# Semi-Supervised Learning for Time Series Collected at a Low Sampling Rate

Minyoung Bae
KAIST
Daejeon, Republic of Korea
mybae@kaist.ac.kr

Yooju Shin
KAIST
Daejeon, Republic of Korea
yooju.shin@kaist.ac.kr

Youngeun Nam
KAIST
Daejeon, Republic of Korea
youngeun.nam@kaist.ac.kr

Young Seop Lee
Samsung Electronics Co., Ltd.
Suwon, Republic of Korea
yseop.lee@samsung.com

Jae-Gil Lee*
KAIST
Daejeon, Republic of Korea
jaegil@kaist.ac.kr

## Abstract

Although time-series classification has many applications in healthcare and manufacturing, the high cost of data collection and labeling hinders its widespread use. To reduce data collection and labeling costs while maintaining high classification accuracy, we propose a novel problem setting, called *semi-supervised learning with low-sampling-rate time series*, in which the majority of time series are collected at a low sampling rate and are unlabeled whereas the minority of time series are collected at a high sampling rate and are labeled. For this novel problem scenario, we develop the *SemiTSR* framework equipped with the *super-resolution* module and the *semi-supervised learning* module. Here, low-sampling-rate time series are upsampled precisely, taking periodicity and trend at each timestamp into account, and both labeled and unlabeled high-sampling-rate time series are utilized for training. In particular, consistency regularization between artificially downsampled time series derived from an original high-sampling-rate time series is effective at overcoming limited sampling rates. We demonstrate that *SemiTSR* significantly outperforms conventional semi-supervised learning techniques by assuring high classification accuracy with low-sampling-rate time series.

## CCS Concepts

• **Mathematics of computing** → **Time series analysis**; • **Theory of computation** → *Semi-supervised learning*.

## Keywords

time series, semi-supervised learning, classification, sampling rate

**ACM Reference Format:**
Minyoung Bae, Yooju Shin, Youngeun Nam, Young Seop Lee, and Jae-Gil Lee. 2024. Semi-Supervised Learning for Time Series Collected at a Low Sampling Rate. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3672033

*Corresponding author.



*KDD '24, August 25–29, 2024, Barcelona, Spain.*

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3672033

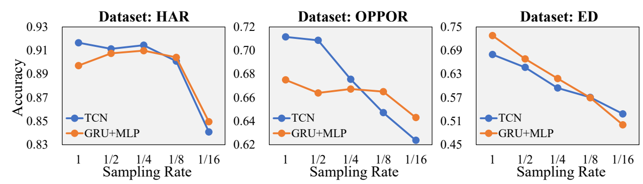



**Figure 1: Negative impact of reduced sampling rates on classification accuracy in three datasets [2, 10, 35].**

## 1 Introduction

### 1.1 Background

The success of time-series classification requires a sufficient amount of high-resolution data and high-quality labels [26]. However, these requirements are not always easily met. Despite advancements in sensor technology, the sampling rate for time series may not be high enough due to limitations in storage space [21], transfer speed [1], and battery capacity [17], particularly in mobile devices. Moreover, due to the complicated temporal dynamics and high-dimensional data structure of time series [30, 36, 59], data labeling typically requires domain experts at a high cost.

Insufficient resolution of a time series, determined by its *sampling rate*, naturally leads to a deterioration in classification accuracy. Figure 1 demonstrates, using the TCN and GRU+MLP methods [3, 13] on the HAR, OPPOR, and ED datasets [2, 10, 35], that the classification accuracy decreases rapidly as the sampling rate decreases—by at least 20% when it is decreased to $\frac{1}{16}$. Thus, there is a trade-off between the cost of data collection and the accuracy of classification, which cannot be attained concurrently.

The deterioration depicted in Figure 1 is primarily attributed to (i) *periodicity blur* and (ii) *phase shift*. First, in Figure 2, consider two time series whose frequency has been reduced to $\frac{1}{8}$ of their original frequency by downsampling. At the original frequency, these two time series from the 'Jogging' and 'Walking' classes are easily distinguishable; however, at the $\frac{1}{8}$ frequency, the unique periodicity in each class has been obscured by the downsampling. Second, in Figure 3, consider two time series whose frequency has been reduced to $\frac{1}{16}$ of their original frequency. Due to the difference in the phase of each data point in the two time series that have been downsampled differently, where the *phase* indicates a position

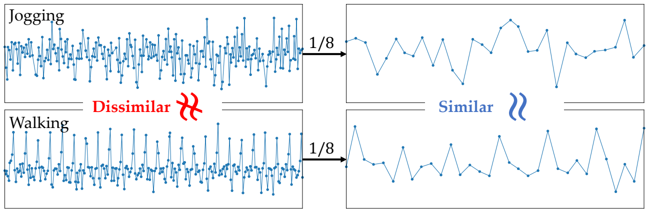



**Figure 2: Example of *periodicity blur* while reducing the sampling rate in the WISDM human activity dataset [57].**

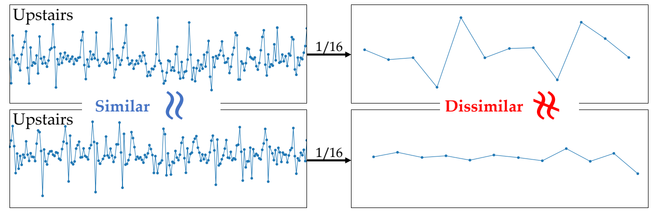



**Figure 3: Example of *phase shift* while reducing the sampling rate in the WISDM human activity dataset [57].**

within each period, the two time series appear very distinct despite belonging to the same class 'Upstairs.'

### 1.2 Main Contributions

To achieve high classification accuracy while keeping low data collection cost, in this paper, we formulate a novel problem, called *semi-supervised learning with low-sampling-rate time series*. In this problem setting, a majority of time series are collected at a low sampling rate and unlabeled, while a minority of time series are collected at a high sampling rate and labeled. The overall cost of data collection remains low because the proportion of the high-sampling-rate time series is small (e.g., 10%), and labeling for this small portion is considered feasible as confirmed in active learning [46]. In addition, both high and low sampling rates can coexist because the sampling rate is adjustable in most data gathering environments [24, 29, 38]. Overall, the problem setting is practically feasible and obviously satisfies low data collection cost.

The remaining challenge is to accomplish high classification accuracy with limited availability of labeled high-sampling-rate time series. To this end, we propose the framework, *Semi-supervised Time-series Super Resolution* (*SemiTSR*), for addressing the novel problem. The *SemiTSR* framework consists of (i) the *super-resolution* module that upsamples low-sampling-rate time series into high-sampling-rate time series and (ii) the *semi-supervised learning* module that trains a classifier using both labeled and unlabeled high-sampling-rate time series. These two modules—upsampler and classifier—are trained in an end-to-end fashion.

The unique contributions for accomplishing high classification accuracy are two fold, addressing the aforementioned two problems.
**Handling of the Periodicity Blur**: We integrate the inherent characteristics of time series into the design of the upsampler. Without loss of generality, we assume that a time series is the sum of trend, periodicity, and random error [42]. When the upsampler estimates missing values at unsampled timestamps, it takes account of other sampled values that have a similar phase within a similar periodicity as well as a similar trend, instead of simply using adjacent values. Accordingly, we propose a *context-aware attention* technique based on temporal embedding to focus on the other timestamps with a similar context (i.e., periodicity and trend). As a result, unsampled values are restored closer to their true values, which aids in resolving the periodicity blur issue depicted in Figure 2.
**Handling of the Phase Shift**: We take *full* advantage of a small amount of labeled high-sampling-rate time series in both the reconstructor and classifier. Using the high sampling rates, the reconstructor is trained to recover the original high-sampling-rate time series from two phase-shifted, downsampled low-sampling-rate time series. In addition, the classifier is trained to generate consistent output for the two reconstructed high-sampling-rate time series by *shift consistency regularization*, which is very effective to relieve the phase shift issue depicted in Figure 3. Using the available labels, the classifier is further trained to predict the correct label for the two reconstructed high-sampling-rate time series.

In conclusion, to achieve both high classification accuracy and low data collection cost, we newly introduce *semi-supervised learning with low-sampling-rate time series* and develop the *SemiTSR* framework for the problem. In particular, *SemiTSR* effectively addresses periodicity blur and phase shift caused by limited sampling rates in time series by employing the context-aware attention technique and the shift consistency regularization, respectively. According to our extensive experiments, when only 10% of the data is labeled and the sampling rate is decreased to $\frac{1}{8}$ of the labeled set, *SemiTSR* significantly outperforms conventional semi-supervised learning methods by 2.9–29.5%.
**Potential Application Scenario**: We would like to wrap up this section with discussing an application scenario to emphasize the usefulness of the framework. Most clinical electrocardiogram (ECG) databases encompass time series data with sampling rates of 500–1000Hz, following a practice endorsed by American Heart Association [9, 45]. These databases are meticulously annotated by cardiologists, incurring significant costs. Conversely, widely-used *wireless* ambulatory ECG devices produce unlabeled time series data at sampling rates below 250Hz [22], which are insufficient to detect long QT syndrome and hypocalcemia [5, 18], due to their power constraint. These wireless ECG devices generate large amounts of data in order to track the patients' twenty-four-hour daily activities. This discrepancy aligns perfectly with our novel problem. *SemiTSR* leverages a substantial volume of unlabeled low-sampling-rate data alongside a smaller volume of labeled high-sampling-rate data. This approach enhances the contribution of the unlabeled, lower-quality data to the diagnosis of sudden cardiac events outside of hospital settings. Please see Section 4.6 for the empirical results of this application scenario.

## 2 Related Work

### 2.1 Semi-Supervised Learning

*Semi-supervised learning* (SSL) leverages large amounts of unlabeled data to enhance deep learning when the labeled data is scarce. The key SSL methods include consistency regularization, which mandates consistency in the model's predictions across augmented instances. This approach facilitates the model's comprehension of the

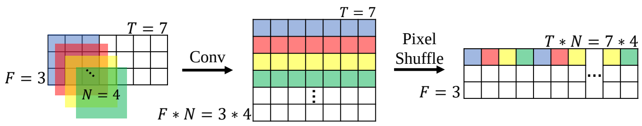



**Figure 4: Limitation of upsampling convolution [47].** $T$, $F$, $N$ **is the number of timestamps at a low sampling rate, the number of features, and the number of kernels for convolution which is same as the upscaling factor.**

unlabeled data distribution. In FixMatch [52], a strongly-augmented instance is assigned to predict a weakly-augmented instance as a pseudo-label. FreeMatch [56], an extension of FixMatch [52], controls the confidence threshold for pseudo-labeling based on the learning status in order to ensure the consistency with the correct pseudo-label.

Another pivotal SSL strategy is the entropy minimization which operates under the premise that the classification decision boundary should evade regions populated densely by data points. The techniques such as Pseudo-Label [32] and MixMatch [7] implicitly achieve entropy reduction by refining the sharpness of the model's predictions. Furthermore, ReMixMatch [8], building upon MixMatch [7], endeavors to synchronize the marginal distribution of unlabeled data predictions with the ground-truth labels, thus enhancing model reliability and prediction accuracy.

Recently, a few semi-supervised learning techniques have been introduced explicitly for time-series data [15, 19, 27, 55]. SemiTime [19] employs the pretext task, predicting whether two segments have past and future relationships, to capture the necessary temporal relations for classification. CrossMatch [48] proposes context attachment as the time-series augmentation strategy in consistency regularization, but does not address periodic features of time series or the challenges presented by varying sampling rates. SSGAN [40] utilizes labeled data to improve the imputation of missing values in time series. The classifier is self-trained using the pseudo-labels inferred from the imputed time series, while the imputation module is supervised using the reconstruction and adversarial losses. However, the goal of this work is mostly imputation, and the classifier primarily serves for the purpose of imputation, which does not suit our problem.

In summary, none of the existing SSL works consider the high costs of time-series collection and labeling at the same time. Compared with these methods, we reduce data collection and labeling costs while leveraging high-sampling-rate time-series data to ensure classification accuracy.

## 2.2 Single-Image Super-Resolution

Single-image super-resolution models [12, 33, 34] aim to reconstruct high-resolution images given low-resolution counterparts. SwinIR [34] uses convolution and attention modules to capture long-range dependencies within varying contents and simultaneously remove border artifacts. Several recent methods utilize implicit neural representation [12, 33, 39, 51] to represent a high-resolution image as a combination of its low-resolution counterpart and a neural network. For example, LIFF [12] employs an MLP to predict the RGB value of a target coordinate based on the nearest low-resolution latent code. LTE [33] reconstructs high-frequency components by mapping the target coordinate and the nearest low-resolution latent code to locally dominant frequency terms.

However, none of these existing methods have been applied to time-series data, where both temporal dependencies and periodic features should be modeled [20, 59]. Particularly in the upscaling stage, methods using implicit neural networks do not consider temporal dependencies as they estimate high-resolution values based on a single coordinate and its corresponding feature. A convolution-based upscaling approach [47] also neglects the sequential relation in the high resolution. As depicted in Figure 4, a low-sampling-rate time series with three features is upsampled by a factor of 4. The first four high-sampling-rate timestamps are generated using different convolution kernels (blue, red, yellow, and green). However, since independent convolution kernels are used to reconstruct the values of adjacent high-resolution coordinates, the sequential relation is disconnected during the upsampling process.

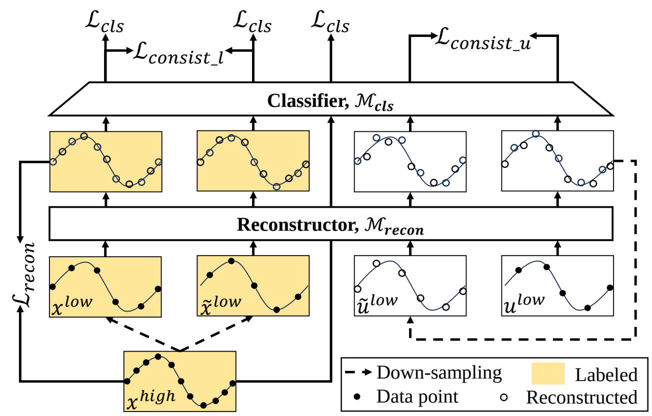



**Figure 5: Overview of the training *SemiTSR* framework for labeled high-sampling-rate data and unlabeled low-sampling-rate data. While testing, *SemiTSR* uses unlabeled low-sampling-rate data as input.**

# 3 Methodology

## 3.1 Problem Definition

We formally define the problem of learning a classification model that operates on low-sampling-rate instances by leveraging high-sampling-rate instances for training. Let $(\boldsymbol{x}^{high} \in \mathbb{R}^{d\times T}, y)$ be a labeled high-sampling-rate instance and $\boldsymbol{u}^{low} \in \mathbb{R}^{d\times\gamma T}$ be an unlabeled low-sampling-rate instance. $d$, $T$, and $\gamma$ denote the number of features, the number of timestamps within a time series collected in the original sampling rate, and the low sampling rate, respectively. The goal of this work is building a classification model $\mathcal{M}(\boldsymbol{u}; \Theta^{recon}, \Theta^{cls})$ that returns the probability of each class given an unlabeled low-sampling-rate instance $\boldsymbol{u} \in \mathbb{R}^{d\times\gamma T}$. The model $\mathcal{M}$ is parameterized by $\Theta^{recon}$ and $\Theta^{cls}$ referring to reconstruction and classification modules.

## 3.2 The Overall Framework of *SemiTSR*

Figure 5 illustrates the overall procedure of *SemiTSR* framework, which consists of the two neural network modules: reconstructor

$\mathcal{M}_{recon}$ and classifier $\mathcal{M}_{cls}$. To recover class-discriminative periodic features, $\mathcal{M}_{recon}$ increases the resolution of a low-sampling-rate time series to match the sampling rate of $\boldsymbol{x}^{high}$. Subsequently, the synthesized high-sampling-rate time series is fed into the $\mathcal{M}_{cls}$ to obtain the probability distribution for each class.

During training, the labeled high-sampling-rate data $\boldsymbol{x}^{high}$ provides supervision for both $\mathcal{M}_{recon}$ and $\mathcal{M}_{cls}$ as shown in the left flow of Figure 5. First, $\boldsymbol{x}^{high}$ is downsampled at the rate of $\gamma$ to synthesize two low-sampling-rate instances of different phases, $\boldsymbol{x}^{low}$ and $\widetilde{\boldsymbol{x}}^{low}$. Then, each $\boldsymbol{x}^{low}$ and $\widetilde{\boldsymbol{x}}^{low}$ are processed by $\mathcal{M}_{recon}$ and $\mathcal{M}_{cls}$ sequentially. The supervised loss for the labeled high-sampling-rate data consists of a classification loss $\mathcal{L}_{cls}$ and a reconstruction loss $\mathcal{L}_{recon} = \lambda_{time}\mathcal{L}_{time} + \lambda_{freq}\mathcal{L}_{freq}$ which are formulated by

$$\begin{aligned}
\mathcal{L}_{cls}(\boldsymbol{x}^{high}, y) =& \frac{1}{2}\sum_{\boldsymbol{x}\in X} H[y, \mathcal{M}_{cls}(y|\mathcal{M}_{recon}(\boldsymbol{x}))] \\
&+ H[y, \mathcal{M}_{cls}(y|\boldsymbol{x}^{high})], \\
\mathcal{L}_{time}(\boldsymbol{x}^{high}) =& \frac{1}{2}\sum_{\boldsymbol{x}\in X} \|\boldsymbol{x}^{high} - \mathcal{M}_{recon}(\boldsymbol{x})\|_2^2 \\
\mathcal{L}_{freq}(\boldsymbol{x}^{high}) =& \frac{1}{2}\sum_{\boldsymbol{x}\in X} \|\mathcal{F}(\boldsymbol{x}^{high}) - \mathcal{F}(\mathcal{M}_{recon}(\boldsymbol{x}))\|_1,
\end{aligned} \tag{1}$$

where $X$ is a set of downsampled instances $\{\boldsymbol{x}^{low}, \widetilde{\boldsymbol{x}}^{low}\}$ and $H[p, q]$ is the cross-entropy between two probability distributions $p$ and $q$, while $\mathcal{F}$ denotes the fast Fourier transform. The classification loss $\mathcal{L}_{cls}$ supervises $\mathcal{M}_{recon}$ and $\mathcal{M}_{cls}$, using a limited quantity of labeled data. Recent studies in the filed of time-series analysis [20, 59] have emphasized the significance of modeling both temporal and frequency domains for accurate time-series forecasting and classification tasks. Hence, the reconstruction errors on both the original time series and Fourier transformed time series, $\mathcal{L}_{time}$ and $\mathcal{L}_{freq}$, are minimized to improve the classification by recovering temporal dependency as well as periodic features.

Now, to resolve the phase shift, we use a consistency regularization between the instances of different phases. The regularization enforces similarity between the classification result of $\boldsymbol{x}^{low}$ and $\widetilde{\boldsymbol{x}}^{low}$, which have different phases derived from a labeled high-sampling-rate instance $\boldsymbol{x}^{high}$, by minimizing the mean squared error. The consistency regularization term for labeled high-sampling-rate data, denoted as $\mathcal{L}_{consist\_l}$ in the top left of Figure 5, serves the *consistency* purpose. Since $\boldsymbol{x}^{low}$ and $\widetilde{\boldsymbol{x}}^{low}$ are supervised using ground-truth labels via $\mathcal{L}_{cls}$, minimizing the cross-entropy between the results of the shifted instances can be redundant. Therefore, we minimize the mean squared error as a loss term to facilitate a more fine-grained consistency [31].

For consistency regularization of the unlabeled low-sampling-rate instance $\boldsymbol{u}^{low}$, a synthesized counterpart $\widetilde{\boldsymbol{u}}^{low}$ is generated, which exhibits a different phase compared to $\boldsymbol{u}^{low}$. This synthesis is achieved by downsampling from $\mathcal{M}_{recon}(\boldsymbol{u}^{low})$, since the ground-truth shifted instance is not available. Then, the classification result of $\boldsymbol{u}^{low}$ serves as a pseudo-label for $\widetilde{\boldsymbol{u}}^{low}$ if the prediction for $\boldsymbol{u}^{low}$ is confident. Self-training limited to confident instances prevents $\mathcal{M}_{cls}$ learning from mis-classified uncertain instances [52]. The loss flow for the unlabeled low-sampling-rate data is shown in the right flow of Figure 5. Overall, the consistency loss is formulated by

**Algorithm 1** *SemiTSR*

**Input:** Labeled batch $\mathcal{D}_L$, unlabeled batch $\mathcal{D}_U$, confidence threshold $\tau$, labeled batch size $B$, unlabeled batch size ratio $\mu$, loss weights $(\lambda_{time}, \lambda_{freq}, \lambda_{consist})$, the number of original timestamps $T$, low-sampling-rate $\gamma$

```
ℒ_cls, ℒ_recon, ℒ_consist_l, ℒ_consist_u = 0, 0, 0, 0
/* Labeled data flow */
for x^high in 𝒟_L do
    x^low = DownSample(x^high; γ)
    x̃^low = DownSample(x_b^high; γ)
    ℒ_cls += 1/2 (H[y, ℳ_cls(y|ℳ_recon(x^low))]
             + H[y, ℳ_cls(y|ℳ_recon(x̃^low))])
    ℒ_recon += λ_time ℒ_time + λ_freq ℒ_freq        /* Eq. (1) */
    ℒ_consist_l += ‖ℳ_cls(y|ℳ_recon(x^low))
                   − ℳ_cls(y|ℳ_recon(x̃^low))‖_2^2
end for
/* Unlabeled data flow */
for u^low in 𝒟_U do
    u^high = ℳ_recon(u^low)
    ũ^low = γ Σ_1^{1/γ} DownSample(u^high; γ)
    q = ℳ_cls(y|u^high)
    ℒ_consist_u += 𝟙(max(q) ≥ τ)
                   · H[argmax_y(q), ℳ_cls(y|ℳ_recon(ũ^low))]
end for
return ℒ_cls/B + ℒ_recon/B + λ_consist(ℒ_consist_l/B + ℒ_consist_u/(μB))
```

$$\begin{aligned}
\mathcal{L}_{consist_l}(x^{low}, \widetilde{x}^{low}) &= \|\mathcal{M}_{cls}(y|\mathcal{M}_{recon}(x^{low})) \\
&\quad - \mathcal{M}_{cls}(y|\mathcal{M}_{recon}(\widetilde{x}^{low}))\|_2^2, \\
\mathcal{L}_{consist_u}(u^{low}, \widetilde{u}^{low}) &= \mathbb{1}(max(q) \geq \tau) \\
&\quad \cdot H[\hat{q}, \mathcal{M}_{cls}(y|\mathcal{M}_{recon}(\widetilde{u}^{low}))],
\end{aligned} \tag{2}$$

where $q$ is $\mathcal{M}_{cls}(y|\mathcal{M}_{recon}(\boldsymbol{u}^{low}))$, $\hat{q}$ is $argmax(q)$, and $\tau$ is the confidence threshold for making a pseudo-label.

The final loss is defined as $\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{recon} + \lambda_{consist}(\mathcal{L}_{consist\_l} + \mathcal{L}_{consist\_u})$. Here, $\lambda_{time}$, $\lambda_{freq}$, and $\lambda_{consist}$ are scalar hyperparameters that determine the relative weights of $\mathcal{L}_{time}$, $\mathcal{L}_{freq}$, and $\mathcal{L}_{consist\_l}+\mathcal{L}_{consist\_u}$, respectively. The pseudo-code implementation of the proposed *SemiTSR* framework is provided in Algorithm 1.

## 3.3 Reconstructor

As we have emphasized in Introduction, the restoration of class-discriminative local periodicity is crucial for enhancing the classification of low-sampling-rate time series. For this purpose, an upsampler within the resconstructor $\mathcal{M}_{recon}$ plays a vital role in restoring the periodicity that has been smoothed out. The reconstructor mainly consists of two components: an encoder and a temporal upsampler. Inspired by image super-resolution techniques [12, 33, 34], the low-sampling-rate time series $\{\boldsymbol{x}_t; t \in (p, p + \frac{1}{\gamma}, \ldots, p + \frac{\gamma T-1}{\gamma}), p \in (1, 2, \ldots, \frac{1}{\gamma})\}$ is first encoded by a

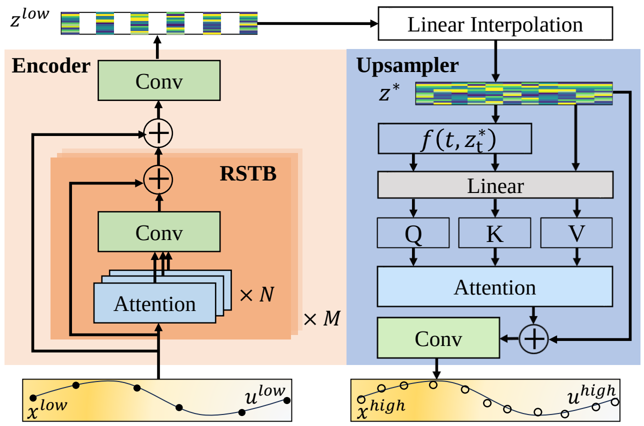


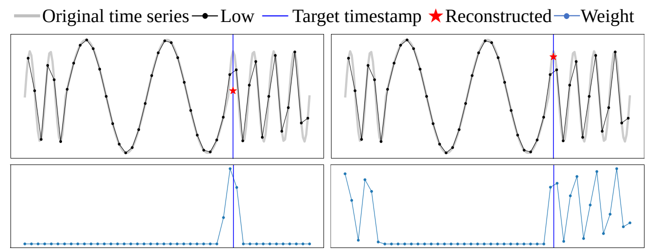

**Figure 6: Overall architecture of high-sampling-rate reconstructor $\mathcal{M}_{recon}$ consisting of a deep feature encoder and a temporal upsampler.**

deep feature encoder. Then, the latent feature map at a low sampling rate, $\{z_t; t \in (p, p+\frac{1}{\gamma}, \ldots, p+\frac{\gamma T-1}{\gamma}), p \in (1, 2, \ldots, \frac{1}{\gamma})\}$, is upsampled at the end of the reconstruction as shown in Figure 6.

*3.3.1 Encoder.* We modify Residual Swin Transformer Block (RSTB) [34] to create a deep feature encoder capable of processing 1-D input. As shown in the left part of Figure 6, the deep feature encoder consists of $M$ RSTBs and a convolution layer. Each RSTB consists of stacked attention modules, repeated $N$ times, and culminates with a convolution layer that includes a skip connection. Both a long skip connection and attention modules enable modeling the long-range dependency within the input time series. While we compose an encoder with RSTB, which is used in state-of-the-art image super-resolution methods, an arbitrary time-series encoder can be used as an encoder.

*3.3.2 Temporal Upsampler.* Meanwhile, for an upsampler that increases the temporal resolution in effect from the low-sampling-rate of $\boldsymbol{u}^{low}$ to the high-sampling-rate of $\boldsymbol{x}^{high}$, it is important to accurately restore local periodic patterns that have been smoothed out. However, techniques widely employed in image super-resolution [12, 47] are inappropriate for time series due to their inability to effectively model temporal dependencies and periodic features. For example, as depicted in in Figure 7(a), sub-pixel convolution applies a fixed kernel of size three to the adjacent observed points, without considering the periodicity and relative phase of the target timestamp. Consequently, the reconstructed value significantly deviates from the ground-truth peak point. That is, upsampling with the observed timestamps should consider the relative phase and periodicity of the target timestamp. To this end, we propose a novel periodic time embedding-based attention.

*3.3.3 Context-Aware Periodic Time Embedding.* To achieve accurate upsampling, the model should learn the periodicity and relative phase associated with each timestamp. For this purpose, we propose a time embedding which captures the local periodicity and trend within a given local context corresponding to each timestamp. Let us denote the output of the encoder on a low-sampling-rate time series




(a) Sub-pixel Convolution. (b) Temporal Upsampler.

**Figure 7: Visualization of the two upsampling methods. The gray line is the original time series at a high sampling rate, and the black points are the observed time series at a low sampling rate. Red stars are the reconstructed values given black observed points.**

as $\{z_t \in \mathbb{R}^{d_z} : t \in (p, p+\frac{1}{\gamma}, \ldots, p+\frac{\gamma T-1}{\gamma}), p \in (1, 2, \ldots, \frac{1}{\gamma})\}$. By applying linear interpolation to $z_t$, we generate a latent feature map with a high sampling rate denoted as $\{z_t^* \in \mathbb{R}^{d_z} : t \in (1, 2, \ldots, T)\}$. Using the interpolated latent feature map, the time embedding $f(t, z_t^*) \in \mathbb{R}^{d_t}$ at time $t$ occurring with the high sampling rate is formulated by

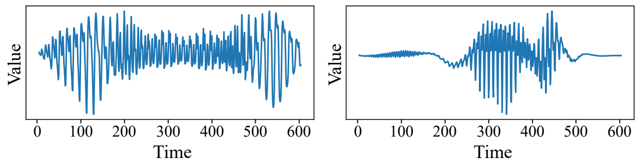
$$f(t, z_t^*) = \begin{bmatrix} W_0 t + b_0 \\ sin(F_1(z_t^*)t + P_1(z_t^*)) \\ \vdots \\ sin(F_{d_t-1}(z_t^*)t + P_{d_t-1}(z_t^*)) \end{bmatrix}, \tag{3}$$

where $W_0$ and $b_0$ are learnable parameters, and $F_i$ and $P_i$ are the MLPs that estimate frequency and phase, respectively, for a dimension $i \in (1, 2, \ldots, d_t - 1)$. In Equation (3), a local trend and the periodicities flowing at time $t$ are embedded in each dimension of the time embedding.

*3.3.4 Time Embedding-Based Attention.* Based on the time embedding, which represents the local periodicity and trend associated with each timestamp, a comparative analysis of the relative phase and period of these timestamps becomes feasible. By employing the time embedding as the *query* and *key* and an interpolated latent feature as the *value*, an attention module [54] can effectively learn the most relevant timestamps and assign appropriate weights to their corresponding values. Specifically, for a given *query* timestamp $t_{query}$, the attention score for each *key* timestamp is computed by evaluating the similarity between $f(t_{query}, z^*_{t_{query}})$ and $\{f(t_{key}, z^*_{t_{key}}) : t_{key} \in (1, \ldots, t_{query}, \ldots, T)\}$ as follows:

$$\begin{gathered} \text{MultiHead}(Q, K, V) = \text{Concat}(head_1, \ldots, head_h)W_O \\ \text{where } head_i = \text{Attention}(\mathrm{f(t}, z_t^*)\mathrm{W_Q}, \mathrm{f(t}, z_t^*)\mathrm{W_K}, z_t^*\mathrm{W_V}) \\ \text{and Attention}(Q, K, V) = \text{Softmax}(\frac{QK^\top}{\sqrt{d_t}})\, V. \end{gathered} \tag{4}$$

Here, $W_Q, W_K \in \mathbb{R}^{d_t \times \frac{d_t}{h}}$ and $W_V \in \mathbb{R}^{d_z \times \frac{d_z}{h}}$ are learnable parameters, and $h$ is the number of heads. Finally, the attention result is combined with a shortcut connection and subsequently subjected to convolutions with dilation sizes of 1, $\frac{1}{2\gamma}$, and $\frac{3}{2\gamma}$. This formulation ensures that the receptive fields cover values beyond the sampling




(a) Quinx (female). (b) Stigma (male).

**Figure 8: Insect Sound dataset [11] visualization of different species and sex.**

**Table 1: Benchmark dataset statistics.**

| Datasets | Applications | Window size | # Class | # Train | # Test | # Feature |
|---|---|---|---|---|---|---|
| Opportunity | Human Activity | 64 | 17 | 5907 | 1604 | 77 |
| InsectSound | Audio | 600 | 10 | 10000 | 5000 | 1 |
| mHealth | Human Activity | 200 | 12 | 2281 | 948 | 23 |
| SAMSUNG | Server Monitoring | 120 | 12 | 6000 | 3000 | 4 |

rate. As the high-sampling-rate time series prediction, the average of the three convolution outputs is used.

Through the mapping of a latent feature to the frequency and phase components associated with the corresponding timestamp, the time embedding $f(t, z_t^*)$ captures the local periodicity, allowing it to effectively model time-varying periodic patterns. Figure 8 visualizes how the frequency and amplitude change over time within a given time series. Simultaneously, the first dimension of $f(t, z_t^*)$ represents the trend component of the time series by means of learned linear transformation. The time embedding-based attention then assigns appropriate weights to the values of contextually and periodically relevant timestamps, as depicted in Figure 7(b).

# 4 Experiments

## 4.1 Experiment Setting

**Datasets**: Table 1 provides a summary of four benchmark datasets used in the experiment: *Opportunity*, *InsectSound*, *mHealth*, and *SAMSUNG*. For more detail of each dataset, see Appendix A. We also conducted experiment on ten datasets from the UCR time-series classification archive[1]. The ten datasets were chosen based on three specific criteria: a large amount of data, with at least 1000 instances in the training and test sets; a long original window length, of at least 96; and a high level of classification difficulty, with at least seven classes.

Instances from the original dataset were used as high-sampling-rate time series. Low-sampling-rate instances are downsampled from the original time series at a fixed sampling rate of $\gamma$. In our experiment, $\gamma$ varies from $\frac{1}{2}$ to $\frac{1}{16}$.

**Evaluation Metrics**: We report the evaluation based on *accuracy*, which is defined by

$$Accuracy(\%) = \frac{\#\, correct\ predictions}{\#\, total\ predictions} \times 100. \tag{5}$$

We repeated each evaluation *five* times using random seeds and distinct validation sets, and then report the mean and standard deviation for each result.

**Baselines**: We compare our method with the state-of-the-art semi-supervised learning methods based on consistency regularization: FixMatch [52], FreeMatch [56], and ReMixMatch [8]. For the augmentation strategy used in each method, see Appendix B.

For the classifier backbone $\mathcal{M}_{cls}$ of the semi-supervised learning, we used TCN [3] and Transformer [54] for their popularity in time-series classification [25, 58]. $\mathcal{M}_{cls}$ trained with 100% labeled high-sampling-rate data is the performance upper bound, while $\mathcal{M}_{cls}$ trained with $l$% labeled low-sampling-rate data is the performance lower bound. Our evaluation is conducted on $l = 20$ and $l = 10$ and, we randomly sampled $l$% from the fully-labeled original dataset to make unlabeled data.

**Model Configurations**: A deep feature encoder in *SemiTSR* consists of four RSTBs[2], and each RSTB consists of four attention layers with four heads. For the time embedding-based attention, $F_i(\cdot)$ and $P_i(\cdot)$ are three layer MLPs with 256 hidden dimensions. ($\lambda_{time}$, $\lambda_{freq}$, $\lambda_{consist}$) is $(2, 0.1, 1)$ for *Opportunity* and $(1, 2, 1)$ for the others since *Opportunity* exhibits a weak periodicity. $d_z$ is 256, and $d_t$ is set to 64 for *mHealth* and 32 for the others.

For the classifier backbone $\mathcal{M}_{cls}$, TCN is composed of eight temporal blocks whose hidden dimensionality is 128 and kernel size is seven. Transformer is composed of two transformer layers with two heads followed by a linear layer.

**Implementations Details**: The batch size of the labeled data is 32, and the number of epochs is 400, except for the *InsectSound* dataset where we set 16 and 150 due to its data size. The unlabeled batch size ratio is 3. Train and validation data are split by 9:1.

For semi-supervised learning baselines, we use SGD with a momentum of 0.9 for the optimizer following [52]. The learning rate is initialized as 0.03 and decayed using a cosine scheduler [37] to $\eta cos(\frac{7\pi k}{16K})$, where $\eta$ is the initial learning rate, $k$ is the current training step, and $K$ is the total number of training steps. All the other hyperparameters are set to default values in each paper. According to [52], semi-supervised learning performance heavily depends on an optimizer, a regularization, and a training scheduler as well as a semi-supervised learning algorithm. We found that Adam [28] works better than SGD [53] in *SemiTSR*; the learning rate is initialized to 0.001 and decayed by 0.5 every 50 epochs.

We conducted our experiment using Pytorch 1.12.1 on an NVIDIA RTX 3090Ti-equipped server. The source code is available at https://github.com/kaist-dmlab/SemiTSR.

## 4.2 Overall Performances

Tables 2 and 3 compare the classification accuracy of *SemiTSR* and semi-supervsied learning baselines as well as fully supervised learning when $\mathcal{M}_{cls}$ is TCN. *SemiTSR* performs best in almost all combinations of the low-sampling-rates and the labeled data ratios. Specifically in the lowest sampling rate when the labeled ratio is 10%, *SemiTSR* outperforms the other semi-supervised methods by 4.00–22.74% average performance margin. This result confirms that recovering high-frequency terms is crucial for improving the low-sampling-rate time-series classification. It is noteworthy that our method mostly caught up 100% fully supervised learning of each sampling rate. Existing semi-supervised learning baselines occasionally beat *SemiTSR* in a *relatively high* low-sampling-rate (e.g., 1/2). In these cases, we conjecture that the low-sampling-rate data

[1]https://www.cs.ucr.edu/~eamonn/time_series_data_2018/

[2]https://github.com/JingyunLiang/SwinIR

**Table 2: Accuracy comparison of semi-supervised learning methods with 20% labeled data when the classifier backbone is TCN.**

| Method | Opportunity | | | | InsectSound | | | mHealth | | | SAMSUNG | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1/2 | 1/4 | 1/8 | 1/16 | 1/2 | 1/4 | 1/8 | 1/2 | 1/4 | 1/8 | 1/2 | 1/4 | 1/8 |
| Fully-Supervised(100%) | 67.24 | 66.66 | 64.83 | 63.37 | 70.26 | 64.74 | 51.01 | 93.08 | 92.51 | 90.57 | 92.86 | 91.38 | 89.81 |
| | (±1.93) | (±1.59) | (±2.20) | (±1.77) | (±0.39) | (±0.66) | (±1.14) | (±3.54) | (±3.67) | (±2.73) | (±0.43) | (±0.20) | (±0.44) |
| Fully-Supervised(20%) | 63.90 | 57.72 | 57.36 | 53.37 | 60.48 | 54.20 | 39.58 | 92.57 | 91.12 | 86.43 | 88.05 | 85.39 | 84.03 |
| | (±1.99) | (±1.94) | (±0.88) | (±1.40) | (±0.81) | (±1.11) | (±0.59) | (±3.03) | (±2.92) | (±1.67) | (±0.50) | (±0.86) | (±0.75) |
| FixMatch | 69.43 | 67.42 | 66.20 | 65.07 | 65.83 | 59.51 | 42.81 | 86.79 | 84.14 | 85.91 | 91.65 | 89.88 | 88.85 |
| | (±2.86) | (±2.02) | (±2.29) | (±1.79) | (±1.00) | (±0.56) | (±0.60) | (±3.93) | (±5.67) | (±3.67) | (±0.67) | ) ± 0.70) | (±0.57) |
| FreeMatch | 66.28 | 65.11 | 65.51 | 64.04 | 66.15 | 60.39 | 43.37 | 88.75 | 91.50 | 88.59 | 92.44 | 89.90 | 88.95 |
| | (±1.52) | (±2.16) | (±1.56) | (±0.77) | (±0.93) | (±0.83) | (±0.28) | (±3.47) | (±3.68) | (±3.27) | (±0.14) | (±0.32) | (±0.45) |
| ReMixMatch | 69.40 | 66.72 | 65.12 | 64.93 | **67.54** | 59.46 | 43.11 | **94.05** | 91.65 | 90.08 | 87.20 | 86.57 | 86.83 |
| | (±1.31) | (±1.98) | (±0.40) | (±1.07) | (±0.91) | (±0.63) | (±0.87) | (±1.89) | (±4.35) | (±2.70) | (±1.06) | (±0.85) | (±0.71) |
| *SemiTSR* | **71.59** | **69.85** | **71.15** | **66.69** | 65.32 | **62.43** | **50.99** | 93.76 | **94.77** | **93.10** | **93.20** | **92.50** | **91.64** |
| | (±2.09) | (±.39) | (±1.88) | (±2.44) | (±0.98) | (±0.77) | (±0.61) | (±1.18) | (±2.45) | (±1.81) | (±0.54) | (±0.57) | (±0.89) |

**Table 3: Accuracy comparison of semi-supervised learning methods with 10% labeled data when the classifier backbone is TCN.**

| Method | Opportunity | | | | InsectSound | | | mHealth | | | SAMSUNG | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1/2 | 1/4 | 1/8 | 1/16 | 1/2 | 1/4 | 1/8 | 1/2 | 1/4 | 1/8 | 1/2 | 1/4 | 1/8 |
| Fully-Supervised(100%) | 67.24 | 66.66 | 64.83 | 63.37 | 70.26 | 64.74 | 51.01 | 93.08 | 92.51 | 90.57 | 92.86 | 91.38 | 89.81 |
| | (±1.93) | (±1.59) | (±2.20) | (±1.77) | (±0.39) | (±0.66) | (±1.14) | (±3.54) | (±3.67) | (±2.73) | (±0.43) | (±0.20) | (±0.44) |
| Fully-Supervised(10%) | 56.66 | 54.15 | 53.98 | 49.35 | 55.18 | 48.79 | 34.16 | 86.92 | 83.78 | 77.07 | 85.06 | 83.77 | 81.23 |
| | (±1.98) | (±1.02) | (±1.41) | (±2.63) | (±0.83) | (±1.96) | (±0.39) | (±2.33) | (±2.90) | (±4.00) | (±0.66) | (±0.99) | (±0.67) |
| FixMatch | 64.89 | 64.20 | 62.42 | 61.22 | 61.24 | 55.76 | 37.38 | 89.64 | 83.82 | 83.14 | 88.50 | 87.98 | 86.71 |
| | (±1.75) | (±3.32) | (±0.87) | (±1.25) | (±1.14) | (±1.05) | (±0.82) | (±2.68) | (±2.70) | (±2.77) | (±0.66) | (±0.63) | (±0.36) |
| FreeMatch | 64.66 | 61.25 | 63.23 | 58.73 | **64.10** | 56.78 | 41.22 | 87.34 | 87.89 | 83.25 | 89.45 | 88.79 | 86.81 |
| | (±1.62) | (±2.84) | (±1.59) | (±2.09) | (±1.14) | (±0.96) | (±1.31) | (±6.67) | (±5.45) | (±3.85) | (±0.58) | (±0.72) | (±0.80) |
| ReMixMatch | 66.52 | 63.75 | 61.66 | 61.22 | 63.95 | 53.42 | 36.30 | **90.49** | 87.91 | 82.91 | 85.99 | 84.24 | 84.27 |
| | (±1.86) | (±1.80) | (±1.70) | (±1.74) | (±1.45) | (±1.85) | (±1.01) | (±4.20) | (±5.63) | (±3.66) | (±0.96) | (±1.06) | (±0.42) |
| *SemiTSR* | **67.39** | **66.25** | **65.33** | **64.59** | 61.34 | **58.04** | **47.01** | 90.41 | **90.34** | **89.42** | **90.21** | **90.40** | **89.69** |
| | (±1.21) | (±2.95) | (±3.43) | (±0.97) | (±1.65) | (±0.78) | (±0.36) | (±3.12) | (±1.93) | (±4.04) | (±0.75) | (±0.74) | (±0.74) |

already contains enough information, eliminating the necessity for reconstruction into a high sampling rate.

The results for the UCR datasets are shown in Table 4. For all datasets, *SemiTSR* outperforms the semi-supervised learning baselines. In particular, the *Mallat* and *NonInvasiveFetalECGThorax1* datasets, which were down-sampled at much lower rates than the other datasets, exhibited substantial performance enhancements. In the absence of a sophisticated upsampling method like *SemiTSR*, it would be difficult to recover the ground-truth time series in harsh down-sampling environments.

## 4.3 Effectiveness of Temporal Upsampler

*4.3.1 Quantitative Analysis.* To show the effectiveness of the temporal upsampler, we compare the classification performance with the existing upsampling methods: sub-pixel convolution [47] and LTE[3] [33]. Model configurations except the upsampler are all identically set. According to Table 5, the temporal upsampler performs better than the other upsampling methods in most combinations. Wide performance gap at the lowest sampling rate implies that our method learns temporal patterns better than the other upsamplers even if a small amount of information is available. In cases where data exhibits clear periodicity (e.g., *mHealth*), we observe that LTE outperforms sub-pixel convolution. We hypothesize that it is because LTE aims to restore the local periodicity accurately. Further analysis on the reconstruction quality including MSE comparison is presented in Appendix D.

[3]https://github.com/jaewon-lee-b/lte

*4.3.2 Qualitative Analysis.* As shown in Figures 9 and 10, the temporal upsampler reconstructs high-sampling-rate time series better than the other upsamplers in effect. In Figure 9, the first column demonstrates that the local periodicity is precisely estimated by the temporal upsampler, whereas the periodicity reconstructed by sub-pixel convolution is shifted in time relative to the ground-truth periodicity. LTE neither precisely estimates frequency nor amplitude. The second column shows that our method better estimates the amplitude of the changing periodicity. For Figure 10, the temporal upsampler restores the meaningful periodicity, whereas the others generate false noisy peaks. Especially in the second column, sub-pixel convolution generates false periodicities and LTE generates overly smoothed time series, whereas the temporal upsampler effectively recovers the periodicity.

## 4.4 Ablation Study

In order to demonstrate the contribution of each loss term in *SemiTSR*, we train the model without a specific loss term. That is, each weight parameter $\lambda_{time}$, $\lambda_{freq}$, $\lambda_{consist\_u}$, and $\lambda_{consist\_l}$

**Table 4: Accuracy comparison of semi-supervised learning methods on ten UCR time-series classification datasets.**

| Data | Label Ratio | Sampling Rate | Fully-Sup (Label Ratio%) | Fully-Sup (100%) | FixMatch | FreeMatch | ReMixMatch | *SemiTSR* |
|---|---|---|---|---|---|---|---|---|
| ElectricDevices | 10 | 1/2 | 55.01 (±0.83) | 64.67 (±0.60) | 58.93 (±1.04) | 57.56 (±1.13) | 58.31 (±1.02) | **60.17** (±1.02) |
| | 10 | 1/4 | 52.89 (±1.86) | 59.53 (±1.05) | 54.41 (±1.74) | 53.36 (±0.98) | 55.52 (±0.64) | **55.95** (±1.34) |
| | 10 | 1/8 | 49.17 (±1.15) | 57.67 (±0.87) | 51.41 (±1.92) | 51.88 (±1.64) | 52.32 (±0.89) | **53.29** (±1.42) |
| SwedishLeaf | 20 | 1/8 | 59.26 (±0.91) | 76.74 (±1.11) | 66.96 (±1.27) | 61.93 (±1.51) | 52.74 (±3.02) | **76.15** (±2.19) |
| FacesUCR | 10 | 1/8 | 61.67 (±2.01) | 86.67 (±3.70) | 66.50 (±5.35) | 65.17 (±3.68) | 69.33 (±1.03) | **73.33** (±1.25) |
| FaceAll | 10 | 1/8 | 50.96 (±2.63) | 81.38 (±1.22) | 56.30 (±1.26) | 55.85 (±1.15) | 56.89 (±2.10) | **65.90** (±0.72) |
| Mallat | 10 | 1/32 | 91.48 (±0.85) | 96.02 (±0.56) | 92.82 (±0.68) | 94.54 (±0.36) | 86.06 (±2.52) | **95.37** (±0.42) |
| NonInvasiveFetalECGThorax1 | 10 | 1/25 | 40.56 (±2.26) | 79.25 (±0.44) | 39.68 (±2.37) | 42.95 (±1.10) | 33.76 (±0.27) | **58.59** (±3.17) |
| MedicalImages | 10 | 1/8 | 49.43 (±2.37) | 63.69 (±1.82) | 55.03 (±1.61) | 54.16 (±0.25) | 53.63 (±0.62) | **55.29** (±1.60) |
| ShapesAll | 10 | 1/16 | 30.78 (±5.85) | 73.78 (±0.91) | 30.00 (±4.77) | 28.28 (±4.11) | 32.17 (±2.23) | **33.64** (±2.37) |
| UWaveGestureLibraryAll | 10 | 1/8 | 83.34 (±1.65) | 96.27 (±0.23) | 84.71 (±1.82) | 84.12 (±1.26) | 61.60 (±10.08) | **87.98** (±0.47) |
| Phoneme | 10 | 1/8 | 12.62 (±1.01) | 23.21 (±0.88) | 13.55 (±2.38) | 12.15 (±0.38) | 13.24 (±0.79) | **16.28** (±1.67) |

**Table 5: Accuracy comparison of our temporal upsampler and the others based on the same deep feature encoder.**

| Labeled Ratio | Upsmapler | Opportunity | | | | InsectSound | | | mHealth | | | SAMSUNG | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1/2 | 1/4 | 1/8 | 1/16 | 1/2 | 1/4 | 1/8 | 1/2 | 1/4 | 1/8 | 1/2 | 1/4 | 1/8 |
| 20% | Sub-Pixel Conv | 71.54 (±1.46) | **71.98** (±1.52) | 70.22 (±2.58) | 65.12 (±1.50) | **65.54** (±0.54) | 62.40 (±1.67) | 50.29 (±0.77) | 93.49 (±0.97) | 92.87 (±1.16) | 90.65 (±2.60) | 93.07 (±0.23) | 92.41 (±0.84) | **91.72** (±0.60) |
| | LTE | 67.15 (±2.31) | 68.58 (±1.22) | 67.23 (±1.95) | 63.40 (±1.28) | 64.89 (±1.17) | 61.33 (±0.84) | 48.31 (±0.49) | 92.82 (±2.82) | 94.41 (±3.62) | 92.63 (±1.54) | 92.84 (±0.18) | 91.85 (±0.19) | 90.84 (±0.47) |
| | *SemiTSR* | **71.59** (±2.09) | 69.85 (±2.39) | **71.15** (±1.88) | **66.69** (±2.44) | 65.32 (±0.98) | **62.43** (±0.77) | **50.99** (±0.61) | **93.76** (±1.18) | **94.77** (±2.4) | **93.10** (±1.81) | **93.20** (±0.54) | **92.50** (±0.57) | 91.64 (±0.89) |
| 10% | Sub-Pixel Conv | 66.92 (±2.88) | 66.22 (±2.20) | 65.22 (±2.49) | 62.86 (±1.71) | 58.88 (±1.76) | 55.82 (±1.07) | 45.72 (±1.15) | 87.05 (±3.10) | 89.27 (±2.02) | 86.22 (±2.03) | 89.53 (±0.80) | 89.48 (±0.69) | 89.49 (±0.72) |
| | LTE | 60.05 (±2.64) | 63.29 (±1.86) | 62.68 (±2.32) | 59.80 (±1.48) | 60.29 (±0.37) | 56.01 (±0.49) | 44.46 (±0.99) | 90.36 (±3.88) | 89.30 (±2.18) | 88.11 (±3.56) | 90.16 (±0.64) | 89.53 (±0.65) | 89.13 (±0.44) |
| | *SemiTSR* | **67.39** (±1.21) | **66.25** (±2.95) | **65.33** (±3.43) | **64.59** (±0.97) | **61.34** (±1.65) | **58.04** (±0.78) | **47.01** (±0.36) | **90.41** (±3.12) | **90.34** (±1.93) | **89.42** (±4.04) | **90.21** (±0.75) | **90.40** (±0.74) | **89.69** (±0.74) |

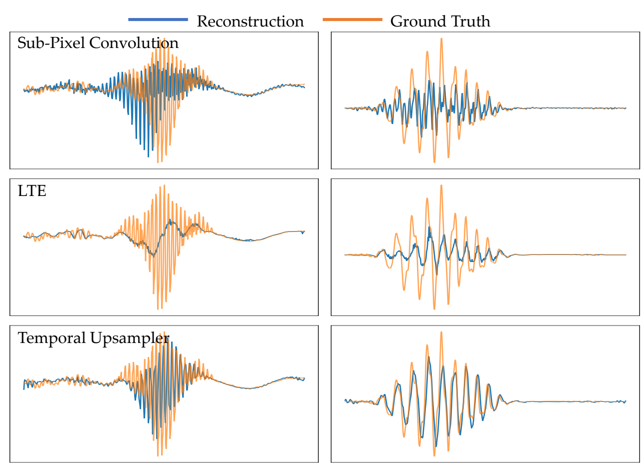



**Figure 9: Visualization of *InsectSound* reconstruction using different upsampling methods. $\gamma$ is $\frac{1}{8}$ and $l$ is 20% . The two columns are different instances.**

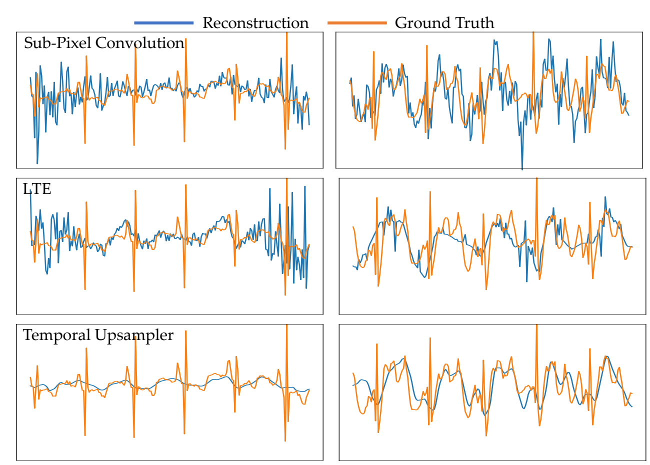



**Figure 10: Visualization of mHealth reconstruction using different upsampling methods. $\gamma$ is $\frac{1}{8}$ and $l$ is 10% . The two columns are different instances.**

was set to zero in (i)–(iv) respectively. Table 6 shows the result. Performance gain by shift consistency regularization verifies that learning various phases from the distribution of unlabeled data improves generalization. Depending on the dataset, the effect of reconstruction loss on Fourier-transformed time series varies. It will be the future work to adaptively adjust the weight on frequency domain reconstruction considering the periodicity of the data. To further demonstrate the contribution of an attention layer, we con-

**Table 6: Ablation study on each loss term where 10% of the data is labeled and the low sampling rate is $\frac{1}{8}$ except for *Opportunity* ($\frac{1}{16}$).**

| | Opportunity | InsectSound | mHealth | SAMSUNG |
|---|---|---|---|---|
| (i) w/o $\mathcal{L}_{time}$ | 59.65 (±1.35) | 45.00 (±1.83) | 88.21 (±4.84) | 83.60 (±4.38) |
| (ii) w/o $\mathcal{L}_{freq}$ | 60.49 (±2.25) | 42.68 (±1.08) | **89.72** (±3.22) | 89.07 (±0.60) |
| (iii) w/o $\mathcal{L}_{consist\_u}$ | 57.54 (±0.50) | 46.39 (±0.78) | 88.99 (±2.95) | 89.23 (±0.95) |
| (iv) w/o $\mathcal{L}_{consist\_l}$ | 62.22 (±3.92) | 40.15 (±0.96) | 87.58 (±3.27) | 88.09 (±0.41) |
| (v) w/o *attention* | 61.82 (±2.03) | 46.63 (±0.60) | 87.09 (±3.61) | 89.22 (±0.49) |
| *SemiTSR* | **64.59** (±0.97) | **47.01** (±0.36) | 89.42 (±4.04) | **89.37** (±0.78) |

**Table 7: Classification accuracy compared to imputation methods when $l$ is 10% and $\gamma$ is $\frac{1}{8}$ except for *Opportunity* ($\frac{1}{16}$).**

| | Opportunity | InsectSound | mHealth | SAMSUNG |
|---|---|---|---|---|
| mTAND | 58.70 (±3.03) | 25.26 (±0.91) | 89.24 (±3.32) | 86.95 (±1.22) |
| HetVAE | 62.73 (±1.63) | 10.18 (±0.48) | 66.56 (±2.39) | 83.34 (±0.51) |
| *SemiTSR* | **64.59** (±0.97) | **47.01** (±0.36) | **89.42** (±4.04) | **89.37** (±0.78) |

duct an ablation study where interpolated latent features directly go through the convolution layer. As shown in (v) of Table 6, attention mechanism based on the time embedding improves the classification accuracy in all datasets.

## 4.5 Comparison with Imputation Methods

To further show the effectiveness of *SemiTSR*, we compare it with existing imputation methods [49, 50] followed by a classifier. To be specific, an imputation model is pre-trained using both the low-sampling-rate and high-sampling-rate data, while the classification loss on the labeled high-sampling-rate data supervises the classifier and the imputation model. Then, the pre-trained imputation model replaces *SemiTSR*'s reconstructor for semi-supervised learning. For mTAND[4] [50], we follow their setting which uses the encoder output as the classifier input. For HetVAE[5] [49], we use the imputed data at a high sampling rate as the classifier input. Table 7 shows that *SemiTSR* with our reconstructor performs better than the imputation-based semi-supervised learning. It demonstrates that the fixed-rate upsampler resolves semi-supervised learning at a low sampling rate more effectively than the imputation methods which assume irregular input time series. In other words, imputation methods followed by a classifier are not optimal for our problem setting.

## 4.6 A Case Study on an ECG Dataset with Heterogeneous Sampling Rates

Extensive healthcare data has been generated as a result of recent developments in wearable medical devices, albeit at heterogeneous sampling rates [60]. This challenge directly fits our novel problem, *semi-supervised learning with low-sampling-rate time series*, so we conducted a case study on an ECG dataset [23, 43, 44] of mixed sampling rates. We specifically used a dataset at 500Hz and 1000Hz, segmented into three-second intervals. The dataset comprises 12 features, 7 categories, 4000 training instances, and 1000 test instances for each sampling rate. According to Table 8, *SemiTSR* outperforms semi-supervised learning baselines as well as other upsampler methods on the inference of the 500Hz data by utilizing the 30% of the labeled 1000Hz data. It demonstrates that *SemiTSR* is extendable not only in situations where the sampling rate of a single device is varied but also in situations where data is generated from multiple devices with various sampling rates.

[4]https://github.com/reml-lab/mTAN
[5]https://github.com/reml-lab/hetvae

**Table 8: Result of the case study. Classification accuracy on 500Hz ECG data compared to semi-supervised learning baselines and other upsamplers.**

| Method | | Accuracy |
|---|---|---|
| Fully-Supervised | TCN(100%) | 71.90 |
| | TCN(30%) | 69.70 |
| Semi-Supervised | FixMatch | 74.00 |
| | FreeMatch | 71.70 |
| | ReMixMatch | 74.40 |
| Upsampler Variation | Sub-Pixel Conv | 73.10 |
| | LTE | 73.50 |
| *SemiTSR* (ours) | | **75.20** |

**More Results in the Appendix:** Appendix C reports (i) the classification accuracy on a different classifier backbone, Transformer, (ii) the computation efficiency of *SemiTSR*, and (iii) the classification accuracy of high-sampling-rate data. Further analysis of reconstruction quality is conducted in Appendix D.

## 5 Conclusion

This paper introduces *SemiTSR*, a semi-supervised learning framework for the low-sampling-rate time series. It aims to recover the lost information caused by reducing the sampling rate by reconstructing the time series at its original sampling rate and performing classification. In particular, our temporal upsampler considers relative phase and periodicity of the target timestamp by comparing the time embedding that reflects local context. Furthermore, consistency regularization on the instances of different phases enables to reconstruct and classify within a class more accurately. Experiments demonstrate improved classification accuracy for various low-sampling-rates and labeled ratios than prior semi-supervised learning methods. We anticipate that our work will contribute to time-series modeling where labels are scarce and sampling rate must be reduced due to practical constraints.

We expect that our novel upsampler can be transferred to enhance any time-series analysis tasks on low sampling rates. Nevertheless, the current design of phase consistency regularization is tailored for classification but is difficult to apply to forecasting. Extending the upsampler for low-sampling-rate forecasting would be a meaningful direction of future work.

## Acknowledgments

This work was partly supported by Mobile eXperience Business, Samsung Electronics Co., Ltd. (Few-shot Anomaly Detection and Root Cause Estimation Development) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) (No. 2023R1A2C2003690).

# References


[1] Richard Andersson et al. 2010. Sampling frequency and eye-tracking measures: how speed affects durations, latencies, and more. *Journal of Eye Movement Research* 3, 3 (2010).
[2] Davide Anguita et al. 2013. A public domain dataset for human activity recognition using smartphones.. In *ESANN*. 437–442.
[3] Shaojie Bai et al. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018).
[4] Oresti Banos et al. 2014. mHealthDroid: A novel framework for agile development of mobile health applications. In *IWAAL*. 91–98.
[5] Mathias Baumert et al. 2016. Effects of ECG sampling rate on QT interval variability measurement. *Biomedical Signal Processing and Control* 25 (2016), 159–164.
[6] Yoshua Bengio et al. 2011. Deep learners benefit more from out-of-distribution examples. In *AISTATS*. 164–172.
[7] David Berthelot et al. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*. 5049–5059.
[8] David Berthelot et al. 2019. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785* (2019).
[9] Joel S Burma et al. 2021. Insufficient sampling frequencies skew heart rate variability estimates: Implications for extracting heart rate metrics from neuroimaging and physiological data. *Journal of Biomedical Informatics* 123 (2021), 103934.
[10] Ricardo Chavarriaga et al. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042.
[11] Yanping Chen et al. 2014. Flying insect classification with inexpensive sensors. *Journal of Insect Behavior* 27 (2014), 657–677.
[12] Yinbo Chen et al. 2021. Learning continuous image representation with local implicit image function. In *CVPR*. 8628–8638.
[13] Kyunghyun Cho et al. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
[14] Ekin D Cubuk et al. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshop*. 702–703.
[15] Lucas de Carvalho Pagliosa and Rodrigo Fernandes de Mello. 2018. Semi-supervised time series classification on positive and unlabeled problems using cross-recurrence quantification analysis. *Pattern Recognition* 80 (2018), 53–63.
[16] Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552* (2017).
[17] William R Dieter et al. 2005. Power reduction by varying sampling rate. In *ISLPED*. 227–232.
[18] Namık Kemal Eryol et al. 2003. Effects of calcium treatment on QT interval and QT dispersion in hypocalcemia. *American Journal of Cardiology* 91, 6 (2003), 750–2.
[19] Haoyi Fan et al. 2021. Semi-supervised time series classification by temporal relation prediction. In *ICASSP*. 3545–3549.
[20] Wei Fan et al. 2022. DEPTS: Deep Expansion Learning for Periodic Time Series Forecasting. In *ICLR*.
[21] Scott Fazackerley et al. 2021. Efficient Flash Indexing for Time Series Data on Memory-constrained Embedded Sensor Devices. In *SENSORNETS*. 92–99.
[22] Erik Fung et al. 2015. Electrocardiographic patch devices and contemporary wireless cardiac monitoring. *Frontiers in Physiology* 6 (2015).
[23] Ary L Goldberger et al. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation* 101, 23 (2000), e215–e220.
[24] Lukas Gorzelniak et al. 2013. Does the low power mode of the actigraph GT3X+ accelerometer influence the device output in sleep research in healthy subjects?. In *MedInfo*. 1172.
[25] Hassan Ismail Fawaz et al. 2019. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery* 33 (2019), 917–963.
[26] Brian Kenji Iwana and Seiichi Uchida. 2021. An empirical survey of data augmentation for time series classification with neural networks. *Plos One* 16, 7 (2021), e0254841.
[27] Shayan Jawed et al. 2020. Self-supervised learning for semi-supervised time series classification. In *PAKDD*. 499–511.
[28] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[29] T Kurp et al. 2012. Adaptive sensing for energy-efficient manufacturing system and process monitoring. *CIRP Journal of Manufacturing Science and Technology* 5, 4 (2012), 328–336.
[30] Guokun Lai et al. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *SIGIR*. 95–104.
[31] Samuli Laine and Timo Aila. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242* (2016).
[32] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*.
[33] Jaewon Lee and Kyong Hwan Jin. 2022. Local texture estimator for implicit representation function. In *CVPR*. 1929–1938.
[34] Jingyun Liang et al. 2021. Swinir: Image restoration using swin transformer. In *ICCV*. 1833–1844.
[35] Jason Lines and Anthony Bagnall. 2014. Ensembles of elastic distance measures for time series classification. In *SDM*. 524–532.
[36] Shizhan Liu et al. 2021. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *ICLR*.
[37] Ilya Loshchilov and Frank Hutter. 2016. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).
[38] Ping Lou et al. 2020. A data-driven adaptive sampling method based on edge computing. *Sensors* 20, 8 (2020), 2174.
[39] Lars Mescheder et al. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*. 4460–4470.
[40] Xiaoye Miao et al. 2021. Generative semi-supervised learning for multivariate time series imputation. In *AAAI*. 8983–8991.
[41] Hangwei Qian et al. 2021. Latent independent excitation for generalizable sensor-based cross-person activity recognition. In *AAAI*. 11921–11929.
[42] Jinwen Qiu et al. 2018. Multivariate Bayesian Structural Time Series Model. *Journal of Machine Learning Research* 19, 1 (2018), 1–33.
[43] Matthew A Reyna et al. 2021. Will two do? Varying dimensions in electrocardiography: the PhysioNet/Computing in Cardiology Challenge 2021. In *2021 Computing in Cardiology (CinC)*, Vol. 48. 1–4.
[44] Matthew A Reyna et al. 2022. Issues in the automated classification of multilead ECGs using heterogeneous labels and populations. *Physiological Measurement* 43, 8 (2022), 084001.
[45] Peter R Rijnbeek et al. 2001. Minimum bandwidth requirements for recording of pediatric electrocardiograms. *Circulation* 104, 25 (2001), 3087–3090.
[46] Burr Settles. 2009. *Active learning literature survey*. Technical Report 1648. University of Wisconsin-Madison.
[47] Wenzhe Shi et al. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*. 1874–1883.
[48] Yooju Shin et al. 2023. Context consistency regularization for label sparsity in time series. In *ICML*. 31579–31595.
[49] Satya Narayan Shukla and Benjamin M Marlin. 2021. Heteroscedastic Temporal Variational Autoencoder For Irregularly Sampled Time Series. *arXiv preprint arXiv:2107.11350* (2021).
[50] Satya Narayan Shukla and Benjamin M Marlin. 2021. Multi-time attention networks for irregularly sampled time series. *arXiv preprint arXiv:2101.10318* (2021).
[51] Vincent Sitzmann et al. 2020. Implicit neural representations with periodic activation functions. In *NeurIPS*. 7462–7473.
[52] Kihyuk Sohn et al. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*. 596–608.
[53] Ilya Sutskever et al. 2013. On the importance of initialization and momentum in deep learning. In *ICML*. 1139–1147.
[54] Ashish Vaswani et al. 2017. Attention is all you need. In *NeurIPS*. 6000–6010.
[55] Haishuai Wang et al. 2019. Time series feature learning with labeled and unlabeled data. *Pattern Recognition* 89 (2019), 55–66.
[56] Yidong Wang et al. 2022. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246* (2022).
[57] Gary M Weiss et al. 2019. Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living. *IEEE Access* 7 (2019), 133190–133202.
[58] Qingsong Wen et al. 2022. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125* (2022).
[59] Gerald Woo et al. 2022. CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting. In *ICLR*.
[60] Qiao Xiao et al. 2023. Deep Learning-Based ECG Arrhythmia Classification: A Systematic Review. *Applied Sciences* 13, 8 (2023), 4964.

# A Datasets

*Opportunity*, *InsectSound*, and *mHealth* are widely-used public datasets in time-series classification, while *SAMSUNG* is a proprietary dataset obtained from a server monitoring system.

**Opportunity** [10] is a human activity recognition dataset measured by wearable, object, and ambient sensors at a sampling rate of 30Hz. We preprocessed the dataset [41] and removed Null class from the 18 original gesture classes: Open / Close Dishwasher / Fridge / Drawer1(2,3) / Door1(2), Drink from Cup, Clean Table, Toggle Switch, and Null.

**InsectSound** [11] consists of audio time-series data derived from insect wingbeat to classify the species and determine the sex. It is collected at a sampling rate of 6000Hz.

**mHealth** [4] records body motions and vital signs measured from 3D accelerometers, 3D gyroscopes, 3D magnetometers, and electrocardiograms, sampled at a rate of 50Hz. It is categorized into 12 distinct activity classes: Stand, Sit, Lie, Walk, Upstairs, Waist Bend, Raise Arms, Crouch, Cycle, Jog, Run, and Jump.

**SAMSUNG** consists of time-series data obtained from a server monitoring system, collected every two minutes. The dataset is labeled by the modules to which the servers belong.

# B Details on Semi-Supervised Learning Baselines

For augmentation strategies, FixMatch and FreeMatch utilize RandAugment [14], and ReMixMatch proposes CTAugment. However, a time series is vulnerable to arbitrary augmentations due to its diverse and unstationary distributions. Also, forcing invariances which are appropriate to the image domain decreases downstream task performance depending on the dataset [26]. Accordingly, we selected the augmentation [6, 16] that works well for each dataset, as described in Table 9. Since unlabeled and labeled data share the window size in typical semi-supervised learning, we downsample the labeled high-sampling-rate data to have the same window size with the unlabeled low-sampling-rate data.

**Table 9: Augmentation strategy for semi-supervised learning baselines. $\epsilon$ is a Gaussian noise variable, $s$ is a scale factor, and $n, h$ are the number of holes and the relative length of the hole to the window.**

| Datasets | Weak Augmentation | Strong Augmentation |
|---|---|---|
| Opportunity | jitter $\epsilon \sim N(0, 0.3)$ | scale $s \sim N(1.1, 0.8)$ |
| InsectSound | cutout $n = 1, h = 0.1$ | cutout $\begin{cases} n = 2, h = 0.1 & \gamma = \frac{1}{2} \\ n = 1, h = 0.2 & \gamma \in (\frac{1}{4}, \frac{1}{8}) \end{cases}$ |
| mHealth | jitter $\epsilon \sim N(0, 0.3)$ | scale $s \sim N(1.1, 0.8)$ |
| SAMSUNG | cutout $n = 1, h = 0.1$ | cutout $n = 2, h = 0.1$ |

# C Additional Experiments

**Classification Accuracy on Different Classifier Backbones:** Table 10 compares the classification accuracy of *SemiTSR* and semi-supervsied learning baselines as well as fully supervised learning when the classifier backbone $\mathcal{M}_{cls}$ is Transformer. *SemiTSR* outperforms the other semi-supervised learning baselines in various datasets, low-sampling-rates, and labeled ratios.

**Table 10: Accuracy comparison of semi-supervised learning methods when the classifier backbone is Transformer.**

| Method | Opportunity | | mHealth | |
|---|---|---|---|---|
| | 1/8 | 1/16 | 1/4 | 1/8 |
| Fully-Supervised(100%) | 65.06 (±1.11) | 62.59 (±1.08) | 91.35 (±1.40) | 92.68 (±1.82) |
| Fully-Supervised(20%) | 59.09 (±2.74) | 55.91 (±1.61) | 85.23 (±3.25) | 85.19 (±1.43) |
| FixMatch(20%) | 67.96 (±2.55) | 64.61 (±0.91) | 91.60 (±2.33) | 89.26 (±1.85) |
| FreeMatch(20%) | 67.71 (±1.94) | 65.50 (±1.33) | 92.43 (±4.57) | 89.85 (±1.84) |
| ReMixMatch(20%) | 68.72 (±0.72) | 66.13 (±1.20) | 94.79 (±1.14) | 94.26 (±2.41) |
| *SemiTSR*(20%) | **69.43** (±2.02) | **68.30** (±1.29) | **96.34** (±2.17) | **94.45** (±1.12) |
| Fully-Supervised(10%) | 54.33 (±2.70) | 51.50 (±1.09) | 79.85 (±3.41) | 76.58 (±4.37) |
| FixMatch(10%) | 63.94 (±2.29) | 61.36 (±1.83) | 84.64 (±3.04) | 81.03 (±5.59) |
| FreeMatch(10%) | 65.10 (±1.19) | 62.93 (±1.50) | 88.04 (±2.84) | 84.01 (±5.40) |
| ReMixMatch(10%) | **68.32** (±1.22) | **64.70** (±1.09) | 88.29 (±3.58) | 86.22 (±6.16) |
| *SemiTSR*(10%) | 67.20 (±1.13) | **64.69** (±1.00) | **92.90** (±2.69) | **90.77** (±3.76) |

**Table 11: Inference time in seconds of low-sampling-rate data classification and SemiTSR (reconstruction+classification).**

| Method\Datasets | Opportunity(1/8) | mHealth(1/8) |
|---|---|---|
| TCN | 0.00313 | 0.00409 |
| SemiTSR | 0.01161 | 0.01412 |

**Table 12: Classification accuracy on high-sampling-rate data.**

| Method | Labeled Ratio | Sampling Rate | Opportunity | mHealth |
|---|---|---|---|---|
| TCN | 20% | 1/1 | 65.01 (±2.00) | 92.26 (±4.80) |
| SemiTSR | 20% | 1/4 | 71.80 (±1.50) | 95.25 (±2.77) |
| SemiTSR | 20% | 1/8 | 71.41 (±1.54) | 93.21 (±2.72) |
| TCN | 10% | 1/1 | 59.20 (±1.80) | 87.64 (±3.35) |
| SemiTSR | 10% | 1/4 | 66.22 (±3.14) | 90.93 (±2.32) |
| SemiTSR | 10% | 1/8 | 65.41 (±3.20) | 91.24 (±3.30) |

**Computation Efficiency:** We evaluated the inference time, in seconds, for the classification model and *SemiTSR*, which involves the classification model subsequent to the reconstruction model. Both the classification model and *SemiTSR* use the low-sampling-rate data as their input. We used *Opportunity* and *mHealth* due to their largest feature dimensionality and sequence length. Table 11 shows the results. When the sampling rate is 1/8, *SemiTSR* took

**Table 13: MSE comparison of our temporal upsampler and the others based on the same deep feature encoder.**

| Labeled Ratio | Upsmapler | Opportunity | | | | InsectSound | | | mHealth | | | SAMSUNG | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1/2 | 1/4 | 1/8 | 1/16 | 1/2 | 1/4 | 1/8 | 1/2 | 1/4 | 1/8 | 1/2 | 1/4 | 1/8 |
| 20% | Sub-Pixel Conv | 0.1296 (±0.0046) | 0.3035 (±0.0009) | 0.5124 (±0.0063) | 0.6799 (±0.0057) | 0.1364 (±0.0024) | 0.5431 (±0.0132) | 1.1083 (±0.0362) | 0.1735 (±0.0130) | 0.2893 (±0.0034) | 0.4140 (±0.0206) | **0.0147** (±0.0003) | **0.0228** (±0.0002) | **0.0300** (±0.0007) |
| | LTE | **0.1072** (±0.0000) | **0.2792** (±0.0004) | 0.5185 (±0.0005) | 0.8056 (±0.0019) | 0.1487 (±0.0009) | 0.5328 (±0.0023) | **0.9469** (±0.0408) | 0.1674 (±0.0054) | 0.3045 (±0.0038) | 0.4567 (±0.0110) | 0.0150 (±0.0008) | 0.0232 (±0.0009) | **0.0300** (±0.0008) |
| | *SemiTSR* | 0.1179 (±0.0006) | 0.2811 (±0.0017) | **0.4795** (±0.0082) | **0.6290** (±0.0025) | **0.1352** (±0.0005) | **0.5288** (±0.0083) | 1.0023 (±0.0225) | **0.1659** (±0.0066) | **0.2836** (±0.0044) | **0.3930** (±0.0039) | **0.0147** (±0.0007) | 0.0249 (±0.0010) | 0.0307 (±0.0013) |
| 10% | Sub-Pixel Conv | 0.1414 (±0.0074) | 0.3308 (±0.0159) | 0.5444 (±0.0057) | 0.7132 (±0.0039) | 0.1573 (±0.0409) | 0.5494 (±0.0125) | 1.1530 (±0.0176) | 0.1833 (±0.0170) | 0.3062 (±0.0043) | 0.4257 (±0.0062) | 0.0150 (±0.0002) | 0.0241 (±0.0001) | 0.0305 (±0.0005) |
| | LTE | **0.1072** (±0.0000) | **0.2792** (±0.0004) | 0.5186 (±0.0005) | 0.8057 (±0.0019) | 0.1503 (±0.0009) | **0.5341** (±0.0028) | 1.0233 (±0.0442) | **0.1684** (±0.0065) | 0.3066 (±0.0035) | 0.4609 (±0.0115) | 0.0156 (±0.0008) | **0.0231** (±0.0005) | **0.0304** (±0.0006) |
| | *SemiTSR* | 0.1225 (±0.0030) | 0.3030 (±0.0090) | **0.5028** (±0.0066) | **0.6971** (±0.0449) | **0.1357** (±0.0005) | 0.5368 (±0.0122) | **0.9588** (±0.0329) | 0.1724 (±0.0101) | **0.2905** (±0.0044) | **0.3599** (±0.0439) | **0.0149** (±0.0007) | 0.0258 (±0.0011) | 0.0316 (±0.0009) |

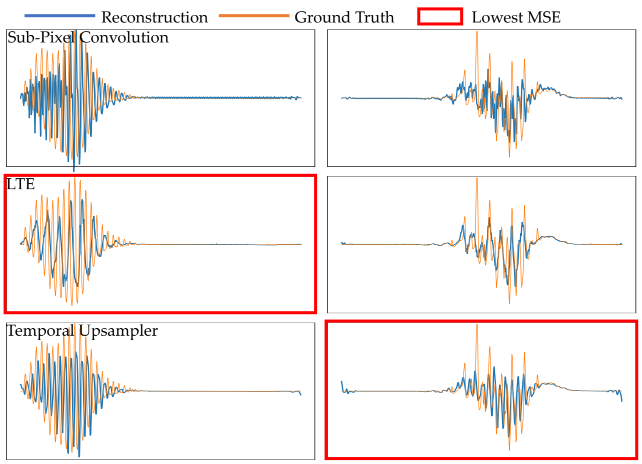



**Figure 11: Visualization of *InsectSound* reconstruction using different upsampling methods. $\gamma$ is $\frac{1}{8}$ and $l$ is 10%. The two columns are different instances.**

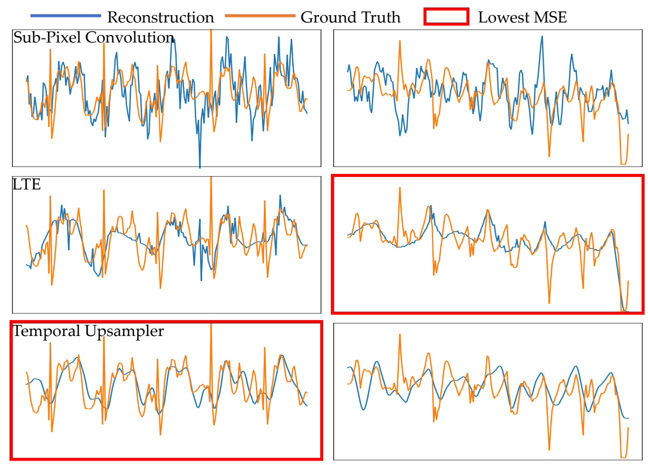



**Figure 12: Visualization of *mHealth* reconstruction using different upsampling methods. $\gamma$ is $\frac{1}{8}$ and $l$ is 10% . The two columns are different instances.**

3.45–3.71 times longer than the classification model only with low-sampling-rate data. Because the inference time per batch takes 8.5–10ms longer, it is worthwhile to incur the additional cost for saving storage and network load while simultaneously increasing accuracy. The train time is also linear to the inference time.

**Classification Accuracy on High-Sampling-Rate Data:** While our primary focus is on classifying low-sampling-rate data to reduce the costs associated with data gathering and labeling, our approach can also be applied in the mixed sampling-rates inference. We additionally conducted inference on high-sampling-rate data using the classification module of *SemiTSR*, which is trained with both the low-sampling-rate and high-sampling-rate data. As evident in Table 12, the performance of inference on high-sampling-rate data, when trained using *SemiTSR*, surpasses that of fully-supervised learning across all labeled ratios.

## D Additional Analysis on Reconstruction Quality

Table 13 shows the mean squared error of the reconstructed and ground-truth high-sampling-rate time series using various upsamplers. The temporal upsampler exhibited lower reconstruction error than the other methods in more than half of the combinations. It is obvious that the high classification accuracy in Table 5 is partly attributed to the high reconstruction quality in Table 13. In the meantime, the temporal upsampler was unable to achieve the lowest reconstruction error despite having the highest classification accuracy in certain combinations. We conjecture that this inconsistency is due to the fact that a low reconstruction error does not always guarantee a high reconstruction quality. In the first column of Figure 11, LTE exhibits the lowest reconstruction MSE despite predicting the frequency incorrectly. The second column of Figure 12 also shows that overly smoothed reconstruction of LTE has the lowest reconstruction MSE, while the temporal upsampler restores the meaningful periodicity.