
Investigating Hallucinations of Time Series Foundation Models through Signal Subspace Analysis: Supplementary Materials

A Knowledge Rules	1
B Properties of CPS Operation	2
C SSIM Algorithm	3
D Experimental Details	3
D.1 Experimental Platform	3
D.2 Dataset Details	3
D.3 Rule Ablations	4
D.4 Additional Baseline	5
D.5 Parameter Study	6
E Additional Experimental Results	6
E.1 Model Size	6
E.2 No Data Filtering	6
F Case Study	7
G Limitations	7
H Impact Statement	8
I Ethics Statement	8
J Reproducibility Statement	8

A Knowledge Rules

The rolling windows we apply on the context always have the same size as the forecasting horizon. We adopt the OLS and ARMA [3] implementations in the statsmodels package¹². We adopt the STFT [4] implementation in the SciPy package³, with unsymmetrical Parzen windows and hop=1.

¹https://www.statsmodels.org/stable/generated/statsmodels.regression.linear_model.OLS.html

²<https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima.model.ARIMA.html>

³<https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.ShortTimeFFT.html#scipy.signal.ShortTimeFFT>

B Properties of CPS Operation

Center-Project-Scale (CPS) is our proposed intervention operation that magnifies the signal information in hidden states with a scaling factor λ . We use \mathbb{S} to denote the identified signal subspaces. CPS has several mathematical properties that help with the effectiveness of hidden state intervention:

Theorem 1 (Mean Invariance). *The mean of hidden state neuron activations across the positions of a layer is unaltered by the CPS operation, i.e., $\mathbb{E}(\tilde{\mathbf{h}}) = \bar{\mathbf{h}}$.*

Proof. Let $\bar{\mathbf{h}}$ denote the mean of hidden states across the positions of a layer. For a hidden state \mathbf{h} , the hidden state after CPS becomes $\tilde{\mathbf{h}} = \mathbf{h} + (\lambda - 1)\Pi_{\mathbb{S}}(\mathbf{h} - \bar{\mathbf{h}})$. The mean of hidden states after CPS is $\mathbb{E}(\tilde{\mathbf{h}}) = \frac{1}{n} \sum_{i=1}^n \tilde{\mathbf{h}}_i = \frac{1}{n} \sum_{i=1}^n [\mathbf{h}_i + (\lambda - 1)\Pi_{\mathbb{S}}(\mathbf{h}_i - \bar{\mathbf{h}})] = \frac{1}{n} \sum_{i=1}^n \mathbf{h}_i + \frac{1}{n} \sum_{i=1}^n (\lambda - 1)\Pi_{\mathbb{S}}(\mathbf{h}_i - \bar{\mathbf{h}})$. Since $\frac{1}{n} \sum_{i=1}^n (\lambda - 1)\Pi_{\mathbb{S}}(\mathbf{h}_i - \bar{\mathbf{h}}) = (\lambda - 1)\Pi_{\mathbb{S}}[\frac{1}{n} \sum_{i=1}^n (\mathbf{h}_i - \bar{\mathbf{h}})] = (\lambda - 1)\Pi_{\mathbb{S}}(\bar{\mathbf{h}} - \bar{\mathbf{h}}) = \mathbf{0}$, we have $\mathbb{E}(\tilde{\mathbf{h}}) = \frac{1}{n} \sum_{i=1}^n \mathbf{h}_i + \mathbf{0} = \bar{\mathbf{h}}$. \square

Theorem 2 (Standard Deviation Scaling). *In CPS, the standard deviation of signal neuron activations across the positions of a layer scales with λ , i.e., $\tilde{\mathcal{A}}(j) = \lambda \mathcal{A}(j)$ for any $j \in \mathbb{S}$.*

Proof. For any $j \in \mathbb{S}$,

$$\begin{aligned} \tilde{\mathcal{A}}(j) &= \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\tilde{\mathbf{H}}_{i,j} - \mathbb{E}(\tilde{\mathbf{h}}_j) \right)^2} \\ &= \sqrt{\frac{1}{n} \sum_{i=1}^n \left[\mathbf{H}_{i,j} + (\lambda - 1)\Pi_{\mathbb{S}}(\mathbf{H}_{i,j} - \bar{\mathbf{h}}_j) - \bar{\mathbf{h}}_j \right]^2} \quad (\text{From Theorem 1}) \\ &= \sqrt{\frac{1}{n} \sum_{i=1}^n \left[\mathbf{H}_{i,j} + (\lambda - 1)(\mathbf{H}_{i,j} - \bar{\mathbf{h}}_j) - \bar{\mathbf{h}}_j \right]^2} \\ &= \sqrt{\frac{1}{n} \sum_{i=1}^n (\lambda \mathbf{H}_{i,j} - \lambda \bar{\mathbf{h}}_j)^2} \\ &= \lambda \sqrt{\frac{1}{n} \sum_{i=1}^n (\mathbf{H}_{i,j} - \bar{\mathbf{h}}_j)^2} \\ &= \lambda \mathcal{A}(j). \end{aligned}$$

\square

Theorem 3 (Cosine Similarity Reduction). *For two hidden states \mathbf{h}_1 and \mathbf{h}_2 that have different projection directions in the signal subspace after subtracting the mean, the CPS operation reduces the cosine similarity of these two hidden states with a sufficiently large scaling factor λ , i.e., $\cos(\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2) < \cos(\mathbf{h}_1, \mathbf{h}_2)$ when $\lambda > \max\left\{-\frac{(\mathbf{s}_1 + \mathbf{s}_2) \cdot \bar{\mathbf{h}}}{\mathbf{s}_1 \cdot \mathbf{s}_2} - 1, -\frac{2\mathbf{s}_1 \cdot \bar{\mathbf{h}}}{\|\mathbf{s}_1\|^2} - 1, -\frac{2\mathbf{s}_2 \cdot \bar{\mathbf{h}}}{\|\mathbf{s}_2\|^2} - 1, 1\right\}$.*

Proof. The original hidden states can be decomposed as $\mathbf{h}_1 = \mathbf{s}_1 + \mathbf{n}_1 + \bar{\mathbf{h}}$ and $\mathbf{h}_2 = \mathbf{s}_2 + \mathbf{n}_2 + \bar{\mathbf{h}}$, where $\mathbf{s}_1 = \Pi_{\mathbb{S}}(\mathbf{h}_1 - \bar{\mathbf{h}})$, $\mathbf{n}_1 = \Pi_{\mathbb{S}^\perp}(\mathbf{h}_1 - \bar{\mathbf{h}})$, $\mathbf{s}_2 = \Pi_{\mathbb{S}}(\mathbf{h}_2 - \bar{\mathbf{h}})$, and $\mathbf{n}_2 = \Pi_{\mathbb{S}^\perp}(\mathbf{h}_2 - \bar{\mathbf{h}})$. We have $\mathbf{h}_1 \cdot \mathbf{h}_2 = \mathbf{s}_1 \cdot \mathbf{s}_2 + (\mathbf{s}_1 + \mathbf{s}_2) \cdot \bar{\mathbf{h}} + \text{residual terms}$. The hidden states after CPS can be written as $\tilde{\mathbf{h}}_1 = \lambda \mathbf{s}_1 + \mathbf{n}_1 + \bar{\mathbf{h}}$ and $\tilde{\mathbf{h}}_2 = \lambda \mathbf{s}_2 + \mathbf{n}_2 + \bar{\mathbf{h}}$. We have $\tilde{\mathbf{h}}_1 \cdot \tilde{\mathbf{h}}_2 = \lambda^2 \mathbf{s}_1 \cdot \mathbf{s}_2 + \lambda(\mathbf{s}_1 + \mathbf{s}_2) \cdot \bar{\mathbf{h}} + \text{residual terms}$.

The residual terms are $\mathbf{n}_1 \cdot \mathbf{n}_2 + (\mathbf{n}_1 + \mathbf{n}_2) \cdot \bar{\mathbf{h}} + \|\bar{\mathbf{h}}\|^2$ in both cases. Then,

$$\begin{aligned}
& \tilde{\mathbf{h}}_1 \cdot \tilde{\mathbf{h}}_2 < \mathbf{h}_1 \cdot \mathbf{h}_2 \\
\iff & \lambda^2 \mathbf{s}_1 \cdot \mathbf{s}_2 + \lambda(\mathbf{s}_1 + \mathbf{s}_2) \cdot \bar{\mathbf{h}} < \mathbf{s}_1 \cdot \mathbf{s}_2 + (\mathbf{s}_1 + \mathbf{s}_2) \cdot \bar{\mathbf{h}} \\
\iff & (\lambda^2 - 1) \mathbf{s}_1 \cdot \mathbf{s}_2 + (\lambda - 1)(\mathbf{s}_1 + \mathbf{s}_2) \cdot \bar{\mathbf{h}} < 0 \\
\iff & (\lambda + 1) \mathbf{s}_1 \cdot \mathbf{s}_2 + (\mathbf{s}_1 + \mathbf{s}_2) \cdot \bar{\mathbf{h}} < 0 \quad (\text{Assume } \lambda > 1) \\
\iff & \lambda > -\frac{(\mathbf{s}_1 + \mathbf{s}_2) \cdot \bar{\mathbf{h}}}{\mathbf{s}_1 \cdot \mathbf{s}_2} - 1. \quad (\text{Assume } \mathbf{s}_1 \cdot \mathbf{s}_2 < 0)
\end{aligned}$$

Also,

$$\begin{aligned}
& \|\tilde{\mathbf{h}}_1\| > \|\mathbf{h}_1\| \\
\iff & \lambda^2 \mathbf{s}_1 \cdot \mathbf{s}_1 + 2\lambda \mathbf{s}_1 \cdot \bar{\mathbf{h}} + \|\bar{\mathbf{h}}\|^2 > \mathbf{s}_1 \cdot \mathbf{s}_1 + 2\mathbf{s}_1 \cdot \bar{\mathbf{h}} + \|\bar{\mathbf{h}}\|^2 \\
\iff & (\lambda^2 - 1) \mathbf{s}_1 \cdot \mathbf{s}_1 + 2(\lambda - 1) \mathbf{s}_1 \cdot \bar{\mathbf{h}} > 0 \\
\iff & (\lambda + 1) \mathbf{s}_1 \cdot \mathbf{s}_1 + 2\mathbf{s}_1 \cdot \bar{\mathbf{h}} > 0 \quad (\text{Assume } \lambda > 1) \\
\iff & \lambda > -\frac{2\mathbf{s}_1 \cdot \bar{\mathbf{h}}}{\|\mathbf{s}_1\|^2} - 1.
\end{aligned}$$

Similarly, we have $\|\tilde{\mathbf{h}}_2\| > \|\mathbf{h}_2\|$ when $\lambda > -\frac{2\mathbf{s}_2 \cdot \bar{\mathbf{h}}}{\|\mathbf{s}_2\|^2} - 1$ and $\lambda > 1$.

Hence, when $\lambda > \max\{-\frac{(\mathbf{s}_1 + \mathbf{s}_2) \cdot \bar{\mathbf{h}}}{\mathbf{s}_1 \cdot \mathbf{s}_2} - 1, -\frac{2\mathbf{s}_1 \cdot \bar{\mathbf{h}}}{\|\mathbf{s}_1\|^2} - 1, -\frac{2\mathbf{s}_2 \cdot \bar{\mathbf{h}}}{\|\mathbf{s}_2\|^2} - 1, 1\}$ and $\mathbf{s}_1 \cdot \mathbf{s}_2 < 0$, we have

$$\cos(\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2) = \frac{\tilde{\mathbf{h}}_1 \cdot \tilde{\mathbf{h}}_2}{\|\tilde{\mathbf{h}}_1\| \|\tilde{\mathbf{h}}_2\|} < \frac{\mathbf{h}_1 \cdot \mathbf{h}_2}{\|\mathbf{h}_1\| \|\mathbf{h}_2\|} = \cos(\mathbf{h}_1, \mathbf{h}_2).$$

□

Since we select signal neurons with top activity scores as the bases of signal subspaces, we often have $\mathbf{s}_1 \cdot \mathbf{s}_2 < 0$ when the time series information encoded in \mathbf{h}_1 and \mathbf{h}_2 is distinct. The CPS operation reduces hidden state homogeneity by increasing the contrast of time series signal information across the positions of a layer.

Corollary 4. *In CPS, the cosine similarity of two hidden states tends to the cosine similarity of their projections the signal subspace as the scaling factor tends to infinity, i.e., $\lim_{\lambda \rightarrow \infty} \cos(\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2) = \cos(\mathbf{s}_1, \mathbf{s}_2)$.*

The proofs above explain how the CPS operation improves the clustering effects of hidden states by magnifying the time series signal information.

C SSIM Algorithm

Algorithm 1 details the full procedures of SSIM. The additional computation overhead at each layer is $O(nk)$, with k being the number of signal neurons at the layer.

D Experimental Details

D.1 Experimental Platform

All experiments are conducted on the Ubuntu 22.04.4 LTS operating system, 16 Intel(R) Core(TM) i7-7820X CPUs, and 4 NVIDIA GeForce RTX 2080 Ti GPUs, with the framework of Python 3.11.9 and PyTorch 1.12.1.

D.2 Dataset Details

Synthetic dataset. The times series we generate takes the form of $x(t) = \text{signal}(t) + \text{trend}(t) + \text{noise}(t)$. For signal component, we adopt common waveforms of sine, square, sawtooth, triangle,

Algorithm 1: SSIM: Signal Subspace Intervention through Magnification

Input : TSFM \mathcal{M}_θ with L layers, context time series $\mathbf{x}_{context}$, signal neurons Sig , reference neuron activity scores \mathcal{A}_{signal}

Output : Forecasts $\hat{\mathbf{x}}$

```
1  $\mathbf{H}^{(0)} \leftarrow \text{Preprocess}(\mathbf{x}_{context})$ 
2 for  $l \leftarrow 1, \dots, L$  do
3    $\mathbf{H}^{(l)} \leftarrow \mathcal{M}_\theta^{(l)}(\mathbf{H}^{(l-1)})$ 
4    $\bar{\mathcal{A}}^{(l)}(\mathbf{x}) \leftarrow \frac{1}{k} \sum_{j \in Sig(l)} \mathcal{A}^{(l)}(j | \mathbf{x})$ 
5    $\bar{\mathcal{A}}_{signal}^{(l)} \leftarrow \frac{1}{k} \sum_{j \in Sig(l)} \mathcal{A}_{signal}^{(l)}(j)$ 
6    $\lambda^{(l)} \leftarrow \frac{\bar{\mathcal{A}}_{signal}^{(l)}}{\bar{\mathcal{A}}^{(l)}(\mathbf{x})}$ 
7   if  $\lambda^{(l)} > 1$  then
8      $\mathbf{H}_c^{(l)} \leftarrow \mathbf{H}^{(l)} - \bar{\mathbf{h}}^{(l)}$  ▷ Center
9      $\mathbf{H}^{(l)} \leftarrow \mathbf{H}^{(l)} + (\lambda^{(l)} - 1)\mathbf{H}_c^{(l)}[:, Sig(l)]$  ▷ Project and scale
10  end
11 end
12  $\hat{\mathbf{x}} \leftarrow \text{Predict}(\mathbf{H}^{(L)})$ 
13 return  $\hat{\mathbf{x}}$ 
```

and pulse waves as implemented in the SciPy package⁴. We vary the number of signal periods in the context input from $\{8, 10, 12, 14, 16, 18, 20\}$. We vary the slope of the trend component from $\{-0.01, 0, 0.01\}$. The noise component is Gaussian noise with mean of 0, and we vary the standard deviation from $\{0, 0.1, 0.2, 0.3, 0.4\}$. In this way, we generate 525 time series instances in total.

Real-world dataset. We adopt the GIFT-Eval benchmark [1]. We discard time series instances with over 10% missing values and impute missing values with the mean of the segment. As explained in §3 of the main text, 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. The datasets are categorized into five different domains, with the bracket showing the number of time series instances after preprocessing:

- Econ/Fin: m4_daily (294), m4_hourly (391), m4_monthly (168), m4_quarterly (13), m4_weekly (42);
- Energy: electricity/15T (235), electricity/D (43), electricity/H (318), solar/10T (43), solar/H (132);
- Nature: kdd_cup_2018_with_missing/H (66), temperature_rain_with_missing (2358);
- Transport: LOOP_SEATTLE/5T (37), LOOP_SEATTLE/H (129), M_DENSE/D (9), M_DENSE/H (25), SZ_TAXI/15T (26), SZ_TAXI/H (12);
- WebOps: bitbrains_fast_storage/5T (261), bitbrains_fast_storage/H (203), bitbrains_rnd/5T (124), bitbrains_rnd/H (127).

To preserve data balance, we randomly sample 500 time series instances from oversized datasets.

D.3 Rule Ablations

In Table 1, we report the ablation results on the effect of each knowledge rule that constitutes our definition of time series forecasting hallucinations in §3 of the main text. We note that all three rules contribute to the differentiation of the forecast quality, with the aggregated Pearson correlation of non-hallucinated forecasts substantially higher than that of hallucinated forecasts in all cases. The pattern+ARMA rule effectively differentiates both R^2 and correlations in all cases. The frequency rule differentiates correlations in all cases but not R^2 , affected by the negative R^2 values from the

⁴<https://docs.scipy.org/doc/scipy/reference/signal.html>

Table 1: Comparison of the aggregated mean performance between hallucinated and non-hallucinated TSFM forecasts by checking with different knowledge rules.

Models	Rules	$R^2 \uparrow$			$Corr \uparrow$		
		<i>Hal</i>	<i>Non-hal</i>	<i>Diff</i>	<i>Hal</i>	<i>Non-hal</i>	<i>Diff</i>
Chronos	All	-161.6823	-16.6599	145.0224	0.1459	0.6943	0.5484
	Trend	-7.4013	-86.5621	-79.1609	0.1512	0.4623	0.3111
	Frequency	-0.6223	-84.2801	-83.6578	0.2505	0.4504	0.1999
	Pattern+ARMA	-168.4090	-16.2990	152.1100	0.1343	0.6851	0.5507
	Pattern	-143.4747	-18.1746	125.3001	0.1536	0.7529	0.5994
	ARMA	-108.4763	-1.0564	107.4199	0.3672	0.6909	0.3237
Chronos-Bolt	All	-1.1647	0.4096	1.5743	0.2441	0.8105	0.5664
	Trend	-1.3373	-0.2686	1.0686	0.1313	0.5753	0.4440
	Frequency	-0.0674	-0.5082	-0.4408	0.1278	0.5974	0.4697
	Pattern+ARMA	-1.1767	0.3555	1.5322	0.2370	0.7940	0.5570
	Pattern	-1.1788	0.4547	1.6335	0.2455	0.8192	0.5737
	ARMA	-0.5322	0.3277	0.8599	0.4677	0.8096	0.3420
TimesFM	All	-10.9572	0.5757	11.5329	0.3429	0.8716	0.5286
	Trend	-24.6562	0.2459	24.9021	0.1132	0.7616	0.6483
	Frequency	-0.2205	-5.3556	-5.1351	0.1027	0.7367	0.6340
	Pattern+ARMA	-11.0627	0.4657	11.5284	0.3416	0.8639	0.5223
	Pattern	-10.5886	0.5633	11.1519	0.3410	0.8869	0.5459
	ARMA	-5.3132	0.3740	5.6872	0.6040	0.8471	0.2431

Table 2: Performance comparison between full-parameter finetuning and SSIM.

Model	Chronos						TimesFM					
	Finetuning			SSIM			Finetuning			SSIM		
Domain	<i>Hal</i> ↓	R^2 ↑	<i>Corr</i> ↑	<i>Hal</i> ↓	R^2 ↑	<i>Corr</i> ↑	<i>Hal</i> ↓	R^2 ↑	<i>Corr</i> ↑	<i>Hal</i> ↓	R^2 ↑	<i>Corr</i> ↑
Synthetic	0.08	0.74	0.90	0.41	0.19	0.72	0.36	0.66	0.88	0.10	0.57	0.92
Econ/Fin	0.40	-20.11	0.54	0.41	-3.20	0.51	0.81	-63.93	0.35	0.38	-0.32	0.78
Energy	0.14	-0.41	0.79	0.12	0.03	0.77	0.21	0.48	0.87	0.12	0.13	0.81
Nature	0.56	-5.46	0.14	0.67	-0.76	0.11	0.92	-1.22	0.17	0.95	-0.09	0.16
Transport	0.36	-0.04	0.64	0.39	-0.22	0.61	0.85	-0.59	0.40	0.57	0.42	0.71
WebOps	0.44	-7906.68	0.35	0.61	-21.84	0.34	0.87	-434686.91	0.17	0.64	-6.75	0.43
Aggregated	0.33	-1524.60	0.55	0.42	-5.08	0.51	0.65	-83519.74	0.48	0.43	-1.25	0.64

misalignment between forecasts and ground truths on some outlier test instances. The synergy of these rules gives the best performance differentiation overall.

We further examine the effect of pattern and ARMA rules separately. From Table 1, both pattern and ARMA rules contribute to the differentiation of TSFM forecast quality on their own, with positive differences of both R^2 and correlations in all cases. The pattern+ARMA rule narrows down the scope of hallucinations by considering forecasts that violate both the pattern and ARMA rules as hallucinations. It captures a smaller set of forecasting hallucinations with poor quality and thus provides superior performance differentiation overall.

D.4 Additional Baseline

As a stronger hallucination mitigation baseline, we full-parameter finetune TSFMs using the validation data of each domain with 1,000 training steps. Table 2 compares the forecasting performance across domains. We observe that the performance of SSIM for Chronos is on par with full-parameter finetuning on most domains and better on some domains such as Energy, while finetuning suffers poor performance on some test instances due to overfitting. For TimesFM, finetuning suffers from even more severe overfitting, leading to inferior performance on all domains except Energy. Energy is a domain with frequent and regular patterns. SSIM saves the overheads of finetuning and better preserves the generalization performance of TSFMs.

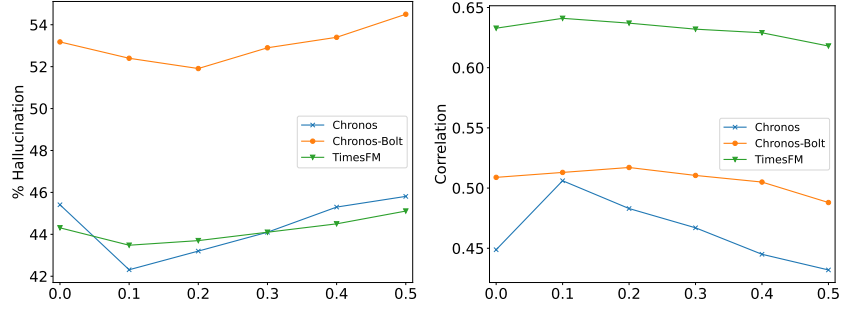


Figure 1: The aggregated mean validation performance under varying proportions of top neurons.

Table 3: Comparison of forecasting performance across domains, with the improvements boldfaced.

Method	Domain	Chronos-Mini			Chronos-Small			Chronos-Base			Chronos-Large		
		Hal ↓	R ² ↑	Corr ↑	Hal ↓	R ² ↑	Corr ↑	Hal ↓	R ² ↑	Corr ↑	Hal ↓	R ² ↑	Corr ↑
Original	Synthetic	0.6024	-0.8781	0.5426	0.5714	-0.7235	0.5623	0.4524	-0.1625	0.6265	0.3048	0.1946	0.7153
	Econ/Fin	0.4307	-2.5519	0.4727	0.4184	-2.8915	0.4786	0.4115	-3.3554	0.4751	0.4321	-4.0115	0.4874
	Energy	0.1567	-13.0010	0.6869	0.1309	-9.1815	0.6807	0.1389	-0.4839	0.7180	0.1486	-0.2673	0.7055
	Nature	0.7792	-14.3961	0.0613	0.7748	-14.3360	0.0674	0.8035	-10.7283	0.0457	0.7815	-10.8791	0.0415
	Transport	0.4560	-1.9068	0.5013	0.3938	-1.5206	0.5415	0.4197	-1.6444	0.5127	0.5130	-1.3101	0.5397
	WebOps	0.6760	-1830.7650	0.1665	0.5941	-13.1433	0.2236	0.5801	-414.8937	0.2762	0.5470	-564.4352	0.2791
	Aggregated	0.4997	-357.4382	0.4076	0.4665	-7.5057	0.4250	0.4531	-82.3762	0.4458	0.4357	-111.1697	0.4604
SSIM	Synthetic	0.5095	-0.1507	0.6166	0.4500	-0.0495	0.6451	0.4145	0.1854	0.7150	0.2452	0.4594	0.7658
	Econ/Fin	0.4360	-2.0534	0.5522	0.4387	-2.2134	0.5336	0.4061	-3.2037	0.5146	0.4319	-1.3095	0.5769
	Energy	0.1373	-0.6871	0.7427	0.1228	-1.3981	0.7258	0.1191	0.0268	0.7707	0.1292	-0.5976	0.7543
	Nature	0.7139	-1.2148	0.0980	0.6894	-0.3257	0.1050	0.6715	-0.7575	0.1082	0.6960	-0.5360	0.0860
	Transport	0.4197	-0.1980	0.5880	0.3893	-0.2187	0.6032	0.3938	-0.2221	0.6081	0.5078	-0.2593	0.5954
	WebOps	0.6400	-2.2954	0.2413	0.5625	-1.9302	0.2899	0.6052	-21.8389	0.3369	0.5261	-1.4402	0.3685
	Aggregated	0.4621	-1.3024	0.4745	0.4320	-1.2709	0.4818	0.4231	-5.0845	0.5061	0.4051	-0.7534	0.5270

D.5 Parameter Study

Figure 1 reports the validation performance selecting varying proportions of top neurons for SSIM. We observe that while intervening a reasonable number of signal neurons boosts the forecasting performance of TSFMs, selecting too many neurons degrades the performance because the activations of noise neurons may also get scaled. Based on the results, we set this parameter to 0.1 for Chronos and TimesFM and 0.2 for Chronos-Bolt.

E Additional Experimental Results

E.1 Model Size

We examine the effectiveness of our proposed approaches on models of different sizes from the Chronos family [2]. From Table 3, SSIM effectively reduces the hallucination rate and improves the forecasting quality of different models. From Table 4, the efficacy of SSAS for hallucination detection and performance prediction is better for larger models. This is because larger models have stronger capability of distinguishing context signals from noises through the processing of larger and more layers, leading to more distinct activation behaviors of signal neurons with respect to different types of inputs.

E.2 No Data Filtering

We study the impact of TSFM hallucinations without filtering the time series instances using the knowledge rules in data preprocessing. We randomly sample 500 time series instances from oversized datasets to preserve data balance. From Tables 5 and 6, we note that although there is a small decline in the overall performance compared with the results with data filtering in Tables 1 and 2 of the main text, the quality of non-hallucinated forecasts is still substantially better than that of hallucinated forecasts at $p < 10^{-5}$ by unpaired t -tests. This demonstrates the general impact of the hallucination problem we have formulated on the forecasting performance of TSFMs.

Table 4: The results of hallucination detection and forecasting performance prediction for TSFMs of different sizes, 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)		
		<i>Hal</i> \uparrow	R^2 \uparrow	<i>Corr</i> \uparrow	<i>Hal</i> \uparrow	R^2 \uparrow	<i>Corr</i> \uparrow	<i>Hal</i> \uparrow	R^2 \uparrow	<i>Corr</i> \uparrow
Chronos-Large	Synthetic	0.7878	0.3553**	0.2976**	0.5719	0.1594**	0.1459**	0.7725	0.4818 **	0.4905 **
	Econ/Fin	0.8619	0.6584**	0.6523**	0.7922	0.6079**	0.6474**	0.8780	0.6965 **	0.7055 **
	Energy	0.6952	0.1309 **	0.1190**	0.6883	-0.1911	-0.0404	0.8190	-0.0030	0.1407 **
	Nature	0.5008	0.3057**	0.1232**	0.5020	0.2979**	0.1239**	0.5479	0.4150 **	0.2080 **
	Transport	0.6228	0.4703**	0.5423**	0.6994	0.5187**	0.5728**	0.7044	0.5523 **	0.6572 **
	WebOps	0.6121	0.2366**	0.3800 **	0.5222	0.2237**	0.2785**	0.5477	0.2532 **	0.3525**
	Aggregated	0.7800	0.5168**	0.5388 **	0.7240	0.4455**	0.5033**	0.8035	0.5787 **	0.6492 **
Chronos-Base	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-Small	Synthetic	0.7748	0.4697**	0.3234**	0.5204	0.4696**	0.5616 **	0.7367	0.4725 **	0.4265**
	Econ/Fin	0.8686	0.6334 **	0.6167 **	0.6337	0.3839**	0.4081**	0.6405	0.4002**	0.4556**
	Energy	0.6909	0.6957 **	0.6036 **	0.6000	0.0611	0.1903**	0.7932	0.4817**	0.5093**
	Nature	0.4989	0.3520**	0.2850**	0.5236	0.3589**	0.3672 **	0.5595	0.3803 **	0.3253**
	Transport	0.7160	0.6267 **	0.6033**	0.7683	0.4515**	0.6326**	0.7822	0.5387**	0.6956 **
	WebOps	0.5695	0.1688 **	0.2756 **	0.4830	-0.0445	0.1239**	0.5524	0.0906*	0.2306**
	Aggregated	0.7733	0.6013 **	0.5874 **	0.6255	0.3299**	0.4396**	0.6876	0.4356**	0.5408**
Chronos-Mini	Synthetic	0.8133	0.4669 **	0.2308 **	0.3921	-0.1445	0.0087	0.5656	0.1675**	0.0889*
	Econ/Fin	0.8479	0.6233 **	0.5881 **	0.2835	-0.1790	-0.1390	0.5310	0.2172**	0.2582**
	Energy	0.7092	0.6510 **	0.5645 **	0.6565	-0.0651	0.1352**	0.7568	0.0121	0.1435**
	Nature	0.5288	0.3352**	0.2117**	0.5769	0.4002**	0.3066**	0.6528	0.5848 **	0.4637 **
	Transport	0.6610	0.4551**	0.3709**	0.6627	0.4066**	0.4655**	0.6817	0.5747 **	0.6522 **
	WebOps	0.5263	0.1289**	0.1983 **	0.5149	0.1870 **	0.1203**	0.5349	0.1148**	0.1532**
	Aggregated	0.7580	0.5664 **	0.5317 **	0.5855	0.1707**	0.2853**	0.6147	0.3091**	0.3975**

Table 5: Comparison of forecasting performance across domains without data filtering.

Domain	Chronos			Chronos-Bolt			TimesFM		
	<i>Hal</i> \downarrow	R^2 \uparrow	<i>Corr</i> \uparrow	<i>Hal</i> \downarrow	R^2 \uparrow	<i>Corr</i> \uparrow	<i>Hal</i> \downarrow	R^2 \uparrow	<i>Corr</i> \uparrow
Synthetic	0.4610	-0.1701	0.6292	0.5448	0.0013	0.5544	0.1105	0.5983	0.9141
Econ/Fin	0.6526	-5.5863	0.3556	0.6825	-2.0782	0.4702	0.6937	-1.0520	0.6423
Energy	0.4078	-2.2455	0.5606	0.3643	-0.4111	0.6521	0.3938	-0.5814	0.6625
Nature	0.8736	-2.8898	0.0875	0.9347	-0.6486	0.1628	0.9432	-0.2972	0.1918
Transport	0.7358	-7.6617	0.2747	0.8752	-0.8877	0.3657	0.8507	-0.3036	0.4527
WebOps	0.8172	-433.0063	0.1532	0.8616	-1.8096	0.2428	0.8407	-6.8100	0.2511
Aggregated Mean	0.6696	-117.9305	0.3228	0.7120	-1.2357	0.4055	0.6804	-2.1914	0.4915

F Case Study

Figures 2-4 illustrate examples of hallucinated and non-hallucinated forecasts by different TSFMs across real-world domains. The inferior performance of hallucinated forecasts is due to the TSFM failing to adequately capture the context information and generating hallucinations in the forecast that do not exist in the context, such as peaks, slopes, or patterns. The hallucinations generally deviate from the ground truths, resulting in low performance evaluation scores.

G Limitations

While our formulation of time series forecasting hallucinations is broadly applicable to all types of TSFMs, the proposed hallucination detection and intervention methods are applicable to white-box models only. A future direction is to include further TSFMs in our study.

Table 6: Performance comparison of hallucinated and non-hallucinated forecasts by TSFMs without data filtering.

Metric	Domain	Chronos			Chronos-Bolt			TimesFM		
		Hal	Non-hal	Diff	Hal	Non-hal	Diff	Hal	Non-hal	Diff
$R^2 \uparrow$	Synthetic	-1.1127	0.6360	1.7487	-0.4383	0.5273	0.9656	-1.3704	0.8428	2.2132
	Econ/Fin	-7.5763	-1.8483	5.7280	-2.9005	-0.3106	2.5899	-1.5933	0.1737	1.7670
	Energy	-3.5878	-1.3212	2.2667	-1.1147	-0.0079	1.1069	-1.5737	0.0632	1.6369
	Nature	-2.9187	-2.6900	0.2286	-0.6573	-0.5235	0.1338	-0.3369	0.3629	0.6999
	Transport	-8.4450	-5.4807	2.9642	-0.9532	-0.4285	0.5247	-0.4530	0.5478	1.0008
	WebOps	-524.3276	-24.6696	499.6580	-1.3156	-4.8857	-3.5701	-7.1521	-5.0045	2.1476
	Aggregated	-173.5672	-5.1648	168.4024	-1.4589	-0.6837	0.7753	-3.0391	-0.3866	2.6524
$Corr \uparrow$	Synthetic	0.3617	0.8581	0.4964	0.3568	0.7909	0.4341	0.7481	0.9347	0.1866
	Econ/Fin	0.1642	0.7151	0.5508	0.3121	0.8100	0.4980	0.5549	0.8403	0.2854
	Energy	0.3303	0.7191	0.3888	0.4583	0.7631	0.3048	0.4491	0.8012	0.3521
	Nature	0.0619	0.2643	0.2024	0.1192	0.7860	0.6668	0.1561	0.7855	0.6295
	Transport	0.1798	0.5388	0.3589	0.3090	0.7630	0.4539	0.3858	0.8336	0.4477
	WebOps	0.1037	0.3744	0.2706	0.1778	0.6474	0.4696	0.1750	0.6530	0.4780
	Aggregated	0.1629	0.6467	0.4838	0.2603	0.7643	0.5040	0.3375	0.8193	0.4818

H Impact Statement

We are the first to formulate and systematically study TSFM hallucinations to our best knowledge. We have formally defined the problem of TSFM hallucinations and outlined a set of procedures to check hallucinations in practice. We have proposed a method to identify the signal subspaces in TSFMs along with a measure to quantify the signal strength in TSFM hidden states. We have also proposed a simple and efficient intervention approach to mitigate hallucinations by magnifying the signal information in hidden states. Our work contributes to deeper understanding of TSFM trustworthiness that could foster future research in this direction.

I Ethics Statement

No human subjects were involved in this research, and all experiments were conducted using publicly available models and datasets, adhering to their respective licenses and use policies.

J Reproducibility Statement

We have provided comprehensive details of our proposed methods in §4 of the main text and Appendices A and C. The experimental setups, including model configurations, datasets, and evaluation metrics, have been thoroughly described in §5.1 of the main text and Appendix D. We utilize publicly available TSFMs and detail any modifications or specific settings used during experimentation. All datasets employed in our evaluation are standard benchmarks.

References

- [1] 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.
- [2] 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.
- [3] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [4] 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.

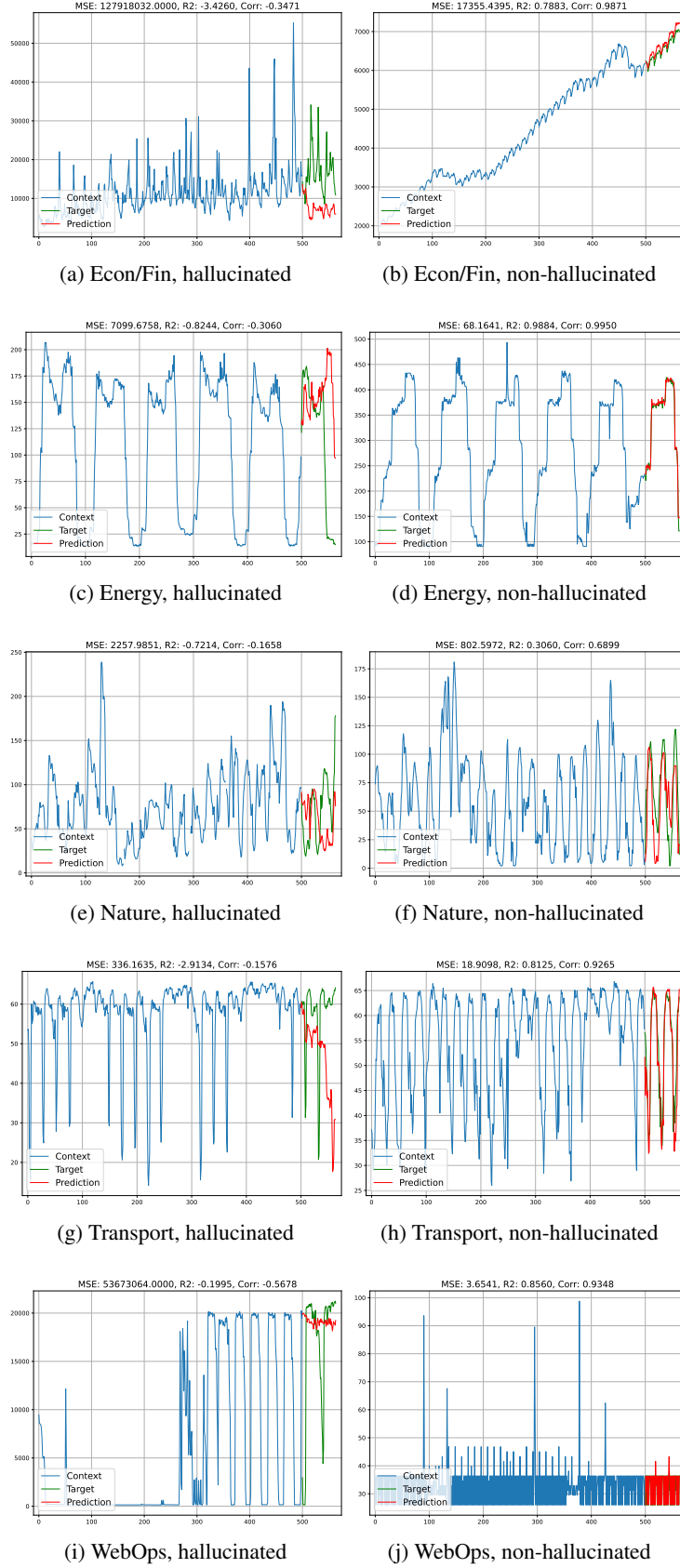


Figure 2: Examples of hallucinated and non-hallucinated forecasts by Chronos across domains.

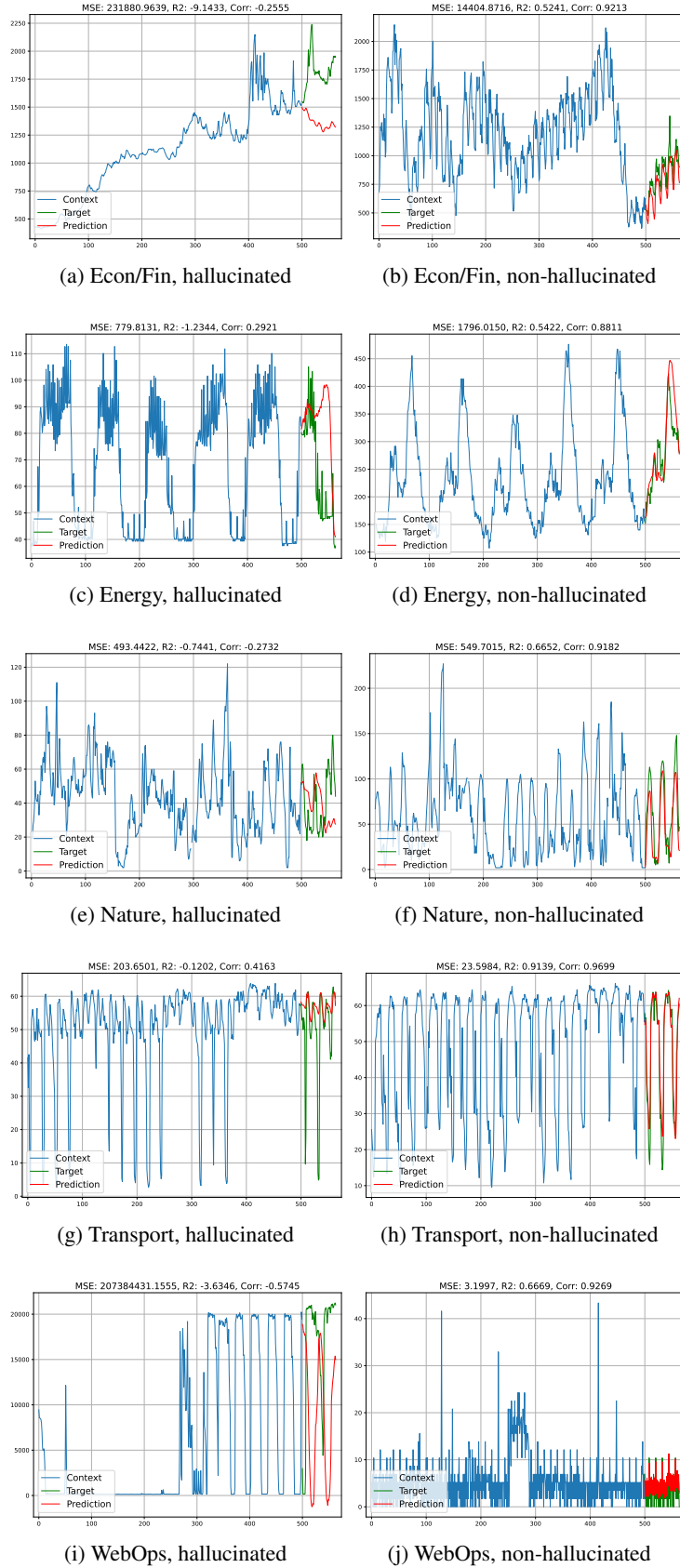


Figure 3: Examples of hallucinated and non-hallucinated forecasts by Chronos-Bolt across domains.

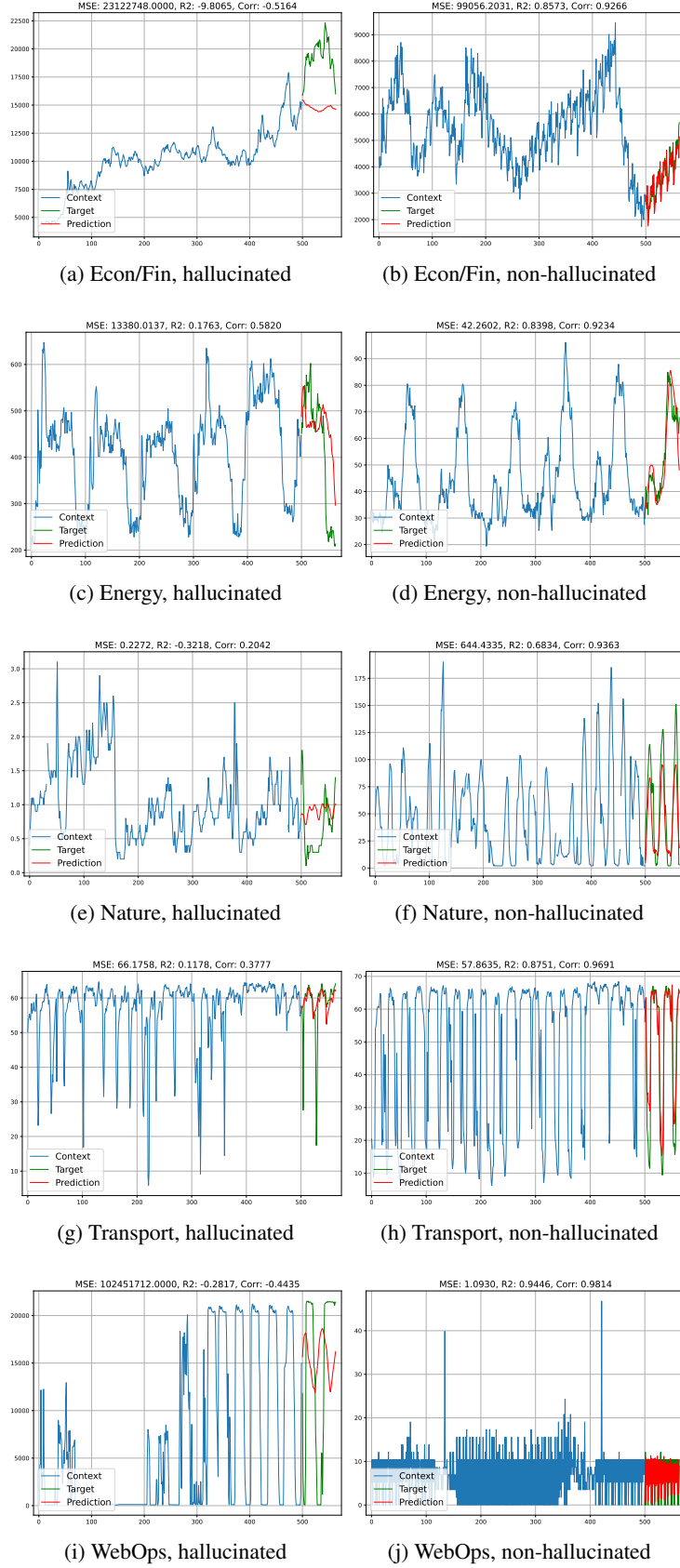


Figure 4: Examples of hallucinated and non-hallucinated forecasts by TimesFM across domains.