# Uncovering Zero-Shot Generalization Gaps in Time-Series Foundation Models Using Real-World Videos

**Lujun Li[1], Lama Sleem[1], Yiqun Wang[1], Yangjie Xu[1], Niccolò Gentile[2], Radu State[1]**

[1]University of Luxembourg
{lujun.li, lama.sleem, yiqun.wang, yangjie.xu, radu.state}@uni.lu
[2]Foyer S.A.
niccolo.gentile@foyer.lu

## Abstract

Recent research on time-series foundation models (TSFMs) has underscored the scarcity of real-world data, often supplemented with synthetic sources in existing datasets, whose generalizability remains however debated. As such, in this work, we propose a novel benchmarking approach: in particular, we aim at building a curated dataset reflecting real world physical temporal dynamics, extracting temporal signals from real-world videos using optical flow. As such, we introduce **REAL-V-TSFM**, a novel dataset designed to capture rich and diverse time series derived from real-world videos. Experimental results on state-of-the-art TSFMs under zero-shot forecasting show that, despite strong performance on conventional benchmarks, these models exhibit performance degradation on the proposed dataset, suggesting limited generalizability to novel datasets. These findings underscore the need for novel approaches to acquiring time series data and highlight the lack of universality in recent TSFMs, while further validating the effectiveness of our video-based time series data extraction pipeline.

**Code** —
https://github.com/DobricLilujun/benchmarking_nature_tsfm
**Datasets** — https://huggingface.co/datasets/Volavion/real-v-tsfm
**Extended version** — https://arxiv.org/abs/2509.26347

## Introduction

**TSFM Generalization.** Time series analysis has historically played a central role in practical applications across finance (Chen et al. 2023; Yu et al. 2023), healthcare (Li et al. 2024; Liu et al. 2023), urban computing (Wang et al. 2023), environmental research (Dong et al. 2023), and numerous other fields (Nie et al. 2024). Foundation Models (FMs) are large pre-trained architectures that learn general patterns from massive data, enabling broad adaptability and strong zero-shot performance across tasks. In natural language processing (NLP), models such as BERT (Devlin et al. 2019) and GPT-3 (Brown et al. 2020) have fundamentally reshaped approaches to text comprehension and generation, and the time-series community is undergoing a "BERT moment", marked by the emergence of foundation Transformer-based models (e.g., Chronos and TSFM variants (Ansari et al. 2024)). In contrast to NLP, whose generalization performance has been validated by vast numbers of users and researchers (Wang et al. 2025), the generalization of TSFMs has been far less tested and verified, largely due to limited dataset diversity and a relatively small user base.

**Research Question.** In (Ansari et al. 2024), the author introduced Chronos, now considered as a cornerstone TSFM. After their evaluation across 42 datasets, both in-domain and zero-shot forecasting, they showed that Chronos surpassed both traditional models and task-specific deep learning approaches. However, the training process mostly rely on synthetic data augmentation(Tan et al. 2024; Liu et al. 2025; Xie et al. 2025). More specifically, said synthetic training data were generated via two augmentation methods—KernelSynth (Duvenaud et al. 2013) and TSMixup (Zhang et al. 2017). However, despite employing multiple augmentation strategies, the generalization of these models remains debated. We therefore ask: **How general can we consider the current TSFMs, and can they really forecast based on real data extracted from daily real events?**

## Dataset Construction and Statistics

### Real World Projection

To answer our aforementioned research question, we consider specifically time series from video. We chose to use videos since other form of time series like sensor data or stock prices have already been widely investigated in the context of TSFMs. Accordingly, we assess these models with videos as an alternative source of time-series signals. Where appropriate, this strategy can markedly expand available resources for the community, given that videos are among the most abundant time-series data in modern settings. More precisely, camera-recorded video projects 3D scenes onto 2D images, eliminating explicit depth according to the pinhole camera model (Sturm 2021). Videos' high dimensionality and multimodality complicate the extraction of informative univariate signals, yet they still embeds rich temporal patterns that reflect underlying physical dynamics (Chari et al. 2019). The above technical point once again motivate our choice to leverage existing videos to enrich to assess existing TSFMs' generalizability skills in real-world.

## Optical Flow Mindset

As a starter, in this paper, we build REAL-V-TSFM, a novel time-series dataset entirely derived using optical flow methods from existing video data. The ultimate goal of optical flow methods is to estimate the motion information of objects within a scene in an image by analyzing changes in pixel intensities over consecutive frames (Fleet and Weiss 2005). This method relies on the brightness constancy assumption, which suggests that a moving point in a scene preserves its pixel intensity across adjacent frames (Horn and Schunck 1981). When dealing with a large number of diverse long-duration videos, the motion of the main objects within the video frames forms continuous motion patterns. For example, a person on a swing exhibits distinct temporal motion sequences for the hands, waist, and head. Each sequence can be represented along the x- and y-axes, and treated either individually (univariate) or jointly as a multivariate time series. Additionally, camera shake can introduce movements in the background, resulting in a multitude of continuous time series. To our knowledge, this work is the first to propose the use of optical flow through the Lucas-Kanade method (Lucas and Kanade 1981) to extract time series signals from pixel trajectories in videos, with particular emphasis on key points. In the following section, we go more in details of the developed workflow to build REAL-V-TSFM.

## Dataset Production Workflow

As a starter, LaSOT (Fan et al. 2019) serves as the primary video source, offering long sequences with guaranteed main subjects (e.g., humans, animals). As shown in Fig. 1, videos are first selected **(Step 1)** and extracted as frame-by-frame images **(Step 2)**. Foreground detection then uses Mixture of Gaussians 2 (MOG2) (Bouwmans, El Baf, and Vachon 2008; Stauffer and Grimson 1999), a GMM-based method (Han and Lin 2005) that models each pixel's color distribution with multiple Gaussians: pixels not matching background models are classified as foreground **(Step 3)**. The resulting mask is applied to suppress background, after which corner detection is performed on subjects **(Step 4)**. Shi–Tomasi's algorithm (Shi et al. 1994) is then adopted, identifying corners via a large minimum eigenvalue of the local structure matrix, ensuring strong gradients in both directions.

Finally, a forward–backward consistency check (Kalal, Mikolajczyk, and Matas 2010) is performed using pyramidal Lucas–Kanade optical flow to filter unstable trajectories (Step 5). This step removes unreliable trajectories, retaining only correspondences that are consistently tracked across frames. The rationale for this step is that, in our experiments, numerous tracking errors were observed, such as target point loss and misidentifications across multiple frames. Applying the forward–backward consistency check significantly reduces these issues, although a certain amount of noise remains unavoidable.

The forward–backward check is defined as:

$$e_{fb}(\mathbf{p}_0) = \left\| \mathbf{p}_0 - \left(f_{\text{backward}} \circ f_{\text{forward}}\right)(\mathbf{p}_0) \right\|_2 , \qquad (1)$$

where $\mathbf{p}_0$ stands for the original pixel (or keypoint) in the first frame and $e_{fb}(\mathbf{p}_0)$ denotes the forward-backward error, derived by calculating the Euclidean distance between forward optical flow followed by backward optical flow and comparing the result with the original point. The forward optical flow $f_{\text{forward}}$ estimates pixel displacement from the first frame to the second frame, while the backward optical flow $f_{\text{backward}}$ estimates the displacement from the second frame back to the first. If $e_{fb}(\mathbf{p}_0) < \epsilon$, the tracking of point $\mathbf{p}_0$ is considered valid, that is to say, the forward and backward tracking is consistent, so the error will be small (close to 0).

In the subsequent post-processing step (**Step 6**), track durations are standardized by linearly interpolating each track to match the length of the longest track within the same video. Despite background masking, camera motion can introduce background corner tracks; therefore, correlations among all tracks are computed and the five least correlated are retained (subjects typically map to one–two corner points), emphasizing informative, diverse motions while suppressing noise. Finally, each track's x- and y-coordinates are treated as two independent time-series and stored in REAL-V-TSFM as separate streams but we also propose a multi-variant version. All the technical details are shown in the Appendix.

## Datasets Statistics

The dataset contains 6,130 time series corresponding to 609 different objects, reflecting substantial categorical diversity. The series vary markedly in length, with an average of 2,043 time steps (i.e., frames) and a coefficient of variation of 0.516 indicating moderate relative dispersion. The average sequence lengths of obtained dataset range from 1,000 to 8,000, reflecting a broad variation in temporal resolution across the dataset. Time series values span a broad dynamic range reflecting the positional information in each frame of videos (mean = 402.13, standard deviation = 281.46) highlights pronounced variability across series. Each time series additionally includes the primary object category (e.g., airplane, boat, cat), derived from the corresponding original video dataset. Together, these characteristics underscore the dataset's rich heterogeneity, making it well-suited for comprehensive time series analysis.

For a comparison with the M4 dataset, the Augmented Dickey–Fuller (ADF) (Mushtaq 2011) unit root test is applied on both datasets, with the null hypothesis corresponding to non-stationarity. Using a 95% confidence level, 44% of the series in the proposed dataset were found to be stationary, compared with only 5% in the M4 dataset. Furthermore, the information entropy is measured at an average of $4.17$ bits for the M4 dataset, while the REAL-V-TSFM dataset shows an average of $3.88$ bits, reflecting a slightly lower degree of variability and uncertainty in the latter.

Finally, we project both datasets using principal component analysis (PCA) and observe markedly different distributions. We align the two time series datasets REAL-V-TSFM and M4 to the same sequence length and performing min–max normalization on each sequence individually, and then we combined them and applied PCA. We visually depict the distribution shapes, concentration regions, and overlap & divergence patterns of the two datasets. As illustrated in Fig. 2, the two time series datasets demonstrate notable differences in their low-dimensional projections: REAL-V-TSFM is distributed more uniformly, whereas M4 displays a clear skew, with its distribution being sparse in the lower-left and right regions of the PCA plane. By contrast, our time series are derived from more variable real-world physical processes, leading to uniform distributional coverage.

# Testing REAL-V-TSFM

## Evaluation Settings

Upon building the REAL-V-TSFM dataset, we also evaluated four forecasting datasets from GIFT_EVAL(Aksu et al. 2024), shown in Table 1. To ensure consistent input dimensions across time series of varying lengths, all data are segmented into fixed-size windows of 500 time steps, where the first 450 steps serve as the contextual input and the remaining 50 as the prediction horizon. For longer sequences exceeding 500 time steps, we apply a sliding-window approach that scans through the series in strides of 500, effectively partitioning the data into multiple overlapping segments. For shorter sequences (i.e., those with fewer than 500 time steps), we perform linear interpolation to extend them to a uniform length of 500. This procedure guarantees that every time series contributes at least one complete 500-step segment for evaluation, ensuring comprehensive coverage across datasets of diverse lengths.

Model performance is assessed using four complementary metrics: (1) **Mean Absolute Percentage Error**: MAPE measures the average magnitude of forecasting errors as a percentage of actual values (2) **Symmetric MAPE**: normalizes errors by the average of actual and predicted values (Chicco, Warrens, and Jurman 2021) (3) **Aggregate Relative Weighted Quantile Loss**: evaluates the accuracy and calibration of quantile forecasts across multiple series, weighted by their importance, and normalized against a reference baseline (Shchur et al. 2023) and finally, (4) **Aggregate Relative Mean Absolute Scaled Error**: Aggregate Relative MASE provides a scaled measure of forecast accuracy by comparing model errors to those of a baseline method, aggregated across all series (Hyndman and Koehler 2006). As baseline, the performance of a Linear Regression model is directly adopted. Three open-source TSFMs from Hugging Face were selected for performance comparison as shown in Table 2, with additional models evaluated as detailed in Table 4.

## REAL-V-TSFM is challenging

As shown in Table 1, we generally observe that performances on REAL-V-TSFM rank either the first or second worst in terms of forecasting across nearly all models. This indicates that our dataset is more challenging than other datasets in the GIFT-EVAL benchmark. When using Agg. Relative WQL as the evaluation metric, which emphasizes distributional characteristics, the difference becomes even more prominent: on chronos-t5-large, the performance gap compared with other datasets reaches 5.45, whereas for other models, it is around 1.0. This result demonstrates that current TSFM models fail to adequately capture the predictive distributions of time series data reflecting real physical laws, thereby revealing their limitations in generalizability.

It is worth noting that in terms of model predictability, the performance disparity is not significantly larger than that observed on other GIFT-EVAL datasets. For instance, in the performance results of the google/timesfm-2.0-500m-pytorch model, the Agg. Relative WQL value on the proposed dataset is even better than that on LOOP_SEATTLE_D, indirectly confirming the predictability of the model. However, overall, the models do exhibit a certain degree of predictive performance degradation on time series that reflect motion dynamics extracted using optical flow methods.

Regarding model comparison, timesfm-2.0-500m-pytorch demonstrates overall better performance than the chronos series, particularly on our dataset, where it exhibits relatively stronger generalization. This advantage may be attributed to its decoder-only architecture. Moreover, the bolt version shows substantial improvement over the t5 version, with significant gains in both performance and generalization ability. Furthermore, as shown in Table 4, models of different sizes within the same series show only limited performance improvements, suggesting that a clear scaling law may not be evident.

# Conclusion

In this work, we propose a novel time series extraction pipeline from real world videos and introduce an open-source dataset, REAL-V-TSFM, built from public video datasets. Temporal signals are extracted from real-world videos using optical flow to bridge the gap between synthetic benchmarks and real dynamics. Experiments show that while TSFMs perform well on standard datasets, their performance drops on REAL-V-TSFM, revealing a gap between synthetic and real-world data. This underscores the need for data-centric benchmarks that more effectively capture real-world complexity, as well as data augmentation strategies leveraging this pipeline on a tremendously rich collection of real-life videos for TSFMs pretraining.

# Discussion and Future Work

To the best of our knowledge, this paper is the first to introduce a novel and practically valuable benchmark derived from real-world physical dynamics via optical flow, and then apply this methodology to evaluate cutting-edge TSFMs in the zero-shot forecasting setting, demonstrating substantial potential for enriching the diversity of time series datasets, particularly in today's video-rich online environment. However, despite the promising generalization capabilities of TSFMs, many real-world deployments still rely on task-specific fine-tuning, so using a portion of this proposed dataset as a training set for few-shot prediction would be informative for assessing both the dataset's utility and TSFMs' practical generalization. Moreover, incorporating a broader set of other TSFMs as baselines, such as N-BEATS(Oreshkin et al. 2019) and PatchTST (Nie et al. 2023), would provide a more comprehensive performance landscape. From a theoretical standpoint, developing expla-

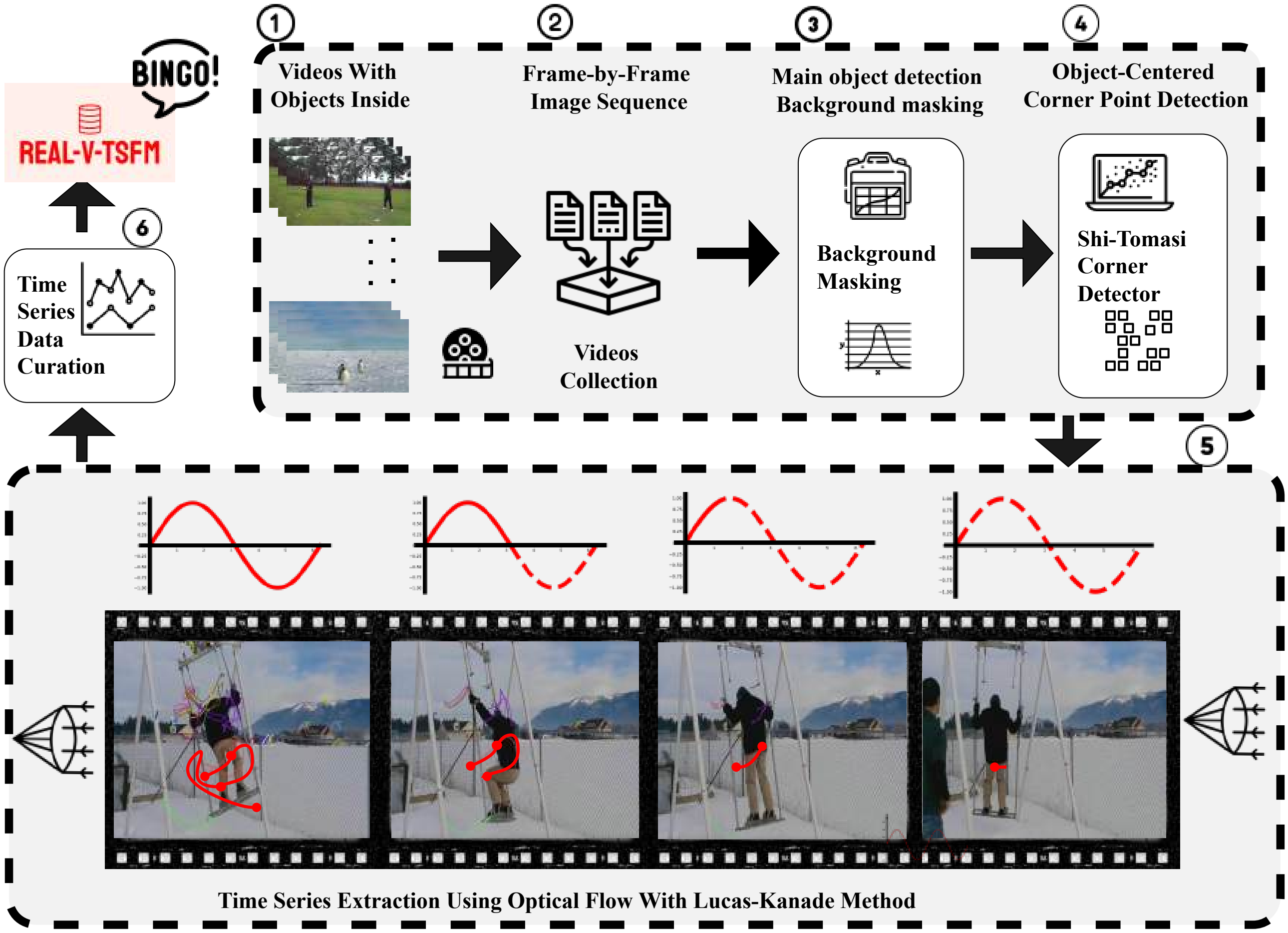
1
Videos With Objects Inside
2
Frame-by-Frame Image Sequence
Videos Collection
3
Main object detection Background masking
Background Masking
4
Object-Centered Corner Point Detection
Shi-Tomasi Corner Detector
5
Time Series Extraction Using Optical Flow With Lucas-Kanade Method
6
Time Series Data Curation
REAL-V-TSFM
BINGO!

Figure 1: Dataset production workflow consisting of six steps

| Model | Datasets | MAPE | sMAPE | Agg. Relative WQL | Agg. Relative MASE |
|---|---|---|---|---|---|
| amazon/chronos-bolt-base | REAL-V-TSFM | **7.32 ± 17.06** | **6.57 ± 9.93** | **0.93 ± 0.90** | **0.67 ± 0.63** |
|  | M4-Weekly | 5.72 ± 3.83 | 5.70 ± 3.83 | 0.79 ± 0.87 | 0.50 ± 0.49 |
|  | M4-Daily | 4.93 ± 3.82 | 5.03 ± 3.22 | **1.00 ± 0.85** | 0.63 ± 0.59 |
|  | electricity_D | 6.52 ± 3.03 | 6.44 ± 2.88 | 0.80 ± 0.41 | 0.62 ± 0.35 |
|  | LOOP_SEATTLE_D | **6.65 ± 2.61** | **6.75 ± 2.70** | 0.90 ± 0.09 | **0.89 ± 0.09** |
| amazon/chronos-t5-large | REAL-V-TSFM | **9.32 ± 18.46** | 8.40 ± 9.69 | **5.45 ± 31.23** | **5.58 ± 34.82** |
|  | M4-Weekly | 8.85 ± 6.17 | **8.70 ± 6.06** | 1.19 ± 1.20 | 0.75 ± 0.76 |
|  | M4-Daily | 7.11 ± 7.02 | 7.18 ± 4.65 | **1.56 ± 1.26** | **0.98 ± 0.88** |
|  | electricity_D | **9.18 ± 3.85** | **8.99 ± 3.44** | 1.19 ± 0.56 | 0.88 ± 0.50 |
|  | LOOP_SEATTLE_D | 7.06 ± 2.83 | 7.18 ± 2.92 | 0.98 ± 0.16 | 0.95 ± 0.13 |
| google/timesfm-2.0-500m-pytorch | REAL-V-TSFM | **6.97 ± 16.63** | 6.24 ± 9.17 | **0.91 ± 1.02** | **0.64 ± 0.65** |
|  | M4-Weekly | **8.30 ± 7.08** | 8.18 ± 6.70 | 0.85 ± 0.96 | 0.60 ± 0.62 |
|  | M4-Daily | 1.9 ± 1.08 | 2.03 ± 2.55 | 0.39 ± 0.22 | 0.23 ± 0.25 |
|  | electricity_D | 6.81 ± 3.63 | **6.77 ± 3.54** | 0.80 ± 0.44 | 0.63 ± 0.39 |
|  | LOOP_SEATTLE_D | 6.79 ± 2.77 | **6.89 ± 2.86** | **0.92 ± 0.06** | **0.90 ± 0.07** |
| LinearRegression | REAL-V-TSFM | **15.52 ± 28.44** | **14.28 ± 20.21** | 1.00 ± 0.00 | 1.00 ± 0.00 |
|  | M4-Weekly | **21.91 ± 23.49** | **19.96 ± 19.67** | 1.00 ± 0.00 | 1.00 ± 0.00 |
|  | M4-Daily | 14.14 ± 24.99 | 14.10 ± 19.99 | 1.00 ± 0.00 | 1.00 ± 0.00 |
|  | electricity_D | 13.56 ± 10.74 | 13.93 ± 14.34 | 1.00 ± 0.00 | 1.00 ± 0.00 |
|  | LOOP_SEATTLE_D | 7.50 ± 2.97 | 7.63 ± 3.13 | 1.00 ± 0.00 | 1.00 ± 0.00 |

Table 1: Performance comparison across models and datasets. Boldface values indicate the lowest and second-lowest performance within each group of models. Each item is reported as $\mu\pm$std, representing the mean and standard deviation, respectively.

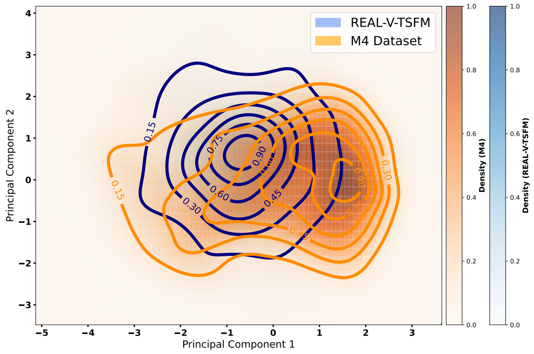

Figure 2: PCA projection of the proposed dataset and the M4-Daily dataset. The color of the heatmap represents the density level, with darker colors indicating regions of higher point concentration. The contour lines (also referred to as iso-density or equal-density curves) connect points sharing the same density value.

| Model | Params (M) | Release Date | Architecture | Reference |
|---|---|---|---|---|
| amazon/chronos-bolt-base | ∼205M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| amazon/chronos-t5-large | ∼709M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| google/timesfm-2.0-500m | ∼500M | 2025 | Decoder-only | (Das et al. 2024) |
| LinearRegression | – | classical | Linear Model | (Baseline) |

Table 2: Comparison of evaluated TSFMs

nations for why real-world video-derived datasets induce performance degradation in current TSFMs is crucial for understanding model limitations; in addition, although this work adopts a single optical-flow method, alternative or more recent variants could serve as the basis for constructing comparable datasets across different extraction pipelines. Finally, broader evaluations on additional tasks such as imputation and classification, together with more diverse real-world video sources, would further substantiate claims regarding the universality and robustness of TSFMs.

## Acknowledgments

All models and resources developed in this work are strictly intended for research and educational purposes; no model weights or derivatives are used — or will be used — for any commercial application. We exclusively utilize publicly available corpora or datasets for which explicit authorization has been obtained from the original data providers. All license terms have been reviewed to ensure full compliance with copyright, attribution, and sharing requirements.

All code, models, and processed data artifacts will be released under an open-source, research-oriented license (e.g., CC BY-NC), accompanied by comprehensive documentation and bias-analysis methodology to promote transparency and reproducibility. We commit to ongoing ethical oversight through periodic reevaluation of datasets and model outputs, prompt updates in response to emerging concerns, and consultation with interdisciplinary advisory boards to ensure adherence to the highest ethical standards.

## References

Aksu, T.; Woo, G.; Liu, J.; Liu, X.; Liu, C.; Savarese, S.; Xiong, C.; and Sahoo, D. 2024. GIFT-Eval: A Benchmark For General Time Series Forecasting Model Evaluation. *arxiv preprint arxiv:2410.10393*.

Ansari, A. F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Arango, S. P.; Kapoor, S.; et al. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Bouwmans, T.; El Baf, F.; and Vachon, B. 2008. Background modeling using mixture of gaussians for foreground detection-a survey. *Recent patents on computer science*, 1(3): 219–237.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chari, P.; Talegaonkar, C.; Ba, Y.; and Kadambi, A. 2019. Visual physics: Discovering physical laws from videos. *arXiv preprint arXiv:1911.11893*.

Chen, Z.; Zheng, L. N.; Lu, C.; Yuan, J.; and Zhu, D. 2023. Chatgpt informed graph neural network for stock movement prediction. *arXiv preprint arXiv:2306.03763*.

Chicco, D.; Warrens, M. J.; and Jurman, G. 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *Peerj computer science*, 7: e623.

Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2024. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 4171–4186.

Dong, J.; Wu, H.; Zhang, H.; Zhang, L.; Wang, J.; and Long, M. 2023. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36: 29996–30025.

Duvenaud, D.; Lloyd, J.; Grosse, R.; Tenenbaum, J.; and Zoubin, G. 2013. Structure discovery in nonparametric regression through compositional kernel search. In *International Conference on Machine Learning*, 1166–1174. PMLR.

Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; and Ling, H. 2019. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5374–5383.

Fleet, D. J.; and Weiss, Y. 2005. Optical Flow Estimation. In Paragios, N.; Chen, Y.; and Faugeras, O., eds., *Handbook of Mathematical Models in Computer Vision*, 239–258. Springer.

Han, B.; and Lin, X. 2005. Update the GMMs via adaptive Kalman filtering. In *Visual Communications and Image Processing 2005*, volume 5960, 1506–1515. SPIE.

Horn, B. K.; and Schunck, B. G. 1981. Determining optical flow. *Artificial intelligence*, 17(1-3): 185–203.

Hyndman, R. J.; and Koehler, A. B. 2006. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4): 679–688.

Kalal, Z.; Mikolajczyk, K.; and Matas, J. 2010. Forward-backward error: Automatic detection of tracking failures. In *2010 20th international conference on pattern recognition*, 2756–2759. IEEE.

Li, J.; Liu, C.; Cheng, S.; Arcucci, R.; and Hong, S. 2024. Frozen language model helps ecg zero-shot learning. In *Medical Imaging with Deep Learning*, 402–415. PMLR.

Liu, X.; Aksu, T.; Liu, J.; Wen, Q.; Liang, Y.; Xiong, C.; Savarese, S.; Sahoo, D.; Li, J.; and Liu, C. 2025. Empowering Time Series Analysis with Synthetic Data: A Survey and Outlook in the Era of Foundation Models. arXiv:2503.11411.

Liu, X.; McDuff, D.; Kovacs, G.; Galatzer-Levy, I.; Sunshine, J.; Zhan, J.; Poh, M.-Z.; Liao, S.; Di Achille, P.; and Patel, S. 2023. Large language models are few-shot health learners. *arXiv preprint arXiv:2305.15525*.

Lucas, B. D.; and Kanade, T. 1981. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, 674–679.

Mushtaq, R. 2011. Augmented dickey fuller test.

Nie, Y.; Kong, Y.; Dong, X.; Mulvey, J. M.; Poor, H. V.; Wen, Q.; and Zohren, S. 2024. A survey of large language models for financial applications: Progress, prospects and challenges. *arXiv preprint arXiv:2406.11903*.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. arXiv:2211.14730.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *ArXiv*, abs/1905.10437.

Shchur, O.; Turkmen, A. C.; Erickson, N.; Shen, H.; Shirkov, A.; Hu, T.; and Wang, B. 2023. AutoGluon–TimeSeries: AutoML for probabilistic time series forecasting. In *International Conference on Automated Machine Learning*, 9–1. PMLR.

Shi, J.; et al. 1994. Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, 593–600. IEEE.

Stauffer, C.; and Grimson, W. E. L. 1999. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, volume 2, 246–252. IEEE.

Sturm, P. 2021. Pinhole camera model. In *Computer Vision: A Reference Guide*, 983–986. Springer.

Tan, M.; Merrill, M. A.; Gupta, V.; Althoff, T.; and Hartvigsen, T. 2024. Are Language Models Actually Useful for Time Series Forecasting? arXiv:2406.16964.

Wang, X.; Antoniades, A.; Elazar, Y.; Amayuelas, A.; Albalak, A.; Zhang, K.; and Wang, W. Y. 2025. Generalization v.s. Memorization: Tracing Language Models' Capabilities Back to Pretraining Data. arXiv:2407.14985.

Wang, X.; Wang, D.; Chen, L.; Wang, F.-Y.; and Lin, Y. 2023. Building transportation foundation model via generative graph transformer. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 6042–6047. IEEE.

Xie, S.; Feofanov, V.; Alonso, M.; Odonnat, A.; Zhang, J.; Palpanas, T.; and Redko, I. 2025. CauKer: classification time series foundation models can be pretrained on synthetic data only. arXiv:2508.02879.

Yu, X.; Chen, Z.; Ling, Y.; Dong, S.; Liu, Z.; and Lu, Y. 2023. Temporal data meets LLM–explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

## Models

As shown in Table 3, we mainly select multiple variants of the Chronos models, and we also evaluate the TimesFM models. The key architectural distinction lies in the fact that TimesFM adopts a *decoder-only* design, while the Chronos models are based on an *encoder–decoder* architecture. Their sizes range from the largest at 709M parameters to the smallest at 7M parameters, representing some of the most commonly used TSFMs in recent studies. It is worth noting that, for TimesFM, we only tested the case with frequency $= 0$, i.e., the high-frequency setting, since we assume that the video-derived time series used in our experiments are predominantly of high frequency.

## Optical Flow Experimental Details

The `calcOpticalFlowPyrLK` function from OpenCV is employed, with a relatively large search window of $(40 \times 40)$ to improve the tracking of fast-moving objects. A three-level image pyramid is adopted to enhance robustness in scenarios involving large displacements. In addition, the convergence criteria were set to a maximum of 30 iterations or an accuracy threshold of 0.01, thus balancing computational cost and precision. Overall, this configuration provides a robust solution for optical flow estimation in the cases of rapid motion or large displacements, albeit at the expense of increased computational complexity.also Forward-backward check is also tested as a quality verification method for optical flow and set the forward-backward error threshold to 50.0 and the single-direction optical flow residual error threshold (ERR_THRESH) to 80.0. Increasing FB_ERR_THRESH is intended to retain more tracked points, even those with relatively large errors, thereby extending the length of the time series, though inevitably introducing some noise from erroneous tracking. Similarly, the

| Model | Params (M) | Release Date | Architecture | Reference |
|---|---|---|---|---|
| amazon/chronos-bolt-tiny | ∼7M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| amazon/chronos-bolt-mini | ∼21M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| amazon/chronos-bolt-small | ∼48M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| amazon/chronos-bolt-base | ∼205M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| amazon/chronos-t5-tiny | ∼8M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| amazon/chronos-t5-mini | ∼20M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| amazon/chronos-t5-small | ∼46M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| amazon/chronos-t5-base | ∼201M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| amazon/chronos-t5-large | ∼709M | 2024 | Encoder-Decoder | (Ansari et al. 2024) |
| google/timesfm-2.0-500m | ∼500M | 2025 | Decoder-only | (Das et al. 2024) |
| LinearRegression | – | classical | Linear Model | (Baseline) |

Table 3: Comparison of evaluated TSFMs details

higher ERR_THRESH allowed for more relaxed error filtering, further increasing the number of valid tracked points and enhancing coverage of the optical flow data at the potential cost of reduced accuracy. In corner detection, OpenCV's `goodFeaturesToTrack` function is applied, setting the `maxCorners` parameter to 30 as a trade-off between performance and efficiency, and the `qualityLevel` parameter to 0.01.

## Time Series Example

Extracting time-series information from videos requires meticulous manual verification. Here, we present an illustrative example of the extracted time series, as shown in Figure 3. We display ten object motion trajectories tracked from the video using the optical flow method. It can be observed that the temporal variation patterns along the X and Y axes may differ significantly. Some stationary time series are visible in the graph, such as uniform camera shake caused by human breathing, while there are also nonstationary sequences, which contribute to increasing the diversity of the dataset.

## Principal Component Analysis (PCA)

The similarity of the REAL-V-TSFM dataset is evaluated in comparison to other datasets, highlighting the enhanced diversity introduced by this newly constructed time series dataset. For both the REAL-V-TSFM and M4 datasets, we first normalize and then merge them prior to performing a PCA. When the two datasets exhibit a closer distribution in the principal projection space, their similarity is considered higher; conversely, the greater the divergence, the lower their similarity. This approach thus reflects whether the datasets reveal comparable structures within the same low-dimensional principal component space.

We align the two time series datasets to the same sequence length and performing min–max normalization on each sequence individually, and then we combined them and applied PCA. Two-dimensional kernel density estimates were then computed separately for the projections of each dataset. By overlaying semi-transparent heatmaps with contour lines in blue (M4) and orange (REAL-V-TSFM), we visually depict the distribution shapes, concentration regions, and overlap & divergence patterns of the two datasets in the first two principal component dimensions. As illustrated in Fig. 2, the two time series datasets demonstrate notable differences in their low-dimensional projections: REAL-V-TSFM is distributed more uniformly, whereas M4 displays a clear skew, with its distribution being sparse in the lower-left and upper-right regions of the PCA plane. This observation not only underscores the limited diversity of existing time series datasets, but also demonstrates the feasibility and significance of extracting diverse and information-rich time series directly from video data. Given the massive availability of video sources, the potential for extracting varied and valuable time series data is both substantial and promising.

## Does model size really matter?

We evaluated the full range of Chronos models of varying sizes and compared their performance against other models, as shown in Table 4. Our findings suggest that increasing model size leads to only limited performance gains. Moreover, the *decoder-only* TimesFM model demonstrates substantially better performance than other models on the M4 dataset, but its degraded performance on REAL-V-TSFM highlights limitations in its generalization ability. Interestingly, smaller models, such as the *tiny* variant, can achieve performance comparable to or even on par with larger counterparts such as *large* or *base*. Consequently, scaling laws do not appear to consistently hold for TSFMs, and this observation provides valuable guidance for continued exploration in this domain.

## Objects in REAL-V-TSFM

Figure 4 illustrates the performance of 3 models on videos containing different objects within the REAL-V-TSFM dataset, where the vertical axis represents the normalized sMAPE. It is evident that the performance varies considerably between different objects, which correspond to diverse types of time series. Interestingly, videos involving animals yield motion-derived time series that are particularly difficult to forecast, resulting in notably lower predictive ac-

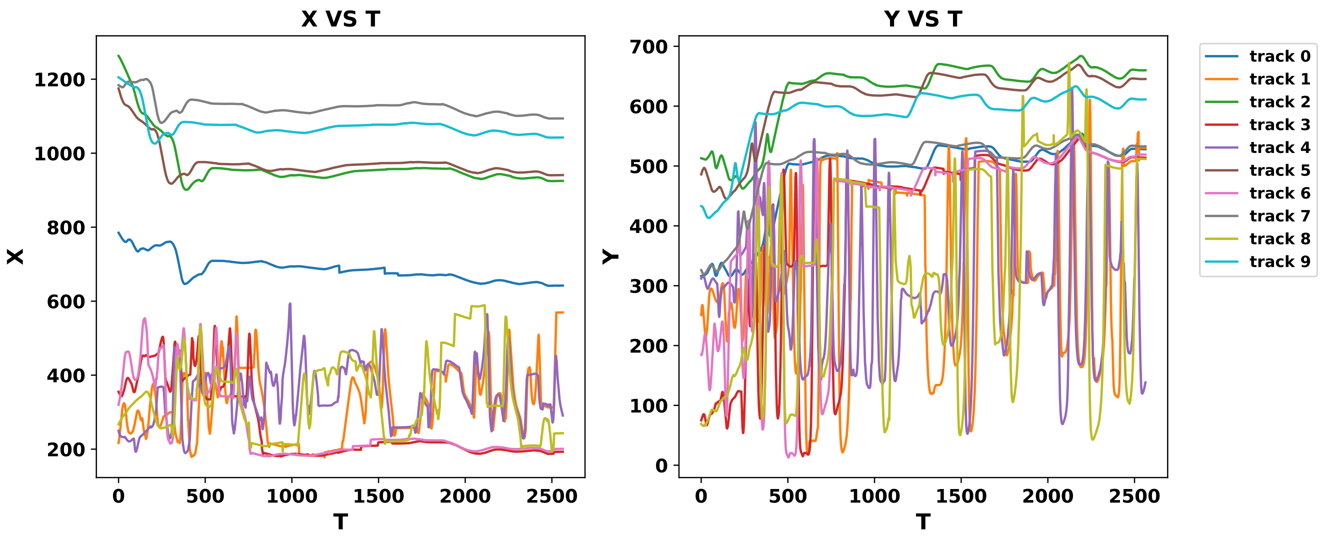

Figure 3: Ten time series data extracted from real video data

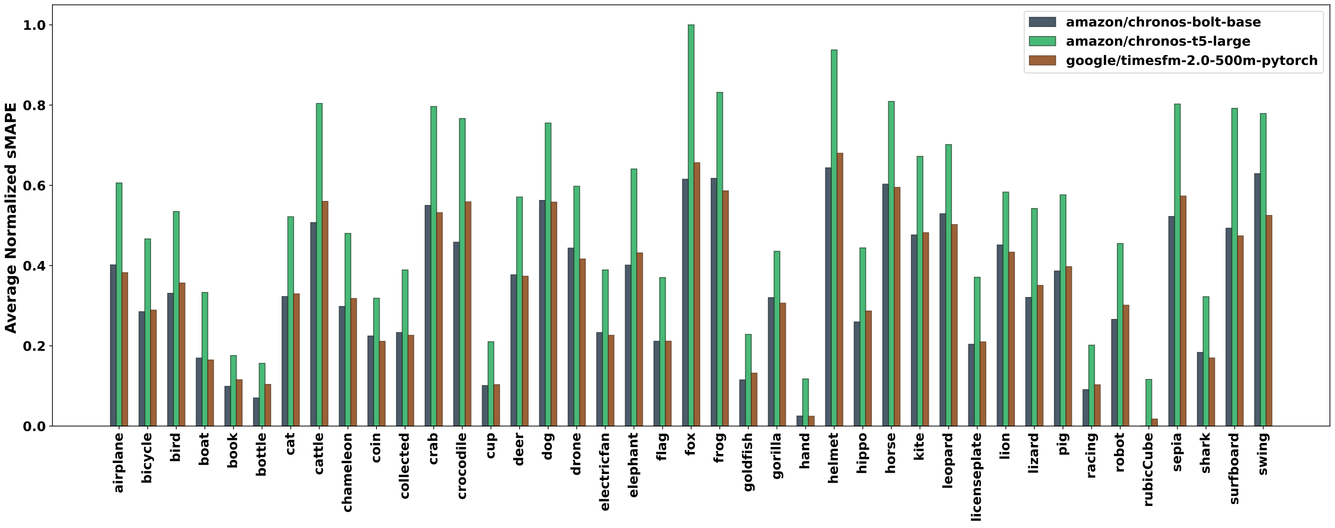

Figure 4: Performance comparison in different objects inside REAL-V-TSFM

| Model | Datasets | MAPE | sMAPE | Agg. Relative WQL | Agg. Relative MASE |
|---|---|---|---|---|---|
| amazon/chronos-bolt-tiny | REAL-V-TSFM | **7.43 ± 17.40** | **6.66 ± 9.92** | **0.92 ± 0.88** | **0.67 ± 0.62** |
| | M4-Weekly | 7.22 ± 5.33 | 7.12 ± 5.27 | 0.91 ± 0.96 | 0.59 ± 0.55 |
| | M4-Daily | 5.07 ± 1.03 | 5.14 ± 4.03 | **0.95 ± 0.77** | 0.63 ± 0.59 |
| | electricity_D | 6.79 ± 3.43 | **6.65 ± 3.18** | 0.82 ± 0.41 | 0.64 ± 0.38 |
| | LOOP_SEATTLE_D | **6.03 ± 2.39** | 6.15 ± 2.49 | 0.82 ± 0.09 | **0.81 ± 0.09** |
| amazon/chronos-bolt-mini | REAL-V-TSFM | **7.37 ± 17.22** | 6.65 ± 10.16 | **0.92 ± 0.90** | **0.67 ± 0.63** |
| | M4-Weekly | 6.88 ± 5.01 | **6.76 ± 4.90** | 0.90 ± 0.98 | 0.57 ± 0.57 |
| | M4-Daily | 5.01 ± 4.88 | 5.08 ± 4.78 | **0.97 ± 0.87** | 0.62 ± 0.62 |
| | electricity_D | 6.73 ± 3.26 | 6.61 ± 3.05 | 0.83 ± 0.43 | 0.65 ± 0.39 |
| | LOOP_SEATTLE_D | **6.69 ± 2.63** | **6.79 ± 2.73** | 0.91 ± 0.08 | **0.90 ± 0.08** |
| amazon/chronos-bolt-small | REAL-V-TSFM | **7.50 ± 18.80** | 6.55 ± 9.38 | **0.92 ± 0.88** | **0.67 ± 0.62** |
| | M4-Weekly | 6.64 ± 4.88 | 6.55 ± 4.86 | 0.87 ± 0.97 | 0.55 ± 0.55 |
| | M4-Daily | 5.00 ± 3.98 | 5.09 ± 3.88 | **0.96 ± 0.88** | 0.62 ± 0.45 |
| | electricity_D | **6.71 ± 3.16** | **6.60 ± 2.96** | 0.81 ± 0.42 | 0.64 ± 0.38 |
| | LOOP_SEATTLE_D | 6.58 ± 2.62 | **6.66 ± 2.70** | 0.89 ± 0.09 | **0.88 ± 0.09** |
| amazon/chronos-bolt-base | REAL-V-TSFM | **7.32 ± 17.06** | **6.57 ± 9.93** | **0.93 ± 0.90** | **0.67 ± 0.63** |
| | M4-Weekly | 5.72 ± 3.83 | 5.70 ± 3.83 | 0.79 ± 0.87 | 0.50 ± 0.49 |
| | M4-Daily | 4.93 ± 3.82 | 5.03 ± 3.22 | **1.00 ± 0.85** | 0.63 ± 0.59 |
| | electricity_D | 6.52 ± 3.03 | 6.44 ± 2.88 | 0.80 ± 0.41 | 0.62 ± 0.35 |
| | LOOP_SEATTLE_D | **6.65 ± 2.61** | **6.75 ± 2.70** | 0.90 ± 0.09 | **0.89 ± 0.09** |
| amazon/chronos-t5-tiny | REAL-V-TSFM | **10.40 ± 20.13** | 9.28 ± 10.85 | **5.40 ± 30.04** | **5.37 ± 33.57** |
| | M4-Weekly | **10.84 ± 8.94** | **10.41 ± 6.81** | 1.32 ± 1.30 | 0.87 ± 0.81 |
| | M4-Daily | 7.43 ± 3.28 | 7.44 ± 4.64 | 0.59 ± 0.89 | 1.00 ± 1.25 |
| | electricity_D | 10.12 ± 4.05 | **9.82 ± 3.82** | **1.35 ± 0.64** | 0.97 ± 0.55 |
| | LOOP_SEATTLE_D | 7.87 ± 3.17 | 8.09 ± 3.41 | 1.10 ± 0.25 | **1.07 ± 0.23** |
| amazon/chronos-t5-mini | REAL-V-TSFM | 9.36 ± 18.15 | 8.66 ± 10.26 | **5.55 ± 32.01** | **5.46 ± 34.38** |
| | M4-Weekly | **9.98 ± 6.57** | **9.77 ± 6.25** | 1.22 ± 1.13 | 0.80 ± 0.69 |
| | M4-Daily | 7.13 ± 4.57 | 7.09 ± 6.22 | **1.48 ± 1.32** | 0.93 ± 0.75 |
| | electricity_D | **9.69 ± 4.12** | **9.53 ± 3.77** | 1.31 ± 0.63 | 0.93 ± 0.54 |
| | LOOP_SEATTLE_D | 7.52 ± 3.03 | 7.74 ± 3.27 | 1.03 ± 0.20 | **1.01 ± 0.18** |
| amazon/chronos-t5-small | REAL-V-TSFM | 9.76 ± 19.56 | 8.68 ± 10.07 | **4.89 ± 30.06** | **4.93 ± 32.30** |
| | M4-Weekly | **9.96 ± 6.42** | **9.70 ± 6.12** | 1.26 ± 1.16 | 0.81 ± 0.70 |
| | M4-Daily | 7.26 ± 6.13 | 7.28 ± 5.22 | **1.53 ± 1.25** | 0.97 ± 0.77 |
| | electricity_D | **9.88 ± 3.74** | **9.64 ± 3.30** | 1.34 ± 0.67 | 0.97 ± 0.58 |
| | LOOP_SEATTLE_D | 7.54 ± 2.88 | 7.66 ± 2.98 | 1.05 ± 0.15 | **1.02 ± 0.13** |
| amazon/chronos-t5-base | REAL-V-TSFM | **9.56 ± 21.12** | 8.38 ± 9.92 | **5.57 ± 32.11** | **5.33 ± 33.89** |
| | M4-Weekly | 9.34 ± 6.41 | **9.17 ± 6.20** | 1.24 ± 1.23 | 0.79 ± 0.75 |
| | M4-Daily | 7.31 ± 6.28 | 7.34 ± 6.39 | **1.55 ± 1.45** | 0.98 ± 0.78 |
| | electricity_D | **9.41 ± 3.47** | **9.21 ± 3.24** | 1.25 ± 0.63 | 0.93 ± 0.57 |
| | LOOP_SEATTLE_D | 7.45 ± 2.86 | 7.60 ± 2.98 | 1.03 ± 0.17 | **1.01 ± 0.15** |
| amazon/chronos-t5-large | REAL-V-TSFM | **9.32 ± 18.46** | 8.40 ± 9.69 | **5.45 ± 31.23** | **5.58 ± 34.82** |
| | M4-Weekly | 8.85 ± 6.17 | **8.70 ± 6.06** | 1.19 ± 1.20 | 0.75 ± 0.76 |
| | M4-Daily | 7.11 ± 7.02 | 7.18 ± 4.65 | **1.56 ± 1.26** | **0.98 ± 0.88** |
| | electricity_D | **9.18 ± 3.85** | **8.99 ± 3.44** | 1.19 ± 0.56 | 0.88 ± 0.50 |
| | LOOP_SEATTLE_D | 7.06 ± 2.83 | 7.18 ± 2.92 | 0.98 ± 0.16 | 0.95 ± 0.13 |
| google/timesfm-1.0-200m-pytorch | REAL-V-TSFM | **58.29 ± 82.96** | 66.11 ± 70.55 | **32.20 ± 75.67** | 32.90 ± 79.88 |
| | M4-Weekly | **92.12 ± 103.91** | **73.04 ± 61.86** | 24.74 ± 61.39 | **25.38 ± 68.59** |
| | M4-Daily | 49.19 ± 55.93 | 55.26 ± 65.22 | 20.77 ± 75.33 | 22.55 ± 26.33 |
| | electricity_D | 36.88 ± 49.90 | 29.84 ± 33.19 | **29.84 ± 33.19** | 4.41 ± 17.42 |
| | LOOP_SEATTLE_D | 22.93 ± 9.86 | 19.85 ± 7.61 | 3.45 ± 1.22 | 3.09 ± 1.14 |
| google/timesfm-2.0-500m-pytorch | REAL-V-TSFM | **6.97 ± 16.63** | 6.24 ± 9.17 | **0.91 ± 1.02** | 0.64 ± 0.65 |
| | M4-Weekly | **8.30 ± 7.08** | **8.18 ± 6.70** | 0.85 ± 0.96 | 0.60 ± 0.62 |
| | M4-Daily | 1.9 ± 1.08 | 2.03 ± 2.55 | 0.39 ± 0.22 | 0.23 ± 0.25 |
| | electricity_D | 6.81 ± 3.63 | 6.77 ± 3.54 | 0.80 ± 0.44 | **0.63 ± 0.39** |
| | LOOP_SEATTLE_D | 6.79 ± 2.77 | **6.89 ± 2.86** | **0.92 ± 0.06** | **0.90 ± 0.07** |
| LinearRegression | REAL-V-TSFM | **15.52 ± 28.44** | **14.10 ± 20.21** | 1.00 ± 0.00 | 1.00 ± 0.00 |
| | M4-Weekly | **21.91 ± 23.49** | **19.96 ± 19.67** | 1.00 ± 0.00 | 1.00 ± 0.00 |
| | M4-Daily | 14.14 ± 24.99 | 14.6 ± 19.99 | 1.00 ± 0.00 | 1.00 ± 0.00 |
| | electricity_D | 13.56 ± 10.74 | 13.93 ± 14.34 | 1.00 ± 0.00 | 1.00 ± 0.00 |
| | LOOP_SEATTLE_D | 7.50 ± 2.97 | 7.63 ± 3.13 | 1.00 ± 0.00 | 1.00 ± 0.00 |

Table 4: Performance comparison across datasets and models.

curacy. By examining the videos, we observed that animal behaviors often exhibit unpredictable irregularities, particularly in species with a higher degree of freedom in body movement, such as frogs or foxes, which possess four limbs and a movable head. In contrast, species like birds demonstrate more predictable motion patterns during flight, characterized by regular wing flapping, making their motion-derived time series easier for TSFMs to forecast. In contrast, videos of inanimate and static objects, such as books, tend to produce more predictable time series, leading to higher performance. Overall, the dataset exhibits substantial diversity, which not only facilitates the evaluation of model generalization, but also provides a novel and valuable perspective on extracting time series from large-scale real-world data.