Investigating Hallucinations of Time Series Foundation Models through Signal Subspace Analysis

Yufeng Zou[†] **Zijian Wang**[§] **Diego Klabjan**[‡] **Han Liu**[†‡♮]
[†]Department of Computer Science, [‡]Department of Industrial Engineering
and Management Sciences, [♮]Department of Statistics and Data Science, Northwestern University
[§]School of Computer Science, The University of Sydney
yufeng.zou@u.northwestern.edu
zwan0998@uni.sydney.edu.au
{d-klabjan, hanliu}@northwestern.edu

Abstract

Times series foundation models (TSFMs) have emerged as a promising paradigm for time series analysis and forecasting, showing remarkable generalization performance across different domains. While efforts have been made on hallucinations of foundation models, the hallucinations of TSFMs have been underexplored. In this paper, we formally define TSFM hallucinations in the zero-shot forecasting setting by examining whether a generated forecast exhibits different dynamics from those of the context. Our study reveals that TSFM hallucinations primarily stem from the loss of context information in hidden states during forward propagation. As such, we propose methodologies to identify signal subspaces in TSFMs and magnify their information through intervention. Extensive experiments demonstrate that our proposed intervention approach effectively mitigates hallucinations and improves forecasting performance. Furthermore, the signal strength measure we compute from signal subspaces has strong predictive power of hallucinations and forecasting performance of the model. Our work contributes to deeper understanding of TSFM trustworthiness that could foster future research in this direction.

1 Introduction

Times series analysis is a major research field that facilitates decision making and scientific inference across a broad range of domains, from energy and weather to economy, transport, and system management. As a key task, time series forecasting has motivated the development of distinct approaches including statistical [6, 32] and deep learning [28, 40] models. Despite competitive performance for specific tasks, these models are typically trained on a single domain, without sufficient capability to generalize to different domains. Inspired by the success of foundation models in fields like natural language processing [1, 7], time series foundation models (TSFMs) have recently emerged as a new paradigm towards universal forecasters [4, 8, 11, 16, 24, 30, 39]. Pretrained on large-scale time series data, TSFMs have shown remarkable few-shot and even zero-shot forecasting performance across numerous domains [3, 18], substantially reducing the need for downstream data. The hidden representations of TSFMs are also useful for downstream tasks through the extraction of context time series information.

Yet, the performance of TSFMs is often plagued by hallucinations, as with other foundation models. Broadly referring to the generation of unsupported statements or nonsensical content, hallucinations are attributed to incorrect knowledge or insufficient inference capability of a model [19]. Among various hallucination detection and mitigation approaches proposed, mechanistic interpretability and

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

test-time intervention require no additional training and have demonstrated effectiveness for large language models (LLMs) [22, 25, 31, 43] and large vision-language models (LVLMs) [23, 42].

In zero-shot forecasting, where a TSFM is tasked with generating extrapolations based on the extracted information of context time series such as trends, periodicity, and patterns [18], accurate processing of the context information is essential for generating high-quality forecasts. As such, we study TSFM hallucinations by examining whether a forecast exhibits drastically different dynamics from those of the context, e.g., Figure 1 (a) versus (b). We investigate the underlying mechanisms of TSFM hallucinations through the lens of hidden representations and develop a novel intervention approach to address the identified causes. As far as we know, limited effort has been made on similar research problems in the existing literature. We strive to address these knowledge gaps and contribute to deeper understanding of TSFM trustworthiness, which could foster future research in this direction.

We formally define TSFM hallucinations in the zero-shot forecasting setting in §3 and outline the knowledge rules for checking hallucinations in practice. In §4.1, we gain insights on TSFM hallucinations through experimental analyses, where we find that hallucinations are mainly caused by a lack of context information in hidden states during forward propagation. We then propose a methodology to identify the signal spaces along with a measure (SSAS) to quantify the signal strength of hidden states in §4.2. Built upon these results, we propose a novel intervention approach (SSIM) that mitigates hallucinations by magnifying the signal information of hidden states in §4.3. Extensive experiments in §5.2 demonstrate that while the forecasting performance of TSFMs suffers from hallucinations, our test-time intervention effectively mitigates hallucinations and improves the quality of forecasts, yielding up to 6.62% reduction in hallucination rate, 93.83% gain in R^2, and 13.52% gain in correlation. Moreover, the signal strength measure we propose has strong predictive power of both hallucinations and forecasting performance of TSFMs.

Our main contributions in this work are as follows. (1) We formally define the problem of TSFM hallucinations and outline a set of procedures to check hallucinations. We are the first to systematically study this problem to our best knowledge. (2) We propose a methodology to identify the signal subspaces in TSFMs along with a measure to quantify the signal strength in TSFM hidden states. (3) We propose a simple and efficient intervention approach to mitigate hallucinations by magnifying the signal information in TSFM hidden states. (4) We conduct extensive experiments on synthetic and real-world datasets across various domains to examine the impact of TSFM hallucinations and demonstrate the effectiveness of our proposed signal strength measure and test-time intervention.

2 Related Work

Times series foundation models. TSFMs represent a promising paradigm towards generalization across different time series domains and tasks leveraging the knowledge from large-scale pretraining data [4, 8, 11, 16, 24, 30, 39]. TSFMs not only substantially reduce the need for downstream data but have also shown capabilities of producing accurate forecasts even in zero-shot scenarios when forecasting on inputs from unseen domains [3, 18]. While most TSFMs are Transformer based [33] and open sourced, they are diverse in architectural design, tokenization strategies, and pretraining objectives. For instance, Chronos [4] and Chronos-Bolt adopt an encoder-decoder architecture, while TimesFM [11] is decoder-only. Chronos-Bolt and TimesFM truncate the normalized time series inputs into patches, while Chronos discretely quantizes the scaled inputs into a fixed vocabulary. Yet, the forecasting performance of TSFMs suffers from hallucinations when they fail to adequately capture the signal information from inputs, an issue which we study on models from both families.

Hallucinations. Hallucination, defined as the generation of unfaithful or nonsensical content, is a fundamental challenge in large foundation models due to their black-box nature [19]. Recent research has examined model hidden representations for hallucination detection and mitigation, based on the hypothesis that factual knowledge is encoded in these states [10, 12, 15]. Studies have identified diagnostic signals in hidden states, showing that outlier or inconsistent activation patterns during generation can indicate potential hallucinations [2, 9, 13, 31, 36]. Complementary approaches focus on hidden state manipulation, demonstrating that truthfulness can be elicited through targeted neuron activation interventions, offering promising directions for reducing hallucinations [22, 23, 31, 41, 42]. We are the first to formally define and systematically study hallucinations of time series foundation models. We develop methodologies to detect and mitigate TSFM hallucinations.

Intervention. Hidden state intervention has emerged as a powerful technique for controlling neural models' behaviors, as these internal representations serve as causal factors influencing model outputs. Research in [21, 23, 43] demonstrates effective control over model outputs through activation steering, which identifies linear-interpretable directions in the representation space and guides hidden states along these pathways. Some achieve model output modification by selectively masking specific neuron activations, preventing corresponding generations from occurring [10, 25, 29, 35]. The approach in [38] alters TSFM outputs but does not address specific challenges of time series forecasting. Differently, we propose a novel intervention approach specifically for TSFM hallucination mitigation, which is context adaptive and selectively intervenes model layers.

3 Definitions and Preliminaries

Formally, we first describe the forecasts of a time series foundation model and the problem of hallucinations.

Definition 1 (**TSFM forecasts**). A pretrained time series foundation model, denoted as $\mathcal{M}_\theta$, takes a time series $\boldsymbol{x}_{context} = [x_1, \ldots, x_p]$ of context length p as the input and generates a forecast $\hat{\boldsymbol{x}} = \mathcal{M}_\theta(\boldsymbol{x}_{context}) = [\hat{x}_{p+1}, \ldots, \hat{x}_{p+q}]$ of horizon q. For an L-layer time series foundation model, we denote the hidden states at different positions of layer l (the outputs of the layer) as a matrix $\boldsymbol{H}^{(l)} = [\boldsymbol{h}_1^{(l)}, \ldots, \boldsymbol{h}_n^{(l)}] \in \mathbb{R}^{n \times d}$, where d is the dimension of hidden states.

Definition 2 (**Time series forecasting hallucinations**). Suppose for a time series $\boldsymbol{x}_{full} = [x_1, \ldots, x_T]$, a knowledge set $\mathbb{K}$ can be inferred from a partial time series $\boldsymbol{x}_{context} = [x_i, \ldots, x_j]$, $1 \leq i < j < T$ for any i and j. The knowledge set $\mathbb{K}$ comprises time-dependent knowledge rules r that hold true for $\boldsymbol{x}_{full}$, i.e., $r(x_i, i) = 1$ for all $1 \leq i \leq T$, or simply $r(\boldsymbol{x}_{full}) = 1$. In zero-shot time series forecasting, we consider a hallucination to be a forecast that does not conform to the knowledge rules inferred from the context time series. We define the set of hallucinations as $Hallu(\boldsymbol{x}_{context}) = \{\hat{\boldsymbol{x}} : \bigwedge_{r \in \mathbb{K}} r(\hat{\boldsymbol{x}}) = 0\}$. The goal of hallucination detection is to define a score function f that discriminates hallucinated forecasts of the model, such that for any $\hat{\boldsymbol{x}} = \mathcal{M}_\theta(\boldsymbol{x}_{context})$, we have $f(\boldsymbol{x}_{context}, \hat{\boldsymbol{x}}, \theta) > \tau$ if an only if $\hat{\boldsymbol{x}} \in Hallu(\boldsymbol{x}_{context})$. We mitigate hallucinations through test-time intervention on hidden states so that with the intervention operation $\mathcal{I}$, we obtain non-hallucinated forecasts $\mathcal{M}_{\theta,\mathcal{I}}(\boldsymbol{x}_{context}) \notin Hallu(\boldsymbol{x}_{context})$.

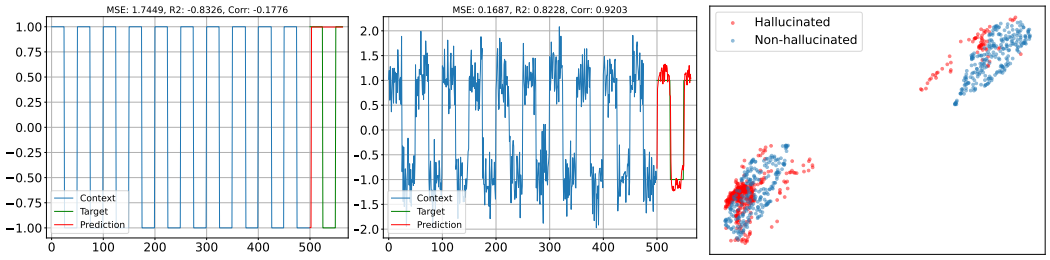
In practice, we sequentially extract a set of knowledge rules $\mathbb{K} = \{r_1, \ldots, r_n\}$ from the context time series to check whether a forecast is hallucinated.

Trend. The trend rule checks whether the trend of the forecast conforms to those of the context. We perform ordinary least-square (OLS) regression on $\hat{\boldsymbol{x}}$ and take the first-degree coefficient $\hat{c}$ as the trend if it is significant with the p-value < 0.01. We then perform OLS on rolling windows of $\boldsymbol{x}_{context}$ and take significant trends $[c_1, \ldots, c_n]$. With the relative difference between trends computed as $diff(c, \hat{c}) = \left|\frac{\hat{c}}{c} - 1\right|$, the trend rule is satisfied iff the minimum relative difference $\min_i diff(c_i, \hat{c}) < \delta$, or neither the forecast nor the context exhibits a significant trend.

Frequency. The frequency rule checks whether the spectral density of the forecast conforms to those of the context. After removing the trend, we compute the spectral densities $[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n]$ of rolling windows on $\boldsymbol{x}_{context}$ using short-time Fourier transform (STFT) [17] and also the spectral density $\hat{\boldsymbol{f}}$ of $\hat{\boldsymbol{x}}$. With the Jaccard distance between spectral densities computed as $\mathcal{D}(\boldsymbol{f}, \hat{\boldsymbol{f}}) = 1 - \frac{\sum_i \min\{f_i, \hat{f}_i\}}{\sum_i \max\{f_i, \hat{f}_i\}}$, the frequency rule is satisfied iff the minimum Jaccard distance $\min_j \mathcal{D}(\boldsymbol{f}_j, \hat{\boldsymbol{f}}) < \delta$.

Pattern. The pattern rule checks whether the pattern of the forecast is similar to those of the context. After removing the trend, we compute the relative absolute errors between the forecast and rolling windows $[\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n]$ on $\boldsymbol{x}_{context}$. With the relative absolute error computed as $RAE(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{\sum_i |x_i - \hat{x}_i|}{\sum_i |x_i - \bar{x}|}$, the pattern rule is satisfied iff the minimum relative error $\min_j RAE(\boldsymbol{w}_j, \hat{\boldsymbol{x}}) < \delta$.

ARMA. The ARMA rule checks whether the ARMA dynamics of the forecast conform to those of the context, which complements the pattern rule since a time series that exhibits strong ARMA dynamics may not have distinct patterns. After removing the trend, we fit a first-order Autoregressive



(a) Hallucinated (b) Non-hallucinated (c) UMAP visualization

Figure 1: (a) (b) Examples of hallucinated and non-hallucinated forecasts by `Chronos`. (c) UMAP visualization of hidden states at the final model layer for both examples.

moving average (ARMA) model [6] on $\boldsymbol{x}_{context}$ and take the AR and MA coefficients ϕ and ψ if both are significant with p-values less than 0.01. Let $\hat{\phi}$ and $\hat{\psi}$ be the first-order ARMA coefficients on $\hat{\boldsymbol{x}}$, the ARMA rule is satisfied iff the relative differences $\left|\frac{\hat{\phi}}{\phi} - 1\right| < \delta$ and $\left|\frac{\hat{\psi}}{\psi} - 1\right| < \delta$.

A TSFM forecast that violates either the trend, frequency, or both the pattern and ARMA rules is considered to be hallucinated, since it is ungrounded in the context time series information. Further details are provided in Appendix §A and §D.3.

4 Methodology

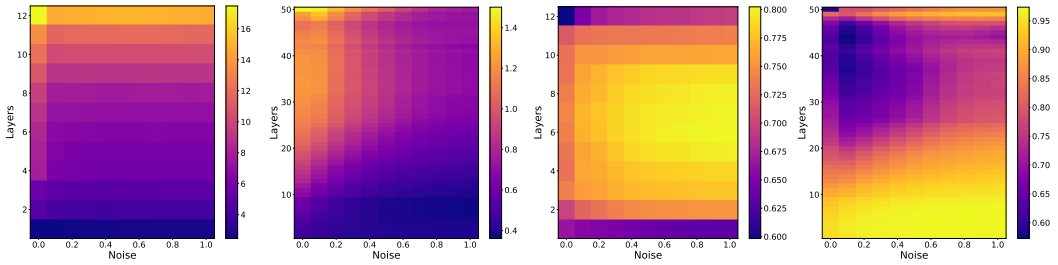
To understand the cause of TSFM hallucinations, we first gain insights from observations, provide intuitive explanations, and then perform experimental analyses to justify our claims. Afterwards, we propose a signal strength measure to help detect hallucinations. Finally, we develop a novel test-time intervention approach that mitigates hallucinations by addressing the identified causes.

4.1 Observations and Analyses

We begin with a brief case analysis. Figure 1 (a) presents a hallucination example where a TSFM fails to generate a forecast consistent with the context time series. The UMAP [26] visualization of hidden states at the final layer in Figure 1 (c) shows irregular patterns, with a high mean pairwise cosine similarity at 0.8763 and a low mean activation std across positions at 11.0570. We speculate that the forecast failure is caused by the loss of context information in hidden states during forward propagation. We find that injecting a small amount of random perturbation to the context time series with Gaussian noise alleviates such information loss, as shown in Figure 1 (b). We observe that the hidden states get more evenly distributed in each cluster of Figure 1 (c), with the mean pairwise similarity substantially reduced to 0.6338 and the mean activation std raised to 14.5772.

To analyze the effects of context signal and noise on a TSFM when no hallucination occurs, we collect the activations over 10 random perturbations to context signal with Gaussian noise of varying magnitudes. From Figure 2 (a)(b), we observe that hidden state activations exhibit the greatest variance across the positions of a layer with clean signal. As the signal gets mixed with more noise, despite the increase of input variance, hidden state activations become less variant. The decline in mean activation variance with noise is more salient at higher layers, suggesting that a TSFM incrementally extracts signals and reduces noises from the input at each layer.

Based on this, we posit that the hidden state space $\mathbb{H}^{(l)}$ at each TSFM layer can be decomposed into signal and noise subspaces $\mathbb{H}^{(l)} = \mathbb{S}^{(l)} \oplus \mathbb{N}^{(l)} \subset \mathbb{R}^d$, respectively handling the signals and noises of the input [14, 20, 27]. In forward model propagation $\boldsymbol{H}^{(1)} \rightarrow \ldots \rightarrow \boldsymbol{H}^{(L)}$, the signal components of a hidden state $\Pi_{\mathbb{S}^{(l)}} \boldsymbol{h}^{(l)}$ are further processed by subsequent layers, while the noise components $\Pi_{\mathbb{N}^{(l)}} \boldsymbol{h}^{(l)}$ get repressed and eventually removed. Since the signal components are more variant and dissimilar across hidden state positions than the noise components, the hidden states would exhibit greater distinctiveness across positions with strong signal presence at a layer (Figure 2 (c)(d)).



(a) `Chronos` (b) `TimesFM` (c) `Chronos` (d) `TimesFM`

Figure 2: (a) (b) The mean standard deviations of hidden state activations across the positions of a layer under varying noise magnitudes. (c) (d) The mean pairwise cosine similarities of hidden states across the positions of a layer. The x-axis represents the standard deviation of Gaussian noise.

Back to the previous case of hallucinated forecasts, the inactivity of signal subspaces of the model leads to homogeneous hidden states across positions. In such case, a proper amount of input random perturbation injects variances that help restore the activity level of signal subspaces and facilitate the propagation of context signal information. Nonetheless, it is hard to determine the optimal amount of perturbation, since too much perturbation obscures the input signal and degrades forecast quality. Moreover, a single perturbation is not robust [23], while performing multiple perturbations hampers efficiency. Our goal is to magnify the signal information in hidden states through test-time intervention, which would enable us to mitigate hallucinations in a controllable and efficient manner.

4.2 Signal Subspace Identification

Now, we develop a novel methodology to identify the signal subspaces in TSFM layers and provide an empirical analysis. We aim to identify a set of hidden state neurons that are most active to context signals by examining the variance of activations across hidden state positions, enlightened by the associations between the activation variance and signal strength we observe in the previous subsection. We compute the activity score of the j-th neuron at layer l given a context input $\boldsymbol{x}$ as:

$$\mathcal{A}^{(l)}(j \mid \boldsymbol{x}) = \sqrt{\frac{1}{n}\sum_{i}(\boldsymbol{H}_{i,j}^{(l)} - \bar{\boldsymbol{h}}_{j}^{(l)})^2} \ . \tag{1}$$

The neuron activity measure we propose is more nuanced compared with the magnitude of neuron activations used in prior works [34, 35], as it not only reflects the overall magnitude but also measures the deviation of neuron activations across time series steps.

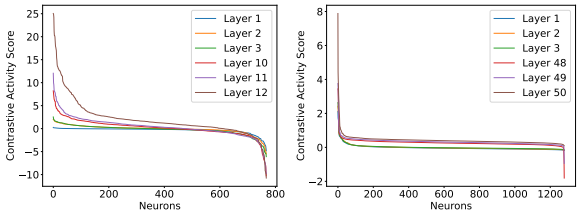
To measure neuron activity in the presence of signals, we collect the activity scores on a synthetic dataset comprising common waveforms that will be described in §5.1. We also vary the magnitude of noises injected to the context signals and initialize them with different random seeds for robustness. With $\mathcal{D}_{signal}$ denoting the set of synthetic time series inputs where no hallucination occurs, we consider neurons with the activity score consistently top ranked across the samples as candidate signal neurons, i.e., $Cand(l) = \bigcap_{\boldsymbol{x}\in\mathcal{D}_{signal}} \{j \mid rank(\mathcal{A}^{(l)}(j \mid \boldsymbol{x})) < \epsilon d\}$. We compute the signal activity score of each neuron using the sample mean $\mathcal{A}_{signal}^{(l)}(j) = \frac{1}{|\mathcal{D}_{signal}|}\sum_{\boldsymbol{x}\in\mathcal{D}_{signal}} \mathcal{A}^{(l)}(j \mid \boldsymbol{x})$.

As hidden state neurons may fulfill multiple roles [5, 37, 42], e.g., signal processing and noise removal in concurrence, we want to identify neurons that are primarily responsible for signal processing. To this end, we further collect the activity scores $\mathcal{A}_{noise}^{(l)}$ with Gaussian noises as the input by similar means and then compute the contrastive neuron activity score between signal and noise:

$$\mathcal{A}_{contrastive}^{(l)}(j) = \mathcal{A}_{signal}^{(l)}(j) - \mathcal{A}_{noise}^{(l)}(j) \ . \tag{2}$$

For each model layer $l \in \{1, \ldots, L\}$, we select the candidate neurons with top-ranked contrastive activity scores as the signal neurons, i.e., $Sig(l) = Cand(l) \cap \{j \mid rank(\mathcal{A}_{contrastive}^{(l)}(j)) < \epsilon d\}$. Figure 3 plots the distributions of the contrastive neuron activity scores across different model layers. We note that at each layer only a small proportion of neurons are exclusively sensitive to signal or noise. Moreover, greater contrasts are observed at higher layers where neurons get more specialized.

Ranking signal neurons by contrastive activity score, we leverage the top signal neuron's activity score at the final layer as a strength measure of signal information the model has processed, i.e., $\mathcal{A}^{(L)}(j \mid \boldsymbol{x})$ with $j = Top_1(Sig(L))$. The final layer is selected as it shows the greatest contrast of neuron activity scores between signal and noise inputs. We call the proposed measure Signal Subspace Activity Score (SSAS) and will verify its efficacy of hallucination detection and forecasting performance prediction in §5.2. With this, we claim that a TSFM implicitly expresses in the hidden state subspaces how much signal information it is able to capture from the context.



(a) `Chronos` (b) `TimesFM`

Figure 3: Distributions of ranked contrastive neuron activity scores across model layers.

4.3 Signal Subspace Intervention

Built upon the previous results, we propose a Center-Project-Scale (CPS) intervention operation to mitigate hallucinations by magnifying the signal information in hidden states. During forward propagation, for the hidden states $\boldsymbol{H}^{(l)} \in \mathbb{R}^{n \times d}$ at a TSFM layer, CPS works as follows:

1. Centering $\boldsymbol{H}^{(l)}$ by subtracting the mean across positions to obtain $\boldsymbol{H}_c^{(l)} = \boldsymbol{H}^{(l)} - \bar{\boldsymbol{h}}^{(l)}$;
2. Computing the projections on signal subspaces $\Pi_{\mathbb{S}^{(l)}} \boldsymbol{H}_c^{(l)}$ at all positions;
3. Scaling the signal components by a factor λ so that $\tilde{\boldsymbol{H}}_c^{(l)} = \boldsymbol{H}_c^{(l)} + (\lambda - 1)\Pi_{\mathbb{S}^{(l)}} \boldsymbol{H}_c^{(l)}$;
4. Adding back the mean to obtain the intervened hidden states $\tilde{\boldsymbol{H}}^{(l)} = \tilde{\boldsymbol{H}}_c^{(l)} + \bar{\boldsymbol{h}}^{(l)}$.

The intervened hidden states are passed as the inputs to the next layer. We center the hidden states in Step 1 to emphasize the activation differences across positions. Arranging the bases of $\mathbb{S}$ into a orthogonal matrix $\boldsymbol{P} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_k] \in \mathbb{R}^{k \times d}$, where $\boldsymbol{e}_i$ is the indicator vector of a signal neuron, the projection in Step 2 can be computed by matrix product. The CPS operation can be formulated simply as $\tilde{\boldsymbol{H}}^{(l)} = \boldsymbol{H}^{(l)} + (\lambda - 1)\boldsymbol{H}_c^{(l)} \boldsymbol{P}^T \boldsymbol{P}$, which can be efficiently computed at each layer in $\mathcal{O}(ndk)$ cost, with $k \ll d$. The cost can be further reduced to $O(nk)$ leveraging the sparsity of $\boldsymbol{P}$.

The CPS operation has desirable properties. First, the mean of hidden state neuron activations is unaltered by the operation, and the standard deviation scales proportionally with λ, which makes the operation easy to control and causes no distribution drift to neuron activations. Moreover, different from prior approaches that add a static steering vector to hidden representations [21, 23, 38], our approach adaptively alters neuron activations based on their distributions, improving the contrast of hidden states and clustering effects. We mathematically show that in many cases the CPS operation can reduce the cosine similarity between hidden states (see proofs in Appendix §B).

We further propose an adaptive scaling approach to help identify the scenarios when it is necessary to apply intervention and determine the scaling magnitude. Since the signal activity scores $\mathcal{A}_{signal}^{(l)}$ measure the neuron activity in the presence of strong signals, we use them as a reference. At each layer, we compute the mean activity scores of the signal neurons $\bar{\mathcal{A}}^{(l)}(\boldsymbol{x}) = \frac{1}{k} \sum_{j \in Sig(l)} \mathcal{A}^{(l)}(j \mid \boldsymbol{x})$. Then we compute the scaling factor as a ratio $\lambda^{(l)} = \bar{\mathcal{A}}_{signal}^{(l)} / \bar{\mathcal{A}}^{(l)}(\boldsymbol{x})$ and apply the intervention when $\lambda^{(l)} > 1$. In this way, we adaptively select the intervened layers with weak signal information and scale the activations of signal neurons to match those of the reference. We call the complete test-time intervention approach Signal Subspace Intervention through Magnification (SSIM).

5 Experiments

In this section, we conduct experiments to address the following questions: (1) How do hallucinations affect the performance of each type of TSFM? (2) How is the effect of our proposed test-time intervention on hallucination mitigation? (3) How is the performance of our proposed signal strength measures? (4) How does each our designed component impact the intervention performance?

5.1 Experimental Settings

Datasets. We curate a synthetic dataset comprising common waveforms of sine, square, sawtooth, triangle, and pulse waves with varying slopes in $\{-0.01, 0, 0.01\}$. We vary the number of periods in the context in $\{8, 10, 12, 14, 16, 18, 20\}$ and the standard deviation of Gaussian noise added to the context signal in $\{0, 0.1, 0.2, 0.3, 0.4\}$. In addition, we adopt read-world datasets from GIFT-Eval [3] benchmark covering various domains. We take a fixed number of final observations from each time series, dividing them into context and ground truth of fixed lengths. We discard time series instances with over 10% missing values and impute missing values with the segment mean. As defined in §3, we retain time series instances whose ground truth satisfies the knowledge rules extracted from the context such that the context contains sufficient information for forecasting. Each dataset is randomly split into validation (20%) and test (80%) sets. Further details are provided in Appendix §D.2.

Baselines. For hallucination mitigation, we compare SSIM with input denoising by smoothing as well as input perturbation with output averaging [23]. For hallucination detection, we compare SSAS with other statistics discussed in §4.1, including the mean pairwise cosine similarity of hidden states and the mean standard deviation of neuron activations.

Evaluation metrics. We evaluate forecast quality with R^2 and Pearson correlation coefficient, which are scale invariant. R^2 measures the goodness of fit to the ground truth, ranging in $(-\infty, 1]$; Pearson correlation coefficient measures the strength and direction of the linear relationship with the ground truth, ranging in $[-1, 1]$ (invalid values are imputed with 0). Whether a forecast is hallucinated is determined according to the knowledge rules defined in §3. We evaluate the effect of hallucination mitigation with hallucination rate reduction and forecast quality improvement. We evaluate the accuracy of hallucination detection with AUROC and forecasting performance prediction with Spearman rank correlation coefficient.

Implementation details. We evaluate on three mainstream TSFMs: *Chronos* [4], *Chronos-Bolt*, and *TimesFM* [11]. We set the context length to 500 and the forecasting horizon to 64 for zero-shot time series forecasting in our main experiments, using the base versions of *Chronos* and *Chronos-Bolt* together with `TimesFM-2.0`. As *Chronos* produces probabilistic forecasts, we set the number of decoding samples to 1 and fix the random seed to ensure reproducibility. We set the frequency configuration of *TimesFM* to 0. For hallucination check, we set the tolerance thresholds δ of the trend, frequency, pattern, and ARMA rules to 0.25, 0.5, 0.5, and 0.25, respectively, based on validation. For SSIM, we perform grid search for the proportion of selected top neurons $\epsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and set it to 0.1 for Chronos and TimesFM and 0.2 for Chronos-Bolt based on validation. For baselines methods, we denoise input time series using the mean of sliding windows of size 5. We perturb input time series by Gaussian noise with a standard deviation of 0.05 times that of the input for 10 runs with different random seeds. We leverage signal strength measures for forecasting performance prediction and their negations for hallucination detection.

5.2 Main Experimental Results

TSFM hallucinations (*RQ1*). Table 1 summarizes the performance of original TSFMs. We note that the hallucination rate varies drastically across domains. On *Energy* domain the time series have more periodic patterns, resulting in low hallucination rates; while on *Nature* domain the time series contain more abrupt changes, making it harder for TSFMs to process the context information. The forecasts appear to have stronger correlations with the ground truths on domains where the hallucination rate is lower. Table 2 compares the performance of hallucinated forecasts versus non-hallucinated forecasts by TSFMs. We note that hallucinated forecasts are consistently outperformed by non-hallucinated forecasts. For Chronos-Bolt and TimesFM, the mean R^2 of non-hallucinated forecasts stays positive on each domain. Hallucinated forecasts have substantially weaker correlations with ground truths than non-hallucinated forecasts for all models, indicating weaker signal information captured from the context. Unpaired t-tests yield $p < 10^5$ against the null hypothesis that the performance of hallucinated and non-hallucinated forecasts is the same. These results reveal how significantly the forecasting performance of TSFMs are affected by hallucinations.

Figure 4 (a) compares the distributions of TSFM hallucinations, with Type 1 referring to the violation of the trend rule, Type 2 to the frequency rule, and Type 3 to both the pattern and ARMA rules. We

Table 1: Comparison of forecasting performance across domains, with the best results boldfaced.

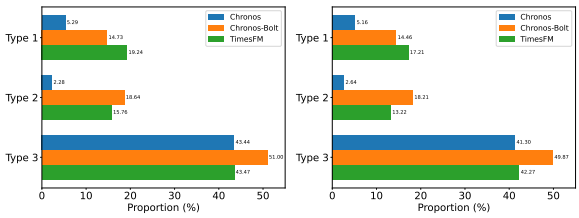
Model	Domain	Original			Denoising			Perturbation Averaging			SSIM (ours)		
		$Hal \downarrow$	$R^2 \uparrow$	$Corr \uparrow$	$Hal \downarrow$	$R^2 \uparrow$	$Corr \uparrow$	$Hal \downarrow$	$R^2 \uparrow$	$Corr \uparrow$	$Hal \downarrow$	$R^2 \uparrow$	$Corr \uparrow$
Chronos	Synthetic	0.4524	-0.1625	0.6265	0.4429	-0.6734	0.5714	0.4333	-0.1053	0.6392	**0.4145**	**0.1854**	**0.7150**
	Econ/Fin	0.4115	-3.3554	0.4751	0.5007	-4.5011	0.3413	0.4609	-3.5727	0.4735	**0.4061**	**-3.2037**	**0.5146**
	Energy	0.1389	-0.4839	0.7180	0.2504	-3.0764	0.5315	0.1212	-0.2073	0.7241	**0.1191**	**0.0268**	**0.7707**
	Nature	0.8035	-10.7283	0.0457	0.9514	-8.3558	0.0575	0.8436	-7.2973	0.0552	**0.6715**	**-0.7575**	**0.1082**
	Transport	0.4197	-1.6444	0.5127	0.7565	-1.4804	0.3295	0.4461	-1.4582	0.5315	**0.3938**	**-0.2221**	**0.6081**
	WebOps	**0.5801**	-414.8937	0.2762	0.8833	-139.8035	0.1559	0.6115	-79.3298	0.2822	0.6052	**-21.8389**	**0.3369**
	Aggregated	0.4531	-82.3762	0.4458	0.5991	-28.6399	0.3336	0.4759	-17.2700	0.4529	**0.4231**	**-5.0845**	**0.5061**
Chronos-Bolt	Synthetic	0.5381	0.0152	0.5589	0.5810	-0.0625	0.5302	0.5500	0.0099	0.5586	**0.5231**	**0.0238**	**0.5672**
	Econ/Fin	0.4856	-1.3759	0.5727	0.4870	**-1.2344**	0.4243	0.4911	-1.4191	**0.5929**	**0.4787**	-1.2891	0.5811
	Energy	0.0985	0.1499	0.7694	0.1712	-0.0411	0.6291	0.1002	0.1033	0.7671	**0.0843**	**0.1508**	**0.7765**
	Nature	0.9426	-0.0744	0.1400	0.9536	-0.6290	0.1057	0.9404	-0.0876	0.1462	**0.9316**	**-0.0657**	**0.1472**
	Transport	0.6684	0.2039	0.6501	0.8446	-0.2129	0.4072	0.6632	0.2015	0.6488	**0.6522**	**0.2124**	**0.6563**
	WebOps	0.6777	-0.6529	0.3591	0.9024	-1.2424	0.1777	0.6707	**-0.4822**	0.3625	**0.6632**	-0.6680	**0.3646**
	Aggregated	0.5308	-0.4260	0.5099	0.6084	-0.6662	0.3848	0.5321	-0.4163	0.5158	**0.5191**	**-0.3766**	**0.5171**
TimesFM	Synthetic	0.1143	0.5661	0.9143	0.1452	0.4685	0.7568	0.1190	0.5688	0.9094	**0.1049**	**0.5699**	**0.9194**
	Econ/Fin	0.3868	-1.8715	0.7793	0.4472	-5.1532	0.3722	0.4005	-1.8277	0.7657	**0.3771**	**-0.3161**	**0.7847**
	Energy	0.1357	0.2745	0.8065	0.1987	-0.0981	0.6150	0.1341	**0.2916**	0.7941	**0.1222**	0.1304	**0.8133**
	Nature	0.9558	-0.1678	**0.1620**	0.9691	-0.2561	0.1302	**0.9492**	-0.1715	0.1451	0.9536	**-0.0902**	0.1559
	Transport	0.5751	**0.4245**	0.7027	0.6321	-0.4210	0.3540	**0.5648**	0.4236	0.7028	0.5733	0.4201	**0.7076**
	WebOps	0.6429	-22.0090	0.4224	0.8676	-20.0077	0.2324	**0.6202**	-7.5448	0.4216	0.6359	**-6.7480**	**0.4291**
	Aggregated	0.4441	-4.5461	0.6368	0.5251	-5.3271	0.4119	0.4418	-2.0435	0.6275	**0.4348**	**-1.2529**	**0.6410**

Table 2: Performance comparison of hallucinated and non-hallucinated forecasts by TSFMs.

Metric	Domain	Chronos			Chronos-Bolt			TimesFM		
		Hal	*Non-hal*	*Diff*	*Hal*	*Non-hal*	*Diff*	*Hal*	*Non-hal*	*Diff*
$R^2 \uparrow$	Synthetic	-1.1206	0.6290	1.7496	-0.4450	0.5512	0.9962	-1.5404	0.8379	2.3783
	Econ/Fin	-6.7964	-0.9492	5.8473	-3.3098	0.4497	3.7595	-5.7181	0.5553	6.2734
	Energy	-2.8794	-0.0974	2.7820	-1.1625	0.2934	1.4559	-0.8971	0.4585	1.3556
	Nature	-12.8573	-2.0209	10.8364	-0.0846	0.0933	0.1779	-0.1923	0.3626	0.5549
	Transport	-3.9208	0.0018	3.9226	-0.0548	0.7253	0.7800	0.2046	0.7223	0.5178
	WebOps	-634.8971	-110.9056	523.9916	-1.1848	0.4657	1.6505	-34.4655	0.4127	34.8783
	Aggregated	-161.6823	-16.6599	145.0224	-1.1647	0.4096	1.5743	-10.9572	0.5757	11.5329
$Corr \uparrow$	Synthetic	0.3478	0.8568	0.5090	0.3596	0.7910	0.4314	0.7436	0.9364	0.1927
	Econ/Fin	0.0853	0.7476	0.6624	0.2558	0.8719	0.6162	0.5927	0.8969	0.3042
	Energy	0.4738	0.7574	0.2836	0.5716	0.7910	0.2194	0.6300	0.8342	0.2043
	Nature	0.0226	0.1401	0.1176	0.0991	0.8118	0.7127	0.1312	0.8289	0.6977
	Transport	0.2776	0.6827	0.4051	0.5407	0.8708	0.3301	0.5744	0.8765	0.3021
	WebOps	0.1036	0.5148	0.4113	0.1760	0.7441	0.5681	0.2135	0.7984	0.5849
	Aggregated	0.1459	0.6943	0.5484	0.2441	0.8105	0.5664	0.3429	0.8716	0.5286

observe that Type 3 hallucinations are the most common ones, since the pattern and ARMA rules demand more sophisticated reasoning of context information. Chronos suffers from fewer Type 1 and Type 2 hallucinations than the other models as it does not apply input patching, which enables more accurate processing of trend and frequency information.

Hallucination mitigation (*RQ2*). Table 1 compares the forecasting performance with SSIM and the baseline methods. SSIM attains the best performance overall, yielding up to 6.62% reduction on hallucination rate, 93.83% gain on R^2, and 13.52% gain on correlation over the original models. Although denoising improves R^2 in some cases by reducing the impact of outliers, it causes higher hallucination rate and lower correlation in general due to the loss of context information. Perturbation averaging improves the forecast quality to some extent, but it does not sufficiently mitigate hallucinations and demands considerably more computation. In comparison, SSIM pre-computes the signal neurons only once for each TSFM and incurs minor additional overheads during test time. The performance margin between SSIM and baselines is statistically significant at $p < 0.01$ by Friedman-Nemenyi test. We further analyze the effect of SSIM on the distributions of hallucinations. From Figure 4 (b), we note that SSIM has the greatest effect on Type 3 hallucinations, yielding up to 5% reduction. The facilitation of signal information propagation helps a TSFM better capture patterns from the context.



(a) Original (b) SSIM

Figure 4: Distributions of hallucination types.

Table 3: The results of TSFM hallucination detection and forecasting performance prediction, with the best results boldfaced. For each method, the first column shows AUROC of hallucination detection and the latter two columns show rank correlations with the performance metrics. The statistical significance of positive rank correlations is indicated with * for $p < 0.05$ and ** for $p < 0.01$.

Model	Domain	Cosine Similarity			Activation Variance			SSAS (Ours)		
		$Hal \uparrow$	$R^2 \uparrow$	$Corr \uparrow$	$Hal \uparrow$	$R^2 \uparrow$	$Corr \uparrow$	$Hal \uparrow$	$R^2 \uparrow$	$Corr \uparrow$
Chronos	Synthetic	0.7847	0.3834**	0.3262**	0.6786	0.3734**	0.4052**	**0.8316**	**0.4299****	**0.5111****
	Econ/Fin	**0.8495**	**0.6501****	**0.6208****	0.6927	0.5096**	0.5507**	0.7833	0.5034**	0.5258**
	Energy	0.7124	**0.5088****	**0.3528****	0.8093	–0.1116	0.0373	**0.8096**	0.1166**	0.0363
	Nature	0.4978	0.3382**	0.1524**	0.5384	**0.3490****	**0.1886****	**0.5925**	0.3430**	0.1507**
	Transport	0.6601	0.4706**	0.5550**	**0.7466**	**0.5351****	**0.6083****	0.6767	0.4158**	0.5234**
	WebOps	0.5542	**0.2328****	**0.3710****	0.5060	0.1363**	0.2720**	**0.5740**	0.1693**	0.2526**
	Aggregated	0.7903	**0.5866****	**0.5804****	0.7226	0.4197**	0.5277**	**0.8086**	0.5082**	0.5758**
Chronos-Bolt	Synthetic	0.4416	–0.1767	–0.0361	0.4011	–0.3806	–0.3569	**0.5282**	**0.2363****	**0.2142****
	Econ/Fin	0.2247	–0.5253	–0.5530	0.3851	–0.3600	–0.4283	**0.8528**	**0.6289****	**0.5979****
	Energy	0.5360	0.2052**	0.1488**	0.4739	**0.4158****	**0.3741****	**0.7451**	–0.0603	–0.0865
	Nature	0.9343	–0.1155	0.3819**	**0.9395**	–0.1003	**0.3842****	0.6340	**0.0895***	0.1690**
	Transport	**0.7183**	0.2601**	0.2394**	0.6656	0.2730**	0.2064**	0.6416	**0.2850****	**0.2858****
	WebOps	**0.6656**	0.1315**	**0.4091****	0.5224	–0.0236	0.1767**	0.6518	**0.2188****	0.3423**
	Aggregated	0.6279	0.0462*	0.2607**	0.6131	0.0789**	0.1840**	**0.7991**	**0.3860****	**0.5037****
TimesFM	Synthetic	0.4892	–0.2210	–0.1684	0.3821	–0.3302	–0.2738	**0.4902**	**–0.0081**	**–0.0359**
	Econ/Fin	0.2210	–0.6896	–0.6568	0.1930	–0.5539	–0.5192	**0.4625**	**–0.0932**	**–0.0987**
	Energy	0.2898	–0.2597	–0.1570	0.2777	–0.1531	–0.1911	**0.7469**	**0.1407****	**0.1042***
	Nature	0.4042	–0.0565	–0.0041	0.7912	0.0587	0.3480**	**0.8934**	**0.1744****	**0.3870****
	Transport	0.4881	–0.0669	–0.0174	0.6121	0.1892**	0.2710**	**0.6529**	**0.3483****	**0.3622****
	WebOps	0.3756	–0.3497	–0.2117	0.5615	–0.1001	0.1046*	**0.6365**	**0.0637**	**0.2418****
	Aggregated	0.3963	–0.2608	–0.1767	0.5211	–0.0517	0.0634*	**0.6890**	**0.2636****	**0.3552****

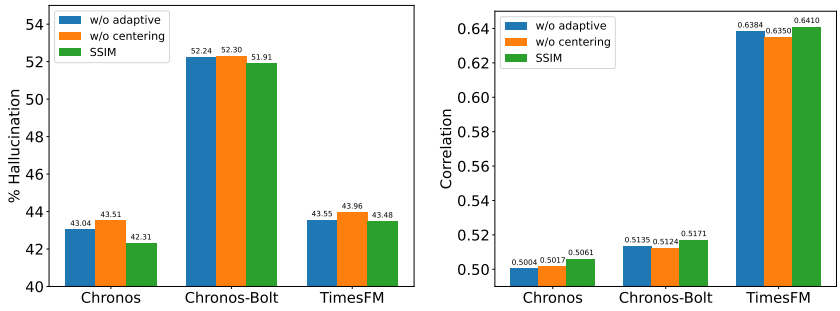


Figure 5: The aggregated mean performance of SSIM and the variants for TSFMs.

Hallucination detection and performance prediction (*RQ3*). We report the performance of different measures for hallucination detection and forecasting performance prediction in Table 3. SSAS has consistently strong predictive power across domains for different TSFMs, with high AUROC for hallucination detection and significantly positive rank correlations with forecasting performance, demonstrating the effectiveness of of our proposed signal strength measure and the critical role of signal neurons in generating high-quality forecasts. Simply using the mean neuron activation variance as a measure yields inferior and less consistent results overall, as it is obscured by the activity of irrelevant neurons. The mean pairwise cosine similarity of hidden states exhibits relatively strong predictive power for Chronos and Chronos-Bolt but fails to generalize to TimesFM.

Ablation study (*RQ4*). We compare the performance of SSIM intervention with the following reduced variants: (1) w/o adaptive scaling: using a constant scaling factor λ for each layer; (2) w/o centering: scaling neuron activations without subtracting the mean across the positions of a layer [10, 35]. From Figure 5, SSIM consistently outperforms the variants. The performance differences are significant at $p < 0.01$ by paired t-tests, highlighting the effectiveness of our design. The adaptive scaling enables more refined control of the intervention, providing greater magnification when weak signal information is detected at a layer. The centering operation places an emphasis on

activation differences that helps reduce the homogeneity of hidden states without causing distribution drifts to the activations of intervened neurons.

6 Conclusion

Time series foundation models represent a promising paradigm for time series analysis and forecasting, yet the issue of hallucinations has been underexplored in existing literature. In this paper, we have formally defined TSFM hallucinations in the zero-shot forecasting setting and outlined the rules for checking hallucinations in practice. We found that hallucinations primarily stem from a lack of context information in hidden states through experimental analyses. We have proposed a methodology to identify the signal spaces along with a measure to quantify the signal strength of hidden states. We have further developed an intervention approach that mitigates hallucinations by magnifying the signal information of hidden states. Extensive experiments across various domains have demonstrated that our test-time intervention effectively mitigates hallucinations and improves the quality of TSFM forecasts. The signal strength measure we proposed has shown strong predictive power of both hallucinations and forecasting performance. Our work contributes to deeper understanding of TSFM trustworthiness that could foster future research in this direction.

Acknowledgments

Y.Z. is partially supported by the Walter P. Murphy Fellowship. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Sumukh K Aithal, Pratyush Maini, Zachary Chase Lipton, and J Zico Kolter. Understanding hallucinations in diffusion models through mode interpolation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[3] Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

[4] Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

[5] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2, 2023.

[6] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, 2020.

[8] Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. VisionTS: Visual masked autoencoders are free-lunch zero-shot time series forecasters. *arXiv preprint arXiv:2408.17253*, 2024.

[9] Shiqi Chen, Miao Xiong, Junteng Liu, Zhengxuan Wu, Teng Xiao, Siyang Gao, and Junxian He. In-context sharpness as alerts: An inner representation perspective for hallucination mitigation. In *International Conference on Machine Learning*, pages 7553–7567. PMLR, 2024.

[10] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8493–8502, 2022.

[11] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, pages 10148–10167. PMLR, 2024.

[12] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, 2021.

[13] Xuefeng Du, Chaowei Xiao, and Yixuan Li. Haloscope: Harnessing unlabeled LLM generations for hallucination detection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[14] Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained model editing for language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[15] Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *International Conference on Machine Learning*, pages 15466–15490. PMLR, 2024.

[16] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *International Conference on Machine Learning*, pages 16115–16152. PMLR, 2024.

[17] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243, 1984.

[18] Nate Gruver, Marc Anton Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[19] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), 2023.

[20] Shuyang Jiang, Yusheng Liao, Ya Zhang, Yanfeng Wang, and Yu Wang. Fine-tuning with reserved majority for noise reduction. In *The Thirteenth International Conference on Learning Representations*, 2025.

[21] Junsol Kim, James Evans, and Aaron Schein. Linear representations of political perspective emerge in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[22] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023.

[23] Sheng Liu, Haotian Ye, and James Zou. Reducing hallucinations in large vision-language models via latent space steering. In *The Thirteenth International Conference on Learning Representations*, 2025.

[24] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *International Conference on Machine Learning*, pages 32369–32399. PMLR, 2024.

[25] Jinqi Luo, Tianjiao Ding, Kwan Ho Ryan Chan, Darshan Thaker, Aditya Chattopadhyay, Chris Callison-Burch, and Rene Vidal. PaCE: Parsimonious concept engineering for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[26] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[27] Tam Minh Nguyen, Tan Minh Nguyen, and Richard Baraniuk. Mitigating over-smoothing in transformers via regularized nonlocal functionals. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[28] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

[29] Haowen Pan, Yixin Cao, Xiaozhi Wang, Xun Yang, and Meng Wang. Finding and editing multi-modal neurons in pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1012–1037, 2024.

[30] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, et al. Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

[31] ZhongXiang Sun, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Weijie Yu, Yang Song, and Han Li. RedeEP: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. In *The Thirteenth International Conference on Learning Representations*, 2025.

[32] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] Chengxin Wang, Yiran Zhao, Shaofeng Cai, and Gary Tan. Investigating pattern neurons in urban time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.

[35] Weixuan Wang, Jingyuan Yang, and Wei Peng. Semantics-adaptive activation intervention for LLMs via dynamic steering vectors. In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] Yiming Wang, Pei Zhang, Baosong Yang, Derek F Wong, and Rui Wang. Latent space chain-of-embedding enables output-free llm self-evaluation. In *The Thirteenth International Conference on Learning Representations*, 2024.

[37] Zijian Wang, Britney Whyte, and Chang Xu. Locating and extracting relational concepts in large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4818–4832, 2024.

[38] Michał Wiliński, Mononito Goswami, Nina Żukowska, Willa Potosnak, and Artur Dubrawski. Exploring representations and interventions in time series foundation models. *arXiv preprint arXiv:2409.12915*, 2024.

[39] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning*, pages 53140–53164. PMLR, 2024.

[40] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

[41] Yuxin Xiao, Wan Chaoqun, Yonggang Zhang, Wenxiao Wang, Binbin Lin, Xiaofei He, Xu Shen, and Jieping Ye. Enhancing multiple dimensions of trustworthiness in llms via sparse activation control. *Advances in Neural Information Processing Systems*, 37:15730–15764, 2024.

[42] Tianyun Yang, Ziniu Li, Juan Cao, and Chang Xu. Mitigating hallucination in large vision-language models via modular attribution and intervention. In *The Thirteenth International Conference on Learning Representations*, 2025.

[43] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Sections 3, 4, and 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix §F.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Appendix §B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5.1 and Appendix §A and §D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.1 and Appendix §D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 5.2. We have reported the significance of statistical tests.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix §D.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Appendix §H.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix §G.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited the papers of the owners.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.