Flow-Guided Neural Operator for Self-Supervised Learning on Time Series Data

Anonymous Author(s)
Affiliation
Address
email

Abstract

Self-supervised learning (SSL) is a powerful paradigm for learning from unlabeled time-series data. However, traditional methods such as masked autoencoders (MAEs) rely on reconstructing inputs from a fixed, predetermined masking ratio. Instead of this static design, we propose treating the corruption level as a new degree of freedom for representation learning. To achieve this, we introduce the Flow-Guided Neural Operator (FGNO), the first framework to combine operator learning with flow matching for SSL training. By leveraging Short-Time Fourier Transform (STFT) to enable computation under different time resolutions, our approach effectively learns mappings in functional spaces. We extract a rich hierarchy of features by tapping into different network layers (l) and generative time steps (s) that apply varying strengths of noise to the input data. This enables the extraction of versatile, task-specific representations—from low-level patterns to high-level semantics—all from a single model. We evaluated our model performance on two different biomedical domains, where our flow-based operator significantly outperforms established baselines. When applied to a sleep health dataset, it achieved 16% RMSE improvement over MAE in skin temperature regression, while showing 1% AUROC gain in classification tasks. On a neural decoding task for binary speech classification, our approach achieves a significant 20% AUROC improvement compared to MAE, highlighting its ability to learn powerful, task-adaptable representations.

1 Introduction

Time-series data are common across domains such as healthcare [1], finance [2], climate and weather forecasting [3]. Learning useful supervised representations from temporal signals can be challenging when labels are scarce [4]. Self-supervised learning (SSL) has become a compelling technique, enabling models to exploit large collections of unlabeled time series data. Prior work adapts ideas from NLP and computer vision, e.g., BERT-style masked modeling [5, 6] and masked autoencoders (MAE) [7], as well as contrastive objectives [8]—and has led to increasingly capable time-series foundation models [9]. However, objectives based on discrete masking ratios or fixed augmentations can make it difficult to recover features spanning multiple temporal and semantic scales within a single pretraining recipe/masking ratio.

Generative modeling offers a complementary perspective. Diffusion- and flow-based methods learn to map simple noise distributions to complex data distributions and are trained with self-supervised signals (denoising [10] or flow matching [11]). Beyond high-quality data generation, their training dynamics expose a continuum of corruption levels that acts as a *continuous* analogue to masking. Recent studies on images suggest that internal representations taken at different noise levels naturally organize from low-level textures to high-level semantics, providing an explicit control knob for multi-scale features [12].

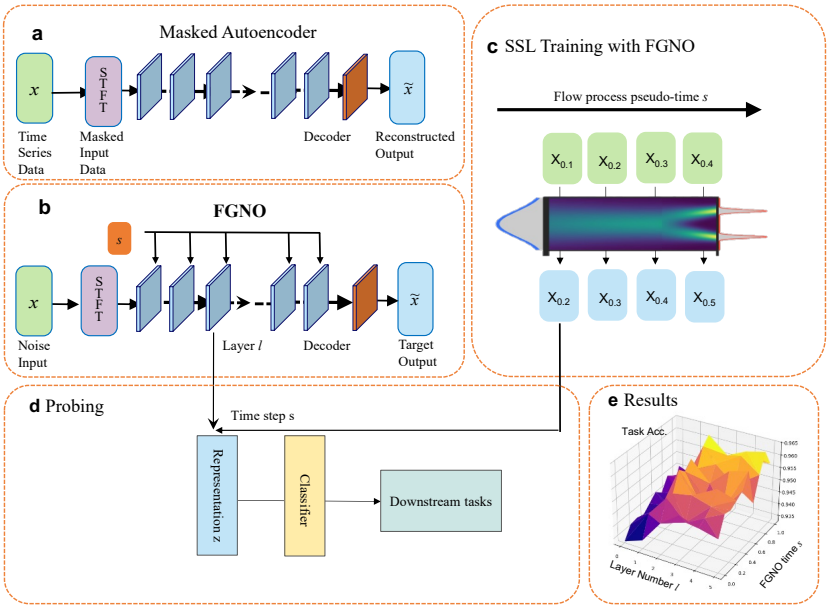

Figure 1: While Masked Autoencoder (**a**) reconstructs inputs from compressed latent representations, Flow Matching (FM), **b** progressively transforms noisy inputs through continuous flows, generating intermediate outputs across time steps. In both architecture, time series data is converted to spectrogram via STFT. Our FM, instantiated as the FGNO, is pretrained in a self-supervised fashion (**d**). The pretrained encoder is probed at different layers and time steps (**d**), with frozen representations evaluated on downstream tasks, across different time and layers (**e**).

Neural operators [13] learn mappings directly in the functional space of signals, offering a natural framework for time-series modeling where data can be viewed as functions over time [14]. This approach has achieved state-of-the-art results across various time-series domains, including faster and more accurate weather forecasts [15] and chaotic time-series prediction [16, 17]. However, FNO-based [18] networks are not well-suited for time series applications that focus on local segmentation or classification tasks, as they primarily capture global patterns in the frequency domain. Short-Time Fourier Transform (STFT) is a widely known approach in signal processing that focuses on local time-frequency analysis, enabling the extraction of both temporal and spectral features at fine-grained resolutions that are crucial for many downstream tasks while being resolution invariant.

The combination of neural operators and flow matching creates a natural synergy for continuous-time modeling, where the functional space perspective of neural operators aligns seamlessly with both the continuous denoising process of flow matching and the continuous nature of time-series data. However, it remains underexplored whether these advantages transfer to time-series data at scale and whether flow matching (FM) can serve as an efficient and effective SSL objective in this setting.

Our approach: We propose the FGNO, a self-supervised framework that pretrains a *flow matching* model on time-series data (spectrograms) and then extracts task-specific representations by selecting both a network layer l and a generative time/noise level s (Fig. 1). We leverage Short-Time Fourier Transform (STFT) to convert raw 1D signals into time-frequency representations that preserve both local and global information across multiple resolutions. Our framework has two phases. The first phase is training/pretraining, where we train an FM model to learn the continuous flow from noise to data on these spectrograms, and treat intermediate features $\phi_{l,s}$ as a hierarchy of representations. The second phase is probing for downstream tasks such as classification and regression. where we freeze the backbone and train a probing head (classifier) on top of $\phi_{l,s}$. This design turns the noise level s into a practical, tunable degree of freedom—analogous to a continuous masking ratio—that allows practitioners to emphasize fine temporal detail (lower s, shallower l) or higher-level semantics (higher s, deeper l) with a *single* pretrained model. During inference, we find the best combination of (l^*, s^*) for the downstream task.

Empirically, we observe that the optimal choice of network layer l and noise level s is task-dependent: tasks requiring precise local timing benefit from lower noise and earlier layers, whereas tasks relying on global context prefer higher noise and deeper layers. Selecting (l, s) per task yields consistent

gains over MAE- and contrastive-style baselines. On DREAMT dataset, our approach improves AUROC metric by **1%** for binary sleep classification and **16%** in RMSE for regression compared to state-of-the-art SSL baselines like MAE [5]. For BrainTree Bank dataset, we achieved a **20%** increase in AUROC against MAE on neural signal decoding where subject is tasked with detecting speech from movies.

In summary, our contributions are: 1. **An SSL framework combining operator learning and flow matching for time series.** We pretrain a single FM model on time–frequency representations of 1D signals and show how to derive a rich, multi-level feature hierarchy by varying the generative time/noise level s and network layer l. 2. **Noise as a gauge to control features.** We demonstrate that s serves as an explicit and practical control over representation granularity, offering a clear advantage over fixed-ratio masking in MAE-style SSL [5]; practitioners can tune (l, s) to the demands of each downstream task.

2 Methods

Our FGNO methodology is a two-stage process for self-supervised representation learning based on the Flow Matching (FM) framework [11]. By operating in the Fourier domain through spectrograms, FGNO learns mappings in the functional space of signals, and can thus be viewed as a neural operator.

Pre-training We first convert raw 1D time series signals into time-frequency representations using a Short-Time Fourier Transform (STFT), resulting in spectrograms. A time-dependent Transformer architecture, $u_\theta(s, g)$, is then pre-trained on these unlabeled spectrograms. The model is optimized with the FM objective, which involves learning a conditional vector field that maps a simple noise distribution to the complex data distribution of the spectrograms. Because this process learns transformations between functions in Fourier space, it naturally takes on the role of a neural operator. This self-supervised task forces the network to capture the rich underlying structure of the time series data across multiple temporal and semantic scales.

Feature Extraction and Probing After pre-training, the weights of the Transformer u_θ are frozen and used as a feature extractor. A key challenge is that the model was trained on noisy inputs, but downstream tasks begin with clean data. To prevent a distributional shift, for a given clean spectrogram f and a desired feature extraction time $s \in [0, 1]$, we first generate a correctly noisy sample:

$$g_s = sf + \sigma_s^f z, \quad \text{where } z \sim \mathcal{N}(0, I). \tag{1}$$

The feature representation h is then extracted from the activations of an intermediate layer l of the network after processing this input: $h = u_\theta^{(l)}(t, g_t)$. This feature vector is fed into a lightweight downstream head (e.g., a linear layer), which is the only component trained on labeled data. This design makes FGNO computationally efficient while enabling systematic probing of (s, l) combinations to uncover task-optimal representations.

Inference To select the most informative features for each downstream task, we evaluate frozen representations across layers and noise levels and choose the optimal pair:

$$(s^*, l^*) = \arg \min_{s \in S, l \in L} \mathcal{L}_{\text{val}}(s, l). \tag{2}$$

3 Experimental Results

3.1 Dataset and Implementation Details

DREAMT Dataset We used the DREAMT dataset [19], which contains synchronized smartwatch and clinical-grade polysomnography (PSG) data from 100 participants, many with sleep disorders. A single model was pre-trained on the smartwatch's Blood Volume Pulse (BVP) and accelerometer (ACC) signals. This model's features were then evaluated on held-out participants for *two downstream tasks*: a binary sleep/wake classification and a skin temperature regression.

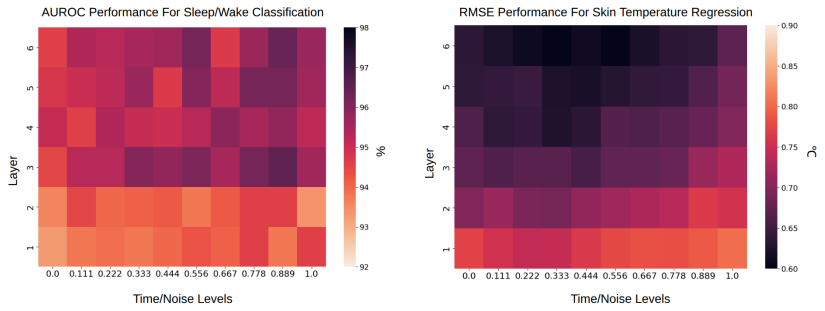

Figure 2: Performance across model layers and feature extraction times in our self-supervised learning framework. **Left:** Sleep classification AUROC (↑). **Right:** Skin temperature regression RMSE (↓).

BrainTree Bank dataset The BrainTree Bank [20] is a large-scale dataset of intracranial neural responses from 10 subjects watching Hollywood movies (43 hours total). The dataset includes extensive linguistic annotations of the movie audio, such as transcripts and word onsets. Using a held-out set of subjects for probing, we evaluate our model on a speech presence detection task.

3.2 Sleep classification and skin temperature prediction with DREAMT

Performance at different layers and noise-level s As shown in Fig. 2, sleep classification performance improves substantially in deeper layers, with layers 3–6 consistently outperforming layers 1–2, and the best AUROC (96.4%) achieved at Layer 3 with low noise ($s = 0.89$). In contrast, skin temperature regression also favors deeper layers but achieves its lowest RMSE (0.599°C) at moderate noise levels ($s \in [0.22, 0.56]$), highlighting that discrete classification tasks benefit from clean, abstract representations while continuous regression tasks rely on partially denoised features that preserve smooth dynamics.

Comparison to baselines Our FGNO approach significantly outperforms baselines in both sleep classification and skin temperature regression. It achieves improved AUROC compared to an MAE baseline across the best-performing layers. Our peak score also surpasses the gradient boosting approach (92.6%) reported in the DREAMT paper [19]. Notably, our model achieved this using only raw time-series data, whereas the DREAMT baseline required additional clinical metadata (Apnea Severity score) [19], highlighting the power of our self-supervised approach. For skin temperature regression, our best RMSE substantially improves upon both the MAE baseline (0.734°C). The results highlight that FGNO not only can leverage layers' depth but also exploits flow time s to extract the most predictive representations, whereas MAE is limited to selecting layers alone.

3.3 Speech Classification with BrainTreeBank

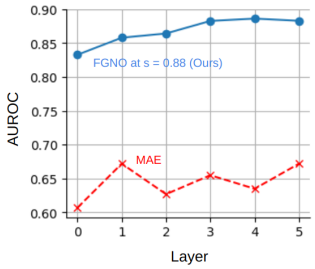

Figure 3: FGNO's and MAE's performance as a function of layer depth l.

By optimizing the combination of model layer and feature extraction time s, we observed a clear performance gradient where our FGNO score improved with both network depth and time, reaching a maximum of 88.6% AUROC. Fig. 3 reveals that the optimal score was not achieved at the final layer or time step, but rather at an intermediate point (layer 4, $s \approx 0.88$). This finding demonstrates that the most discriminative features arise from a specific trade-off in processing depth and time, allowing our approach to significantly outperform the MAE baseline by identifying the most potent feature representations within the network.

Summary We propose the FGNO framework for time-series SSL that treats corruption as a continuous variable. A single pre-trained backbone provides features, while task-specific representations are obtained by selecting an optimal layer (l) and noise level (s). This approach is more flexible than fixed-corruption baselines like MAE and achieves state-of-the-art results on diverse tasks. Its main limitation is the empirical grid search needed to find (l, s) for new tasks. Future work will automate this selection and extend FGNO to new modalities.

References

[1] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[2] Ruey S Tsay. Analysis of financial time series. 2005.

[3] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024.

[4] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. Label-efficient time series representation learning: A review. *IEEE Transactions on Artificial Intelligence*, 5(12):6027–6042, December 2024.

[5] Christopher Wang, Vighnesh Subramaniam, Adam Uri Yaari, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. Brainbert: Self-supervised representation learning for intracranial recordings, 2023.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[7] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[8] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025.

[9] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[11] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

[12] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion, 2023.

[13] Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6(5):320–328, 2024.

[14] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[15] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[16] Qixin Wang, Lin Jiang, Lianshan Yan, Xingchen He, Jiacheng Feng, Wei Pan, and Bin Luo. Chaotic time series prediction based on physics-informed neural operator. *Chaos, Solitons & Fractals*, 186:115326, 2024.

[17] Xin-Yi Li and Yu-Bin Yang. Gafno: Gated adaptive fourier neural operator for task-agnostic time series modeling. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1133–1138. IEEE, 2023.

[18] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[19] Kexin Wang, Jialu Yang, Aakash Shetty, and Jess Dunn. DREAMT: Dataset for Real-time sleep stage EstimAtion using Multisensor wearable Technology (version 1.0.0), 2024.

[20] Christopher Wang, Adam Uri Yaari, Aaditya K Singh, Vighnesh Subramaniam, Dana Rosenfarb, Jan DeWitt, Pranav Misra, Joseph R. Madsen, Scellig Stone, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. Brain treebank: Large-scale intracranial recordings from naturalistic language stimuli, 2024.

Appendix

Training details The pre-trained model is a 6-layer Transformer designed to process the output of a Short-Time Fourier Transform (STFT). The model's architecture was specifically configured to match the STFT output tensor shape: the model's input dimension of 132 corresponds to the number of frequency bins, and the sequence length of 21 corresponds to the number of time frames. Other key hyperparameters include a hidden dimension of 768, 12 attention heads, a feedforward dimension of 3072, a dropout rate of 0.1, and a learning rate of 0.0001.

Evaluation metrics We evaluated the performance on the two downstream tasks using standard metrics. For the binary sleep classification task (awake vs. asleep), we used the Area Under the Receiver Operating Characteristic curve (AUROC). For the skin temperature regression task, we used Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) to quantify the model's predictive accuracy.

Layer Number	FGNO (Best AUROC % @ Time)	MAE AUROC %
1	94.6 @ s=1.00	**95.8**
2	94.6 @ s=0.89	**95.6**
3	**96.5** @ s=0.89	95.7
4	**95.9** @ s=0.67	95.4
5	**96.2** @ s=0.78	95.5
6	**96.4** @ s=0.89	95.8

Table 1: AUROC (↑) comparison between our model and MAE on DREAMT data for sleep classification task

Layer Number	FGNO (Best RMSE °C @ Time)	MAE RMSE °C
0	**0.743** @ s=0.22	0.790
1	**0.691** @ s=0.33	0.775
2	**0.656** @ s=0.44	0.735
3	**0.625** @ s=0.33	0.782
4	**0.619** @ s=0.44	0.738
5	**0.600** @ s=0.56	0.744

Table 2: Best RMSE (↓) values against MAE for DREAMT on skin temperature regression task

Layer Number	FGNO (Best AUROC % @ Time)	MAE AUROC %
0	**83.3** @ s=0.778	60.7
1	**85.8** @ s=0.778	67.2
2	**86.4** @ s=0.778	62.7
3	**88.3** @ s=0.889	65.5
4	**88.6** @ s=0.889	63.5
5	**88.3** @ s=0.889	67.2

Table 3: AUROC (↑) comparison at optimal extraction time for BrainTreeBank data in speech detection task