

variations, limiting their practicality to downstream tasks.

We note a crucial distinction of time series from images or languages, as each time step consists of only a finite number of scalar values. This implies that the most vital information in time series is preserved in the temporal correlations, highlighting the importance of temporal modeling (Nie et al., 2023; Wu et al., 2023a). Therefore, *the critical point of time series pre-training is optimizing encoders to accurately capture temporal correlations*, which has not been adequately addressed in previous masking or contrastive methods.

To address the insufficiency in temporal modeling, we present TimeSiam, a simple yet effective self-supervised pre-training framework. Unlike prior, as shown in Figure 1, TimeSiam proposes to sample pairs of subseries across different timestamps from the same time series, termed “*Siamese subseries*”. Then, it leverages Siamese networks as encoders to capture correlations between temporally distanced subseries. With simple data augmentation such as masking, TimeSiam further improves the diversity and distinctiveness of Siamese subseries, which natively derives a past-to-current reconstruction task, thereby enforcing the encoder to learn temporally related information and capture correlations among past and current series. Besides, to cover different distanced Siamese subseries, we propose learnable lineage embeddings to enhance the encoder capacity for learning diverse time-dependent representations. Eventually, a decoder that integrates cross-attention and self-attention mechanisms is applied to ensure a precise reconstruction of the (masked) Siamese subseries.

Importantly, TimeSiam is not constrained by proximity information in the time series. Instead, benefiting from our Siamese subseries sampling procedure, it can effectively model the correlation among distanced subseries, which empowers the model with a more thorough understanding of the whole time series. With the above designs, TimeSiam remains simple but achieves consistent state-of-the-art against prior time series pre-training methods across various downstream tasks, including time series forecasting and classification, covering both in- and cross-domain settings. Overall, our contributions are summarized as follows:

- In the spirit of learning temporal correlations, we propose TimeSiam, a simple but effective pre-training framework that leverages Siamese networks to capture correlations among temporally distanced subseries.
- With Siamese encoders to reconstruct current masked subseries based on past observation and lineage embeddings to capture subseries disparity, TimeSiam can learn diverse time-dependent representations.
- TimeSiam achieves consistent state-of-the-art fine-tuning performance across thirteen standard benchmarks, excelling in various time series analysis tasks.

2. Related Work

2.1. Time Series Self-supervised Pre-training

Self-supervised pre-training has demonstrated its ability to learn valuable and generalizable representations from large-scale unlabeled datasets in various domains, such as natural language processing (NLP) (Devlin et al., 2018; Radford et al., 2019; Raffel et al., 2020; Brown et al., 2020; Gao et al., 2020) and computer vision (CV) (He et al., 2020; Liu et al., 2021; Xie et al., 2022; He et al., 2022), which can significantly reduce labeling expenses and benefit diverse downstream tasks. Recently, self-supervised pre-training has empowered many breakthroughs in time series analysis by introducing well-established techniques into time series, such as masked modeling and contrastive learning.

Masked Modeling As a fundamental technique in self-supervised pre-training methods, masked modeling enables deep models to learn essential representations by reconstructing masked parts from the visible context. Drawing inspiration from notable advances in NLP and CV, extensive time series pre-training approaches focus on time series masked modeling, which helps the model learn effective time series representations to facilitate various downstream analysis tasks. For instance, TST (Zerveas et al., 2021) and Ti-MAE (Li et al., 2023b) propose to randomly mask segments and points in time series and pre-train the model with the reconstruction task. PatchTST (Nie et al., 2023) divides the temporal dimension into multiple patches and treats time series as independent variates. Additionally, it incorporates a non-overlapping patch-level masked self-supervised strategy for temporal representation learning. HiMTM (Zhao et al., 2024) proposes a novel hierarchical masked time series pre-training method to capture the multi-scale nature of time series. Additionally, SimMTM (Dong et al., 2023) introduces a multi-masking modeling paradigm, which reconstructs original time series through the weighted aggregation of multiple masked time series, thus being able to learn both points-wise and series-wise temporal representations. Despite these advances, all of these approaches solely focus on the modeling of one individual time series, disregarding the intrinsic temporal correlations and dynamical variations of the whole time series. In contrast, TimeSiam proposes to reconstruct the current sub-series based on past observations, which can naturally integrate the time-dependent information during reconstruction pre-training.

Contrastive Learning Unlike masked modeling, this approach enables model pre-training by optimizing the similarity among instance-level representations. It leverages different data augmentations to construct positive and negative pairs from data, where positive pairs are optimized to be close to each other and negative pairs are encouraged to be distant from each other during pre-training (Tang

et al., 2020; He et al., 2020; Chen & He, 2021; Gao et al., 2021). Current time series contrastive learning methods are mainly based on diverse data augmentations tailored to the domain-specific characteristics of time series. CPC (Oord et al., 2018) introduced contrastive predictive coding, which uses model-predicted timesteps as positive samples and randomly-sampled timesteps as negative samples to obtain advantageous time series representations for downstream tasks. Franceschi et al. (2019) combined a causal dilated convolutions-based encoder with a novel triplet loss that employs time-based negative sampling. TNC (Tonekaboni et al., 2021) learns the representations by ensuring that signals from within a neighborhood are distinguishable from the distribution of non-neighboring signals in the latent space using a debiased contrastive loss. TS2Vec (Yue et al., 2022) divides time series into patches, defining contrastive tasks at both the individual instance and patch levels. Mixing-up (Wickstrøm et al., 2022) exploits a data augmentation scheme in which new samples are generated by mixing two data samples. LaST (Wang et al., 2022) aims to separate seasonal and trend components in time series data within the latent space. Additionally, CoST (Woo et al., 2022) utilizes contrastive losses in both time and frequency domains to learn distinct seasonal and trend representations. Furthermore, TF-C (Zhang et al., 2022) introduces a novel time-frequency consistency architecture, optimizing for proximity between time-based and frequency-based representations of the same data sample. However, existing contrastive learning methods for time series heavily rely on intricate data augmentation techniques to generate diverse views of the original data for self-supervision. Also, the instance-level representation learning may fall short in downstream low-level tasks. In TimeSiam, we utilize the native temporally distanced subseries to build reconstruction tasks, thereby freeing from complex augmentation techniques and also considering the low-level representation.

2.2. Siamese Networks

Siamese networks (Bromley et al., 1993) are particular neural network architectures with shared model parameters. This design makes Siamese networks well-suited for comparing and distinguishing two input samples based on a single neural network. They have been widely used in contrastive learning to model the relationship between paired samples (Chen & He, 2021). The combination of Siamese networks and contrastive learning has been widely used in many applications, particularly in tasks requiring instance-level representations (Chen et al., 2020; He et al., 2020; Wang et al., 2023). However, in the field of time series pre-training, this combination generally focuses on recognizing subtle differences between various augmented views of the series itself, overlooking the essence of time series, that is temporal correlation modeling. In this paper, we

explore using shared-weight Siamese autoencoders to establish correlations between past and current subseries. This methodology enables a more efficient understanding of temporal relations in time series and enforces the model to learn time-dependent representations.

3. TimeSiam

To enhance the time-dependent representation learning, TimeSiam is designed to capture correlations between temporally distant subseries based on Siamese networks. This framework can natively derive a past-to-current reconstruction task with simple masked augmentation. In addition, learnable lineage embeddings are incorporated to dynamically capture the disparity among different distanced subseries pairs, which can enhance the model’s capacity to cover different temporal correlations. Hereafter, we will detail the pre-training and fine-tuning stages in TimeSiam.

3.1. Pre-training

TimeSiam pre-training involves the following two modules: Siamese subseries sampling and Siamese modeling.

Siamese Subseries Sampling Typically, previous time series pre-training approaches focus solely on modeling the individual series itself, neglecting the inherent correlations among temporally related time series. This deficiency in the pre-training phase will lead to insufficient extraction of generalizable time-dependent representations. In contrast, our TimeSiam is designed to focus on modeling temporal correlations of subseries across different timestamps, capturing the intrinsic time-correlated information of time series.

As shown in Figure 2, we construct Siamese subseries pairs by randomly sampling a past sample \mathbf{x}^{past} preceding the current sample \mathbf{x}^{curr} in the same time series. Each sample in a Siamese pair, termed “*Siamese subseries*” each other, contains T timestamps and C observed variables. Notably, one \mathbf{x}^{curr} can correspond to multiple \mathbf{x}^{past} subseries due to the random sampling process. We focus on constructing correlations and capturing temporal variations between these Siamese subseries, which benefits intrinsically time-dependent representation learning during pre-training. The relative distance between the past and current subseries, denoted as d , is crucial in representing the correlation and disparities between Siamese subseries. Furthermore, we adopt a simple masking augmentation to generate augmented current subseries $\tilde{\mathbf{x}}^{\text{curr}}$ that further improves the diversity and the disparity of Siamese subseries pairs, ensuring a more robust and sufficient pre-training phase. The above process can be formalized as follows:

$$(\mathbf{x}^{\text{past}}, \tilde{\mathbf{x}}^{\text{curr}}) = \text{Mask-Augment}((\mathbf{x}^{\text{past}}, \mathbf{x}^{\text{curr}})). \quad (1)$$

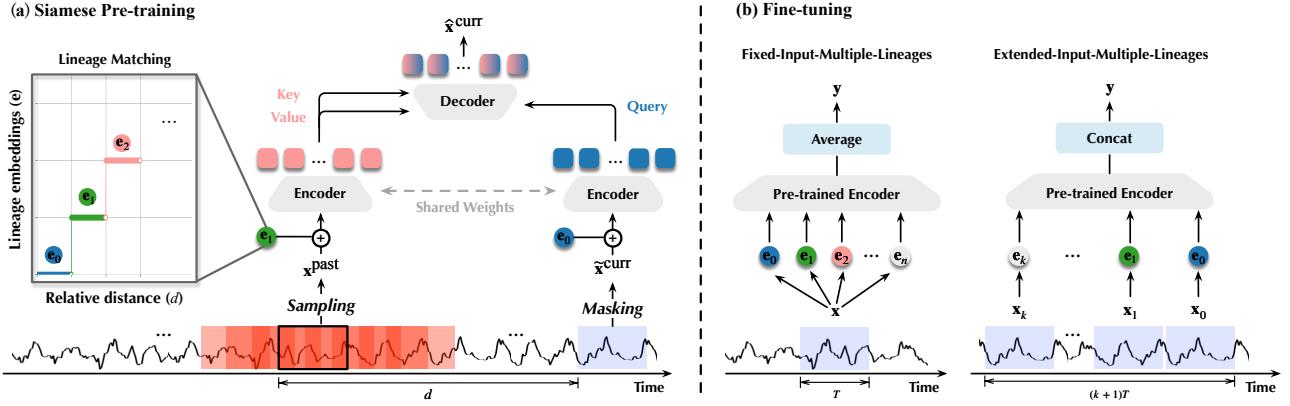


Figure 2. The overall design of TimeSiam, which establishes correlations between subseries randomly sampled from different timestamps using Siamese encoders. It integrates learnable lineage embeddings to enhance the capacity for temporal-related representation learning.

Siamese Modeling After constructing mask-augmented Siamese pairs, as shown in Figure 2, we further integrate learnable lineage embeddings during pre-training to effectively capture the disparity among different Siamese pairs. This design can enhance the model’s capacity to extract diverse temporal-related representations. Given N learnable lineage embeddings $\{e_i^{\text{lineage}}\}_{i=1}^N, e_i^{\text{lineage}} \in \mathbb{R}^{1 \times D}$ and D represents the dimension of lineage embeddings. For the past sample x^{past} , we apply the LineageMatching(\cdot) function to dynamically match a certain lineage embedding based on its temporal distance d to the current series. As for the current sample \tilde{x}^{curr} , we use a special lineage embedding to represent a degeneration situation as $d = 0$:

$$\begin{aligned} e_i^{\text{lineage}} &= \text{LineageMatching}(d) \\ z^{\text{past}} &= \text{Embed}(x^{\text{past}}) \oplus e_i^{\text{lineage}} \\ \tilde{z}^{\text{curr}} &= \text{Embed}(\tilde{x}^{\text{curr}}) \oplus e_0^{\text{lineage}}, \end{aligned} \quad (2)$$

where $e_0^{\text{lineage}} \in \mathbb{R}^{1 \times D}$ is the specific embedding for current subseries and $z^{\text{past}}, \tilde{z}^{\text{curr}} \in \mathbb{R}^{T \times D}$ denote the embedded Siamese features. Different base models correspond to different Embed(\cdot). Regarding PatchTST (Nie et al., 2023), the patch-wise embedding function PatchEmbed(\cdot) is used to divide each variable into several patches and each patch is mapped to a patch token. As for iTransformer (Liu et al., 2024), it uses the variable-wise embedding VariateEmbed(\cdot) and maps the entire variable into a temporal token. Note that lineage embeddings are used to identify the temporal distance between Siamese subseries. It is shared along the time dimension when being added to Siamese subseries features. Here, \oplus represents the addition operation with the temporal dimension broadcast.

Next, TimeSiam utilizes Siamese encoders to process pairs of Siamese pair features, which can be instantiated as advanced time series models, e.g. PatchTST (Nie et al., 2023) or iTransformer (Liu et al., 2024). After the Siamese encoder layer, we can obtain pairs of representations of past

and masked current subseries as follows:

$$h_e^{\text{past}} = \text{Encoder}(z^{\text{past}}), \tilde{h}_e^{\text{curr}} = \text{Encoder}(\tilde{z}^{\text{curr}}), \quad (3)$$

where $h_e^{\text{past}}, \tilde{h}_e^{\text{curr}} \in \mathbb{R}^{T \times D}$ are from the Siamese encoder.

Note that our Siamese sampling strategy natively derives a past-to-current reconstruction task. As shown in Figure 2, we use a decoder that integrates cross-attention and self-attention mechanisms (Vaswani et al., 2017) to incorporate past information into the current subseries for reconstruction, which can inherently capture the temporal correlations. Besides, this design can also enrich the limited context of masked current series to ensure accurate reconstruction for representation learning. Concretely, $\tilde{h}_e^{\text{curr}}$ serves as the query, and h_e^{past} acts as both the key and value, generating the decoder representation of the current time subseries, denoted as \hat{h}_d . This representation undergoes further refinement through a self-attention layer and a Feed-Forward Network (FFN). We formalize the decoder process as follows:

$$\begin{aligned} \hat{h}_d &= \text{LayerNorm} \left(\tilde{h}_e^{\text{curr}} + \text{Cross-Attn} (\tilde{h}_e^{\text{curr}}, h_e^{\text{past}}, h_e^{\text{past}}) \right) \\ h'_d &= \text{LayerNorm} \left(\hat{h}_d + \text{Self-Attn} (\hat{h}_d, \hat{h}_d, \hat{h}_d) \right) \\ h_d &= \text{LayerNorm} \left(h'_d + \text{FFN} (h'_d) \right). \end{aligned} \quad (4)$$

We summarize this process as $h_d = \text{Decoder}(\tilde{h}_e^{\text{curr}}, h_e^{\text{past}})$. Finally, the output of the decoder $h_d \in \mathbb{R}^{T \times D}$ is used to reconstruct the masked current subseries through a linear projection layer, which can be formalized as:

$$\hat{x}^{\text{curr}} = \text{Projector}(h_d). \quad (5)$$

Benefiting from our design, TimeSiam can be supervised by a simple reconstruction loss function and inherently learn time-dependent representations by past-to-current temporal correlation modeling. The loss for each Siamese pair is

$$\mathcal{L}_{\text{reconstruction}} = \|x^{\text{curr}} - \hat{x}^{\text{curr}}\|_2^2. \quad (6)$$

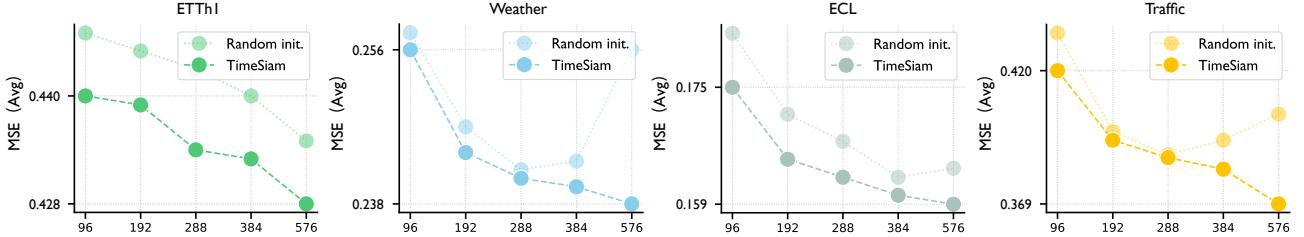


Figure 4. Fine-tuning the pre-trained model to the inputs with extended length {96, 192, 288, 384, 576} based on iTransformer (Liu et al., 2024). The MSE averaged from all predicted horizons {96, 192, 336, 720} is reported. Additional results are in the Appendix F.

in Table 7, reveal that the model performance is promoted significantly, empowering by TimeSiam pre-training (TimeSiam vs. random init.). Furthermore, although TSLD_{500M} does not exhibit a significant advantage over TimeSiam under the in-domain setting initially, we observed a marked enhancement in performance as the dataset size increased (TSLD_{1G} vs. TSLD_{500M}), and TimeSiam_{Large} significantly outperforms TimeSiam_{Base} in the TSLD_{1G} finetuning scenarios, especially in the Traffic benchmark. This observation highlights the efficacy of TimeSiam and the positive correlation between data-model size and the final performance.

Adapt to Extended-Length Input As illustrated in Eq. (8), TimeSiam can natively adapt to longer inputs. Figure 4 shows that the standard prediction framework may degenerate under extended input length, which may be because of the noises in longer series. Contrarily, benefiting from an ingenious integration of Siamese modeling and lineage embeddings, TimeSiam achieves more accurate predictions, even when predicting from extended input series.

Linear Probing As an important finetuning setting, we also experiment with the linear probing, where we fix the pre-trained encoder and only finetune the newly added projector at the end of the model. Figure 5 illustrates that TimeSiam demonstrates superior performance compared to other baselines in terms of overall *linear probing* performance. Interestingly, by only fine-tuning the model head, the average forecasting performance across the four ETT subsets is already comparable with the results obtained through full fine-tuning, and significantly outperforms training from random initialization (MSE: 0.365 vs. 0.383). This finding further validates the effectiveness of TimeSiam in learning generalizable representations for various downstream tasks.

Embedding Effectiveness To elucidate the advantages of employing varying numbers of lineage embeddings within a fixed sampling range for prediction, as illustrated in Figure 6, our findings consistently demonstrate that the incorporation of lineage embeddings enhances prediction performance. Furthermore, augmenting the number of embeddings to encompass a greater extent of lineage within reasonable limits reinforces the efficacy of long-term prediction. Experimental results validate that lineage embeddings introduce more

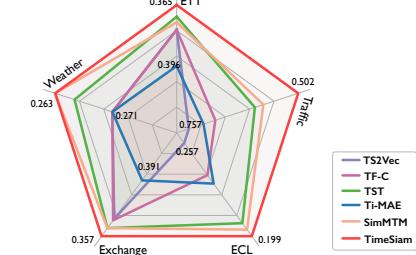


Figure 5. Linear probing on in-domain forecasting setting. Average results (MSE) are reported. Full results are shown in Table 17.

diverse temporal semantic information, enabling discrimination between different temporally distanced Siamese series, thereby boosting long-term prediction outcomes.

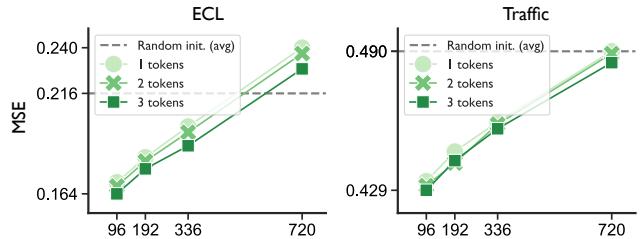


Figure 6. Increasing number of lineage embeddings on the ECL and Traffic. All results are under the “input-96” in-domain setting.

5. Conclusion

This paper proposes a simple, effective self-supervised pre-training framework named TimeSiam, uniquely focusing on temporal correlation modeling. TimeSiam employs Siamese networks as share encoders for randomly sampled past and current Siamese subsamples. It further enhances data diversity through masking augmentation, which can also foster time-dependent representation learning by reconstructing current subsamples from past observations. Additionally, we implement learnable lineage embeddings that efficiently capture disparities among Siamese subsamples under different distances, enhancing the model’s ability to cover diverse temporal correlations. Experimentally, TimeSiam demonstrated remarkable performance on various time series analysis tasks, consistently outperforming existing state-of-the-art baselines in both in- and cross-domain scenarios.

Acknowledgments

This work was supported by the National Key Research and Development Plan (2021YFB1715200), the National Natural Science Foundation of China (62022050 and U2342217), the BNRist Innovation Fund (BNR2024RC01010), and the National Engineering Research Center for Big Data Software. We are grateful to our colleagues Yipeng Huang and Zhiyao Cen for their support in the experimental efficiency.

Impact Statement

This paper focus on developing practical time series pre-training methods. We presents a novel approach based on Siamese networks, which could provide some inspiration for future research. The experimental results demonstrate the effectiveness of our approach across various domains and its potential value for real-world applications. It is essential to note that our work focuses solely on scientific issues, and we also ensure that ethical considerations are carefully taken into account. All the medical-related datasets are publicly available for scientific research. Thus, we believe that there is no ethical risk associated with our research.

References

- Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.
- Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. Signature verification using a "siamese" time delay neural network. In *NeurIPS*, 1993.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. *NeurIPS*, 2020.
- Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.
- Chen, X. and He, K. Exploring simple siamese representation learning. In *CVPR*, 2021.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2018.
- Dong, J., Wu, H., Zhang, H., Zhang, L., Wang, J., and Long, M. Simmtm: A simple pre-training framework for masked time-series modeling. In *NeurIPS*, 2023.
- Escudero, J., Abásolo, D., Hornero, R., Espino, P., and López, M. Analysis of electroencephalograms in alzheimer's disease patients with multiscale entropy. *Physiological measurement*, 27(11):1091, 2006.
- Franceschi, J.-Y., Dieuleveut, A., and Jaggi, M. Unsupervised scalable representation learning for multivariate time series. In *NeurIPS*, 2019.
- Gao, T., Fisch, A., and Chen, D. Making pre-trained language models better few-shot learners. In *IJCNLP*, 2020.
- Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*, 2021.
- Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.
- He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.
- He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.
- Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *SIGIR*, 2018.
- Li, Y., Fan, H., Hu, R., Feichtenhofer, C., and He, K. Scaling language-image pre-training via masking. In *CVPR*, 2023a.
- Li, Z., Rao, Z., Pan, L., Wang, P., and Xu, Z. Ti-mae: Self-supervised masked time series autoencoders. *arXiv preprint arXiv:2301.08871*, 2023b.
- Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., and Tang, J. Self-supervised learning: Generative or contrastive. *TKDE*, 2021.
- Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. itransformer: Inverted transformers are effective for time series forecasting. In *ICLR*, 2024.
- Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.
- Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.
- PeMS. Traffic Dataset. <http://pems.dot.ca.gov/>.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. In *OpenAI blog*, 2019.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(1):5485–5551, 2020.
- Tang, C. I., Perez-Pozuelo, I., Spathis, D., and Mascolo, C. Exploring contrastive learning in human activity recognition for healthcare. In *NeurIPS*, 2020.
- Tonekaboni, S., Eytan, D., and Goldenberg, A. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.
- UCI. UCI Electricity Load Time Series Dataset. <https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014>.
- Van Dijk, H., Van Wingen, G., Denys, D., Olbrich, S., Van Ruth, R., and Arns, M. The two decades brainclinics research archive for insights in neurophysiology (tdbrain) database. *Scientific data*, 9(1):333, 2022.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.
- Wang, Y., Han, Y., Wang, H., and Zhang, X. Contrast everything: A hierarchical contrastive framework for medical time-series. In *NeurIPS*, 2023.
- Wang, Z., Xu, X., Zhang, W., Trajcevski, G., Zhong, T., and Zhou, F. Learning latent seasonal-trend representations for time series forecasting. In *NeurIPS*, 2022.
- Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., and Xu, H. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.
- Wetterstation. Weather Dataset. <https://www.bgc-jena.mpg.de/wetter/>.
- Wickstrøm, K., Kampffmeyer, M., Mikalsen, K. Ø., and Jenssen, R. Mixing up contrastive learning: Self-supervised representation learning for time series. *PRL*, 2022.
- Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *ICLR*, 2022.
- Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, 2021.
- Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023a.
- Wu, H., Zhou, H., Long, M., and Wang, J. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, 2023b.
- Xiao, T., Wang, X., Efros, A. A., and Darrell, T. What should not be contrastive in contrastive learning. In *ICLR*, 2021.
- Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022.
- Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., and Xu, B. TS2Vec: Towards Universal Representation of Time Series. In *AAAI*, 2022.
- Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., and Eickhoff, C. A transformer-based framework for multivariate time series representation learning. In *SIGKDD*, 2021.
- Zhang, X., Zhao, Z., Tsiligkaridis, T., and Zitnik, M. Self-supervised contrastive pre-training for time series via time-frequency consistency. In *NeurIPS*, 2022.
- Zhao, S., Jin, M., Hou, Z., Yang, C., Li, Z., Wen, Q., and Wang, Y. Himtm: Hierarchical multi-scale masked time series modeling for long-term forecasting. *arXiv preprint arXiv:2401.05012*, 2024.
- Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.
- Zhou, T., Niu, P., Wang, X., Sun, L., and Jin, R. One fits all: Power general time series analysis by pretrained lm. In *NeurIPS*, 2023.

A. Implementation Details

In this paper, all experiments were conducted on a single NVIDIA A100 SXM4 80GB GPU and implemented using the PyTorch framework (Paszke et al., 2019) for five repetitions. We evaluated performance using Mean Square Error (MSE) and Mean Absolute Error (MAE) for time series forecasting. For classification tasks, we comprehensively assessed model performance by measuring accuracy, precision, recall, F1 score, AUROC, and AUPRC.

A.1. Baseline Implementation

We have followed and compared the official implementations of all baselines to our approach. We have maintained the original configurations outlined in these papers to ensure a fair comparison. Note that we utilized an unofficial coding version of Ti-MAE (Li et al., 2023b) due to the unavailability of its official open-source implementation.

Table 9. Categories and open-source implementations of all baselines.

CATEGORIES	METHODS	OFFICIAL CODE LINK
Contrastive Learning	TS2VEC (Yue et al., 2022)	https://github.com/yuezhihan/ts2vec
	CoST (Woo et al., 2022)	https://github.com/salesforce/CoST
	LAST (Wang et al., 2022)	https://github.com/zhycs/LaST
	TF-C (Zhang et al., 2022)	https://github.com/mims-harvard/TFC-pretraining
	COMET (Wang et al., 2023)	https://github.com/DL4mHealth/COMET
Masked Modeling	TST (Zerveas et al., 2021)	https://github.com/gzerveas/mvts_transformer
	TI-MAE (Li et al., 2023b)	https://github.com/asmodaay/ti-mae
	PATCHTST (Nie et al., 2023)	https://github.com/yuqinie98/PatchTST
	SIMMTM (Xie et al., 2022)	https://github.com/thuml/simmtm

A.2. Training Configuration

We construct two types of pre-training and fine-tuning scenarios, in-domain and cross-domain, based on benchmarks for prediction and classification tasks to compare the effectiveness of our method with other time series pre-training methods. In the pre-training phase, we pre-train the model with different learning rates and batch sizes based on the pre-trained dataset. We then fine-tune it for downstream prediction and classification tasks supervised by L2 and Cross-Entropy loss, respectively. The configuration details are in Table A.3. Also, considering the size of the fine-tuned dataset and consistency with existing works, we fine-tune the model for 10 epochs for the prediction task and 50 epochs for the classification task.

Table 10. Pre-training and fine-tuning configurations in forecasting and classification tasks.

TASKS	PRE-TRAINING			FINE-TUNING			
	learning rate	batch size	epochs	learning rate	loss function	batch size	epochs
Forecasting	1e-4	32	50	1e-4	L2	{8, 16, 32}	10
Classification	1e-4	256	100	1e-4	Cross-Entropy	{32, 64, 128}	50

A.3. Model Configuration

We compare TimeSiam against eight state-of-the-art baselines for an unbiased and comprehensive comparison. To ensure the fairness of the evaluation, we choose state-of-the-art time series analysis models as a unified backbone for these pre-trained methods. Specifically, PatchTST (Nie et al., 2023) and iTransformer (Liu et al., 2024) are adopted for forecasting and employ Temporal Convolution Network (TCN) (Bai et al., 2018) for classification following the setup in (Wang et al., 2023).

In addition, we performed a hyperparameter search for all baselines, adhering to their official configuration in the in-domain setting. For Siamese encoders, we explored various configurations by adjusting the number of encoder layers (e_{layers}) and decoder layers (d_{layers}) from $\{1, 2, 3, 4\}$, selecting hidden dimensions (d_{model}) from $\{16, 32, 64, 128, 256, 512\}$ and attention heads (n_{heads}) from $\{8, 16, 32\}$. In the case of TCN models, we investigated different numbers of residual blocks, considering configurations of $\{5, 8, 10\}$. During the fine-tuning stage, we carefully consider the learning rate (lr) from $\{1e-3, 5e-4, 1e-4, 1e-5\}$, and head dropout (dropout) from $\{0, 0.1, 0.2, 0.3\}$ in order to enhance the adaptability of our pretrained model to diverse datasets.

Primarily, two model configurations with different sizes are explored in the cross-domain forecasting setting, that is **TimeSiam-Base** and **TimeSiam-Large**. These two models are used to evaluate the impact of model capacity on forecasting performance, specifically in the context of cross-domain pre-training and fine-tuning on large-scale data.

Table 11. Two experimental configurations of TimeSiam with different model sizes.

TYPES	CONFIGURATION					PARAMETERS
	e_{layers}	d_{layers}	d_{model}	d_{ff}	n_{heads}	
TimeSiam _{Base}	3	1	128	256	8	709,344
TimeSiam _{Large}	5	2	128	1024	16	2,554,720

B. Dataset Description

We conduct experiments on eleven well-established datasets and two newly constructed datasets covering two primary tasks in time series analysis: forecasting and classification. These datasets cover a variety of application scenarios, different types of signals, multivariate channel dimensions, varying time series lengths, large-span sampling frequencies, and different data sizes. The detailed descriptions of these datasets are summarized in Table 12.

B.1. Forecasting Datasets

- (1) **ETT (4 subsets)** (Zhou et al., 2021) contains a group of four subsets oil temperature and power load collected by electricity transformers from July 2016 to July 2018 with minutes or hourly recorded frequency.
- (2) **Weather** (Wetterstation) includes meteorological time series with 21 weather indicators collected every 10 minutes from the Weather Station of the Max Planck Biogeochemistry Institute in 2020.
- (3) **Electricity** (UCI) records the hourly electricity consumption of 321 clients from 2012 to 2014.
- (4) **Traffic** (PeMS) encompasses the hourly measures of road occupancy rates obtained from 862 sensors situated in the San Francisco Bay area freeways between January 2015 and December 2016.
- (5) **Exchange** (Lai et al., 2018) records the daily exchange rates of eight different countries ranging from 1990 to 2016.

B.2. Classification Datasets

- (1) **AD** (Escudero et al., 2006) has electroencephalography (EEG) recordings from 12 Alzheimer’s patients and 11 healthy controls. Each patient has around 30 trials, each lasting for 5 seconds with 1280 timestamps (sampled at 256Hz) and includes 16 channels.

Table 12. Dataset descriptions. *Samples* are organized in (Train/Validation/Test).

TASKS	DATASETS	CHANNELS	SERIES LENGTH	SAMPLES	CLASSES	INFORMATION	FREQUENCY
Forecasting	ETTh1,ETTh2	7	{96,192,336,720}	8,545/2,881/2,881	-	Electricity	Hourly
	ETTm1,ETTm2	7	{96,192,336,720}	34,465/11,521/11,521	-	Electricity	15 Mins
	Weather	21	{96,192,336,720}	36,792/5,271/10,540	-	Weather	10 Mins
	Exchange	8	{96,192,336,720}	5,120/665/1,422	-	Exchange rate	Daily
	Electricity	321	{96,192,336,720}	18,317/2,633/5261	-	Electricity	Hourly
	Traffic	862	{96,192,336,720}	12,185/1,757/3,509	-	Transportation	Hourly
	TSLD-500M	1	{96,192,336,720}	369,030/31,872/-	-	Multi-domain	Mixing
	TSLD-1G	1	{96,192,336,720}	13,984,175/1,061,806/-	-	Multi-domain	Mixing
Classification	AD	16	256	4,329/891/747	3	EEG	256 Hz
	TDBrain	33	256	8,208/1,824/1,824	3	EEG	500 Hz
	PTB	15	300	53,950/3,400/5,020	3	ECG	1000 Hz

(2) **PTB** (Goldberger et al., 2000) has electrocardiogram (ECG) recordings from 290 patients with 15 channels sampled at 1000 Hz. This paper focuses on a subset of the dataset that includes 198 patients with heart diseases: Myocardial infarction and healthy controls.

(3) **TDBrain** (Van Dijk et al., 2022) monitors brain signals of 1274 patients with 33 channels during EC (Eye closed) and EO (Eye open) tasks. It includes 60 types of diseases, but this paper focuses on a subset of 25 Parkinson’s disease patients and 25 healthy controls. Only the EC task trials are used for representation learning.

B.3. Merged Large Scale Datasets

To further substantiate the significance of time series pre-training on large-scale data and showcase its benefits in diverse and extensive datasets, we have amalgamated multiple non-overlapping time series datasets from various domains to construct the **Time Series Large Datasets (TSLD)**. In this paper, we present two versions of TSLD to validate our approach.

(1) **TSLD-500M** is a composite dataset comprising 400,902 samples from 12 time series datasets across the domains of Electricity, Transport, Energy, Climate, and others.

(2) **TSLD-1G**, building upon the TSLD-500M dataset, incorporates additional diverse datasets from domains such as Society, IoT, and Web. With an impressive sample count of 15,045,981 observations, TSLD-1G surpasses the size of datasets commonly used in time series analysis and provides greater diversity.

C. Masking Strategy

In this paper, we explored five different mask rules: binomial, channel binomial, continuous, channel continuous, and only masking the last to assess their impact on TimeSiam, illustrated in Figure 7.

(1) **Binomial masking:** Generate a mask by employing a binomial distribution across all channels within a given sample.

(2) **Channel binomial masking:** Generate a mask based on a binomial distribution that selectively masks individual channels at different timestamps within the sample.

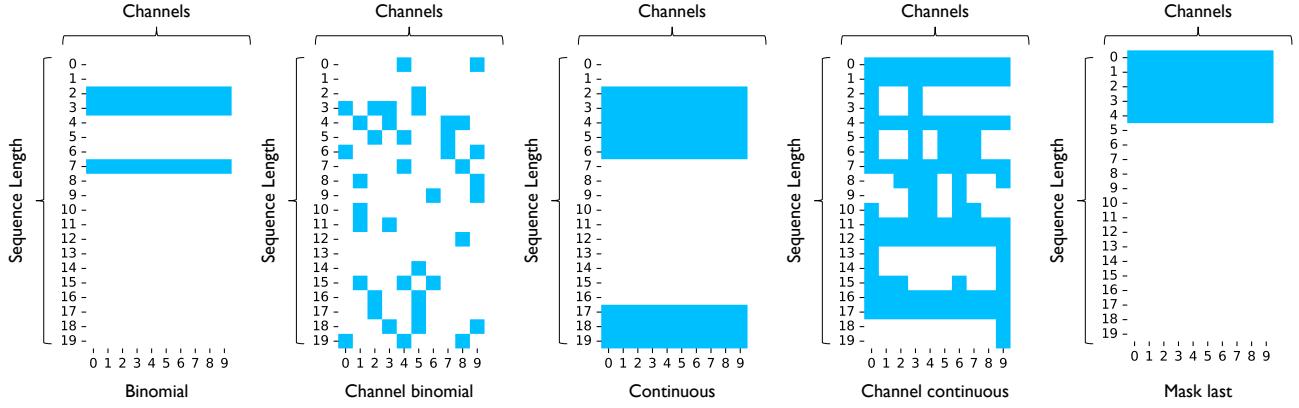


Figure 7. Showcases of various masking rules (75% masked ratio). The x-axis shows the channels and the y-axis represents the sequence length of the time series. Blue blocks indicate unmasked time stamps while white blocks represent masked ones.

- (3) **Continuous masking:** Generate a mask by employing a geometric distribution across all channels within a given sample.
- (4) **Channel continuous masking:** Generate a mask based on a geometric distribution that selectively masks individual channels at different timestamps within the sample.
- (5) **Masking last:** Only mask the tail of time series in all channels.

D. Linear Probing and Full Fine-tuning

The results depicted in Figure 8 unequivocally demonstrate that both fine-tuning and linear probing methodologies utilizing TimeSiam outperform fully supervised learning from random initiation. Moreover, the findings suggest that full fine-tuning consistently yields superior results compared to linear probing across most datasets, with ETTh2 being a notable exception, where both approaches exhibit comparable performance.

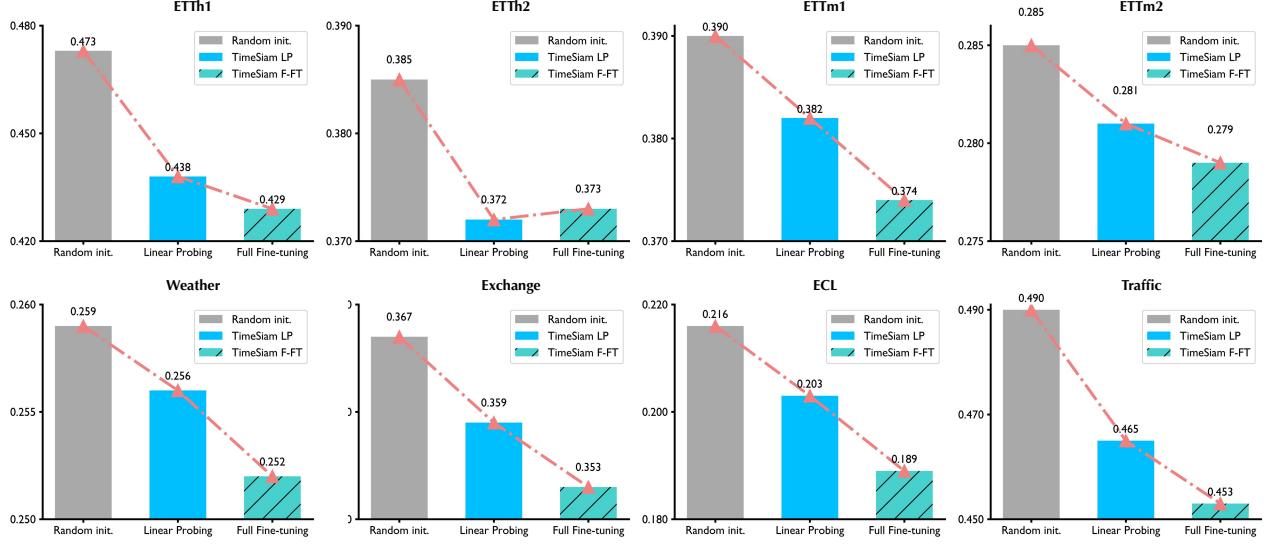


Figure 8. Comparison is made between the performance of linear probing pre-trained from TSLD-1G on various datasets and pre-training followed by fine-tuning on the same dataset. The mean squared error (MSE) is computed across all prediction lengths and serves as a measure of performance. A lower averaged MSE indicates superior predictive capability.

E. Multiple Lineages Representation Visualization

We employ Principal Components Analysis (PCA) to elucidate the distribution of temporal representations on the ECL dataset. We will only train the learned lineage embeddings during the pre-training phase. However, during downstream fine-tuning or linear probing, we will keep them fixed and not update them. It is worth noting that the embedded feature will be the same without different lineage embeddings. However, when time series is fed into a pre-trained Siamese network with different lineage embeddings, the model generates divergent temporal representations that representations derived from the same lineage embeddings tend to be closely clustered together, while representations from different lineage embeddings exhibit significant dissimilarity. Upon visual analysis, we have observed that the representations generated based on the same data but with different lineage embeddings exhibit a high level of diversity. This observation effectively validates the effectiveness of combining a pre-trained Siamese network with different lineage embeddings, which can enlarge the representation diversity.

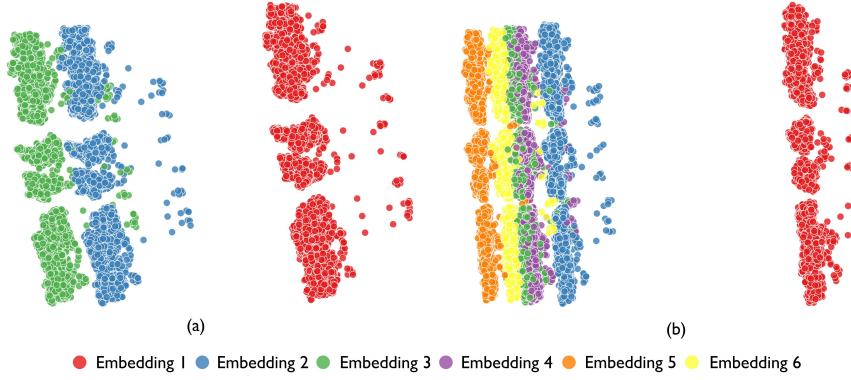


Figure 9. Visualizing the effect of temporal shift representations. (a) Visualization of test distribution under three types of lineage embeddings for ECL. (b) Visualization of test distribution under six types of lineage embeddings for ECL.

F. Adapt Extended Input Length

To facilitate the performance of Timesiam on fine-tuning scenarios with extended input lengths, we choose the input length to be an integral multiple of the pre-training length. In practice, the series length is not restricted to be an integral multiple of the pre-training length. TimeSiam can handle flexible input lengths, as different lineage embeddings can be shared across different time segments.

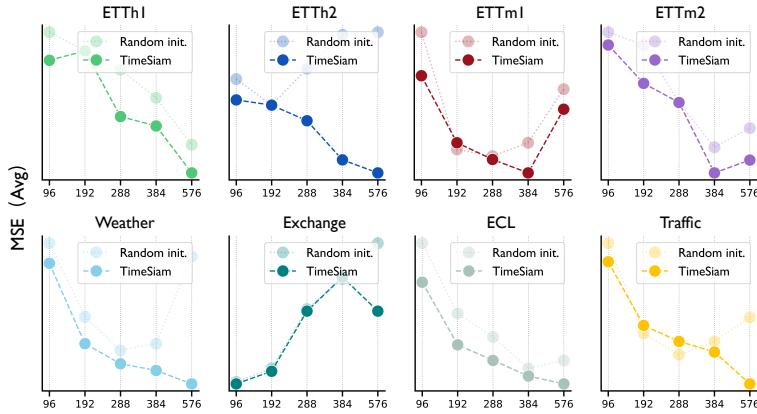


Figure 10. Full results for fine-tuning the pre-trained model with extended input length, where the input length is selected from $\{96, 192, 288, 384, 576\}$. The MSE averaged from four future lengths $O \in \{96, 192, 336, 720\}$ is reported.

G. Full Results

Due to the limited length of the text, we summarize the main experiments as follows:

Table 13. The main results for time series forecasting and classification tasks.

EXPERIMENTS CATEGORIES	TASKS	EVALUATION	TABLES NAME
The main experiment	Forecasting	In-domain	Table 14, 15
		Cross-domain	Table 16
	Classification	In-domain	Table 18
		Cross-domain	Table 19

H. ShowCases

H.1. Different Masked Ratios

To investigate the reconstruction process of TimeSiam, we visually represent past time series, masked current time series, and reconstructed current time series with varying mask ratios using validation data from diverse datasets. Figure 11 demonstrates the reconstruction effects of TimeSiam at different mask ratios applied to the current time series. The context information is obtained by random sampling based on the current series, and the reconstruction becomes more challenging as the mask ratio increases due to the limited available information. Nevertheless, our TimeSiam model consistently achieves accurate reconstruction of masked current time series despite the scarcity of data and significant variation in temporal dimension between past and present. This accomplishment highlights the effectiveness of our approach in learning internal time-dependent representations through a past-to-current reconstruction.

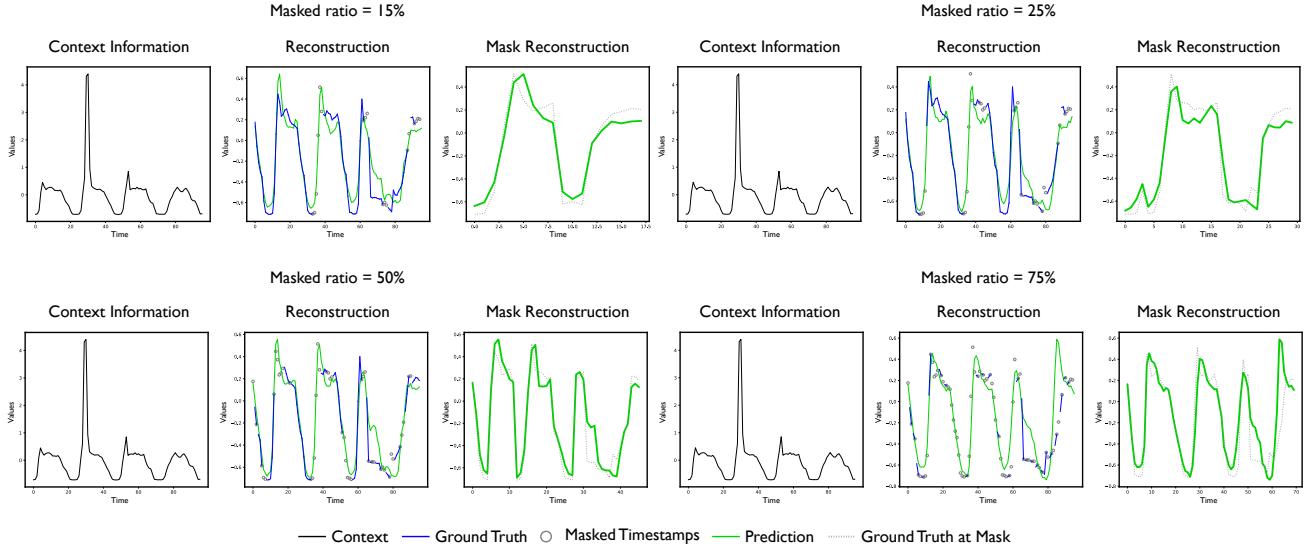


Figure 11. Showcases of TimeSiam in reconstructing time series with different masked ratios from Traffic.

H.2. Different Datasets

We further demonstrate the reconstruction effect across various datasets with different data distributions, as detailed below.

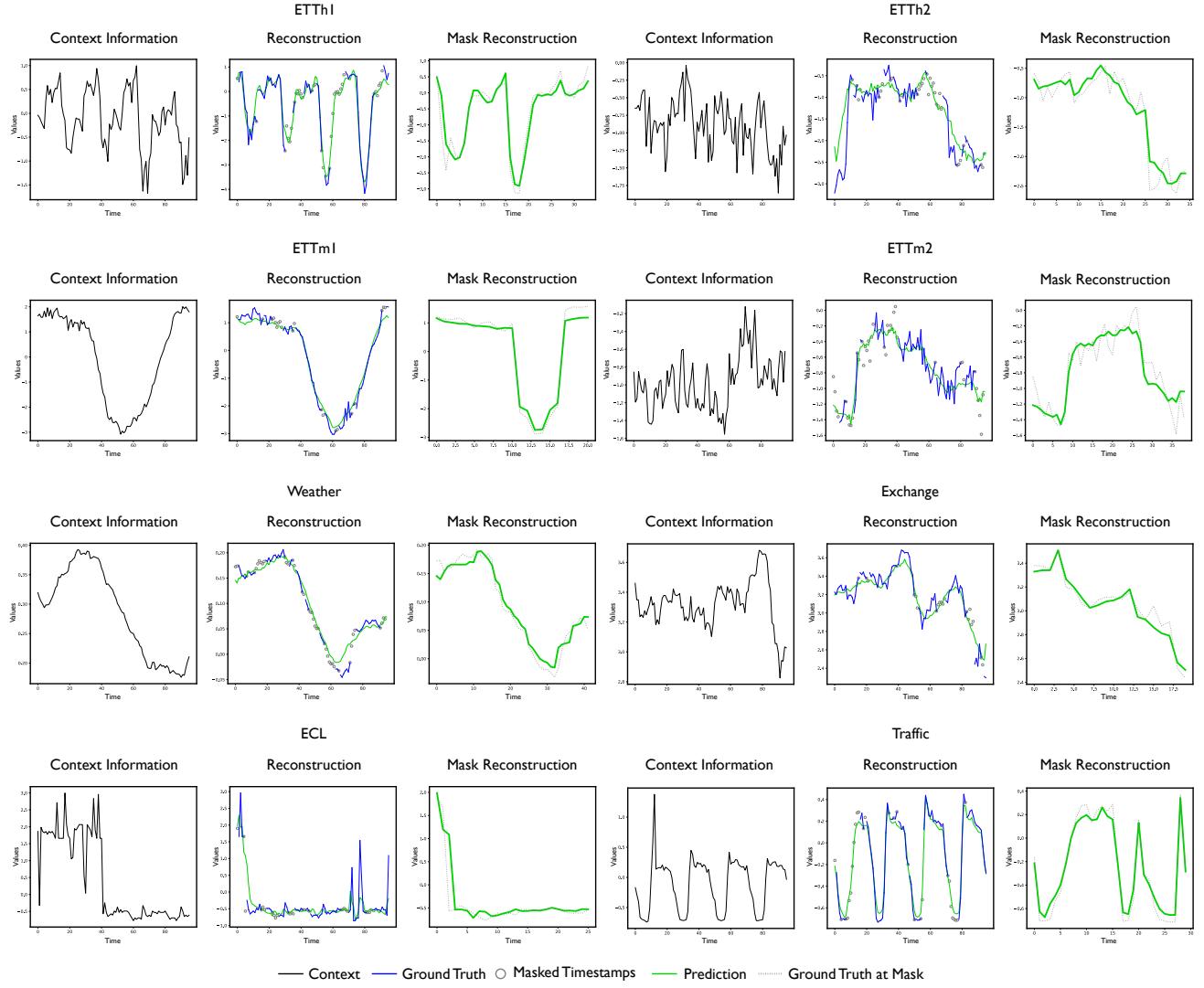


Figure 12. Showcases of TimeSiam in reconstructing time series from different datasets with 25% masked ratio.

Table 20. Ablation studies were conducted on TimeSiam. “W/o Siamese” refers to solely focusing on modeling subseries itself, without incorporating Siamese modeling. “W/o Masking” indicates the absence of mask augmentation in the current subseries.

INPUT-96		RANDOM INIT.		W/O SIAMESE		W/O MASKING		TIMESIAM	
PREDICT- <i>O</i>		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTH1	96	0.420	0.423	0.377	0.401	0.381	0.403	0.378	0.401
	192	0.465	0.449	0.423	0.430	0.430	0.431	0.422	0.430
	336	0.504	0.470	0.458	0.451	0.466	0.452	0.459	0.452
	720	0.502	0.492	0.471	0.478	0.470	0.474	0.459	0.437
	Avg	0.473	0.458	0.432	0.440	0.437	0.440	0.429	0.437
ETTH2	96	0.297	0.345	0.291	0.346	0.289	0.339	0.293	0.345
	192	0.388	0.400	0.375	0.396	0.378	0.393	0.370	0.392
	336	0.426	0.434	0.416	0.432	0.412	0.426	0.410	0.424
	720	0.431	0.446	0.421	0.446	0.420	0.441	0.418	0.440
	Avg	0.385	0.406	0.376	0.405	0.375	0.400	0.373	0.400
ETTM1	96	0.330	0.368	0.317	0.359	0.333	0.368	0.319	0.360
	192	0.369	0.385	0.363	0.387	0.367	0.385	0.353	0.379
	336	0.400	0.407	0.385	0.403	0.400	0.409	0.383	0.402
	720	0.460	0.439	0.444	0.438	0.459	0.442	0.440	0.436
	Avg	0.390	0.400	0.377	0.397	0.390	0.410	0.374	0.394
ETTM2	96	0.175	0.258	0.177	0.262	0.177	0.261	0.175	0.261
	192	0.247	0.307	0.241	0.303	0.243	0.303	0.241	0.303
	336	0.309	0.345	0.302	0.343	0.307	0.347	0.300	0.341
	720	0.408	0.403	0.398	0.398	0.405	0.404	0.399	0.398
	Avg	0.285	0.328	0.280	0.327	0.283	0.329	0.279	0.326
WEATHER	96	0.177	0.218	0.174	0.219	0.176	0.219	0.171	0.213
	192	0.225	0.259	0.221	0.258	0.224	0.259	0.217	0.253
	336	0.278	0.297	0.275	0.296	0.279	0.299	0.272	0.293
	720	0.354	0.384	0.353	0.345	0.356	0.350	0.348	0.343
	Avg	0.259	0.281	0.256	0.280	0.259	0.282	0.252	0.276
EXCHANGE	96	0.084	0.201	0.089	0.209	0.083	0.201	0.084	0.203
	192	0.187	0.307	0.196	0.314	0.173	0.297	0.176	0.300
	336	0.337	0.422	0.334	0.419	0.341	0.424	0.310	0.404
	720	0.858	0.695	0.856	0.700	0.856	0.698	0.842	0.690
	Avg	0.367	0.406	0.369	0.411	0.363	0.405	0.353	0.399
ECL	96	0.193	0.291	0.164	0.250	0.165	0.253	0.164	0.245
	192	0.199	0.297	0.175	0.261	0.177	0.258	0.173	0.256
	336	0.216	0.312	0.191	0.278	0.190	0.274	0.189	0.275
	720	0.257	0.345	0.230	0.312	0.232	0.311	0.229	0.310
	Avg	0.216	0.331	0.190	0.275	0.191	0.274	0.189	0.272
TRAFFIC	96	0.472	0.305	0.433	0.281	0.438	0.283	0.429	0.279
	192	0.474	0.304	0.446	0.287	0.447	0.287	0.442	0.282
	336	0.491	0.331	0.459	0.288	0.459	0.289	0.456	0.288
	720	0.523	0.327	0.490	0.306	0.494	0.307	0.486	0.307
	Avg	0.490	0.317	0.457	0.291	0.460	0.292	0.453	0.289