach in terms of predictive accuracy (Figure A.7a), tilts the average classification scores somewhat towards the narrative prior (Figure 10b), and retains a large overlap with the (in-sample generated) structural decomposition implied by the BH model (Figure A.8). Applying the estimated policy function also improves the model’s ability to produce meaningful narrative attribution. This is illustrated in Table 4, which report the predicted class, sentiment, and headline and text of news stories with the highest probability using the RL-based policy for dates with relatively low entropy (Figure A.7b Appendix A). Similarly, focusing on three periods with large changes in the price of oil, and article content weighted by their selection probability, results in sensible period-level “abstractive” summaries (Figures 10c–10e): Negative international economic activity focus in late 2014, a surprising change in Saudi leadership and potential strategic rethink in the spring of 2016, and a notable change in climate focus and its potential consequences for energy markets in early 2019. Still, we conjecture that the quality of the narrative attribution could be further improved using more targeted text data input.

<sup>13</sup>In terms of hyperparameters, we let  $\lambda_N = 1.2$ ,  $\lambda_F = \lambda_P = 0$ , but keep the reminding parameters as defined in Table E.1 (Appendix E.2). To limit the computational complexity associated with policy optimization over long time series and large action sets, the policy is trained on monthly data covering the period 2005M1 to 2009M12 - preserving post 2010 data for out-of-sample evaluation, as before.

**Table 4.** The headline, first sentences, date, class, and sentiment of selected news stories using the RL-based policy. The news texts are translated from Norwegian to English using GPT-4.5.

| Date    | Class      | Sent. | Headline and text  |
|---------|------------|-------|--|
| 2012M1  | Econ. act. | -0.18 | <i>A Very Dangerous Phase: “Warning: The global economy has entered a very difficult and dangerous phase, the World Bank warns. Emerging economies with strong growth are cautioned against believing they are immune to the debt crisis in parts of the EU region and the United States...”</i>                                       |
| 2013M6  | Oil supply | 0.20  | <i>Secured a Major Contract: “The oil exploration company EMGS has signed a contract worth 600 million Norwegian kroner with one of the world’s largest oil companies. The contract is the third largest in EMGS’s history...”</i>   |
| 2014M11 | Econ. act. | -0.39 | <i>Weak Report: “The export and oil service supplier industry expects weaker growth going forward than it did three months ago, according to Norges Bank’s regional network survey... Economists believe the report is clearly weaker than Norges Bank had anticipated...”</i>   |
| 2015M6  | Oil supply | 0.16  | <i>OPEC Moves in Unison: “Although the oil price is forty percent below the level at the same time last year, OPEC countries chose to keep production unchanged at their meeting in Vienna. OPEC and Saudi Arabia are no longer trying to support prices by shutting off oil taps. OPEC will no longer act as a swing producer...”</i> |
| 2016M11 | Oil supply | 0.12  | <i>Maintains the Forecast: “OPEC is keeping its estimate for global oil demand in 2016 unchanged at 94.40 million barrels per day...”</i>  |

## 7 Conclusion

Human perception and decision-making are inherently multimodal; traditional empirical models in economics, however, are largely unimodal. In macroeconomics, for example, modern time-series tools deliver high-quality business cycle estimates and structural decompositions of its fluctuations, but typically lack a narrative interpretation that textual data might help provide. Against this background, we explore how recent advances in deep learning and NLP - specifically, encoder-decoder Transformer architectures - can be used to construct a multimodal supervised filter for narrative attribution.

While inherently domain agnostic, we call our proposed model the *Narrater* and demonstrate how it jointly processes (news) text and (macroeconomic) time series via contextualized embedding representations. At the core of this framework is the Transformer-based attention mechanism, which integrates modalities with potential different frequencies and reporting lags while preserving temporal context. This makes the architecture well suited to macroeconomic settings with asynchronous and high-dimensional informa-

tion. In the process, we show how the model can be used together with Reinforcement Learning techniques to produce summaries of the high-frequency and high-dimensional flow of textual information encountered in real-word contexts.

Empirically, on Norwegian data, the model yields structural decompositions and narratives that align with well-known historical episodes; it also attains quasi real-time predictive performance on par with, or modestly better than, competitive benchmarks. Permutation tests, ablations, and hyperparameter experiments show that both data modalities matter, larger embeddings improve fit within resource limits, fine-tuning the language encoder is critical, and multi-task learning improves model performance. In additional applications we demonstrate how the framework also can be used to produce reasonable filtering of international oil market dynamics and less structural topic extraction.

Our contribution is foremost methodological: a general multimodal supervised filtering framework for measurement and narrative attribution. We do not claim model-free identification of causal relationships from multimodal data. Rather, we attribute narratives consistent with a maintained structural view. This is both a strength and limitation. Limitations include reliance on labels and dynamics derived from underlying structural constraints. However, this is also the model’s strength because it makes the framework general and easily adaptable to different user needs and structural views.

Going forward we envision numerous improvements and extensions based on this type of model architecture, which also can be useful in other settings where text (or other unstructured streams) co-evolves with numeric time series.

## References

- Abdolmaleki, A., J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. Riedmiller (2018). Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*.
- Angeletos, G.-M., F. Collard, and H. Dellas (2020). Business-Cycle Anatomy. *American Economic Review* 110(10), 3030–70.
- Antolín-Díaz, J. and J. F. Rubio-Ramírez (2018). Narrative sign restrictions for SVARs. *American Economic Review* 108(10), 2802–29.
- Arias, J. E., J. F. Rubio-Ramírez, and D. F. Waggoner (2018). Inference Based on Structural Vector Autoregressions Identified With Sign and Zero Restrictions: Theory and Applications. *Econometrica* 86(2), 685–720.
- Aruoba, S. B., F. X. Diebold, and C. Scotti (2009). Real-Time Measurement of Business Conditions. *Journal of Business & Economic Statistics* 27(4), 417–427.

- Ash, E., S. Hansen, Y. Muvdi, and C. Marangon (2026). *Large Language Models in Economics*, pp. 191–210. Springer Nature Switzerland.
- Bai, J. and P. Wang (2014). Identification theory for high dimensional static and dynamic factor models. *Journal of Econometrics* 178(2), 794–804.
- Baker, S. R., N. Bloom, and S. J. Davis (2016). Measuring economic policy uncertainty. *The Quarterly Journal of Economics* 131(4), 1593–1636.
- Bañbura, M., D. Giannone, and L. Reichlin (2010). Large Bayesian Vector Auto Regressions. *Journal of applied Econometrics* 25(1), 71–92.
- Banbura, M., D. Giannone, and L. Reichlin (2011). *Nowcasting*. The Oxford Handbook of Economic Forecasting. New York: Oxford University Press.
- Barsky, R. B. and E. R. Sims (2012). Information, Animal Spirits, and the Meaning of Innovations in Consumer Confidence. *American Economic Review* 102(4), 1343–77.
- Baumeister, C. and J. D. Hamilton (2019). Structural interpretation of vector autoregressions with incomplete identification: Revisiting the role of oil supply and demand shocks. *American Economic Review* 109(5), 1873–1910.
- Baxter, M. and R. G. King (1999). Measuring business cycles: Approximate band-pass filters for economic time series. *Review of Economics and Statistics* 81(4), 575–593.
- Bernanke, B., J. Boivin, and P. S. Elias (2005). Measuring the effects of monetary policy: A Factor-augmented Vector Autoregressive (FAVAR) approach. *The Quarterly Journal of Economics* 120(1), 387–422.
- Bertsekas, D. P. (2019). *Reinforcement Learning and Optimal Control*. Belmont, MA: Athena Scientific.
- Bertsekas, D. P. and J. N. Tsitsiklis (1996). *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific.
- Bjørnland, H. C. and L. A. Thorsrud (2016). Boom or gloom? Examining the Dutch disease in two-speed economies. *The Economic Journal* 126(598), 2219–2256.
- Blanchard, O. (1993). Consumption and the recession of 1990-1991. *The American Economic Review* 83(2), 270–274.
- Breiman, L. (2001). Random forests. *Machine learning* 45, 5–32.

- Burri, M. and D. Kaufmann (2020). A daily fever curve for the Swiss economy. *Swiss Journal of Economics and Statistics* 156, 1–11.
- Bybee, L., B. Kelly, A. Manela, and D. Xiu (2024). Business news and business cycles. *The Journal of Finance* 79(5), 3105–3147.
- Carriero, A., D. Pettenuzzo, and S. Shekhar (2024). Macroeconomic forecasting with large language models. *arXiv preprint arXiv:2407.00890*.
- Caruana, R. (1997). Multitask learning. *Machine learning* 28(1), 41–75.
- Chahrour, R., K. Nimark, and S. Pitschner (2021). Sectoral Media Focus and Aggregate Fluctuations. *American Economic Review* 111(12), 3872–3922.
- Cochrane, J. H. (1994). Shocks. *Carnegie-Rochester Conference Series on Public Policy* 41, 295–364.
- Das, A., W. Kong, R. Sen, and Y. Zhou (2024). A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*.
- Dell, M. (2024). Deep learning for economists. Technical report, National Bureau of Economic Research.
- Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Diebold, F. X. and R. S. Mariano (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics* 13(3), 253–63.
- Estrella, A. and G. A. Hardouvelis (1991). The term structure as a predictor of real economic activity. *The Journal of Finance* 46(2), 555–576.
- Evans, M. D. D. (2005). Where Are We Now? Real-Time Estimates of the Macroeconomy. *International Journal of Central Banking* 1(2), 127–175.
- Fisher, A., C. Rudin, and F. Dominici (2019). All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research* 20(177), 1–81.
- Forni, M., M. Hallin, M. Lippi, and L. Reichlin (2000). The generalized dynamic-factor model: Identification and estimation. *Review of Economics and statistics* 82(4), 540–554.



- Friedman, M. and A. J. Schwartz (1963). *A Monetary History of the United States, 1867-1960*. Princeton University Press.
- Gambacorta, L., B. Kwon, T. Park, P. Patelli, and S. Zhu (2024). CB-LMs: language models for central banking. Technical report, Bank for International Settlements.
- Giannone, D., L. Reichlin, and D. Small (2008). Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics* 55(4), 665–676.
- Glorot, X. and Y. Bengio (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256.
- Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press.
- Gorodnichenko, Y., T. Pham, and O. Talavera (2023). The voice of monetary policy. *American Economic Review* 113(2), 548–84.
- Gruver, N., M. Finzi, S. Qiu, and A. G. Wilson (2023). Large language models are zero-shot time series forecasters. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.
- Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 1861–1870.
- Hamilton, J. D. (2018). Why you should never use the Hodrick-Prescott filter. *Review of Economics and Statistics* 100(5), 831–843.
- Hansen, S. and M. McMahon (2016). Shocking language: Understanding the macroeconomic effects of central bank communication. *Journal of International Economics* 99(S1), 114–133.
- Hansen, S., M. McMahon, and A. Prat (2018). Transparency and deliberation within the fomc: A computational linguistics approach. *The Quarterly Journal of Economics* 133(2), 801–870.
- Hodrick, R. J. and E. C. Prescott (1997). Postwar U.S. Business Cycles: An Empirical Investigation. *Journal of Money, Credit and Banking* 29(1), 1–16.
- Hollmann, N., S. Müller, K. Eggenberger, and F. Hutter (2022). TabPFN: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*.

- Jin, M., S. Wang, L. Ma, Z. Chu, J. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-f. Li, S. Pan, et al. (2024). Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *International Conference on Learning Representations*.
- Kappen, H. J. (2005). Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment* (11), P11011.
- Kummervold, P. E., J. De la Rosa, F. Wetjen, and S. A. Brygfjeld (2021). Operationalizing a National Digital Library: The Case for a Norwegian Transformer Model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pp. 20–29.
- Larsen, V. H. and L. A. Thorsrud (2019). The value of news for economic developments. *Journal of Econometrics* 210(1), 203 – 218.
- Larsen, V. H., L. A. Thorsrud, and J. Zhulanova (2021). News-driven inflation expectations and information rigidities. *Journal of Monetary Economics* 117, 507–520.
- Li, L., W. Chu, J. Langford, and R. E. Schapire (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pp. 661–670.
- Lim, B., S. Ö. Arik, N. Loeff, and T. Pfister (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting* 37(4), 1748–1764.
- Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81.
- Liu, Z., D. Huang, K. Huang, Z. Li, and J. Zhao (2021). FinBERT: A pre-trained financial language representation model for financial text mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pp. 4513–4519.
- Lu, J., D. Batra, D. Parikh, and S. Lee (2019). ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems* 32.
- Makhzani, A., J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

- Marcellino, M., M. Porqueddu, and F. Venditti (2016). Short-Term GDP Forecasting With a Mixed-Frequency Dynamic Factor Model With Stochastic Volatility. *Journal of Business & Economic Statistics* 34(1), 118–127.
- Mariano, R. S. and Y. Murasawa (2003). A new coincident index of business cycles based on monthly and quarterly series. *Journal of Applied Econometrics* 18(4), 427–443.
- Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Minaee, S., N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao (2021). Deep learning-based text classification: a comprehensive review. *ACM computing surveys (CSUR)* 54(3), 1–40.
- Müller, S., N. Hollmann, S. Pineda Arango, J. Grabocka, and F. Hutter (2021). Transformers can do bayesian inference. *arXiv preprint arXiv:2112.10510*.
- Nagler, T. (2023). Statistical foundations of prior-data fitted networks. In *Proceedings of the 40th International Conference on Machine Learning*, Volume 202, pp. 25660–25676.
- Nimark, K. P. and S. Pitschner (2019). News media and delegated information choice. *Journal of Economic Theory* 181, 160 – 196.
- Pennington, J., R. Socher, and C. Manning (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.
- Peters, J., K. Mülling, and Y. Altün (2010). Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 10)*, pp. 1607–1612.
- Ramey, V. A. (2011). Identifying government spending shocks: It’s all in the timing. *The Quarterly Journal of Economics* 126(1), 1–50.
- Requeima, J., J. Bronskill, D. Choi, R. Turner, and D. K. Duvenaud (2024). Llm processes: Numerical predictive distributions conditioned on natural language. *Advances in Neural Information Processing Systems* 37, 109609–109671.
- Romer, C. D. and D. H. Romer (1989). Does monetary policy matter? A new test in the spirit of Friedman and Schwartz. *NBER macroeconomics annual* 4, 121–170.
- Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

- Schularick, M. and A. M. Taylor (2012). Credit booms gone bust: monetary policy, leverage cycles, and financial crises, 1870–2008. *American Economic Review* 102(2), 1029–1061.
- Schulman, J., S. Levine, P. Abbeel, M. Jordan, and P. Moritz (2015). Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 1889–1897.
- Shapiro, A. H., M. Sudhof, and D. J. Wilson (2022). Measuring news sentiment. *Journal of Econometrics* 228(2), 221–243.
- Shiller, R. J. (2017). Narrative economics. *American Economic Review* 107(4), 967–1004.
- Sims, C. A. (1980). Macroeconomics and reality. *Econometrica* 48(1), 1–48.
- Smets, F. and R. Wouters (2007). Shocks and frictions in US business cycles: A Bayesian DSGE approach. *American economic review* 97(3), 586–606.
- Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1), 1929–1958.
- Stock, J. and M. Watson (2012). Disentangling the channels of the 2007-2009 recession. *Brookings Papers on Economic Activity Spring 2012*, 81–135.
- Stock, J. H. and M. W. Watson (1989). New indexes of coincident and leading economic indicators. In O. J. Blanchard and F. Stanley (Eds.), *NBER Macroeconomics Annual*, NBER Chapters, pp. 351–394. Cambridge, MA: The MIT Press.
- Stock, J. H. and M. W. Watson (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics* 20(2), 147–62.
- Stock, J. H. and M. W. Watson (2016). Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. In J. B. Taylor and H. Uhlig (Eds.), *Handbook of Macroeconomics*, Volume 2, pp. 415–525. Elsevier.
- Thorsrud, L. A. (2018). Words are the new numbers: A newsy coincident index of the business cycle. *Journal of Business & Economic Statistics*, 1–17.
- Todorov, E. (2006). Linearly-solvable markov decision problems. *Advances in Neural Information Processing Systems*, 1369–1376.

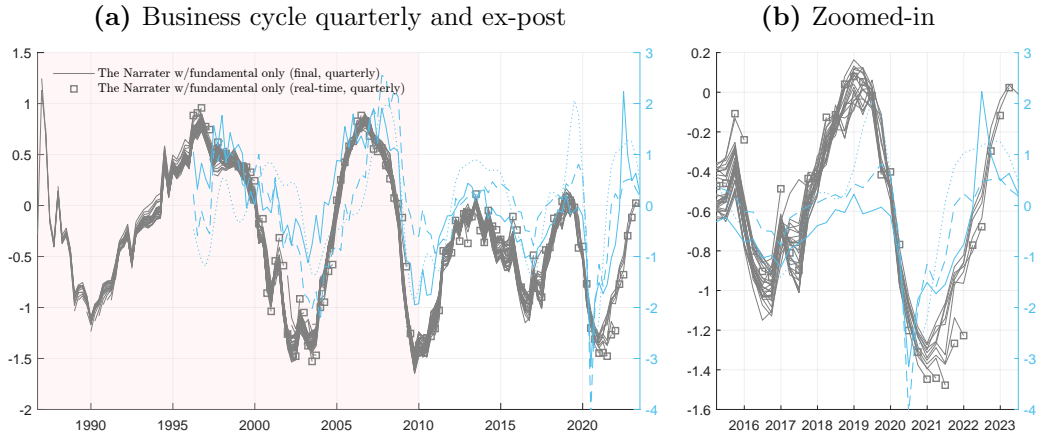
- Tsai, Y.-H. H., S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov (2019). Multimodal Transformer for Unaligned Multimodal Language Sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6558–6569.
- Turc, I., M.-W. Chang, K. Lee, and K. Toutanova (2019). Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.
- Veldkamp, L. L. (2011). *Information choice in macroeconomics and finance*. Princeton University Press.
- Wen, Q., T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun (2023). Transformers in time series: a survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*.
- Woo, G., C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo (2024). Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*.
- Xu, P., X. Zhu, and D. A. Clifton (2023). Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45(10), 12113–12132.
- Zhang, H., P. S. Yu, and J. Zhang (2025). A systematic survey of text summarization: From statistical methods to large language models. *ACM Computing Surveys* 57(11).
- Ziebart, B. D. (2010). *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. Ph. D. thesis, Carnegie Mellon University.

# Appendices for online publication

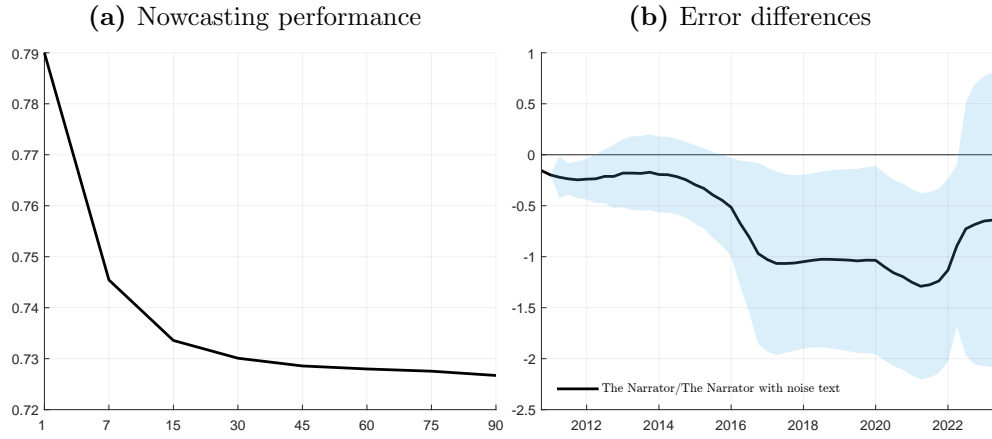
## Appendix A Additional results

**Table A.1.** Correspondence and correlation between *Narrater-greedy* and different implementations of *Narrater-RL*; default, only RMSE rewards ( $\lambda_F = \lambda_N = \lambda_P = 0$ ), and only RMSE and NPRIOR rewards ( $\lambda_F = \lambda_P = 0$ ).

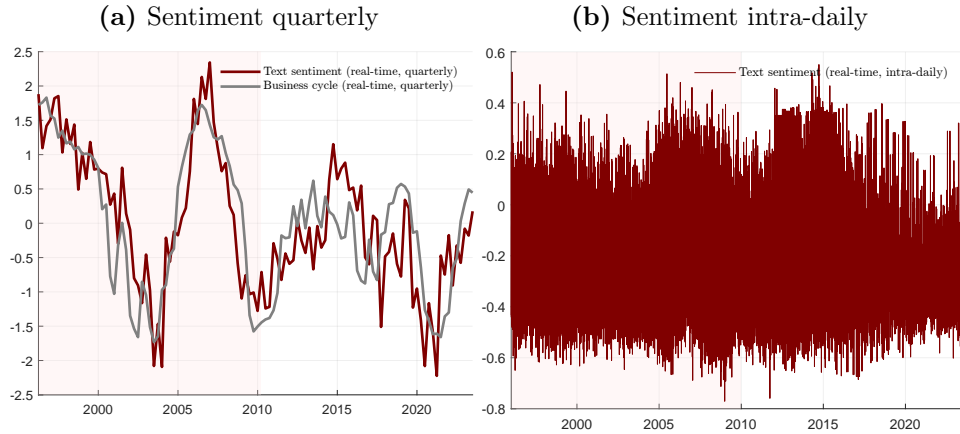
|                 | <i>Default</i> | <i>Only RMSE rewards</i> | <i>Only RMSE and NPRIOR rewards</i> |
|-----------------|----------------|--------------------------|-------------------------------------|
| Class/selection | 0.65           | 0.72                     | 0.70                                |
| Business cycle  | 0.97           | 0.95                     | 0.95                                |



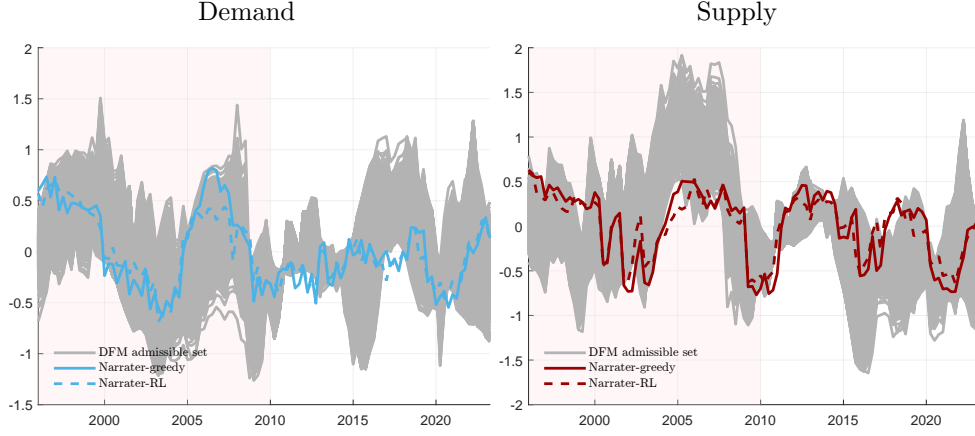
**Figure A.1.** The graphs report the real-time (filtered) estimates  $\hat{y}_{T_f+1,reg}^{(i)}$  together with the smoothed versions  $\hat{y}_{1:T_f+1,reg}^{(i)}$ . The white areas are time periods with data not seen during training and validation.



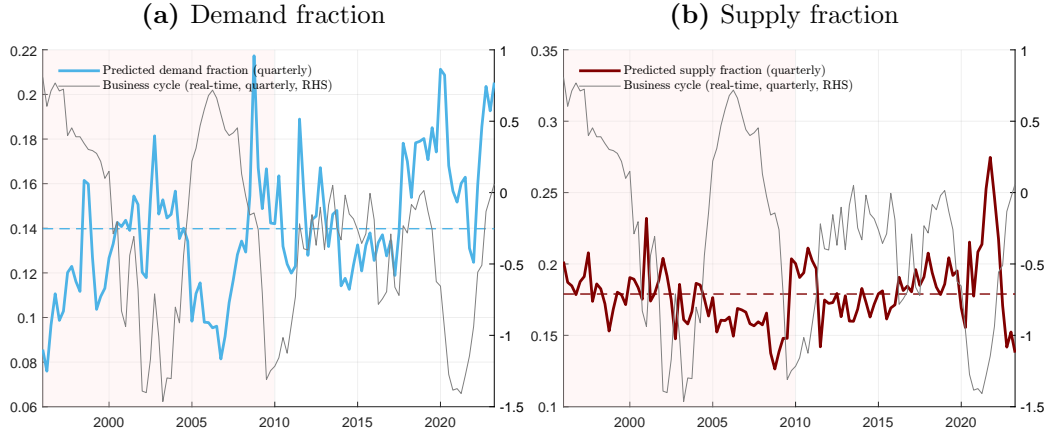
**Figure A.2.** Figure A.2a reports the predictive performance of the *Narrater* relative to the DFM, when the former predictions are constructed as averages of 1 - 90 daily (x-axis) predictions. Figure A.2b reports the cumulative squared prediction error difference between the *Narrater* and a version of the model using a generic and recurrent text sequence. A falling value implies a relative *Narrater* improvement. The shaded areas are 95% confidence intervals of differences in predictive performance, computed following Diebold and Mariano (1995) with HAC corrected standard errors.



**Figure A.3.** Figure A.3a reports the normalized quarterly mean of the *Narrater*'s real-time sentiment predictions. Figure A.3b reports the quarterly real-real time estimates of  $\hat{\mathbf{y}}_{T_f+1, \text{sent}}^{(i)}$  implied by all incoming news. The white areas are time periods with data not seen during training and validation.

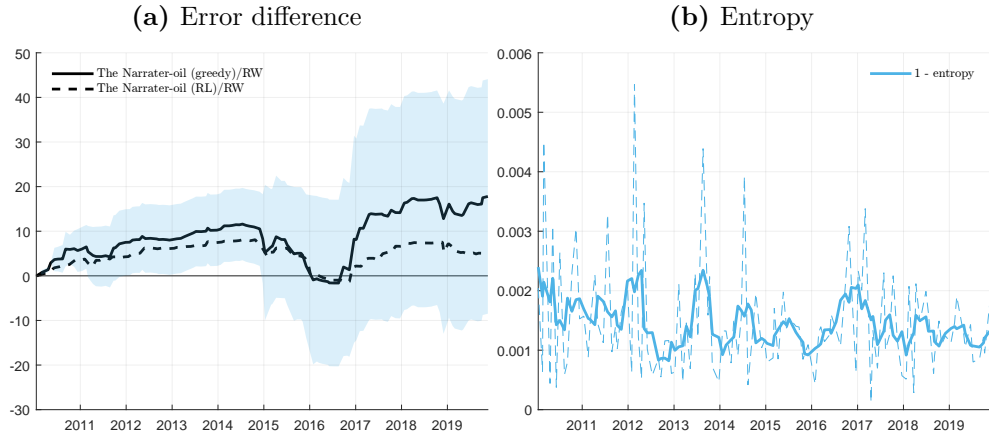


**Figure A.4.** Real-time (predictive) demand and supply decomposition produced by the *Narrater* together with the similar objects produced by the DFM, described in Appendix B. For the DFM these are ex-post estimates (because they are computed with data for the whole sample available). Since the DFM is identified using sign restrictions, we report the set of historical shock decompositions associated with the impulse response functions fulfilling the sign restrictions, i.e., the admissible set. The white areas are time periods with data not seen during training and validation of the *Narrater*.

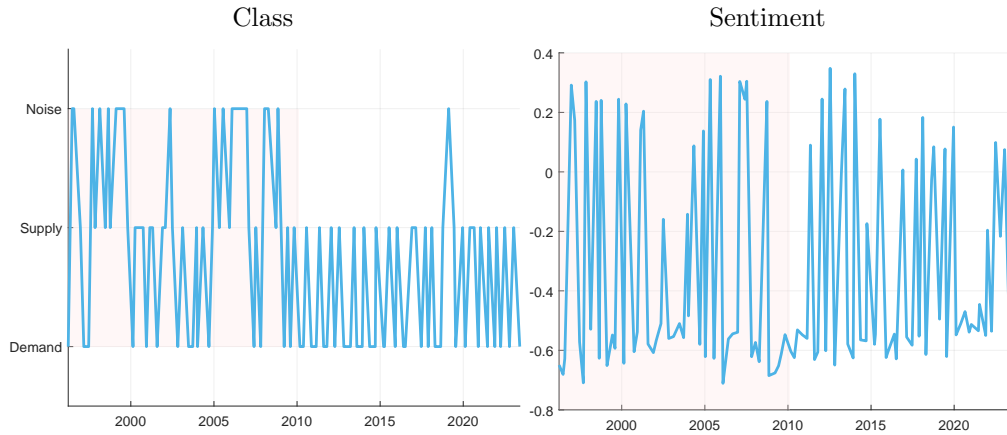


**Figure A.5.** Fraction of news classified as either demand or supply using  $\hat{\mathbf{y}}_{T_f+1, class}^{(i)}$  and each quarterly time segment. The white areas are time periods with data not seen during training and validation.

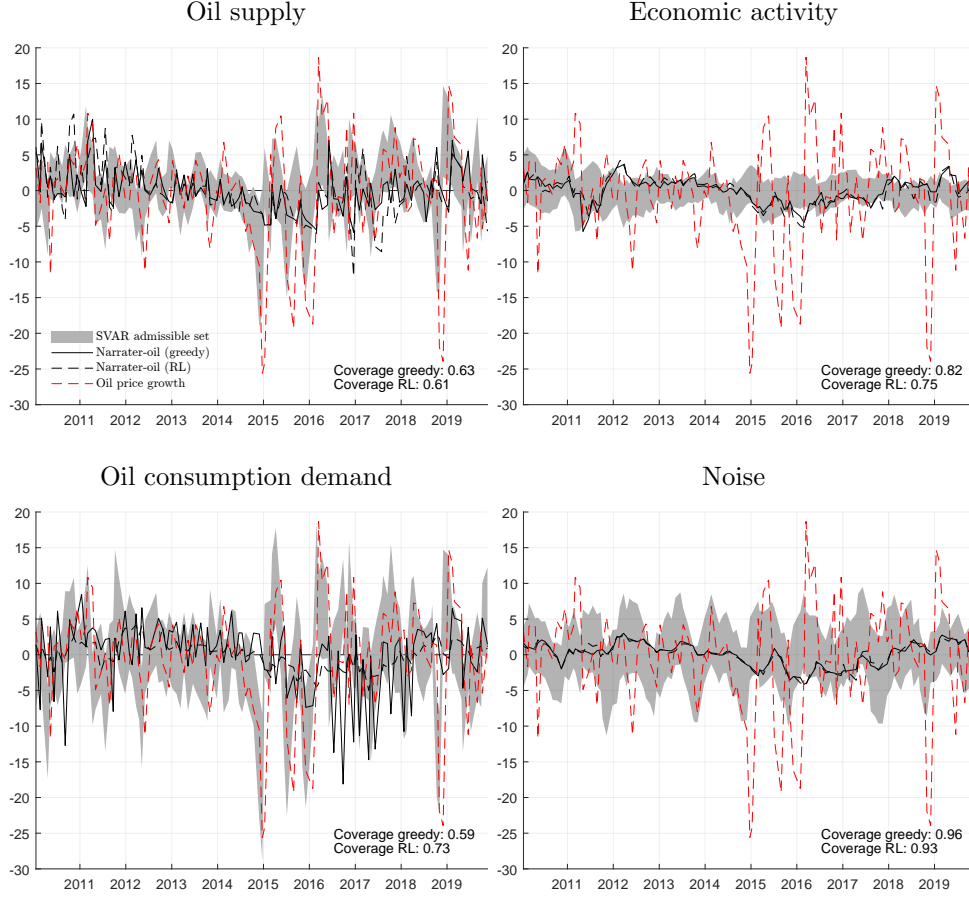




**Figure A.7.** Figure A.7a reports the cumulative squared prediction error difference between the *Narrater-oil* and the RW benchmark. A falling value implies a relative *Narrater-oil* improvement. The shaded areas are 95% confidence intervals of differences in predictive performance, computed following Diebold and Mariano (1995) with HAC corrected standard errors. Figure A.7b reports  $1 - \hat{\mathcal{H}}_{t_s}$ , where the broken line is the raw estimate and the solid line is smoothed with a 4-period moving average for visual clarity.



**Figure A.6.** Quarterly real-time estimates of the class and sentiment of article  $\hat{i}$ . The white areas are time periods with data not seen during training and validation.



**Figure A.8.** Real-time (predictive) structural decompositions produced by the *Narrater-oil* model together with the similar objects produced by the SVAR proposed by [Baumeister and Hamilton \(2019\)](#). For the SVAR these are ex-post estimates (because they are computed with data for the whole sample available). Since the SVAR is identified using sign restrictions, we report the set of historical shock decompositions associated with the impulse response functions fulfilling the sign restrictions, i.e., the admissible set.

## Appendix B Generating training data

As described in Section 4.1, synthetic training data is generated by simulating time series data from an estimated Dynamic Factor Model and mapping text to time series using the implied historical shock decomposition. Prior to this, the class and sentiment of a large set of text data used for training and validation is automatically recorded using embeddings. Below we provide the details of each step.

### B.1 Business cycle simulations

To simulate synthetic macroeconomic data we assume a state space system where the dynamics of the observable data are described by a set of latent state variables evolving according to an autoregressive process. In particular, let  $\mathbf{x}_{t_{ts}}^{ts}$  be a  $D \times 1$  vector of observed

time series at time  $t_{ts}$  and  $\mathbf{f}_{t_{ts}}$  a  $r \times 1$  vector of latent factors, such that:

$$\begin{aligned}\mathbf{x}_{t_{ts}}^{ts} &= \mathbf{\Lambda} \mathbf{f}_{t_{ts}} + \mathbf{u}_{t_{ts}}, \quad \mathbf{u}_{t_{ts}} \sim \mathcal{N}(0, \mathbf{R}), \\ \mathbf{f}_{t_{ts}} &= \mathbf{\Phi}(L) \mathbf{f}_{t_{ts}-1} + \mathbf{H} \varepsilon_{t_{ts}}, \quad \varepsilon_{t_{ts}} \sim \mathcal{N}(0, \mathbf{Q}),\end{aligned}\tag{17}$$

where  $\mathbf{\Lambda}$  is a  $D \times r$  matrix of factor loadings,  $\mathbf{u}_{t_{ts}}$  is a  $D \times 1$  vector of idiosyncratic errors (or noise),  $\mathbf{\Phi}(L)$  is a lag polynomial matrix of autoregressive coefficients.  $\mathbf{R}$  and  $\mathbf{Q}$  are the variance-covariance matrices of the error terms. We assume a diagonal structure for  $\mathbf{R}$  but not for  $\mathbf{Q}$ . Moreover, we write the transition equation in (17) using the companion form, and let  $q$  denote the number of latent factors such that  $q \leq r$  and the number of lags of the factors is  $p$  (with  $p \times q = r$ ). Thus,  $\varepsilon_{t_{ts}}$  is a  $q \times 1$  vector of innovations (or shocks) to the factors and  $\mathbf{H}$  is a  $r \times q$  selection matrix.

For our particular experiment, we focus on two fundamental shocks ( $q = 2$ ), namely aggregate demand and supply shocks, and identify the latent factors as an aggregate business cycle index and an underlying inflation measure. The factors are identified following the unit identification scheme described in, e.g., [Bai and Wang \(2014\)](#), implying that the upper left  $q \times q$  block of  $\mathbf{\Lambda}$  is the identify matrix. The relationship between the reduced form errors and the structural shocks is given by  $\varepsilon_{t_{ts}} = \mathbf{A} \mathbf{e}_{t_{ts}}$ . Because  $(\mathbf{A} \mathbf{e}_{t_{ts}})(\mathbf{A} \mathbf{e}_{t_{ts}})^\top = \mathbf{Q}$ , identification is achieved from  $E(\mathbf{e}_{t_{ts}} \mathbf{e}_{t_{ts}}^\top) = \mathbf{I}$  and by restricting the elements of  $\mathbf{A}$  using sign restrictions following [Arias et al. \(2018\)](#). Accordingly,  $\mathbf{A}$  is restricted such that the following relationship holds:

$$\begin{bmatrix} \varepsilon_{1,t_{ts}} \\ \varepsilon_{2,t_{ts}} \end{bmatrix} = \begin{bmatrix} + & + \\ + & - \end{bmatrix} \begin{bmatrix} e_{t_{ts}}^d \\ e_{t_{ts}}^s \end{bmatrix}\tag{18}$$

where a positive demand innovation moves prices and the business cycle in the same direction while a positive supply shock moves these factors in the opposite direction.

To simulate training data from the model we first estimate it using the data described in Section 4.3 and covering the sample 1986 to 2010. We allow for  $p = 8$  lags and apply a two-step estimation procedure, similar to in [Bernanke et al. \(2005\)](#): Common factors are first estimated using Principal Components Analysis (PCA) and identified as described above. Next, these factors are included in the VAR to estimate the time series dynamics using OLS. Although sign restrictions only give set identification we focus on the structural parameters associated with the median draw when sampling admissible impulse responses following (18).

In total we simulate 60000 time series segments from the model. The length of each segment is determined by the memory constraint of the *Narrater*, while we always remove the first five years of simulated data to avoid dependence on starting values. Simulating data directly from (17) produces synthetic data for the observable macroeconomic indicators ( $\mathbf{x}_{t_{ts}}^{ts}$ ) and the latent factors, including the business cycle, where  $\mathbf{x}_{t_f}^f = \mathbf{f}_{1,t_{ts}}$ . To

**Table B.1.** The class and sentiment sequences and key-words used to generate benchmark embeddings and Boolean search, respectively. For the key-word lists only the most relevant terms are reported. See the text for details.

| Narrative | Text sequence   | Key-words   |
|-----------|---|---|
| Demand    | <i>“demand, sentiment, expectations”</i>                  | <i>shopping, income, demand, consumption, wage growth, tourism, interest rate hike, wage settlement, transfers, consumer loan,...</i> |
| Supply    | <i>“productivity, efficiency, innovation, technology”</i> | <i>technology, labor, strike, sanctions, regulation, storm, sick leave, leave of absence, recruitment, overcapacity,...</i>           |
| Positive  | <i>“positive, strong, good”</i>                           |   |
| Negative  | <i>“negative, weak, bad”</i>                              |   |

also produce synthetic data for the associated historical shock decomposition we use:

$$\mathbf{f}_{tts} = \sum_{j=0}^t \Psi_j \mathbf{A} \mathbf{e}_{tts-j}, \quad (19)$$

where  $\Psi_j$  are the impulse response coefficient matrices (from the moving average (MA) representation of the VAR), and consider the effect of each shock individually on the implied decomposition of  $\mathbf{f}_{1,tts}$ . To also capture the noise component we use the element in  $\mathbf{u}_{tts}$  associated with GDP growth.

## B.2 Creating labeled text data

To create labeled text data that can be mapped to the simulated time series dynamics we proceed in three steps. First we define a set of key-word sequences for the demand and supply categories and good and bad sentiment. These are listed in Table B.1. Next, we use the OpenAI API, and the `text-embedding-3-small` model, to produce embedding representations of each of these benchmark text sequences. The intuition for this approach is that the embedding representations capture shared context, and thus captures broader aspects and abstractions than the key-words themselves. At the same time, using key-word-based sequences instead of, e.g., prototype sentences, reduces the risk of introducing unintended sentiment, framing, or policy biases in the categorizations. Finally, we randomly sample text sequences from the DN news corpus described in Section 4.3, but only consider news reported within the sample 1986 to 2010, and compute each sampled article’s similarity with the benchmark embeddings.

More formally, let  $\mathcal{X}$  denote the set of sampled news texts from the DN corpus (restricted to 1986–2010), and let  $\phi : \mathcal{T} \rightarrow \mathbb{R}^d$  be the embedding map induced by the

`text-embedding-3-small` model. Likewise, let  $\mathcal{B}^{\text{cls}} = \{b_c\}_{c=1}^{C-1}$  be the benchmark keyword sequences used for categorization (e.g., demand and supply), and let  $\mathcal{B}^{\text{sent}} = \{b^+, b^-\}$  be the two sentiment benchmarks used for scoring. Then, for  $x \in \mathcal{X}$  and  $c \in \{1, \dots, C-1\}$  we define class-wise and sentiment similarity as:

$$s_c(x) = \cos(\phi(x), \phi(b_c)) \quad s^+(x) = \cos(\phi(x), \phi(b^+)), \quad s^-(x) = \cos(\phi(x), \phi(b^-)),$$

where, for any  $u, v \in \mathbb{R}^d$ ,  $\cos(u, v)$  is cosine similarity. Then, to motivate distinct article classes we set a class-specific threshold:

$$\tau_c = Q_{0.9}(\{s_c(x) : x \in \mathcal{X}\}), \quad c = 1, \dots, C-1,$$

where  $Q_{0.9}(\cdot)$  denote the empirical 90th percentile, and assign an article  $x$  to a distinct class if its top score clears that class's threshold:

$$y(x)_{\text{class}} = \begin{cases} c^*(x), & \text{if } s_{c^*(x)}(x) \geq \tau_{c^*(x)} \text{ and } c^*(x) \text{ is unique,} \\ \text{noise,} & \text{otherwise,} \end{cases}$$

with  $c^*(x) := \arg \max_c s_c(x)$ . Because  $\tau_k$  is the 90th percentile for each class, the majority of sampled texts are labeled as noise.<sup>14</sup> We have experimented with using different thresholds, finding that if we do not define a relatively high threshold value, labeled articles will be difficult to discriminate. Finally, independently of the class assignment, we compute:

$$sd(x) = s^+(x) - s^-(x), \quad y(x)_{\text{sent}} = \tanh(sd(x)) \in (-1, 1),$$

and use  $y(x)_{\text{sent}}$  as a measure of the article's sentiment.

The procedure for creating labeled text data is simple and fast, and allows labeling a large set of articles automatically. Arguably, the procedure could be improved and refined in a number of directions. As an alternative we have also explored creating labeled text data using a Boolean search-based approach. First, two domain-specific dictionaries of Norwegian terms were manually compiled. One dictionary was intended to capture *aggregate demand* news and the other *aggregate supply* news. The final demand dictionary contains 72 unique tokens, while the supply dictionary contains 129. The 10 most frequent words in each list are displayed in the last column of Table B.1. The full lists are available upon request.

Second, each article was lower-cased, stripped of punctuation, and tokenised. We then counted the number of hits from the demand and supply dictionaries for every article, producing two variables `demand_count` and `supply_count`. An article is labelled as **demand**

<sup>14</sup>To avoid poor performance for the minority classes we use a weighted loss function when training the model. In particular, a class is assigned a weight inversely proportional to its frequency in the training data. These weights are then used to scale the loss contribution of each training sample so that misclassifying a minority class costs more than misclassifying a majority class.

|                          |                       |        |       |       |       |       |
|--------------------------|-----------------------|--------|-------|-------|-------|-------|
| Manual adult class label | demand                | 3      |       | 24    | 11.1% | 88.9% |
|                          | supply                | 1      | 4     | 17    | 18.2% | 81.8% |
|                          | noise                 | 3      | 4     | 34    | 82.9% | 17.1% |
|                          |                       |        |       |       |       |       |
|                          |                       | 42.9%  | 50.0% | 45.3% |       |       |
|                          |                       | 57.1%  | 50.0% | 54.7% |       |       |
|                          | demand                | supply | noise |       |       |       |
|                          | Automated class label |        |       |       |       |       |

**Figure B.1.** Manual audit’s confusion matrix when a Boolean search-based approach has been used to create labeled text data.

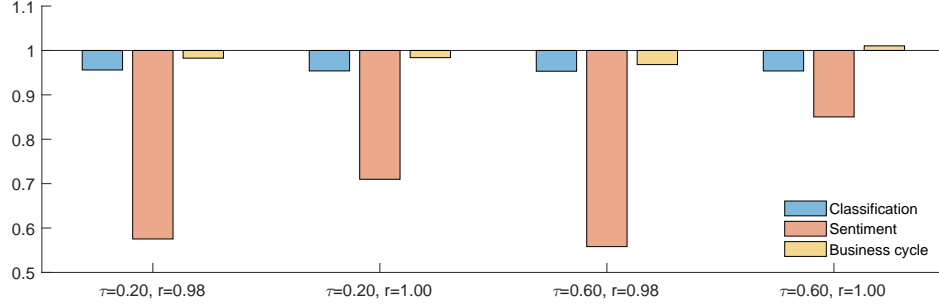
when `demand_count` > 0 and `supply_count` = 0, and as **supply** when `supply_count` > 0 and `demand_count` = 0. Articles that contain keywords from *both* lists or from *neither* list are classified as noise. To avoid extremely long feature articles with many tangential topics, we discard texts longer than 1 000 words before applying the rule. The Boolean approach therefore focuses on relatively short, single-topic news items.

Figure B.1 reports the results from the same type of manual audit as described in Section 4.2, but now applied to the text data labeled using the Boolean search-based approach. The overall accuracy is only 0.46, compared to 0.79 for the embedding-based approach. Similarly, the results presented in Section 6 of the main paper are positive, suggesting that the embedding-based labeling procedure is reasonable. We thus leave it for future research to investigate potentially more adequate methods for automatically creating labeled text data.

### B.3 Simulation alternatives

To examine how training data assumptions shape the model’s use of multimodal information, we have simulated alternative datasets that explicitly manipulate the sharpness and noise in the text–time-series mapping. Instead of linearly linking the probability of drawing each text class to the size of the historical shock decompositions, we apply a temperature-controlled softmax with  $\tau = 0.2$  or  $\tau = 0.6$ . In addition, for each  $\tau$  we introduce noise by randomly breaking the link between the true class and the chosen text, governed by  $r \sim \text{Bernoulli}(p_{\text{rel}})$  with  $p_{\text{rel}} = 0.98$  or 1.00.

The additional rows in Figure 4 (Section 5.2) report permutation tests for models trained on these alternative data. A clear pattern emerges: the sharper and more deterministic the mapping between shock decompositions and texts, the more predictive performance deteriorates when the time-series input is permuted. Similarly, injecting noise into the textual mapping shifts the model’s reliance toward the time series in the



**Figure B.2.** Performance scores relative to the benchmark *Narrater* model. All model versions are evaluated on the same validation data. A value below one indicates that the alternative model is better.

text-prediction tasks.

Drawing on the multi-head attention mechanism in the Transformer architecture, an interesting avenue for future research is to train a single model directly on data generated under varying assumptions. Such a setup would allow different attention heads to specialize in distinct regimes, potentially yielding a final model that is robust across a broader range of data-generating processes. Speaking to this argument, Figure B.2 reports relative predictive scores when all models analyzed in Figure 4 are evaluated on a test data set containing random draws from all five simulation experiments (including the benchmark described in Section 4.1). Interestingly, under these assumptions, the benchmark model is worse than all of the alternatives (except in one case), and the best model would be the one trained on data using the  $\tau = 0.6$  and  $p_{\text{rel}} = 0.98$  assumption.

## Appendix C The oil market and alternative model configurations

To demonstrate the flexibility of the framework, and how it is easily adaptable and capable of addressing diverse user needs, we also test it in a setting where the goal is to capture global oil market dynamics as identified in Baumeister and Hamilton (2019), and describe two other, less structural, configurations exemplifying news topic identification and mixed-frequency data processing. The former application is presented in Section 6.4, with further details below, while the two latter examples are fully presented here.

### C.1 Topic modeling and mixed frequency data

To exemplify how the proposed framework can be used for news topic identification and mixed-frequency data processing we consider two alternative configurations of the model: the *Narrater-topic* and the *Narrater-mixed*.

In *Narrater-topic* we keep the benchmark setup but expand the number of text classes



to  $C = 9$ . Both demand and supply are divided into four topical subclasses - *policy/regulation*, *labor market*, *prices*, and *trade/energy*. These subclasses are less “structural” than the originals but enrich the narrative content and show how the model can also serve as a supervised topic model.

In *Narrater-mixed* we highlight the architecture’s mixed-frequency properties and focus on extractive summarization. Here all text sequences  $\mathbf{s}_t$  are intra-quarterly, but only one is informative for predicting the business cycle, while  $\mathbf{x}_{t_{ts}}^{ts}$  is observed monthly. Because text and time series are non-overlapping, this model predicts only the latent business cycle index rather than reconstructing the full history, but enables efficient filtering of high-frequency text via cross-attention, with article selection replacing the benchmark classification task. As such, the RL approach adopted for news summation becomes redundant for this model specification. Allowing simultaneous classification, selection, and overlapping time windows is possible but would require substantially more memory.

To keep the computational burden and memory requirements for the *Narrater-mixed* reasonable we only consider  $T = C = 10$ , and since the model does not give any structural decomposition of the business cycle in this configuration  $R = 1$ . Moreover, since we do not assume any potential temporal relationship between the  $C$  articles, equation (2) is removed and the model head in (10) is attached to the model’s text encoder. Thus, the sentiment of each of the  $C$  articles is predicted, but only one article is relevant for the business cycle at time  $T_f + 1$ . To facilitate this type of selection we further let the text classification head be a direct function of the time-series-text-embedding cross-attention scores.

Letting  $\mathbf{q}_2^f$  denote the  $1 \times d$  decoder query vector for time period  $T_f + 1$ , the scores are:

$$\mathbf{a} = \text{softmax} \left( \frac{\mathbf{q}_2^f \mathbf{K}^{txt^\top}}{\sqrt{d}} \right) \in \mathbb{R}^{1 \times T}, \quad (20)$$

and  $\mathbf{a}$  replaces  $\mathbf{h}_{(3)}^f$  in (9). This prevents the network from mixing the information in the two data modalities before identifying the relevant text. In contrast, using the aggregated attention output from (21) as input for the text selection head would imply that the model loses the individual identity of the texts before trying to select the correct one. As before  $T_f = 40$  quarters, but  $T_{ts} = 120$  monthly observations, and we assume that  $t_{ts} = t_f + 1$  such that (realistically) monthly observations are available for the current quarter.

## C.2 Generating training data

Training and validation data are generated with the simulation scheme described in Section 4.1 and Appendix B, with minor modifications described below.

To create simulated time series data for training the *Narrater-oil* model we exchange

the DFM with the SVAR specification proposed by [Baumeister and Hamilton \(2019\)](#). The reduced form representation of the model is used to generate simulated time series which are then decomposed into the historical shock decomposition for the real price of oil using the model’s structural parameters. For estimation of model parameters we used data provided here <https://sites.google.com/site/cjsbaumeister/research>, but truncate the sample to cover the period 1958 to 2009. For comparison, we also estimate a version of the model using data ending in 2019, which is the last period in the publicly available dataset.

To create labeled text data for the *Narrater-oil* and *Narrater-topic* experiments we follow the same type of embedding-based labeling procedure as before. We first define text sequences for each of the oil market shocks and topic categories of interest. These are listed in Table C.1. Then, for each sampled text sequence we again use the “text-embedding-3-small” model to obtain its embedding representation and compute the cosine similarity with the benchmark embeddings. Finally, we attach a new class label for each text depending on the index of the maximum cosine similarity score and the original classifications. For the *Narrater-topic* model this implies that we obtain four subclasses for both “demand” and “supply”.

For the *Narrater-mixed* experiment we construct a mapping between the text and time series data so that the sentiment of the true text is as close to  $\tanh(y_{T_f+1,reg})$  as possible, where  $y_{T_f+1,reg}$  is the business cycle index. Then, to construct a set of false articles for each training and validation observation we simply select the  $C - 1$  articles whos sentiment differs the most with the true sentiment. This clearly simplifies the classification problem. For practitioners wanting to apply this model a potentially more realistic approach for creating the set of false articles could be to randomly select articles from the same real word time index as the true article.

Monthly time series data are sourced to reflect the quarterly time series when possible. We use the monthly frequency of U, CPI, SPREAD, and OIL, while the reminding variables are “replaced” by (log) yearly changes in monthly industrial production, retail sales, and money growth, and the level of OECD’s business tendency survey for Norway. Then to simulate monthly  $\mathbf{x}_{ts}^{ts}$  time series data we first use PCA and the monthly variables to compute one common factor, factor loadings, and the variance of the idiosyncratic noise. Because the monthly variables have somewhat different sample lengths, we use data covering the period 1993 to 2010 to derive these objects. Next we simply assume that the mean of the monthly factors are equal to the quarterly business cycle index from the the original training data, and use the estimated factor loadings, the variance of the idiosyncratic noise, and the normal distribution, to simulate monthly observations that are consistent with the dynamics of the original quarterly business cycle index.

**Table C.1.** The table reports the text sequences used to generate the oil- or topic-based embeddings. See the text for details. Note that terms are translated from Norwegian to English, and that Norwegian frequently combines two or more words to create a single new word with a specific meaning. E.g. “monetary policy” is one single word in Norwegian.

|                       | Narrative              | Text sequence  |
|-----------------------|------------------------|--|
| <i>Narrater-oil</i>   | Oil supply             | “oil, petroleum, OPEC, supply, production, surprise”                                   |
|                       | Economic activity      | “global, economy, activity, expectations, forecast, growth”                            |
|                       | Oil consumption demand | “oil, petroleum, risk, inventory, speculation, demand”                                 |
| <i>Narrater-topic</i> | Policy/regulation      | “interest rate, central bank, budget, tax, regulation, monetary policy, fiscal policy” |
|                       | Labor market           | “unemployment, employment, jobs, permits, wage, labor market”                          |
|                       | Prices                 | “inflation, consumer price index, cost of living, price level, price development”      |
|                       | Trade/energy           | “export, import, trade, trade balance, oil price, energy, war, geopolitics”            |

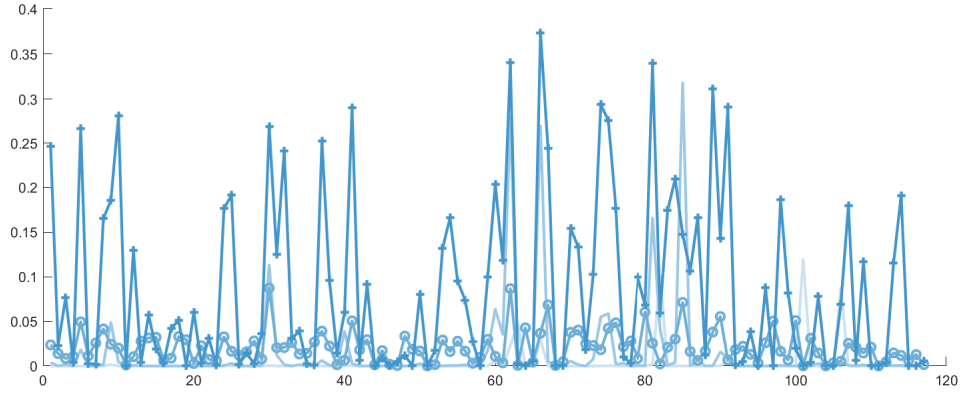
**Table C.2.** Alternative model specifications and validation data performance. For the *Narrater-topic* and *Narrater-mixed* models the sentiment (Sent.) and business cycle (BusC.) tasks are similar while the classification task is different.

|                            | <i>Narrater</i> | <i>Narrater-topic</i> | <i>Narrater-mixed</i> | <i>Narrater-oil</i> |
|----------------------------|-----------------|-----------------------|-----------------------|---------------------|
| Class/selection (accuracy) | 0.99            | 0.99                  | 0.99                  | 0.99                |
| Sent. (RMSE)               | 0.02            | 0.03                  | 0.03                  | 0.03                |
| BusC. /Oil price (RMSE)    | 0.18            | 0.18                  | 0.19                  | 0.5                 |

### C.3 Results

Table C.2 reports the different model configurations’ performance on the validation data. Of main interest here is the classification task. As discussed earlier, the benchmark model achieves close to perfect accuracy. Although the alternative model configurations perform different classification (*Narrater-topic* and *Narrater-oil*) or text selection (*Narrater-mixed*) tasks, their accuracy is also close to one, exemplifying how these model configurations can efficiently identify, e.g., user-defined news topics or be used to automatically perform accurate extractive news summaries.

For regression tasks, all alternatives perform worse than the benchmark in predicting sentiment, and the *Narrater-mixed* also underperforms on the business cycle. This decline is not due to poor handling of monthly time series - Figure C.1 shows that cross-attention assigns weight across the full monthly sequence - but likely due to the removal of structural decomposition (supply, demand, noise) in the output which previously provided implicit



**Figure C.1.** Attention scores from equation (6) and the *Narrater-mixed* model for period  $T_f + 1$  and one particular validation set observation. The four lines reflect the different attention heads, while the x-axis reflect the sequence length of the monthly data.

regularization via multi-task learning (Caruana, 1997; Ruder, 2017) - as documented in Table 2.

Finally, to illustrate the *Narrater-topic* output more explicitly, Figure C.2 reports the fraction of articles classified as one of the four topics across time when the model is used to process the non-synthetic data: We do, for example, observe an elevated focus on politics and regulations following the introduction of inflation targeting and the EU enlargement in the early 2000s, a larger fraction of labor market oriented news during turbulent times such as the financial crisis and Covid-19 period, and a substantial increase in articles about prices and inflation towards the latter part of the sample.

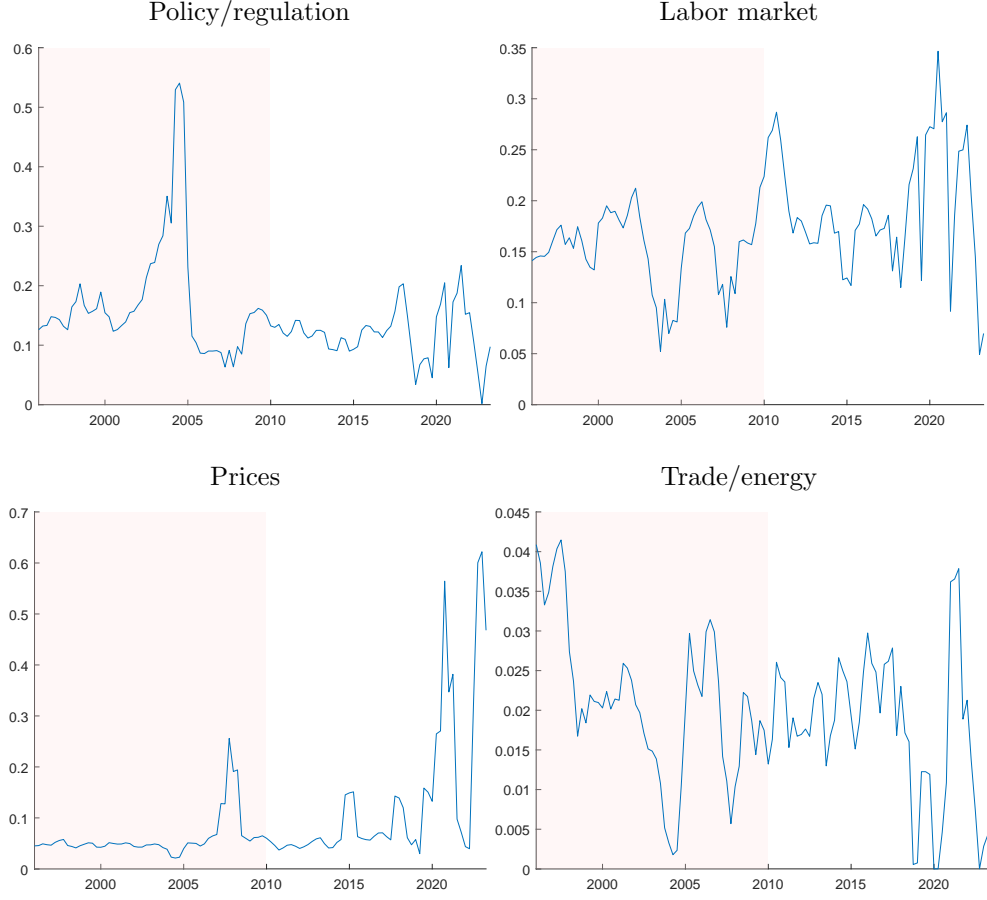
## Appendix D Transformers and BERT

In the proposed architecture we combine a BERT language encoder with a Transformer-based encoder-decoder structure for processing time series data. Below we shortly describe the general Transformer block in a neural network and the BERT structure.

### D.1 The Transformer

A Transformer block, sometimes called a Transformer encoder layer, is a key computational unit originally introduced in the Transformer architecture (Vaswani et al., 2017), which has since become foundational in a wide range of sequence modeling tasks.

At its core, a Transformer block processes an input sequence representation through a self-attention mechanism, followed by a position-wise feed-forward network (FFN). The input to the block is first normalized using a layer normalization operation. Let the input embedding sequence be represented as a matrix  $\mathbf{X} \in \mathbb{R}^{n \times d}$ , where  $n$  is the sequence length



**Figure C.2.** Fraction of news articles, aggregated to quarterly frequency, classified by the *Narrator-topic* model to be about policy/regulation, the labor market, prices, or trade/energy. The white areas are time periods with data not seen during training and validation.

and  $d$  is the feature dimension. After applying layer normalization, a so-called multi-head attention module computes a weighted sum of values across positions, enabling each element to attend to other relevant elements in the sequence. By allowing for multiple heads, the model can focus on information from different representation subspaces at different positions, capturing a richer variety of dependencies compared to single-head attention. This is achieved by projecting the input  $\mathbf{X}$  into three matrices: queries  $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$ , keys  $\mathbf{K} = \mathbf{X}\mathbf{W}_K$ , and values  $\mathbf{V} = \mathbf{X}\mathbf{W}_V$ , where  $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d_k}$  are learnable parameters, and  $d_k = d/h$  for  $h$  attention heads. The scaled dot-product attention is then computed as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) \mathbf{V}. \quad (21)$$

For multi-head attention, this operation is performed  $h$  times in parallel and the results are concatenated to form a  $d$ -dimensional output. A residual connection is added to the layer input, and the output is again normalized.

In many sequence modeling settings, attention is required to be causal, which in

this setting implies that each element in the sequence can only attend to its current and preceding positions, not future ones. Causality is typically enforced by applying a masking operation to the attention matrix, ensuring that  $\mathbf{QK}^\top$  entries associated with future positions are set to  $-\infty$  before the softmax is applied. This restricts the flow of information and prevents “peeking” into the future, a property critical for tasks such as language modeling or time series prediction. In contrast, non-causal (or bidirectional) attention imposes no such constraints, allowing each element in the sequence to attend to both past and future elements in the sequence, which can be advantageous for tasks like machine translation or masked language modeling where full contextual information is available.

Following the attention layer, a FFN is applied to each position independently. The FFN typically consists of two linear transformations with a non-linear activation function, such as ReLU, in between:

$$\text{FFN}(\mathbf{Z}) = \max(0, \mathbf{Z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \quad (22)$$

where  $\mathbf{Z} \in \mathbb{R}^{n \times d}$  is the output of the multi-head attention module, and  $\mathbf{W}_1 \in \mathbb{R}^{d \times d_f}$ ,  $\mathbf{W}_2 \in \mathbb{R}^{d_f \times d}$ ,  $\mathbf{b}_1 \in \mathbb{R}^{d_f}$ , and  $\mathbf{b}_2 \in \mathbb{R}^d$  are learnable parameters. Here  $d_f$  is often chosen to be larger than  $d$  to increase representational capacity. As with the attention layer, the FFN output is combined with a residual connection and layer normalization. Thus, each Transformer block systematically refines the hidden representations through attention-based context aggregation and non-linear transformations.

## D.2 BERT

BERT is a widely-used Transformer-based model for natural language understanding tasks. Introduced by [Devlin et al. \(2018\)](#), BERT leverages a deep stack of Transformer encoders to produce rich, contextualized embeddings of input tokens. Unlike previous popular models that relied on a unidirectional context, or did not use an attention mechanism at all ([Mikolov et al., 2013](#); [Pennington et al., 2014](#)), BERT employs bidirectional attention, allowing each token to attend to all other tokens in both directions. This property is achieved through a masked language modeling pre-training objective, where a certain percentage of tokens are masked and the model learns to predict these masked tokens from their surrounding context.

One of the primary strengths of BERT-like models is that they are provided pre-trained on large-scale text corpus, but can be fine-tuned for a broad range of downstream tasks with minimal architectural modification, often achieving state-of-the-art or near state-of-the-art results.<sup>15</sup> This transfer learning capability stems from the generality and richness

<sup>15</sup>BERT was originally trained on a corpus which included the English Wikipedia (approximately 2.5 billion

of the pre-trained token representations, allowing developers and researchers to focus on task-specific data without the need for extensive task-specific model architectures. As a result, BERT has seen widespread adoption and has significantly influenced the direction and methodologies of NLP research.

BERT is provided in several model sizes to accommodate different resource constraints and performance targets. For example, BERT<sub>BASE</sub> consists of 12 Transformer encoder layers, each with 12 attention heads and a hidden size of 768 dimensions (totaling roughly 110 million parameters). BERT<sub>LARGE</sub>, in contrast, uses 24 layers, 16 attention heads, and a hidden size of 1024, leading to about 340 million parameters. These models can be further adapted or distilled into smaller variants, enabling more efficient deployment on resource-limited devices.

Here we utilize BERT<sub>TINY</sub> (Turc et al., 2019), which present a family of more compact BERT models that trade off model size and computation for slightly reduced accuracy, allowing for more efficient deployments while still maintaining strong performance on many NLP tasks. In particular, the BERT<sub>TINY</sub> version we use consists of only 2 Transformer encoder layers, each with 4 attention heads and a hidden size of 128 dimensions (totaling roughly 4 million parameters).

In terms of processing a text sequence, BERT<sub>TINY</sub> first uses the same WordPiece tokenizer as the original BERT model.<sup>16</sup> Let a given input sequence consist of  $n$  tokens:  $(w_1, w_2, \dots, w_n)$ , where each token  $w_i$  is drawn from a fixed vocabulary  $V$ . Then BERT first maps these tokens into a sequence of token embeddings  $\mathbf{E} \in \mathbb{R}^{n \times d}$ , where  $d$  is the hidden dimension and  $\mathbf{E}$  is a learnable parameter matrix. In addition to token embeddings, BERT uses learnable position embeddings  $\mathbf{P} \in \mathbb{R}^{n \times d}$  to encode the positional information of each token. Thus, each token  $w_i$  is represented as:

$$\mathbf{x}_i = \mathbf{e}_i + \mathbf{p}_i,$$

where  $\mathbf{e}_i$  and  $\mathbf{p}_i$  are the  $i^{th}$  row vectors of  $\mathbf{E}$  and  $\mathbf{P}$ , respectively. Stacking these vectors for all  $i \in \{1, \dots, n\}$  results in the input embedding matrix  $\mathbf{X} \in \mathbb{R}^{n \times d}$ , which serves as the input to the first Transformer encoder layer of BERT.

To further prepare the input, and facilitate batch-processing, each input sequence of length  $n$  is typically padded to a fixed length  $N \geq n$  by appending special padding tokens. A corresponding binary attention mask is then defined and ensures that the model only attends to non-padding tokens. In BERT<sub>TINY</sub>  $N = 512$  is the maximum sequence length, words) and the BookCorpus (approximately 800 million words).

<sup>16</sup>This tokenizer splits input text into subword units based on statistical properties derived from large-scale text corpora. By applying the same WordPiece vocabulary and tokenization strategy, BERT<sub>TINY</sub> maintains compatibility with BERT’s input format and leverages the same robust subword segmentation to effectively handle a broad variety of words, including rare and out-of-vocabulary terms.



and sequences that are longer than this will automatically be truncated. In addition to padding and masking, BERT also incorporates special tokens into the input sequences. For a single sequence or a pair of sentences, BERT prepends a  $[CLS]$  token at the start of the sequence and inserts a  $[SEP]$  token after each sentence. The token  $[CLS]$  provides a convenient vector representation for the entire input sequence, and each  $[SEP]$  helps the model distinguish between sentence boundaries. For our purpose, the former  $[CLS]$  token is important because it facilitates sequence classification.

### D.3 Transformers for time series

Originally developed for natural language processing, the Transformer architecture has increasingly been applied to time series modeling due to its ability to capture long-range dependencies without relying on recurrence. Indeed, by leveraging self-attention mechanisms, Transformer-based models have demonstrated strong performance across a variety of time series tasks (Wen et al., 2023).

Key challenges in applying Transformers to time series include encoding temporal structure and handling multivariate data. For the former, a common approach, adopted here, is to treat time steps as discrete tokens and apply either fixed or learned positional embeddings, analogous to token position encoding in NLP models. In particular, let  $\mathbf{x}_{t_{ts}}^{ts} \in \mathbb{R}^D$  denote a  $D$ -dimensional vector of macroeconomic variables at time  $t_{ts} \in \{1, \dots, T_{ts}\}$ . Each vector is then projected into a  $d$ -dimensional embedding space via:

$$\mathbf{e}_{t_{ts}}^{ts} = \mathbf{W}_{ts} \mathbf{x}_{t_{ts}}^{ts} + \mathbf{b}_{ts}, \quad (23)$$

where  $\mathbf{W}_{ts} \in \mathbb{R}^{d \times D}$  and  $\mathbf{b}_{ts} \in \mathbb{R}^d$  are learned parameters. The resulting embeddings are stacked into  $\mathbf{E}^{ts} \in \mathbb{R}^{T_{ts} \times d}$  and serve as the input sequence to the Transformer, with positional encodings added in the usual manner.<sup>17</sup>

<sup>17</sup>To improve temporal awareness, additional features, such as day of the week, month, or other seasonal indicators, can be incorporated into the embeddings, either through addition or concatenation (Lim et al., 2021). In settings with long sequences, "patching" techniques, inspired by computer vision, can also be employed. Here, consecutive time steps are grouped into segments (patches), reducing the sequence length seen by the model while preserving intra-patch dependencies. We leave exploring the adequacy of these model designs for macroeconomic modeling for future research.

# Appendix E Reinforcement Learning for Narrative Attribution

## E.1 Connection to optimal control

Formally, the out-of-sample narrative attribution problem can be cast as a finite-horizon Markov decision process with state  $s_t$ , action  $a_t \in \mathcal{A}_t$ , and reward  $r_t(s_t, a)$  given by (12). Full RL with optimal control would optimize long-horizon returns, but with  $N_T \gg 2000$  this is infeasible. We therefore adopt an approximation with short rollouts of horizon  $H$ , inspired by classical work in stochastic control (Bertsekas and Tsitsiklis, 1996; Bertsekas, 2019). A basic (hard) rollout estimator for the state-action value is:

$$\widehat{Q}_H(s_T, a) := \mathbb{E} \left[ \sum_{h=0}^{H-1} \gamma^h r_{T+h}(s_{T+h}, a_{T+h}) \mid a_T = a \right], \quad (24)$$

with discount factor  $\gamma$ .

Standard optimal control relies on the *hard* Bellman recursion:

$$Q^*(s, a) = r(s, a) + \gamma \mathbb{E}_{s'} \left[ \max_{a'} Q^*(s', a') \right].$$

However, in high-dimensional settings the max operator easily induce brittle policies and poor exploration. Entropy-regularized or “soft” control instead augments reward with an entropy or KL-to-prior bonus (Todorov, 2006; Kappen, 2005; Ziebart, 2010; Haarnoja et al., 2018), yielding:

$$Q^*(s, a) = r(s, a) + \gamma \mathbb{E}_{s'} [V^*(s')], \quad (25)$$

$$V^*(s) = \tau \log \sum_{a'} \exp \left\{ \frac{1}{\tau} Q^*(s, a') \right\}, \quad (26)$$

with temperature  $\tau > 0$ . The induced soft-optimal policy is:

$$\pi^*(a \mid s) = \frac{\exp \left\{ \frac{1}{\tau} Q^*(s, a) \right\}}{\sum_{a'} \exp \left\{ \frac{1}{\tau} Q^*(s, a') \right\}}, \quad (27)$$

which recovers the hard Bellman policy as  $\tau \downarrow 0$ .

To align rollout targets with (27), we map truncated returns to probabilities via a Boltzmann transform:

$$q_T(a) \propto \exp \left\{ \frac{1}{\tau_r} \widehat{Q}_H(s_T, a) \right\}. \quad (28)$$

Equation (28) is the finite-horizon, rollout-based analogue of (27), replacing the exact  $Q^*$  with the  $H$ -step estimator (24).

Training then projects the policy onto these rollout targets using the KL-regularized objective (15), in line with trust-region methods (Peters et al., 2010; Schulman et al.,

**Table E.1.** Hyperparameters and baseline values used in training.

| Hyperparameter                 | Baseline value      | Hyperparameter                 | Baseline value  |
|--------------------------------|---------------------|--------------------------------|-----------------|
| Learning rate ( $\eta$ )       | $5 \times 10^{-3}$  | Discount factor ( $\gamma$ )   | 0.97            |
| Softmax inv. temp. ( $\beta$ ) | 1.6                 | Rollout horizon ( $H$ )        | 3               |
| Target temp. ( $\tau_r$ )      | 0.8                 | Entropy weight ( $\lambda_H$ ) | 0.01            |
| FUND weight ( $\lambda_F$ )    | 0.8                 | KL weight ( $\lambda_{KL}$ )   | 0.1             |
| NPRIOR weight ( $\lambda_N$ )  | 0.1                 | Steps of training              | 250             |
| PER weight ( $\lambda_P$ )     | 0.1                 | Refresh interval               | 5 iterations    |
| Continuation strategy          | deterministic (max) | Eval. metric                   | expected reward |

2015; Abdolmaleki et al., 2018). With  $q_{t_s}$  of the form (28), the optimizer of (15) is driven toward the soft-optimal policy (27), with  $\pi_0$  acting as a stabilizing prior and  $\lambda_H$  controlling entropy regularization.

Since  $\widehat{Q}_H$  is truncated, noisy, and expensive to recompute for every  $(s, a)$  and period  $T$ , we use a parameterized softmax policy  $\pi_\theta(a | s) = \text{softmax}_a\{\beta \phi(s, a)^\top \theta\}$  as a compact approximation class. This projects noisy rollout targets onto a smooth policy, generalizes across states via features, and amortizes rollout costs so that deployment does not require recomputing  $\widehat{Q}_H$ . If the class is sufficiently rich, the projection recovers (27) exactly.

## E.2 Application details and practical considerations

The training algorithm is outlined in Algorithm 1. We adopt a two-timescale training scheme common in RL: policy parameters are updated every iteration, while the environment is refreshed and new state caches are generated only every few iterations. This setup, often referred to as using multiple epochs per batch or semi-on-policy training, reduces computational cost while still providing stable learning. Similar strategies are standard in policy gradient and fitted policy iteration methods.

Table E.1 lists the parameters used for estimation. The temperature settings are tuned to avoid large jumps in the learning curves while the weights in the objective function (15) allow for a small degree of smoothing and conservatism. The weights in the reward function (12) reflect user specific preferences, and do in our benchmark application reflect a high weight on fundamental news and relatively little weight on the narrative prior and persistence.

We implement policy learning on slates of candidate articles rather than on the full set of available texts. This eases the computational burden: each period typically contains thousands of articles, making it intractable to evaluate every candidate in each policy update. By restricting attention to a smaller, representative slate - obtained through

---

**Algorithm 1** Reinforcement Learning for Narrative Attribution

---

**Require:** Dataset  $\{(\mathcal{A}_{t_s}, \mathbf{X}_{t_s}^{ts})\}_{t_s=1}^{T_s}$  and for  $t_s = 1$  initial states of  $(\mathbf{s}_{1:T-1}, \mathbf{x}_{1:T_f-1}^f)$

- 1: Initialize policy parameters  $(\theta)$ .
- 2: **for** iter = 1, 2, ... **do**
- 3:   **if** iter **mod** 5 = 0 **then**
- 4:     Build a deterministic prefix trajectory  $\{s_{t_s}\}_{t_s=1}^{T_s}$ , with  $s_{t_s} = (\mathbf{s}_T, \mathbf{x}_{T_s}^{ts}, \mathbf{x}_{T_f}^f)$ , by rolling forward with  $\arg \max_a \pi_\theta(a \mid s_{t_s})$ .
- 5:     **for**  $t_s = 1$  **to**  $T_s$  **do**
- 6:       Compute approximate action values by  $H$ -step rollouts:

$$\hat{Q}_{t_s}(a) \leftarrow \text{ROLLOUT}(s_{t_s}, a, \gamma, H; \pi_\theta).$$

- 7:       Form target distribution:

$$q_{t_s}(a) \propto \exp\{\hat{Q}_{t_s}(a)/\tau_r\}.$$

- 8:     **end for**
- 9:   **end if**
- 10:   Update  $\theta$  by one step of (stochastic) gradient descent on:

$$\mathcal{L}(\theta) = \sum_{t_s=1}^{T_s} \text{KL}(q_{t_s} \parallel \pi_\theta(\cdot \mid s_{t_s})) + \lambda_{\text{KL}} \text{KL}(\pi_\theta(\cdot \mid s_{t_s}) \parallel \pi_0(\cdot \mid s_{t_s})) + \lambda_H \mathcal{H}(\pi_\theta(\cdot \mid s_{t_s})).$$

- 11:   **Convergence check:** stop when  $\mathcal{L}$  or validation score stabilizes.
  - 12: **end for**
  - 13: **return** trained policy  $\pi_\theta(a \mid s_{t_s}) \propto \exp\{\beta(\phi(s_{t_s}, a)^\top \theta)\}$ .
- 

cheap pre-ranking heuristics using the *Narrater-greedy* approach - we can train efficiently while still exposing the policy to diverse alternatives. Here we consider 100 articles in each time period. 10% of these are among the best candidates according to the pre-ranking heuristics and the rest are randomly selected from the full set of candidates. Learning convergence is monitored by evaluating the policy function using expected rewards on fixed slates - keeping the best performing parameter set for later application.

In the reward function accuracy is defined as:

$$\text{RMSE}(G\hat{D}P_{1:T_f}^{(i)}, GDP_{1:T_f}) = \sqrt{\frac{1}{T_f} \sum_{t_f=1}^{T_f} (G\hat{D}P_{1:T_f}^{(i)} - GDP_{1:T_f})^2},$$

where  $G\hat{D}P_{1:T_f}^{(i)}$  is the implied growth prediction when using article  $a_{T,i}$  as input to the

filter in period  $T$ . The narrative prior is measured using the ROUGE-1 score (Lin, 2004):

$$\text{NPRIOR}(a, \tilde{a}_T) = \frac{2 \cdot \sum_{w \in V} \min(\text{Count}_a(w), \text{Count}_{\tilde{a}}(w))}{|a| + |\tilde{a}|},$$

where  $|a|$  and  $|\tilde{a}|$  denote the total number of unigrams in the candidate and reference article, respectively. We further define the degree to which the candidate article is fundamental as:

$$\text{FUND}(a) = \sigma((p_{fund}(a) - p_{noise}(a))/0.5),$$

where  $p_{fund} = \sum_{c \in (\text{demand}, \text{supply})} p(c|a)$  and  $\sigma()$  is the sigmoid function. Thus,  $\text{FUND}(a)$  favors articles with higher probability scores for the demand and supply classes. Finally, we use cosine similarity to measure:

$$\text{PER}(a, \mathcal{H}_{T-1}) = \frac{1}{J} \sum_{j=1}^J \cos(\bar{e}(a), \bar{h}_j)$$

where  $\bar{e}(a)$  is the sequence embedding from (1) in period  $T$  for decision point  $t_s$ , and  $\bar{h}_j$  is the associated embedding for the  $t_s - j$  periods earlier with  $J = 4$ . To further avoid getting stuck in repeating noise, we gate the  $\text{PER}(a, \mathcal{H}_{T-1})$  measure by interacting it with  $\text{FUND}(a)$  before computing the aggregate reward.

## Appendix F LLM prompt

I would like your help in forecasting the quarterly business cycle (F) for Norway alongside a more narrative account of this  
 $\hookrightarrow$  prediction. I will provide you with historical data consisting of 38 quarterly values of structural supply,  
 $\hookrightarrow$  demand, and noise series in a CSV file titled `timeSeriesF_{number}.csv`. These values are ordered chronologically from  
 $\hookrightarrow$  earliest to latest, with the first and latest values representing time index 2 and 39, respectively. For each quarter, the  
 $\hookrightarrow$  sum of the supply and demand series correspond to the business cycle indicator. Your task is to predict the business cycle  
 $\hookrightarrow$  indicator (F) for the next quarter (the 40th time index).

Additionally, I will supply you with data on 8 macroeconomic variables, consisting of 39 quarterly values, in a separate CSV  
 $\hookrightarrow$  file, `timeSeriesX_{number}.csv`. These variables collectively form the basis for computing the business cycle indicator (F)  
 $\hookrightarrow$  for time index 1 to 39.

You will also be provided with 39 recent news stories as text data, formatted as a list of strings in a CSV file,  
 $\hookrightarrow$  `text_{num}.csv`. Each news story is associated with one quarter, ranging from time index 2 to 40, and can assist you in  
 $\hookrightarrow$  inferring the historical correlation patterns as well as making your business cycle prediction for time index 40. The  
 $\hookrightarrow$  class of each text is either `**supply**`, `**demand**`, or `**noise**`, and there is statistical mapping between the  
 $\hookrightarrow$  structural three-part decomposition of the values in `timeSeriesF_{number}.csv` and the classes and sentiment of the texts.  
 $\hookrightarrow$  To help you understand this mapping, the true class and sentiment for each of the texts for time index 2 to 39 is provided  
 $\hookrightarrow$  in the file `classSentiment_{number}.csv`. These values are ordered chronologically from earliest to latest, with the 39th  
 $\hookrightarrow$  value representing the most recent quarter for which we have data. The classes are identified using numbers, with  
 $\hookrightarrow$  demand=1, supply=2, and noise=3.

Alongside predicting the business cycle indicator (F), please also provide:

1. An economic sentiment prediction (S) for the 40th quarter, expressed as a value between -1 and 1, reflecting the overall  
 $\hookrightarrow$  economic sentiment which in this context is highly correlated with the business cycle indicator.
2. A classification (C) of the primary driving force behind the economy for the 40th quarter, categorized as one of the  
 $\hookrightarrow$  following:  
`**Supply**`: Economy driven primarily by supply-side factors, defined here as residing in the semantic territory of  
 $\hookrightarrow$  "technology, innovation, productivity".  
`**Demand**`: Economy driven primarily by demand-side factors, defined here as residing in the semantic territory of "demand,  
 $\hookrightarrow$  preferences, sentiment".  
`**Noise**`: No specific supply or demand shocks dominate the economic conditions this quarter.

Your predictions should be informed by the relationship implicit in the historical data among macroeconomic variables, news  
 $\hookrightarrow$  stories, and the business cycle indicator.

Please provide your predictions structured clearly in the following JSON format:

```
```json
{
  "F": <predicted business cycle indicator value>,
  "S": <predicted economic sentiment value between -1 and 1>,
  "C": "Supply" | "Demand" | "Noise"
}
```