Spectral Energy Allocation Enables Source-Free Domain Adaptation in Time Series Forecasting

Zhimin Mei[1] **Gezheng Xu**[1,*] **Mohammad Hossein Moslemi**[1] **Ruiyi Fang**[1]
Nima Hosseini Dashtbayaz[1] **Ziyan Wang**[1] **Jincheng Zhou**[1] **Haolin Jiang**[1]
Song Tang[2] **Charles Ling**[1] **Boyu Wang**[1,3]
[1]Western University, London, ON, Canada
[2]University of Shanghai for Science and Technology, Shanghai, China
[3]Vector Institute, Toronto, ON, Canada
{zmei22,gxu86,mohammad.moslemi,rfang32}@uwo.ca,
{nhosse5,zwan3444,jzhou774,hjian224,charles.ling}@uwo.ca,
tangs@usst.edu.cn bwang@csd.uwo.ca

Abstract

Source-free domain adaptation (SFDA) adapts a pre-trained source model to an unlabeled target domain without access to source data. However, existing SFDA studies have predominantly focused on classification tasks, leaving time series forecasting (TSF), a fundamentally different problem with continuous outputs and complex temporal dependencies, largely unexplored. In this work, we identify unique characteristics and key challenges of SFDA in TSF: 1) forecasting labels correspond to future temporal segments of the same underlying signal, enabling us to split target input and construct a self-supervised learning task, thereby training an accurate short-term forecasting model to provide high-quality pseudo-labels; 2) however, such supervision remains inherently *local* rather than *global* for long-horizon source model adaptation. To fill this gap, we propose the notion of Spectral Energy Allocation (SEA) pattern, defined over a short-long signal pair. The SEA pattern provides a structured correspondence that bridges signals across different temporal horizons, enabling reliable short-horizon pseudo-labels to guide long-horizon adaptation. Extensive experiments demonstrate that our method substantially outperforms baselines across different forecasting backbones.

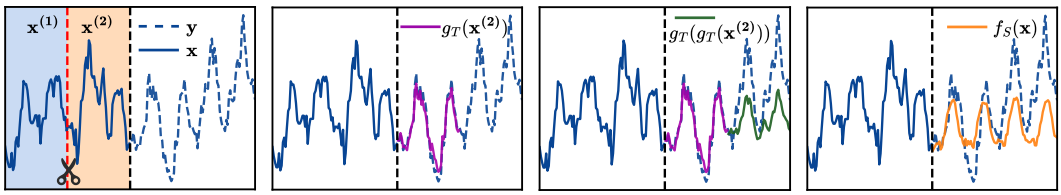
1 Introduction

Time series forecasting (TSF) plays a crucial role across various real-world domains, including weather [1–4], traffic [5–8], and finance [9–11], where accurate forecasting facilitates informed decision-making. Recently, numerous TSF deep models have been proposed to capture intricate patterns in time series data [12–20]. However, training such models relies heavily on a large amount of annotated data [21], and their performance can degrade substantially when deployed in target environments exhibiting distribution shifts [22]. A promising solution for this challenge is source-free domain adaptation (SFDA), which adapts a model pre-trained on source domain to an unlabeled target domain without access to source data [23, 24]. For instance, a neurophysiological signal forecasting model trained in a resource-rich hospital can be adapted to support seizure anticipation in a rural hospital [25, 26], where labeled data are scarce or unavailable.

However, existing SFDA methods are primarily designed for classification tasks, including time series classification, by exploiting the clustered structure of feature representations induced by categorical outputs [27–30]. By contrast, such characteristics are no longer present in TSF, as both

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) self-supervised model g_T training task

(b) high-quality pseudo labels from g_T

(c) error accumulation phenomenon

(d) source model zero-shot prediction

Figure 1: Feasibility and limitations of SFDA in TSF. The black dashed line separates the accessible target input $\mathbf{x}$ from the inaccessible target label $\mathbf{y}$. (a) We split $\mathbf{x}$ into history $\mathbf{x}^{(1)}$ and future $\mathbf{x}^{(2)}$ to train a target self-supervised model g_T. (b) g_T generates reliable short-horizon pseudo labels $g_T(\mathbf{x}^{(2)})$. (c) Rolling g_T for long-horizon prediction, e.g., $g_T(g_T(\mathbf{x}^{(2)}))$, suffers from error accumulation and cannot provide effective supervision. (d) The long-horizon source model prediction $f_S(\mathbf{x})$ is inaccurate under distribution shift.

inputs and outputs are continuous sequences with complex temporal dependencies, leading to poor adaptation performance with existing SFDA methods.

To bridge this gap, we identify the key characteristics and challenges of SFDA in TSF, as shown in Fig. 1: 1) In TSF, the input sequences and their prediction labels correspond to the same physical signal at different timestamps. This enables splitting the target input $\mathbf{x}$ into two temporal segments $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ to construct a self-supervised learning task (Fig. 1a), from which an accurate *short-term* forecasting model g_T can be trained. Crucially, since g_T is trained entirely from target input, it does not suffer from distribution shift and can therefore provide high-quality pseudo-labels for the early part of the target sequence (Fig. 1b) to guide source model f_S adaptation. 2) However, such guidance is inherently *local* rather than *global*: due to error accumulation, g_T cannot perform accurate long-horizon forecasting in the later part of the target sequence (Fig. 1c and 2a), and therefore fails to provide effective supervision for long-horizon f_S adaptation.

On the other hand, f_S performs long-horizon forecasting, but it is also inaccurate due to distribution shifts (Fig. 1d). This raises a fundamental question: **how can reliable short-horizon supervision be leveraged to facilitate long-horizon source model adaptation?** A key challenge is that short- and long-horizon sequences differ in length, making direct temporal alignment infeasible. However, in the frequency domain, both sequences share the same underlying spectral components but at different resolutions: the energy concentrated at a single frequency in a short sequence is resolved into multiple neighboring frequencies in the longer one. This provides a natural basis for establishing cross-horizon correspondence. We therefore introduce Spectral Energy Allocation (SEA) pattern to formalize this process by jointly modeling energy allocation coefficients and inter-frequency distances. Moreover, we empirically show the consistency of SEA patterns within target sequences, indicating that they capture domain-specific spectral characteristics. We are therefore motivated to correct $\mathbf{SEA}(g_T(\mathbf{x}^{(2)}), f_S(\mathbf{x}))$, aligning it with $\mathbf{SEA}(\mathbf{x}^{(1)}, \mathbf{x})$, thereby enabling global supervision for source model adaptation with short-horizon guidance. Our contributions can be summarized as:

(1) We characterize both feasibility and intrinsic limitations of SFDA in TSF, showing that target-only self-supervision provides reliable but inherently local guidance due to error accumulation.

(2) We introduce the notion of SEA pattern, which characterizes the spectral correspondence between a short-long signal pair rather than describing the property of a single signal. This allows long-horizon source model adaptation via short-horizon supervision. To the best of our knowledge, this is the first work exploring SFDA in TSF.

(3) We perform extensive experiments in diverse domain adaptation settings and validate our method consistently outperforms 13 existing SFDA and unsupervised domain adaptation baselines.

2 Related Work

Unsupervised Domain Adaptation in TSF. Most existing UDA methods for time series data are developed for time series classification, including adversarial adaptation methods such as DANN [31],

VRADA [32], CoDATS [33], SASA [34], AdvSKM [35], CLUDA [36], recent temporal-frequency methods such as RAINCOAT [37], ACON [38], and EDEN [39], as well as pseudo-labeling and causal-disentanglement methods such as TransPL [40] and CauDiTS [41]. In contrast, UDA for TSF has received less attention, despite its practical importance in real-world forecasting scenarios. Representative works include AdaRNN [42], which addresses temporal covariate shift to improve forecasting under distribution shift, DAF [43], which shares attention across domains to transfer knowledge, and GCA [44], which assumes stable causal structures across domains. However, these methods require access to source data, which is often impractical in privacy-sensitive applications.

Source-Free Domain Adaptation in Classification. Existing SFDA methods can be grouped into two categories: 1) generation-based approaches that synthesize source-like samples from the source model during adaptation [45–49]. 2) pseudo-label correction-based approaches that progressively refine the pseudo-label [23, 50–53]. Representative classification-oriented SFDA methods include SF(DA)^2 [29], ProDe [54], and DUET [55], which rely on augmentation, proxy-based adaptation, or decision-boundary regularization to improve target adaptation. Recently, SFDA has been explored for time series classification, including LEMON [56], MAPU [27], E-MAPU [57], and TERSE [28]. Despite their effectiveness, these methods rely on clustering structure in the feature space to facilitate pseudo-labeling. However, such assumptions no longer holds in forecasting.

Frequency-Domain Perspective in Time Series. Existing frequency-domain forecasting methods exploit the spectrum of individual signals for representation learning, filtering, normalization, or frequency-bias mitigation [14, 58–62]. However, directly aligning spectra does not resolve SFDA in TSF, since target forecasting labels are unavailable. In contrast, our proposed Spectral Energy Allocation (SEA) is not a spectrum-alignment objective: it does not describe the spectrum of a single signal, but characterizes the cross-resolution energy allocation relationship within a short-long signal pair. This pairwise formulation enables reliable short-horizon pseudo labels to guide long-horizon source-model adaptation, and constitutes one of the core contributions of this work. More importantly, SEA provides a label-free structural reference from accessible target input, making frequency-domain guidance feasible under the SFDA setting.

3 Methodology

In this section, we first present a self-supervised training strategy to provide reliable short-horizon supervision. However, such supervision is insufficient for long-horizon source model adaptation. Therefore, we further introduce Spectral Energy Allocation (SEA) pattern alignment. For simplicity, we assume the look-back window and the forecasting horizon are both L so that $f_S(\cdot;\theta_S) : \mathbb{R}^L \to \mathbb{R}^L$. We further show that our method naturally extends to general cases in the Appendix B.2.

3.1 Problem Formulation

Time Series Forecasting. Given a historical time series $\mathbf{x} = \{x_1, \ldots, x_L\} \in \mathbb{R}^{L\times C}$, where L is the lookback window length and C is the number of channels. The goal of Time Series Forecasting is to learn a model $f : \mathbb{R}^{L\times C} \to \mathbb{R}^{H\times C}$ that predicts the future H time steps $\hat{\mathbf{y}} = \{\hat{x}_{L+1}, \ldots, \hat{x}_{L+H}\} \in \mathbb{R}^{H\times C}$ approximating the ground truth $\mathbf{y} = \{x_{L+1}, \ldots, x_{L+H}\} \in \mathbb{R}^{H\times C}$. We denote $\mathbf{x}_{i:j} = \{x_i, x_{i+1}, \ldots, x_j\}$ as a slice of $\mathbf{x}$. For notation simplicity, we focus on the univariate case ($C = 1, \mathbf{x} \in \mathbb{R}^L$). For multivariate data, the loss is computed per channel and averaged.

Source-Free Domain Adaptation in Time Series Forecasting. Under SFDA, we are given a *source forecasting model* f_S pre-trained on a *labeled source dataset* $\mathcal{D}_S \triangleq \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_S}$, along with an *unlabeled target dataset* $\mathcal{D}_T \triangleq \{\mathbf{x}_j\}_{j=1}^{N_T}$. Our goal is to obtain a *target forecasting model* f_T that performs well on $\mathcal{D}_T$ by adapting f_S to $\mathcal{D}_T$ without access to $\mathcal{D}_S$ or target labels. For brevity, we denote an example in $\mathcal{D}_T$ as $\mathbf{x}$ when no ambiguity arises.

3.2 Self-Supervised Forecasting in the Target Domain

Target Sequence Segmentation. In TSF, input sequences and forecasting targets correspond to the same underlying physical signal at different timestamps. Motivated by this property, for each unlabeled target example $\mathbf{x} = \{x_1, x_2, \ldots, x_L\} \in \mathcal{D}_T$, we split it into two ordered segments $\mathbf{x}^{(1)} =$

$\mathbf{x}_{1:L_1} \in \mathbb{R}^{L_1}$ and $\mathbf{x}^{(2)} = \mathbf{x}_{L_1+1:L} \in \mathbb{R}^{L2}$, where $L_1 + L_2 = L$ and $k = L_1/L \in (0, 1)$ is the segmentation ratio controlling the relative lengths of the two segments. Applying this operation to all target examples reorganizes the original unlabeled dataset $\mathcal{D}_T = \{\mathbf{x}_i\}_{i=1}^{N_T}$ into:

$$\tilde{\mathcal{D}}_T = \{(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})\}_{i=1}^{N_T}. \tag{1}$$

Self-Supervised Model Training. Given $\tilde{\mathcal{D}}_T$, we construct a self-supervised learning task in the target domain. Specifically, for each pair $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \in \tilde{\mathcal{D}}_T$, the first segment $\mathbf{x}^{(1)}$ is treated as the input sequence, while the second segment $\mathbf{x}^{(2)}$ serves as its corresponding label. This naturally defines a short-term forecasting task, from which we train an auxiliary forecasting model $g_T(\cdot; \phi_T) : \mathbb{R}^{L_1} \to \mathbb{R}^{L_2}$ in the target domain by solving:

$$\min_{\phi_T} \mathbb{E}_{(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \sim \tilde{\mathcal{D}}_T} \Big[\mathrm{MSE}\big(g_T(\mathbf{x}^{(1)}; \phi_T),\ \mathbf{x}^{(2)}\big) \Big], \tag{2}$$

where ϕ_T denotes the learnable parameters of g_T, and $\mathrm{MSE}(\cdot, \cdot)$ is the Mean Squared Error loss [63]. We term g_T a *self-supervised model* since its training labels $\mathbf{x}^{(2)}$ are obtained by manually segmenting the unlabeled target sequences $\mathcal{D}_T$.

Pseudo Labels from Self-Supervised Model. As g_T is trained entirely on the target domain, it can provide reliable pseudo labels without being affected by the source-target distribution shift. For each unlabeled target example $\mathbf{x} \in \mathcal{D}_T$, we extract its most recent segment of length L_1 as input:

$$\mathbf{x}^{\text{in}} = \mathbf{x}_{L-L_1+1:L} \in \mathbb{R}^{L_1}, \qquad \hat{\mathbf{y}}^g = g_T(\mathbf{x}^{\text{in}}; \phi_T) \in \mathbb{R}^{L_2}, \tag{3}$$

where $\hat{\mathbf{y}}^g$ is the pseudo label produced by g_T. The segmentation ratio k determines the input and output lengths through $k = L_1/L$ and $L_2 = L - L_1$, and therefore controls the trade-off between pseudo-label horizon and accuracy. A smaller k yields a longer pseudo-label horizon but makes the forecasting task harder, while a larger k improves pseudo-label accuracy but shortens the available supervision horizon. To achieve a favorable trade-off, we fix $k = 0.5$ (i.e., $L_1 = L_2 = L/2$) across all experiments without any tuning, which eliminates the need to tune this hyperparameter in realistic SFDA scenarios where target-domain validation labels are unavailable. The empirical effect of k is reported in Fig. 3(b), which supports the effectiveness of this default choice.

Note that f_S performs long-horizon forecasting with output length L, whereas the pseudo label $\hat{\mathbf{y}}^g$ from g_T has length only $L/2 < L$, leading to a horizon mismatch. A naive idea is to recursively apply g_T twice in a rolling manner to obtain a length-L pseudo label. However, such a rolling strategy is unreliable because prediction errors accumulate over rolling times. The empirical evidence is provided in Fig. 2a, where the second-half pseudo-label MSE is consistently larger than that of first-half across 4 datasets. Here, the first-half and second-half pseudo-labels are obtained by applying g_T once and twice respectively, yielding $g_T(\mathbf{x}^{\text{in}})$ and $g_T(g_T(\mathbf{x}^{\text{in}}))$. This error accumulation phenomenon indicates that we can only trust $\hat{\mathbf{y}}^g$ as *local guidance* to supervises the first half of $\hat{\mathbf{y}}^f$, resulting in the **S**elf-**S**upervised **M**odel **P**seudo-**L**abel Loss **(SSM-PL)**:

$$\mathcal{L}_{\text{SSM-PL}} = \mathrm{MSE}\Big(\hat{\mathbf{y}}^f_{1:L/2},\, \hat{\mathbf{y}}^g\Big), \tag{4}$$

where $\hat{\mathbf{y}}^f = f_S(\mathbf{x}; \theta_S) \in \mathbb{R}^L$ is the long-horizon prediction from the source model.

3.3 Spectral Energy Allocation Pattern Alignment

Spectral Energy Allocation (SEA) Pattern. *To leverage the reliable short-horizon supervision provided by g_T to facilitate long-horizon source model adaptation*, we seek a structural correspondence between short and long horizons that can bridge local guidance and long-range forecasting. Specifically, consider two time series with different horizons sampled from the same underlying physical signal at the same sampling rate: a short-horizon sequence $\mathbf{x}_{1:L_s} = \{x_1, x_2, \ldots, x_{L_s}\} \in \mathbb{R}^{L_s}$ and a long-horizon sequence $\mathbf{x}_{1:L_l} = \{x_1, x_2, \ldots, x_{L_l}\} \in \mathbb{R}^{L_l}$, where $L_l > L_s$. We apply the real Fourier transform (rFFT) along the temporal dimension. For a time series $\mathbf{x}_{1:L} \in \mathbb{R}^L$, the rFFT produces a frequency-domain representation:

$$\begin{aligned} &\mathbf{X} = \mathrm{rFFT}(\mathbf{x}_{1:L}) = [X_1, X_2, \ldots, X_{L_f}] \in \mathbb{C}^{L_f}, \quad L_f = L/2 + 1, \\ &f_k = k/L \in \left[0, \tfrac{1}{2}\right], \quad k = 0, 1, \ldots, L/2 \end{aligned} \tag{5}$$

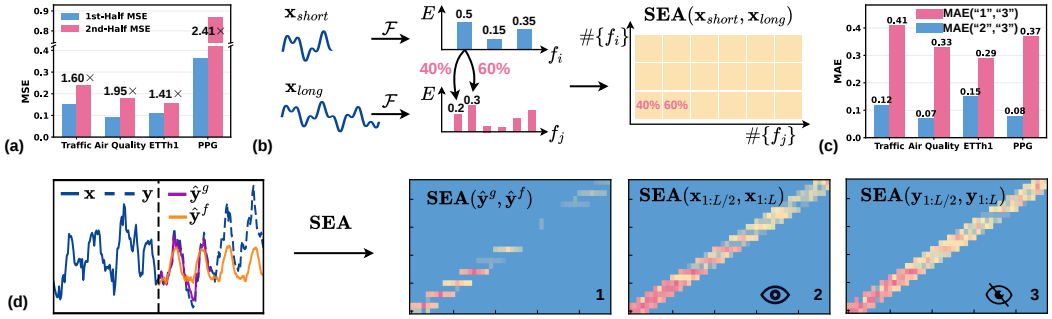


Figure 2: **(a) Error accumulation.** The first-half MSE is measured from $g_T(\mathbf{x}^{(2)})$, while the second-half MSE is measured from the rolling prediction $g_T(g_T(\mathbf{x}^{(2)}))$. **(b) Definition of spectral energy allocation.** SEA models how energy from one short-sequence frequency bin is allocated to multiple neighboring long-sequence frequency bins. **(c, d) SEA consistency.** Quantitative and visual comparisons under the ETTh2→ETTh1 setting show that the input-induced SEA is consistent within the label-induced SEA, motivating us to recover the inaccessible label property from accessible target input. More visualizations are shown in Appendix F.1.

The corresponding frequency resolution is defined as the spacing between adjacent bins:

$$\Delta f = f_{k+1} - f_k = 1/L \tag{6}$$

This shows that the long sequence admits a denser normalized-frequency grid over $[0, 0.5]$, so that energy concentrated at a single short-sequence frequency will be dispersed across multiple neighboring long-sequence frequencies, as illustrated in Fig. 2b. This observation raises a natural question: *Can we model the above "spectral energy allocation" process to bridge the short-horizon pseudo-labels from g_T and the long-horizon predictions of f_S?*

To answer it, we define the **S**pectral **E**nergy **A**llocation (SEA) pattern progressively as follows. We first compute the normalized spectral energy at each frequency f_k:

$$E(f_k) = |X_k|^2, \quad \bar{E}(f_k) = E_k \,/\, \sum_{u=0}^{L_f - 1} E(f_u) \tag{7}$$

Given a short sequence $\mathbf{x}_{1:L_s}$ and a longer one $\mathbf{x}_{1:L_l}$, their frequencies are $f_i^{(s)} = i/L_s$ and $f_j^{(l)} = j/L_l$ according to Eq. 5. The SEA pattern is defined as a matrix whose elements are the energy allocation coefficients between short–long frequency pairs $(f_i^{(s)}, f_j^{(l)})$:

$$\begin{aligned} &\mathbf{SEA}(\mathbf{x}_{1:L_s}, \mathbf{x}_{1:L_l}) \in \mathbb{R}^{L_f^{(s)} \times L_f^{(l)}} \\ &[\mathbf{SEA}(\mathbf{x}_{1:L_s}, \mathbf{x}_{1:L_l})]_{(i,j)} = \bar{E}^{(l)}(f_j^{(l)}) \,/\, \bar{E}^{(s)}(f_i^{(s)}) \end{aligned} \tag{8}$$

Considering energy is more likely to be allocated to neighboring frequency than distant ones [64], we introduce a Gaussian kernel weight with bandwidth $\sigma = 1/L$ matching the long-horizon frequency resolution, which gives more weight to close frequency, the effect of σ is shown in Appendix E.2:

$$\begin{aligned} &w(i,j) = \exp(-|f_i^{(s)} - f_j^{(l)}|^2 / 2\sigma^2) \\ &[\mathbf{SEA}(\mathbf{x}_{1:L_s}, \mathbf{x}_{1:L_l})]_{(i,j)} = w(i,j) * \bar{E}^{(l)}(f_j^{(l)}) / \bar{E}^{(s)}(f_i^{(s)}) \end{aligned} \tag{9}$$

The SEA pattern thus defines the mapping:

$$\mathbf{SEA} : \mathbb{R}^{L_s} \times \mathbb{R}^{L_l} \longrightarrow \mathbb{R}^{(L_s/2+1)\times(L_l/2+1)}. \tag{10}$$

Therefore, we could use the SEA pattern matrix to represent the spectral correspondence between a short time series and a longer one.

SEA Pattern Alignment. Building on the above formulation, we further investigate the properties of SEA patterns. Specifically, we construct and compare three types of SEA patterns:

Algorithm 1 Overall Adaptation Algorithm

Input: Source-pretrained model $f_S(\cdot;\theta_S)$; randomly initialized self-supervised model $g_T(\cdot;\phi_T)$; unlabeled target dataset $\mathcal{D}_T = \{\mathbf{x}_i\}_{i=1}^{N_T} \subset \mathbb{R}^L$.
Parameter: Balancing weight λ; segmentation ratio $k = 0.5$.
Output: Adapted source model parameters θ_S^*.

/ Stage 1: Self-supervised training of g_T on the target domain */*
Build $\tilde{\mathcal{D}}_T$ from $\mathcal{D}_T$ by splitting each $\mathbf{x}_i$ into $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ ▷ segmentation by ratio k
for each minibatch from $\tilde{\mathcal{D}}_T$ until convergence **do**
  $\phi_T \leftarrow \arg\min_{\phi_T} \mathbb{E}_{(\mathbf{x}_i^{(1)},\mathbf{x}_i^{(2)})\sim\tilde{\mathcal{D}}_T}\big[\mathrm{MSE}\big(g_T(\mathbf{x}_i^{(1)};\phi_T),\, \mathbf{x}_i^{(2)}\big)\big]$
end for
/ Stage 2: Source model adaptation via SSM-PL and SEA alignment */*
for each minibatch $\mathbf{x} \sim \mathcal{D}_T$ until convergence **do**
  $\hat{\mathbf{y}}^f = f_S(\mathbf{x};\theta_S) \in \mathbb{R}^L$ ▷ unreliable long-horizon prediction from f_S
  $\mathbf{x}^{\text{in}} = \mathbf{x}_{L/2+1:L}, \quad \hat{\mathbf{y}}^g = g_T(\mathbf{x}^{\text{in}};\phi_T) \in \mathbb{R}^{L/2}$ ▷ reliable short-horizon pseudo label
  $\mathcal{L}_{\text{SSM-PL}} = \mathrm{MSE}\big(\hat{\mathbf{y}}^f_{1:L/2}, \hat{\mathbf{y}}^g\big), \quad \mathcal{L}_{\text{SEA}} = \mathrm{MAE}\big(\mathbf{SEA}(\hat{\mathbf{y}}^g, \hat{\mathbf{y}}^f),\, \mathbf{SEA}(\mathbf{x}_{1:L/2}, \mathbf{x})\big).$
  $\theta_S \leftarrow \arg\min_{\theta_S} \lambda\,\mathcal{L}_{\text{SSM-PL}} + (1-\lambda)\,\mathcal{L}_{\text{SEA}}, \quad \theta_S^* \leftarrow \theta_S$ ▷ update source model
end for

① *Pseudo label-induced SEA*. Computed from the short-horizon prediction of the self-supervised model g_T and the long-horizon prediction of the source model f_S, i.e., $\mathbf{SEA}(\hat{\mathbf{y}}^g, \hat{\mathbf{y}}^f)$.

② *input-induced SEA*. Computed from the accessible target input sequence utilizing temporal segmentation, i.e., $\mathbf{SEA}(\mathbf{x}_{1:L/2}, \mathbf{x})$.

③ *label-induced SEA*. Computed from the true target future sequence, i.e., $\mathbf{SEA}(\mathbf{y}_{1:L/2}, \mathbf{y})$, which is inaccessible during adaptation.

Critically, Fig. 2(d) provides a qualitative comparison among the three SEA patterns. We observe that the input-induced SEA, $\mathbf{SEA}(\mathbf{x}_{1:L/2}, \mathbf{x})$, is highly similar to the label-induced SEA, $\mathbf{SEA}(\mathbf{y}_{1:L/2}, \mathbf{y})$, while the pseudo label-induced SEA, $\mathbf{SEA}(\hat{\mathbf{y}}^g, \hat{\mathbf{y}}^f)$, shows a clear deviation from both of them. To further quantify this observation, we compare two MAE distances across four benchmarks in Fig. 2c: $\mathrm{MAE}(\mathbf{SEA}(\hat{\mathbf{y}}^g, \hat{\mathbf{y}}^f), \mathbf{SEA}(\mathbf{y}_{1:L/2}, \mathbf{y}))$ and $\mathrm{MAE}(\mathbf{SEA}(\mathbf{x}_{1:L/2}, \mathbf{x}), \mathbf{SEA}(\mathbf{y}_{1:L/2}, \mathbf{y}))$. The results show that the input-induced SEA is much closer to the inaccessible label-induced SEA than the pseudo label-induced SEA.

The above experimental observations indicate that the SEA pattern captures domain-specific spectral characteristics and exhibits strong consistency within the same physical signal. This consistency can be leveraged to correct the long-horizon pseudo-labels generated by the source model, using both the target input and the shift-free short-horizon prediction. Specifically, the accessible input-induced SEA serves as a proxy for the inaccessible label-induced SEA, allowing us to adapt f_S by aligning the pseudo label-induced SEA with the input-induced SEA as follows:

$$\mathcal{L}_{\text{SEA}} = \mathrm{MAE}(\mathbf{SEA}(\hat{\mathbf{y}}^g, \hat{\mathbf{y}}^f), \mathbf{SEA}(\mathbf{x}^{(1)}, \mathbf{x})). \tag{11}$$

By minimizing this loss, we leverage the structured correspondence that the SEA pattern establishes across temporal horizons, enabling effective long-horizon source model adaptation.

Overall Adaptation Objective. Algorithm 1 summarizes our two-stage adaptation pipeline. In Stage 1, we train a target self-supervised model g_T by manually segmenting the unlabeled target sequences, which has the capability to provide reliable short-horizon pseudo labels. In Stage 2, g_T is fixed and we adapt the source-pretrained model f_S by combining temporal pseudo-label supervision with SEA pattern alignment as follows:

$$\mathcal{L} = \lambda\mathcal{L}_{\text{SSM}-\text{PL}} + (1-\lambda)\mathcal{L}_{\text{SEA}}, \tag{12}$$

Here, $\mathcal{L}_{\text{SSM}-\text{PL}}$ provides direct temporal supervision and thus preserves phase information while $\mathcal{L}_{\text{SEA}}$ provides spectral-energy supervision. This complementarity is supported by the ablation study in Table. 4. The balance weight λ between the two objectives is a hyperparameter whose effect is further studied in Sec. 4.4, as shown in Fig. 3(a).

Table 1: Cross-domain evaluation MSE and MAE on Air Quality and PPG-DaLiA datasets. We use iTransformer as our primary forecasting backbone, experiment across more backbones is in Tab 3.

	Methods	Venue	Air Quality								PPG-DaLiA							
			B→S		T→B		G→T		S→G		S1→S2		S3→S6		S5→S10		S14→S4	
			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
	Source Only	–	0.875	0.804	0.893	0.681	0.762	0.630	0.632	0.559	0.484	0.487	0.218	0.324	0.217	0.328	0.527	0.576
UDA-TSF	DANN-RNN	JMLR16	0.716	0.577	0.886	0.615	0.733	0.568	0.571	0.503	0.449	0.459	0.201	0.311	0.175	0.295	0.516	0.570
UDA-TSF	AdaRNN	CIKM21	0.719	0.578	0.875	0.612	0.698	0.554	0.550	0.499	0.394	0.430	0.197	0.308	0.164	0.285	0.498	0.517
UDA-TSF	DAF	ICML22	0.705	0.564	0.879	0.614	0.671	0.543	0.541	0.495	0.370	0.413	0.177	0.292	0.156	0.279	0.501	0.521
UDA-TSF	RAINCOAT	ICML23	0.776	0.579	0.891	0.651	0.742	0.572	0.595	0.534	0.468	0.469	0.213	0.322	0.190	0.311	0.513	0.568
UDA-TSF	CauDiTS	ICML24	0.726	0.573	0.874	0.611	0.698	0.553	0.539	0.491	0.382	0.425	0.186	0.302	0.162	0.286	0.496	0.516
UDA-TSF	ACON	NeurIPS24	0.779	0.678	0.892	0.618	0.740	0.581	0.594	0.511	0.476	0.472	0.211	0.319	0.187	0.305	0.519	0.572
UDA-TSF	TransPL	ICML25	0.722	0.581	0.883	0.628	0.696	0.546	0.568	0.508	0.387	0.419	0.193	0.296	0.169	0.287	0.511	0.554
UDA-TSF	EDEN	NeurIPS25	0.713	0.571	0.885	0.616	0.725	0.565	0.566	0.501	0.465	0.467	0.209	0.318	0.183	0.302	0.509	0.566
SFDA	MAPU	KDD23	0.823	0.795	0.891	0.621	0.753	0.599	0.607	0.521	0.479	0.475	0.213	0.320	0.205	0.331	0.516	0.570
SFDA	SF(DA)^2	ICLR24	0.749	0.635	0.884	0.617	0.689	0.549	0.578	0.509	0.452	0.461	0.210	0.319	0.193	0.318	0.507	0.565
SFDA	TERSE	KDD25	0.795	0.686	0.889	0.613	0.713	0.562	0.584	0.510	0.459	0.466	0.203	0.316	0.206	0.335	0.507	0.565
SFDA	ProDe	ICLR25	0.747	0.618	0.887	0.624	0.662	0.544	0.573	0.507	0.365	0.434	0.198	0.306	0.184	0.304	0.512	0.566
SFDA	DUET	NeurIPS25	0.788	0.683	0.893	0.622	0.728	0.568	0.594	0.519	0.465	0.456	0.204	0.314	0.198	0.312	0.518	0.564
	Ours	–	**0.686**	**0.559**	**0.872**	**0.604**	**0.645**	**0.532**	**0.523**	**0.465**	**0.361**	**0.396**	**0.145**	**0.228**	**0.135**	**0.261**	**0.481**	**0.493**

Note: Lower is better. Darker and lighter blue backgrounds denote the best and second-best results, respectively.

Table 2: Cross-domain evaluation MSE and MAE on Bay Area Traffic and ETT datasets. For ETT, H1/H2 denote ETTh1/ETTh2, and M1/M2 denote ETTm1/ETTm2.

	Methods	Venue	Bay Area Traffic								ETT							
			Spr.→Aut.		Sum.→Win.		Aut.→Sum.		Win.→Spr.		H1→H2		H2→H1		M1→M2		M2→M1	
			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
	Source Only	–	0.455	0.425	0.670	0.521	0.661	0.537	0.479	0.465	0.361	0.401	0.578	0.518	0.224	0.312	0.616	0.493
UDA-TSF	DANN-RNN	JMLR16	0.439	0.331	0.436	0.435	0.655	0.407	0.443	0.459	0.324	0.364	0.402	0.419	0.200	0.278	0.408	0.412
UDA-TSF	AdaRNN	CIKM21	0.424	0.326	0.421	0.441	0.650	0.415	0.439	0.458	0.319	0.361	0.397	0.411	0.199	0.278	0.389	0.408
UDA-TSF	DAF	ICML22	0.403	0.338	0.416	0.428	0.596	0.409	0.426	0.414	0.314	0.358	0.398	0.414	0.195	0.277	0.394	0.403
UDA-TSF	RAINCOAT	ICML23	0.448	0.371	0.509	0.464	0.653	0.412	0.469	0.473	0.340	0.377	0.443	0.447	0.217	0.293	0.536	0.465
UDA-TSF	CauDiTS	ICML24	0.412	0.336	0.417	0.426	0.607	0.409	0.418	0.384	0.321	0.363	0.396	0.410	0.201	0.279	0.388	0.402
UDA-TSF	ACON	NeurIPS24	0.444	0.340	0.463	0.450	0.659	0.414	0.445	0.456	0.333	0.374	0.435	0.442	0.212	0.289	0.502	0.452
UDA-TSF	TransPL	ICML25	0.421	0.339	0.424	0.434	0.612	0.408	0.426	0.422	0.323	0.366	0.408	0.417	0.204	0.282	0.396	0.413
UDA-TSF	EDEN	NeurIPS25	0.401	0.326	0.420	0.433	0.605	0.404	0.419	0.385	0.327	0.364	0.416	0.428	0.204	0.281	0.462	0.436
SFDA	MAPU	KDD23	0.449	0.394	0.542	0.490	0.644	0.415	0.462	0.486	0.348	0.380	0.449	0.451	0.221	0.297	0.547	0.469
SFDA	SF(DA)^2	ICLR24	0.421	0.360	0.517	0.476	0.630	0.422	0.434	0.475	0.338	0.372	0.468	0.462	0.209	0.287	0.511	0.456
SFDA	TERSE	KDD25	0.445	0.377	0.529	0.482	0.650	0.407	0.451	0.475	0.350	0.381	0.457	0.457	0.211	0.296	0.527	0.450
SFDA	ProDe	ICLR25	0.432	0.356	0.498	0.468	0.631	0.419	0.456	0.463	0.342	0.373	0.394	0.409	0.213	0.294	0.386	0.401
SFDA	DUET	NeurIPS25	0.441	0.371	0.531	0.479	0.637	0.421	0.467	0.472	0.346	0.379	0.458	0.443	0.216	0.295	0.493	0.439
	Ours	–	**0.392**	**0.325**	**0.388**	**0.374**	**0.571**	**0.403**	**0.411**	**0.381**	**0.308**	**0.357**	**0.387**	**0.405**	**0.189**	**0.276**	**0.381**	**0.396**

Note: Lower is better. Darker and lighter blue backgrounds denote the best and second-best results, respectively.

4 Experiments

4.1 Experiment Settings

Datasets and Evaluation Metrics. To systematically investigate domain adaptation in time series forecasting, we conduct experiments on four real-world datasets: Air Quality [65], PPGDaLiA [66], Bay Area Traffic [67], and ETT [13]. These datasets cover weather, medicine, traffic, and energy domains. To evaluate the performance of our methods, we employ two widely used evaluation metrics in time series forecasting tasks: Mean Absolute Error (MAE) and Mean Squared Error (MSE), detailed description is in appendix C.1. **More details are in appendix C.3 and C.4.**

Domain Adaptation Settings. (1) The Air Quality dataset contains recordings from four Chinese cities: Beijing (B), Tianjin (T), Guangzhou (G), and Shenzhen (S). We consider four adaptation scenarios: (i) B→S, (ii) T→B, (iii) G→T, and (iv) S→G. We set the forecasting horizon $L = 24$ to predict air quality for the next 24 hours. (2) The PPG-DaLiA dataset contains physiological data from 15 subjects: S1→S2 (male→male), S3→S6 (male→female), S5→S10 (female→female), and S14→S4 (female→male). We set $L = 300$ to predict heart rate for the next 300 seconds. (3) The Bay Area Traffic dataset records hourly road occupancy from 862 sensors on San Francisco Bay Area freeways. We consider seasonal domain shifts: (i) Spring→Fall, (ii) Summer→Winter,

Table 3: Adaptation evaluation with four different time series forecasting backbones. The MSE results of source only f_S and our adaptation method are reported. To save space, we use compact scenario abbreviations: S1 is abbreviated as 1 for PPG-DaLiA; S, U, A, W denote Spring, Summer, Autumn, and Winter in Bay Area Traffic; and the ETT prefix is omitted for the ETT dataset.

Backbone	Method	Air Quality				PPG-DaLiA				Bay Area Traffic				ETT			
		B→S	T→B	G→T	S→G	1→2	3→6	5→10	14→4	S→A	U→W	A→U	W→S	H1→H2	H2→H1	M1→M2	M2→M1
iTransformer	f_S	0.875	0.893	0.762	0.632	0.484	0.218	0.217	0.527	0.455	0.670	0.661	0.479	0.361	0.578	0.224	0.616
	Ours	**0.686**	**0.872**	**0.645**	**0.523**	**0.361**	**0.145**	**0.135**	**0.481**	**0.392**	**0.388**	**0.571**	**0.411**	**0.308**	**0.387**	**0.189**	**0.381**
PatchTST	f_S	0.881	0.890	0.774	0.702	0.455	0.219	0.247	0.616	0.456	0.669	0.673	0.496	0.344	0.570	0.237	0.561
	Ours	**0.698**	**0.858**	**0.646**	**0.539**	**0.348**	**0.128**	**0.151**	**0.512**	**0.405**	**0.406**	**0.578**	**0.421**	**0.295**	**0.374**	**0.181**	**0.368**
DLinear	f_S	0.880	0.861	0.780	0.623	0.497	0.224	0.192	0.577	0.475	0.679	0.660	0.446	0.314	0.455	0.218	0.488
	Ours	**0.704**	**0.794**	**0.669**	**0.514**	**0.394**	**0.153**	**0.104**	**0.494**	**0.407**	**0.392**	**0.569**	**0.393**	**0.279**	**0.368**	**0.173**	**0.350**
PhaseFormer	f_S	0.893	0.904	0.756	0.711	0.469	0.232	0.236	0.623	0.486	0.684	0.646	0.509	0.352	0.548	0.244	0.579
	Ours	**0.677**	**0.875**	**0.642**	**0.525**	**0.361**	**0.152**	**0.133**	**0.478**	**0.378**	**0.382**	**0.570**	**0.405**	**0.311**	**0.385**	**0.186**	**0.369**

Table 4: Ablation study on four datasets using iTransformer as the backbone. MSE results are reported. We follow the same compact scenario abbreviations as in Table 3.

$\mathcal{L}_{\text{SSM-PL}}$	$\mathcal{L}_{\text{SEA}}$	Air Quality				PPG-DaLiA				Bay Area Traffic				ETT			
		B→S	T→B	G→T	S→G	1→2	3→6	5→10	14→4	S→A	U→W	A→U	W→S	H1→H2	H2→H1	M1→M2	M2→M1
✗	✗	0.875	0.893	0.762	0.632	0.484	0.218	0.217	0.527	0.455	0.670	0.661	0.479	0.361	0.578	0.224	0.616
✓	✗	0.745	0.883	0.672	0.556	0.407	0.174	0.184	0.502	0.433	0.434	0.645	0.451	0.320	0.446	0.205	0.447
✗	✓	0.717	0.882	0.650	0.563	0.389	0.165	0.149	0.508	0.427	0.397	0.606	0.444	0.323	0.418	0.200	0.411
✓	✓	**0.686**	**0.872**	**0.645**	**0.523**	**0.361**	**0.145**	**0.135**	**0.481**	**0.392**	**0.388**	**0.571**	**0.411**	**0.308**	**0.387**	**0.189**	**0.381**

Note: Lower is better. Darker and lighter blue backgrounds denote the best and second-best results, respectively.

(iii) Fall→Summer, and (iv) Winter→Spring. We set $L = 96$ to predict traffic conditions for the next 96 hours. (4) The ETT dataset contains power load features from two stations. We consider station-level adaptation: (i) ETTh1→ETTh2, (ii) ETTh2→ETTh1, (iii) ETTm1→ETTm2, and (iv) ETTm2→ETTm1. We set $L = 96$ to predict Transformer Temperature for the next 96 timesteps.

Baselines. As an underexplored topic, SFDA for TSF lacks established baselines. We therefore compare with representative methods from two most-related adaptation settings. (1) Unsupervised Domain Adaptation (UDA) for time series, we adopt DAF [43], ACON [38], DANN-RNN(DANN [31] with RNN backbones), AdaRNN [42], EDEN [39], RAINCOAT [37], TransPL [40], and CauDiTS [41]. For (2) SFDA for classification, we adopt MAPU [27], TERSE [28], SF(DA)^2 [29], ProDe [54], and DUET [55]. To adapt these baselines to our TSF setting, we replace classification heads with forecasting heads when needed. Augmentation-based methods, such as SF(DA)^2, are directly applied to time series inputs.

4.2 Main Results

Comprehensive experimental results on four datasets are presented in Tables 1 and 2, covering diverse adaptation scenarios including cross-city, cross-subject, cross-season, and cross-area. We compare our method with six UDA baselines and four SFDA baselines, and several observations can be made: 1) Across all datasets and adaptation settings, our method consistently achieves the best performance, with relative reductions in MSE and MAE ranging from 1% to 22% over the strongest baselines. Compared to the source model, substantially larger gains with over 40% relative MSE reductions are observed for the Summer→Winter transfer in the Bay Area Traffic dataset, demonstrating strong adaptation capability. These results validate the synergy between local guidance from the target-domain self-supervised model and global guidance induced by the SEA pattern. 2) Performance gains are more pronounced in challenging scenarios, such as Summer→Winter transfer and cross-subject adaptation from a male subject (V3) to a female subject (V6), demonstrating the robustness of our approach under severe domain shifts. 3) Existing SFDA methods are not designed for forecasting tasks and thus generally exhibit limited adaptation performance. In contrast, the UDA forecasting method DAF achieves the strongest performance among baselines in most scenarios, second only to our method. In addition, EDEN benefits from modeling inter-channel correlations and performs relatively well on multi-channel datasets such as Bay Area Traffic.

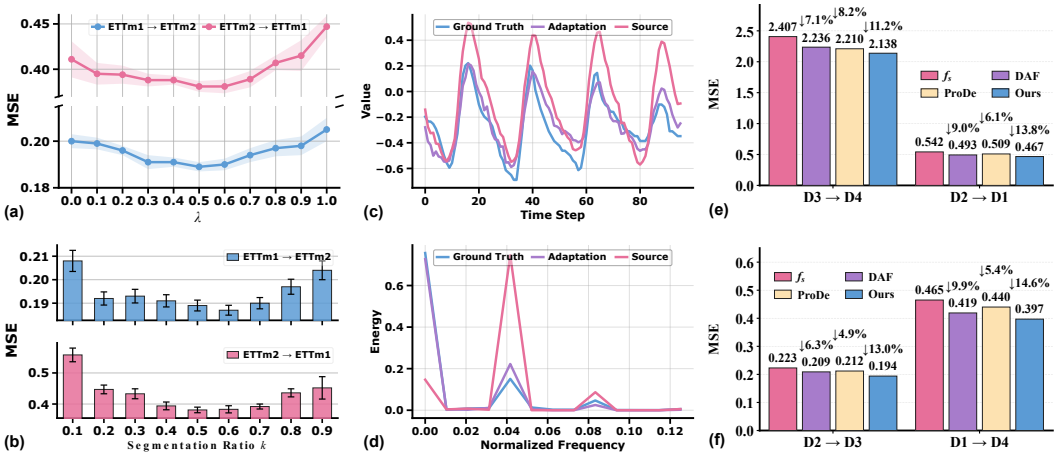


Figure 3: (a) Sensitivity to the score weight λ. (b) Sensitivity to the segmentation ratio k. (c) and (d) present temporal and spectral comparisons of adaptation results on ETTm2→ETTm1, respectively. (e) and (f) evaluate the cross-domain adaptation performance of f_S, ProDe, DAF, and our method on the non-stationary datasets ILI and Exchange, respectively. Percentages indicate the relative MSE reduction compared with f_S.

4.3 Evaluation Under Different Backbones

We conduct evaluations on the same four datasets across 16 adaptation scenarios and report the MSE results. We use three well-established TSF models: iTransformer [15], PatchTST [16], and DLinear [12], which represent temporal dynamics through time-token, patch-token, and channel-token modeling, respectively. We further include a latest patch-token based model, PhaseFormer [68], to examine the generality of our method on newer architectures. As shown in Table 3, our method consistently outperforms the source model f_S across all datasets, adaptation scenarios, and backbone architectures. Quantitatively, the relative improvements over f_S reach up to 42.1%, 41.0%, 45.8%, and 44.2% for iTransformer, PatchTST, DLinear, and PhaseFormer, respectively. These results demonstrate consistently substantial improvements regardless of the underlying architecture, validating the model-agnostic nature and broad applicability of our method.

4.4 Additional Experiments

Ablation Study. Table 4 analyzes the contribution of each loss. When both losses are removed, the source model f_S is zero-shot transferred to the target domain, leading to degraded performance under domain shift. Using either $\mathcal{L}_{\text{SSM-PL}}$ or $\mathcal{L}_{\text{SEA}}$ improves this baseline, reducing the average MSE by 12.4% and 15.3%, respectively: the former provides short-horizon pseudo-label supervision from target input, while the latter encourages long-horizon predictions to preserve the target-domain SEA pattern. Combining the two losses achieves the best results, yielding a 19.8% average MSE reduction over zero-shot f_S and a further 8.4% reduction over the $\mathcal{L}_{\text{SSM-PL}}$ only variant. This demonstrates their complementary roles, as $\mathcal{L}_{\text{SSM-PL}}$ provides phase information which is not captured by $\mathcal{L}_{\text{SEA}}$. Notably, $\mathcal{L}_{\text{SEA}}$ is more beneficial for longer horizons, with the additional mean MSE reduction over the $\mathcal{L}_{\text{SSM-PL}}$ only variant increasing from 4.78% at $H = 24$ to 14.70% at $H = 300$, suggesting that SEA patterns provide more effective fine-grained spectral guidance for long-horizon forecasting.

Sensitivity Analysis and Case Study. We study the effects of the score weight λ and the segmentation ratio k on the ETTm1→ETTm2 and ETTm2→ETTm1 settings with iTransformer as the backbone. We report results averaged over five runs with different random seeds. As shown in Figure 3(a), $\lambda = 0$ or $\lambda = 1$ reduces the objective to pure $\mathcal{L}_{\text{SEA}}$ or pure $\mathcal{L}_{\text{SSM-PL}}$, both leading to degraded performance. In contrast, $\lambda \in [0.3, 0.6]$ consistently yields the best results, confirming that the two losses provide collaborative supervision. Figure 3(b) shows that k controls the trade-off between pseudo-label accuracy and pseudo-label length: a small k makes the self-supervised task difficult, whereas a large k produces accurate but overly short pseudo-labels. The moderate choice $k = 0.5$ offers a good balance, and we therefore keep $k = 0.5$ fixed in all other experiments without tuning. Finally, Figure 3(c,d) presents a case study on ETTm2→ETTm1. The adapted prediction

better matches the ground truth in the time domain, which evidences SEA alignment effectiveness. Additional examples is in Appendix F.2.

Evaluation on Non-Stationary Datasets. The proposed SEA pattern models the spectral correspondence between short- and long-horizon signal pair, without assuming that the spectral characteristics of the underlying time series remain stable. To further evaluate its effectiveness under non-stationary setting, we conduct experiments on two widely used datasets ILI and Exchange in the literature [69, 60], comparing our method against three baselines: f_s zero-shot, DAF (the strongest UDA-TSF baseline), and ProDe (the strongest SFDA baseline), as identified in Tables 1 and 2. Exchange contains daily exchange rates from 1990 to 2016, while ILI records weekly influenza-like illness ratios from 2002 to 2021, with forecasting horizons of 96 and 36, respectively. We chronologically divide each dataset into four non-overlapping domains (D1 to D4), based on which we evaluate cross-domain adaptation between different periods. As shown in Fig. 3(e) and (f), our method consistently achieves the lowest MSE across all adaptation scenarios, reducing MSE by 11.2%–14.6% compared with f_S zero-shot and by 3.3%–7.2% compared with the strongest competing baseline. These results demonstrate the robustness of our method to real-world non-stationary.

Exploration of Frequency Shift on Synthetic Data. To investigate the limitation of SEA under spectral drift, we construct a synthetic benchmark with one stationary source domain (D_S) and two target domains (D_G and D_A) with different frequency-shift patterns. Each episode contains 192 time steps, with 96 history steps and 96 forecasting steps. We generate 2,000 episodes per domain with randomly sampled initial phases and Gaussian noise. The source domain D_S consists of sinusoidal signals with a fixed frequency $f_0 = 1/24$. The two target domains share the same initial frequency f_0 and frequency change magnitude $\Delta f = d/96$, where $d \in \{0.5, 2, 4\}$ controls the shift severity, but differ in how the frequency changes: (1) Gradual frequency drift (D_G), where the frequency gradually increases from f_0 to $f_0 + \Delta f$ throughout the sequence, and (2) Abrupt frequency shift (D_A), where the frequency suddenly jumps from f_0 to $f_0 + \Delta f$ at a randomly selected point within the forecasting window. Detailed generation formulas are provided in the Appendix D .

The results are presented in Table 5, our method consistently achieves the lowest MSE across all shift types and severity levels. For gradual frequency drift (D_G), where the frequency change is observable in the input window, our method reduces MSE by 55.1% over the f_S zero-shot and 18.0% over the strongest baseline on average. In contrast, for abrupt regime shifts (D_A) occurring entirely within the forecasting window, the improvements decrease to 5.9% and 1.8%, respectively. These results indicate that although SEA avoids negative transfer under abrupt frequency shifts that are unobservable in history, its performance gains are substantially reduced in such challenging cross-domain scenarios.

Table 5: MSE comparison under synthetic frequency shifts. Best results are bolded.

d	Domain	f_S	DAF	ProDe	Ours
0.5	$D_S \to D_G$	0.679	0.355	0.375	**0.311**
	$D_S \to D_A$	0.367	0.338	0.342	**0.327**
2	$D_S \to D_G$	0.788	0.401	0.450	**0.332**
	$D_S \to D_A$	0.445	0.435	0.433	**0.429**
4	$D_S \to D_G$	0.656	0.415	0.405	**0.306**
	$D_S \to D_A$	0.450	0.442	0.445	**0.436**

5 Conclusion

We present the first exploration of SFDA in TSF. Leveraging a unique characteristic of TSF, we train a self-supervised model in the target domain that provides reliable short-horizon pseudo labels. However, the adaptation objective remains long-horizon forecasting for the source model. To facilitate long-horizon source model adaptation via short-horizon supervision, we introduce the SEA pattern to model spectral correspondence between time series spanning different horizons. Different from direct spectrum alignment, SEA serves as a bridge between short-horizon pseudo-label supervision and long-horizon source model adaptation. Our results highlight SEA as a core mechanism for transferring reliable local guidance to global forecasting adaptation in the source-free setting.

Limitations and Extensions This work is currently limited to time series forecasting and has not yet examined whether the proposed SEA pattern generalizes to other sequence learning tasks. Moreover, as observed in our synthetic experiments, SEA provides limited improvements when frequency shifts occur only within the forecasting window. To address this limitation, a promising future direction is to explore utilization of SEA under test-time adaptation settings, where the model can continuously adapt to newly observed time steps.

Acknowledgments and Disclosure of Funding

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Discovery Grants program.

References

[1] Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, 2023.

[2] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 2021.

[3] Yisong Fu, Fei Wang, Zezhi Shao, Boyu Diao, Lin Wu, Zhulin An, Chengqing Yu, Yujie Li, and Yongjun Xu. On the integration of spatial-temporal knowledge: A lightweight approach to atmospheric time series forecasting. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[4] Xuming He, Zhiyuan You, Junchao Gong, Couhua Liu, Xiaoyu Yue, Peiqin Zhuang, Wenlong Zhang, and Lei Bai. Radarqa: Multi-modal quality analysis of weather radar forecasts. *arXiv preprint arXiv:2508.12291*, 2025.

[5] Weiyang Kong, Ziyu Guo, and Yubao Liu. Spatio-temporal pivotal graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2024.

[6] Yuchen Fang, Yuxuan Liang, Bo Hui, Zezhi Shao, Liwei Deng, Xu Liu, Xinke Jiang, and Kai Zheng. Efficient large-scale traffic forecasting with transformers: A spatial data management perspective. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2025.

[7] Wenzhen Yue, Yong Liu, Xianghua Ying, Bowei Xing, Ruohao Guo, and Ji Shi. Freeformer: Frequency enhanced transformer for multivariate time series forecasting. *arXiv preprint arXiv:2501.13989*, 2025.

[8] Jiaming Ma, Binwu Wang, Guanjun Wang, Kuo Yang, Zhengyang Zhou, Pengkun Wang, Xu Wang, and Yang Wang. Less but more: Linear adaptive graph learning empowering spatiotemporal forecasting. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[9] Hongbin Huang, Minghua Chen, and Xiao Qiao. Generative learning for financial time series with irregular and scale-invariant patterns. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[10] Hongjie Xia, Huijie Ao, Long Li, Yu Liu, Sen Liu, Guangnan Ye, and Hongfeng Chai. Cisthpan: Pre-trained attention network for stock selection with channel-independent spatio-temporal hypergraph. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2024.

[11] Yifan Hu, Yuante Li, Peiyuan Liu, Yuxia Zhu, Naiqi Li, Tao Dai, Shu-tao Xia, Dawei Cheng, and Changjun Jiang. Fintsb: A comprehensive and practical benchmark for financial time series forecasting. *arXiv preprint arXiv:2502.18834*, 2025.

[12] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2023.

[13] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2021.

[14] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

[15] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

[16] Y Nie. A time series is worth 64words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

[17] Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S Jensen, Zhenli Sheng, et al. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *arXiv preprint arXiv:2403.20150*, 2024.

[18] Yisong Fu, Zezhi Shao, Chengqing Yu, Yujie Li, Zhulin An, Qi Wang, Yongjun Xu, and Fei Wang. Selective learning for deep time series forecasting. *arXiv preprint arXiv:2510.25207*, 2025.

[19] Hao Wang, Licheng Pan, Zhichao Chen, Xu Chen, Qingyang Dai, Lei Wang, Haoxuan Li, and Zhouchen Lin. Time-o1: Time-series forecasting needs transformed label alignment. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[20] Xiaowen Ma, Zhenliang Ni, Shuai Xiao, and Xinghao Chen. Timepro: Efficient multivariate long-term time series forecasting with variable-and time-aware hyper-state. *arXiv preprint arXiv:2505.20774*, 2025.

[21] Tal Gonen, Itai Pemper, Ilan Naiman, Nimrod Berman, and Omri Azencot. Time series generation under data scarcity: A unified generative modeling approach. *arXiv preprint arXiv:2505.20446*, 2025.

[22] Zhen Fang, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. Is out-of-distribution detection learnable? In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[23] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[24] Sanqing Qu, Tianpei Zou, Lianghua He, Florian Röhrbein, Alois Knoll, Guang Chen, and Changjun Jiang. Lead: Learning decomposition for source-free universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[25] Ankit N Khambhati, Edward F Chang, Maxime O Baud, and Vikram R Rao. Hippocampal network activity forecasts epileptic seizures. *Nature Medicine*, 2024.

[26] Niklas Brake, Flavie Duc, Alexander Rokos, Francis Arseneau, Shiva Shahiri, Anmar Khadra, and Gilles Plourde. A neurophysiological basis for aperiodic eeg and the background spectral trend. *Nature communications*, 2024.

[27] Mohamed Ragab, Emadeldeen Eldele, Min Wu, Chuan-Sheng Foo, Xiaoli Li, and Zhenghua Chen. Source-free domain adaptation with temporal imputation for time series data. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD)*, 2023.

[28] Peiliang Gong, Yucheng Wang, Min Wu, Zhenghua Chen, Xiaoli Li, and Daoqiang Zhang. Temporal restoration and spatial rewiring for source-free multivariate time series domain adaptation. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2025.

[29] Uiwon Hwang, Jonghyun Lee, Juhyeon Shin, and Sungroh Yoon. Sf(da)^2: Source-free domain adaptation through the lens of data augmentation. In *Proceedings of the International conference on learning representations (ICLR)*, 2024.

[30] Gezheng Xu, Li Yi, Pengcheng Xu, Jiaqi Li, Ruizhi Pu, Changjian Shui, Charles Ling, A. Ian McLeod, and Boyu Wang. Unraveling the mysteries of label noise in source-free domain adaptation: Theory and practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2025.

[31] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 2016.

[32] Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. Variational recurrent adversarial deep domain adaptation. In *Proceedings of the International conference on learning representations (ICLR)*, 2017.

[33] Garrett Wilson, Janardhan Rao Doppa, and Diane J Cook. Multi-source deep domain adaptation with weak supervision for time-series sensor data. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery & data mining (KDD)*, 2020.

[34] Ruichu Cai, Jiawei Chen, Zijian Li, Wei Chen, Keli Zhang, Junjian Ye, Zhuozhang Li, Xiaoyan Yang, and Zhenjie Zhang. Time series domain adaptation via sparse associative structure alignment. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, volume 35, 2021.

[35] Qiao Liu and Hui Xue. Adversarial spectral kernel matching for unsupervised time series domain adaptation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[36] Yilmazcan Ozyurt, Stefan Feuerriegel, and Ce Zhang. Contrastive learning for unsupervised domain adaptation of time series. *arXiv preprint arXiv:2206.06243*, 2022.

[37] Huan He, Owen Queen, Teddy Koker, Consuelo Cuevas, Theodoros Tsiligkaridis, and Marinka Zitnik. Domain adaptation for time series under feature and label shifts. In *Proceedings of the International conference on machine learning (ICML)*, 2023.

[38] Mingyang Liu, Xinyang Chen, Yang Shu, Xiucheng Li, Weili Guan, and Liqiang Nie. Boosting transferability and discriminability for time series domain adaptation. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[39] Mingyang Liu, Xinyang Chen, Xiucheng Li, Weili Guan, and Liqiang Nie. Breakthrough sensor-limited single view: Towards implicit temporal dynamics for time series domain adaptation. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[40] Jaeho Kim and Seulki Lee. Transpl: Vq-code transition matrices for pseudo-labeling of time series unsupervised domain adaptation. *arXiv preprint arXiv:2505.09955*, 2025.

[41] Shiliang Sun et al. Caudits: Causal disentangled domain adaptation of multivariate time series. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[42] Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the ACM international conference on information & knowledge management (CIKM)*, 2021.

[43] Xiaoyong Jin, Youngsuk Park, Danielle Maddix, Hao Wang, and Yuyang Wang. Domain adaptation for time series forecasting via attention sharing. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

[44] Zijian Li, Ruichu Cai, Tom ZJ Fu, Zhifeng Hao, and Kun Zhang. Transferable time-series forecasting under causal conditional shift. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023.

[45] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision (CVPR)*, 2021.

[46] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision (CVPR)*, 2021.

[47] Shiqi Yang, Shangling Jui, Joost Van De Weijer, et al. Attracting and dispersing: A simple approach for source-free domain adaptation. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[48] Yixin Zhang, Zilei Wang, and Weinan He. Class relationship embedded learning for source-free unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[49] Mattia Litrico, Alessio Del Bue, and Pietro Morerio. Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[50] Yongxiang Li, Yanglin Feng, Yuan Sun, Dezhong Peng, Xi Peng, and Peng Hu. Learning source-free domain adaptation for visible-infrared person re-identification. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[51] Shiqi Yang, Joost Van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Proceedings of the Advances in neural information processing systems (NeurIPS)*, 2021.

[52] Nazmul Karim, Niluthpol Chowdhury Mithun, Abhinav Rajvanshi, Han-pang Chiu, Supun Samarasekera, and Nazanin Rahnavard. C-sfda: A curriculum learning aided self-training framework for efficient source free domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2023.

[53] Chao Su, Huiming Zheng, Dezhong Peng, and Xu Wang. Dica: Disambiguated contrastive alignment for cross-modal retrieval with partial labels. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[54] Song Tang, Wenxin Su, Yan Gan, Mao Ye, Jianwei Zhang, and Xiatian Zhu. Proxy denoising for source-free domain adaptation. *arXiv preprint arXiv:2406.01658*, 2024.

[55] Jae Yun Lee, Jae Hyeon Park, Gyoomin Lee, Bogyeong Kim, Min Hee Cha, Hyeok Nam, Joo Hyeon Jeon, Hyunse Lee, and Sung In Cho. Duet: Dual-perspective pseudo labeling and uncertainty-aware exploration & exploitation training for source-free domain adaptation. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[56] Yingyi Zhong, Wen'an Zhou, and Liwen Tao. Source-free time series domain adaptation with wavelet-based multi-scale temporal imputation. *Neural Networks*, 2025.

[57] Mohamed Ragab, Peiliang Gong, Emadeldeen Eldele, Wenyu Zhang, Min Wu, Chuan-Sheng Foo, Daoqiang Zhang, Xiaoli Li, and Zhenghua Chen. Evidentially calibrated source-free time-series domain adaptation with temporal imputation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2025.

[58] Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Proceedings of the Advances in neural information processing systems (NeurIPS)*, 2022.

[59] Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*, 2023.

[60] Wei Fan, Kun Yi, Hangting Ye, Zhiyuan Ning, Qi Zhang, and Ning An. Deep frequency derivative learning for non-stationary time series forecasting. *arXiv preprint arXiv:2407.00502*, 2024.

[61] Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency debiased transformer for time series forecasting. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD)*, 2024.

[62] Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[63] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 2015.

[64] Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. *Signals & systems*. Pearson Educación, 1997.

[65] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. Forecasting fine-grained air quality based on big data. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD)*, 2015.

[66] Attila Reiss, Ina Indlekofer, and Philip Schmidt. Ppg-dalia. UCI Machine Learning Repository, 2019. DOI: https://doi.org/10.24432/C53890.

[67] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *Proceedings of the International ACM SIGIR conference on research & development in information retrieval (SIGIR)*, 2018.

[68] Yiming Niu, Jinliang Deng, and Yongxin Tong. Phaseformer: From patches to phases for efficient and effective time series forecasting. *arXiv preprint arXiv:2510.04134*, 2025.

[69] Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[70] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[71] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[72] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Appendix

A Theoretical Justification of the SEA Proxy

Our method uses the input-induced SEA pattern $\mathbf{SEA}(\mathbf{x}_{1:L/2}, \mathbf{x})$ as a proxy for the unavailable label-induced SEA pattern $\mathbf{SEA}(\mathbf{y}_{1:L/2}, \mathbf{y})$. Here, we provide a theoretical analysis to show that the discrepancy between these two SEA patterns can be bounded under two explicit assumptions on the target-domain spectra.

Let

$$p_x^s(i) = \bar{E}_{\mathbf{x}_{1:L/2}}(f_i^{(s)}), \quad p_x^l(j) = \bar{E}_{\mathbf{x}}(f_j^{(l)}), \tag{13}$$

denote the normalized spectral energy at the i-th short-window frequency bin and the j-th long-window frequency bin, respectively. We define $p_y^s(i)$ and $p_y^l(j)$ analogously for $(\mathbf{y}_{1:L/2}, \mathbf{y})$.

Assumption 1 (Bounded spectral variation). The normalized spectral variation between the input and forecasting windows is bounded:

$$\|p_x^s - p_y^s\|_\infty \le \delta_s, \quad \|p_x^l - p_y^l\|_\infty \le \delta_l. \tag{14}$$

This assumption allows non-stationary spectral evolution, provided that the input–future spectral variations remain bounded.

Assumption 2 (Non-degenerate short-window spectrum). The normalized spectral energy at each short-window frequency bin is lower-bounded:

$$\min_i \{p_x^s(i), p_y^s(i)\} \ge m > 0. \tag{15}$$

This condition prevents the SEA coefficients from becoming unbounded. In practice, it is guaranteed by the ϵ-stabilization term, since the normalized spectral energies are nonnegative and thus $m = \epsilon > 0$ provides a valid lower bound.

Proposition (Bounded SEA proxy discrepancy). Under Assumptions 1 and 2, the discrepancy between the input-induced and label-induced SEA patterns satisfies

$$\left\|\mathbf{SEA}(\mathbf{x}_{1:L/2}, \mathbf{x}) - \mathbf{SEA}(\mathbf{y}_{1:L/2}, \mathbf{y})\right\|_{\max} \le \frac{\delta_l}{m} + \frac{\delta_s}{m^2}, \tag{16}$$

where $\|A\|_{\max} = \max_{i,j} |A_{ij}|$.

Proof. For brevity, denote

$$\mathbf{S}_x = \mathbf{SEA}(\mathbf{x}_{1:L/2}, \mathbf{x}), \quad \mathbf{S}_y = \mathbf{SEA}(\mathbf{y}_{1:L/2}, \mathbf{y}). \tag{17}$$

According to the SEA definition in Eq. (9),

$$[\mathbf{S}_x]_{(i,j)} = w(i,j)\frac{p_x^l(j)}{p_x^s(i)}, \quad [\mathbf{S}_y]_{(i,j)} = w(i,j)\frac{p_y^l(j)}{p_y^s(i)}, \tag{18}$$

where $0 < w(i,j) \le 1$. For each frequency pair (i,j), we have

$$\begin{aligned} |[\mathbf{S}_x]_{(i,j)} - [\mathbf{S}_y]_{(i,j)}| &= w(i,j)\left|\frac{p_x^l(j)}{p_x^s(i)} - \frac{p_y^l(j)}{p_y^s(i)}\right| \\ &\le w(i,j)\left(\frac{|p_x^l(j) - p_y^l(j)|}{p_x^s(i)} + \frac{p_y^l(j)|p_x^s(i) - p_y^s(i)|}{p_x^s(i)p_y^s(i)}\right) \\ &\le \frac{\delta_l}{m} + \frac{\delta_s}{m^2}, \end{aligned} \tag{19}$$

where the last inequality follows from Assumptions 1 and 2, together with $0 < w(i,j) \le 1$ and $0 \le p_y^l(j) \le 1$. Therefore,

$$\|\mathbf{S}_x - \mathbf{S}_y\|_{\max} = \max_{i,j} |[\mathbf{S}_x]_{(i,j)} - [\mathbf{S}_y]_{(i,j)}| \le \frac{\delta_l}{m} + \frac{\delta_s}{m^2}. \tag{20}$$

□

Discussion. The above result establishes an explicit upper bound on the SEA proxy discrepancy in terms of the input–future spectral variations δ_s and δ_l and the minimum short-window spectral energy m. The bound also characterizes a limitation of the proxy. An abrupt frequency shift occurring only in the future window can increase δ_s and δ_l, resulting in a looser discrepancy bound. This is consistent with our synthetic frequency-shift experiment shown in Fig. 3(e,f) and Table 5, in which such shifts severely limit the improvement obtained from SEA alignment.

B Additional Method Details

This appendix provides complementary details of the proposed source-free adaptation procedure. We first summarize the two-stage adaptation pipeline, and then describe how the same SEA-based objective can be applied when the input length and forecasting horizon are different.

B.1 Detailed Adaptation Process

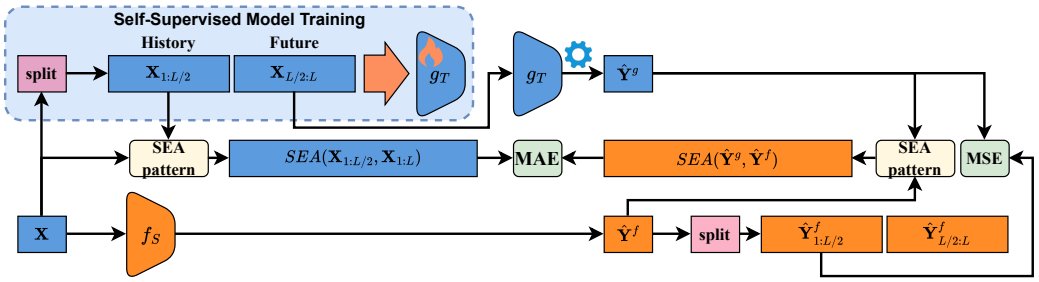


Figure 4: Overview of the proposed SEA-guided source-free adaptation framework.

Fig. 4 summarizes the overall adaptation pipeline. Given an unlabeled target dataset $\mathcal{D}_T$ and a source forecasting model f_S pre-trained on the source domain, the proposed method proceeds in two stages.

Stage I: Target-domain self-supervised training. For each target sequence $\mathbf{x} \in \mathcal{D}_T$, we apply temporal segmentation to obtain $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$. The auxiliary target model g_T is trained to predict $\mathbf{x}^{(2)}$ from $\mathbf{x}^{(1)}$ by minimizing the self-supervised forecasting loss. This stage only uses target input and does not require target labels.

Stage II: Source model adaptation with local supervision and SEA alignment. After Stage I, g_T is fixed and used to generate reliable short-horizon pseudo labels. For a target example $\mathbf{x}$, g_T produces a pseudo-label prediction $\hat{\mathbf{y}}^g$, while the source model f_S produces a long-horizon prediction $\hat{\mathbf{y}}^f$. We then update f_S with two complementary objectives. The SSM-PL loss locally matches the early part of $\hat{\mathbf{y}}^f$ to $\hat{\mathbf{y}}^g$, while the SEA loss aligns $\mathrm{SEA}(\hat{\mathbf{y}}^g, \hat{\mathbf{y}}^f)$ with the input-induced pattern $\mathrm{SEA}(\mathbf{x}^{(1)}, \mathbf{x})$. The overall objective is

$$\mathcal{L} = \lambda \mathcal{L}_{\text{SSM-PL}} + (1-\lambda)\mathcal{L}_{\text{SEA}}, \tag{21}$$

where λ controls the contribution of the local pseudo-label loss and the global SEA alignment loss.

B.2 Extension to General TSF Settings

In the main text, we use the simplified setting $L = H$ for clarity, where L is the historical input length and H is the forecasting horizon. Here, we describe the general case with arbitrary L and H. For notation simplicity, we use the experimental setting $k = 0.5$ and denote $L_g = \lfloor L/2 \rfloor$, so that $\mathbf{x}^{(1)} = \mathbf{x}_{1:L_g}$ and g_T produces a pseudo label $\hat{\mathbf{y}}^g$ with length L_g. The case with other segmentation ratios follows by replacing L_g with $\lfloor kL \rfloor$.

The local pseudo-label supervision is always applied on the overlapping horizon:

$$\mathcal{L}_{\text{SSM-PL}} = \mathrm{MSE}\left(\hat{\mathbf{y}}^f_{1:m}, \hat{\mathbf{y}}^g_{1:m}\right), \qquad m = \min(H, L_g). \tag{22}$$

The SEA alignment is adjusted according to the relative lengths of L, H, and L_g.

Case 1: $L < H$**.** The source model prediction is longer than both the pseudo label and the accessible target input. The prediction-induced SEA pattern and the input-induced SEA pattern have different long-horizon frequency resolutions:

$$\begin{aligned} &\mathrm{SEA}(\hat{\mathbf{y}}^g, \hat{\mathbf{y}}^f) \in \mathbb{R}^{(\lfloor L_g/2 \rfloor+1)\times(\lfloor H/2 \rfloor+1)}, \\ &\mathrm{SEA}(\mathbf{x}^{(1)}, \mathbf{x}) \in \mathbb{R}^{(\lfloor L_g/2 \rfloor+1)\times(\lfloor L/2 \rfloor+1)}. \end{aligned} \tag{23}$$

To compare these two patterns, we introduce a spectral downsampling operator along the long-horizon frequency axis:

$$\mathcal{D}_{H\to L} : \mathbb{R}^{(\lfloor L_g/2 \rfloor+1)\times(\lfloor H/2 \rfloor+1)} \to \mathbb{R}^{(\lfloor L_g/2 \rfloor+1)\times(\lfloor L/2 \rfloor+1)}. \tag{24}$$

The SEA loss is then defined as

$$\mathcal{L}_{\mathrm{SEA}} = \mathrm{MAE}\Big(\mathcal{D}_{H\to L}\big[\mathrm{SEA}(\hat{\mathbf{y}}^g, \hat{\mathbf{y}}^f)\big], \mathrm{SEA}(\mathbf{x}^{(1)}, \mathbf{x})\Big). \tag{25}$$

Case 2: $L \geq H$**.** In this setting, the accessible target input is not shorter than the forecasting horizon. Two subcases arise depending on whether the pseudo-label horizon L_g is shorter or longer than H.

(i) $L_g \leq H$. The pseudo label is the shorter sequence and the source prediction is the longer sequence. We use the first H time steps of the target input as the long sequence for constructing the input-induced SEA pattern:

$$\mathcal{L}_{\mathrm{SEA}} = \mathrm{MAE}\Big(\mathrm{SEA}(\hat{\mathbf{y}}^g, \hat{\mathbf{y}}^f), \mathrm{SEA}(\mathbf{x}^{(1)}, \mathbf{x}_{1:H})\Big). \tag{26}$$

(ii) $L_g > H$. The source prediction is shorter than the pseudo label. Thus, the pseudo label-induced SEA pattern is constructed by treating $\hat{\mathbf{y}}^f$ as the short sequence and $\hat{\mathbf{y}}^g$ as the long sequence. The input-induced counterpart is constructed from the first H steps of $\mathbf{x}^{(1)}$ and the full $\mathbf{x}^{(1)}$:

$$\mathcal{L}_{\mathrm{SEA}} = \mathrm{MAE}\Big(\mathrm{SEA}(\hat{\mathbf{y}}^f, \hat{\mathbf{y}}^g), \mathrm{SEA}(\mathbf{x}^{(1)}_{1:H}, \mathbf{x}^{(1)})\Big). \tag{27}$$

Across all cases, the core mechanism remains unchanged. We use reliable short-horizon pseudo labels to provide local supervision and use input-induced SEA patterns to provide global spectral structural guidance for adapting f_S without source data or target labels.

C Reproduction Details

This appendix reports the datasets, cross-domain adaptation scenarios, backbone models, baseline methods, implementation details, evaluation metrics, and hyperparameter settings.

C.1 Dataset Descriptions

Our empirical evaluation covers four real-world multivariate time series forecasting benchmarks. Each benchmark defines domains in a different way, including cities, subjects, seasons, and transformer stations. Table 6 summarizes the dataset characteristics and cross-domain adaptation settings.

Air Quality Dataset. The air quality dataset [65] is collected from the Urban Air project, covering the period from May 1, 2014 to April 30, 2015. It contains multi-source measurements, including air quality data, meteorological data, and weather forecast data. The dataset includes four major Chinese cities: Beijing (B), Tianjin (T), Guangzhou (G), and Shenzhen (S). We use all air quality stations and treat each city as a domain. Following the original setting, we employ air quality and meteorological variables for forecasting, and apply Z-score normalization [70] independently for each city domain.

PPG-DaLiA Dataset. The PPG-DaLiA dataset [66] is a publicly available multimodal physiological dataset for PPG-based heart rate estimation. It contains physiological and motion data recorded from 15 volunteers using both a chest-worn device and a wrist-worn device. The chest-worn device provides electrocardiogram, respiration, and tri-axial acceleration signals, while the wrist-worn device provides blood volume pulse, electrodermal activity, body temperature, and tri-axial acceleration signals. We treat each subject as an individual domain and construct cross-subject domain adaptation scenarios. Z-score normalization is applied independently for each subject domain.

Table 6: Summary of datasets and cross-domain adaptation settings.

Dataset	D	Forecast length	Frequency	Cross-domain scenario
Air Quality	12	24	Hourly	We treat each city as a separate domain and evaluate B→S, T→B, G→T, and S→G.
PPG-DaLiA	8	300	Varying	We treat each subject as a separate domain and evaluate S1→S2, S3→S6, S5→S10, and S14→S4.
Traffic	862	96	Hourly	We treat each season as a separate domain and evaluate Spring→Autumn, Summer→Winter, Autumn→Summer, and Winter→Spring.
ETTh1 / ETTh2	7	96	Hourly	We treat each station as a separate domain and evaluate ETTh1→ETTh2 and ETTh2→ETTh1.
ETTm1 / ETTm2	7	96	15 min	We treat each station as a separate domain and evaluate ETTm1→ETTm2 and ETTm2→ETTm1.
ILI	7	36	Weekly	We divide the data into four chronological domains and evaluate D3→D4 and D2→D1.
Exchange	8	96	Daily	We divide the data into four chronological domains and evaluate D2→D3 and D1→D4.

Note: D denotes the number of variates. Forecast length denotes the number of predicted time steps. Frequency denotes the sampling interval. For PPG-DaLiA, sampling intervals vary across variables. For Air Quality, B, T, G, and S denote Beijing, Tianjin, Guangzhou, and Shenzhen, respectively. For ILI and Exchange, D1–D4 denote four chronologically ordered, non-overlapping periods.

Traffic Dataset. The Traffic dataset [67] records hourly road occupancy rates from 862 sensors deployed on San Francisco Bay Area freeways. Following standard long-term forecasting protocols, we formulate it as a high-dimensional multivariate forecasting task. Since the dataset is not originally designed for domain adaptation, we construct seasonal domains according to the timestamp of each sample: Spring, Summer, Autumn, and Winter. This setting evaluates cross-domain adaptation under realistic seasonal distribution shifts.

ETT Dataset Series. The Electricity Transformer Temperature (ETT) dataset [13] contains two-year monitoring data collected from two transformer stations in China. It consists of seven channels, including power load features and oil temperature. The four subsets are ETTh1, ETTh2, ETTm1, and ETTm2, where "h" denotes hourly sampling, "m" denotes 15-minute sampling, and "1" and "2" denote different transformer stations. We adopt these subsets as station-level domains and evaluate cross-station adaptation.

Exchange Dataset. The Exchange dataset [67] contains daily exchange rates of eight foreign currencies from 1990 to 2016. It consists of eight channels, each representing the exchange rate of a different currency. We chronologically divide the dataset into four non-overlapping periods (D1–D4), treating each period as a separate domain. We evaluate two cross-period adaptation scenarios: D2→D3 and D1→D4.

ILI Dataset. The Influenza-Like Illness (ILI) dataset [71] contains weekly records of influenza-like illness in the United States from 2002 to 2021, collected by the Centers for Disease Control and Prevention (CDC). It consists of seven channels describing influenza activity and related statistics. We chronologically divide the dataset into four non-overlapping periods (D1–D4), treating each period as a separate domain. We evaluate two adaptation scenarios: D3→D4 and D2→D1.

C.2 Backbones and Baselines

We use iTransformer as the primary forecasting backbone and additionally evaluate representative architectures, including PatchTST, DLinear, and PhaseFormer, to verify that the proposed adaptation strategy is not tied to a specific forecasting model. Table 7 summarizes the backbone models and their official code links.

For baseline comparisons, we include 13 representative domain adaptation methods from two most-related adaptation settings: UDA methods for time series data and SFDA methods originally devel-

Table 7: Backbone models used in our experiments.

Backbone	Type	Code
iTransformer [15]	Transformer based	https://github.com/thuml/iTransformer
PatchTST [16]	Patch based	https://github.com/yuqinie98/PatchTST
DLinear [12]	Linear based	https://github.com/cure-lab/LTSF-Linear
PhaseFormer [68]	Phase domain	https://github.com/neumyor/PhaseFormer_TSL

oped for classification or time series classification. Specifically, for UDA methods, we adopt DAF, ACON, DANN-RNN, AdaRNN, EDEN, RAINCOAT, TransPL, and CauDiTS. For SFDA methods, we adopt MAPU, TERSE, SF(DA)^2, ProDe, and DUET. Table 8 summarizes the baseline methods, references, and public code availability.

Table 8: Baseline methods used in our experiments.

Baseline	Setting	Code
DAF [43]	UDA for TSF	https://github.com/leejoonhun/daf
AdaRNN [42]	UDA for TSF	https://github.com/jindongwang/transferlearning/tree/master/code/deep/adarnn
DANN-RNN [31]	UDA for time series	https://github.com/GRAAL-Research/domain_adversarial_neural_network; RNN forecasting variant implemented by us
RAINCOAT [37]	UDA for time series	https://github.com/mims-harvard/Raincoat
ACON [38]	UDA for time series	https://github.com/iLearn-Lab/NeurIPS24-ACON
EDEN [39]	UDA for time series	https://github.com/iLearn-Lab/NeurIPS25-EDEN
TransPL [40]	UDA for time series	https://github.com/eai-lab/TransPL
CauDiTS [41]	UDA for time series	Implemented according to the original paper
MAPU [27]	SFDA for TS classification	https://github.com/mohamedr002/MAPU_SFDA_TS
TERSE [30]	SFDA for TS classification	https://github.com/Tokenmw/TERSE-master
SF(DA)^2 [29]	SFDA for classification	https://github.com/shinyflight/SFDA2
ProDe [54]	SFDA for classification	https://github.com/tntek/source-free-domain-adaptation
DUET [55]	SFDA for classification	https://github.com/l3umblee/duet-sfda

Note: For methods not originally designed for forecasting, we replace the classification head with a forecasting head and optimize them using the same regression objective and forecasting horizon.

C.3 Experiment Setup

The experiments are implemented on the PyTorch platform using an Intel® Xeon® Silver 4210R CPU @ 2.40 GHz and an NVIDIA GeForce RTX A5000 GPU with 24 GB memory. We primarily adopt iTransformer as the backbone architecture in our experiments, and additionally conduct evaluations on other representative time series forecasting models, including DLinear, PatchTST, and PhaseFormer. All models are trained using the Adam optimizer [72] with the same batch size 32. The initial learning rate is selected from $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}\}$ based on validation performance, and is decayed by a factor of 0.5 every 2 epochs. To prevent overfitting, the dropout rate is set to 0.2. For fairness, the same hyperparameter settings are adopted for all cross-domain time series forecasting methods in the experiments. All models are trained and evaluated 5 times using different random seeds, and the average performance is reported in terms of mean squared error (MSE) and mean absolute error (MAE), defined as

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N}\left\|\hat{\mathbf{y}}_n - \mathbf{y}_n\right\|_2^2, \qquad \mathrm{MAE} = \frac{1}{N}\sum_{n=1}^{N}\left\|\hat{\mathbf{y}}_n - \mathbf{y}_n\right\|_1, \tag{28}$$

where N denotes the number of test samples, and $\hat{\mathbf{y}}_n$ and $\mathbf{y}_n$ are the predicted and ground-truth forecasting sequences of the n-th sample, respectively.

C.4 Hyperparameter Selection

We use a unified hyperparameter configuration across datasets and adaptation scenarios unless otherwise specified. The key method-specific hyperparameters are the segmentation ratio k, the balancing weight λ, and the SEA Gaussian kernel bandwidth σ. Table 9 summarizes the main hyperparameter settings and the corresponding sensitivity analysis results.

Table 9: Summary of the main hyperparameter settings and sensitivity analysis results.

Parameter	Description	Setting and sensitivity result
Optimizer	Optimization algorithm	Adam optimizer [72] is used for all models.
B	Batch size	Fixed to 32 in all experiments.
lr	Initial learning rate	Search range: $\{1\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}, 5\times10^{-3}\}$; selected by validation performance.
η	Learning rate decay coefficient	$\eta = 0.5$, decayed every 2 epochs.
p_{drop}	Dropout rate	Fixed to 0.2 to prevent overfitting.
Seeds	Number of independent runs	Results are averaged over 5 random seeds.
k	Segmentation ratio for the self-supervised target-domain forecasting task	Search range: $[0, 1]$. Across 2 scenarios, $k \in [0.4, 0.7]$ performs consistently well, with the best result at $k = 0.5$. We fix $k = 0.5$ in all 16 scenarios.
λ	Balancing weight between $\mathcal{L}_{\text{SSM-PL}}$ and $\mathcal{L}_{\text{SEA}}$	Search range: $[0, 1]$. Across 2 scenarios, $\lambda \in [0.3, 0.6]$ performs consistently well; $\lambda = 0.5$ and 0.6 yield nearly identical best average performance. We fix $\lambda = 0.5$ in all 16 scenarios.
σ	Bandwidth of the Gaussian kernel weight in SEA pattern estimation	Search range: $[0.01/L, 100/L]$. Across 4 scenarios, $\sigma \in [0.8/L, 1.5/L]$ performs consistently well, with the best result at $\sigma = 1/L$. We fix $\sigma = 1/L$ in all 16 scenarios.

D Synthetic Benchmark Generation Process for Frequency Shift

We provide the detailed formula of the synthetic benchmark used in Sec. 4.4. Each episode consists of $T = 192$ time steps, where the first 96 steps serve as the input $\mathbf{x} = z_{0:95}$ and the remaining 96 steps as the forecasting target $\mathbf{y} = z_{96:191}$. We generate 2,000 episodes per domain, independently sampling the initial phase $\phi \sim \mathcal{U}(0, 2\pi)$ and Gaussian noise $\epsilon_t \sim \mathcal{N}(0, 0.05^2)$ at each time step.

Stationary Source Domain (D_S). The source domain consists of sinusoidal signals with a fixed frequency $f_0 = 1/24$, generated as

$$z_t^S = \sin(\phi + 2\pi f_0 t) + \epsilon_t, \quad t = 0, \ldots, 191. \tag{29}$$

Non-Stationary Target Domains (D_G and D_A). We construct two target domains with different frequency-shift patterns, sharing the same initial frequency f_0 and frequency change magnitude $\Delta f = d/96$, where $d \in \{0.5, 2, 4\}$ controls the shift severity.

For the gradual frequency drift domain (D_G), the instantaneous frequency increases linearly from f_0 to $f_0 + \Delta f$ throughout the sequence:

$$f_t^G = f_0 + \Delta f \frac{t}{191}. \tag{30}$$

For the abrupt frequency shift domain (D_A), the frequency remains fixed at f_0 before suddenly changing to $f_0 + \Delta f$ at a randomly selected time step s within the forecasting window where $s \sim \text{DiscreteUniform}\{96, \ldots, 191\}$:

$$f_t^A = \begin{cases} f_0, & t < s, \\ f_0 + \Delta f, & t \geq s, \end{cases} \tag{31}$$

Given the time-varying frequencies, the target signals are generated by accumulating the instantaneous frequency into the signal phase:

$$z_t^j = \sin\left(\phi + 2\pi \sum_{u=0}^{t-1} f_u^j\right) + \epsilon_t, \quad j \in \{G, A\}. \tag{32}$$

For D_G, the frequency drift is observable within the input window, whereas for D_A, the frequency shift occurs only within the forecasting window and cannot be directly observed from the history.

Cross-Domain Evaluation. We evaluate adaptation from the stationary source domain to each non-stationary target domain, i.e., $D_S \to D_G$ and $D_S \to D_A$, under three severity levels $d \in \{0.5, 2, 4\}$. All methods are evaluated using MSE over the 96-step forecasting window.

E More Experiment Results

E.1 Cross-domain under PhaseFormer backbone

Table 10: Cross-domain evaluation on Air Quality/PPG-DaLiA under PhaseFormer backbone.

	Methods	Venue	Air Quality								PPG-DaLiA							
			B→S		T→B		G→T		S→G		S1→S2		S3→S6		S5→S10		S14→S4	
			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
	Source Only	–	0.893	0.815	0.904	0.686	0.756	0.627	0.711	0.596	0.469	0.476	0.232	0.334	0.236	0.339	0.623	0.614
UDA-TSF	DANN-RNN	JMLR16	0.721	0.583	0.895	0.621	0.728	0.569	0.585	0.512	0.446	0.458	0.206	0.313	0.178	0.296	0.535	0.571
UDA-TSF	AdaRNN	CIKM21	0.715	0.577	0.886	0.617	0.699	0.556	0.557	0.503	0.394	0.431	0.171	0.307	0.163	0.276	0.507	0.523
UDA-TSF	DAF	ICML22	0.701	0.560	0.889	0.618	0.660	0.542	0.533	0.472	0.367	0.417	0.176	0.291	0.154	0.276	0.502	0.519
UDA-TSF	RAINCOAT	ICML23	0.777	0.585	0.901	0.655	0.741	0.574	0.614	0.542	0.458	0.465	0.218	0.323	0.188	0.312	0.521	0.569
UDA-TSF	CauDiTS	ICML24	0.708	0.558	0.884	0.615	0.653	0.554	0.537	0.479	0.382	0.425	0.184	0.301	0.150	0.271	0.496	0.516
UDA-TSF	ACON	NeurIPS24	0.782	0.679	0.897	0.625	0.738	0.582	0.602	0.515	0.469	0.471	0.213	0.320	0.187	0.306	0.527	0.573
UDA-TSF	TransPL	ICML25	0.713	0.581	0.887	0.620	0.681	0.540	0.562	0.485	0.388	0.419	0.190	0.288	0.167	0.286	0.511	0.555
UDA-TSF	EDEN	NeurIPS25	0.707	0.573	0.883	0.613	0.721	0.566	0.551	0.494	0.462	0.416	0.210	0.318	0.181	0.302	0.494	0.565
SFDA	MAPU	KDD23	0.828	0.800	0.901	0.626	0.746	0.603	0.612	0.523	0.461	0.472	0.216	0.321	0.203	0.331	0.518	0.570
SFDA	SF(DA)^2	ICLR24	0.746	0.637	0.889	0.618	0.687	0.552	0.578	0.509	0.447	0.459	0.209	0.319	0.194	0.318	0.506	0.563
SFDA	TERSE	KDD25	0.793	0.688	0.894	0.614	0.711	0.563	0.586	0.512	0.455	0.465	0.203	0.316	0.204	0.334	0.508	0.564
SFDA	ProDe	ICLR25	0.692	0.623	0.891	0.620	0.662	0.548	0.570	0.505	0.372	0.418	0.198	0.305	0.184	0.304	0.512	0.512
SFDA	DUET	NeurIPS25	0.786	0.684	0.898	0.622	0.727	0.568	0.594	0.518	0.464	0.456	0.204	0.314	0.197	0.312	0.516	0.562
	Ours	–	**0.677**	**0.554**	**0.875**	**0.606**	**0.642**	**0.529**	**0.525**	**0.467**	**0.361**	**0.395**	**0.152**	**0.239**	**0.133**	**0.257**	**0.478**	**0.491**

Note: Lower is better. Darker and lighter blue backgrounds denote the best and second-best results, respectively.

Table 11: Cross-domain on Bay Area Traffic/ETT datasets under PhaseFormer backbone. For ETT, H1/H2 denote ETTh1/ETTh2, and M1/M2 denote ETTm1/ETTm2.

	Methods	Venue	Bay Area Traffic								ETT							
			Spr.→Aut.		Sum.→Win.		Aut.→Sum.		Win.→Spr.		H1→H2		H2→H1		M1→M2		M2→M1	
			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
	Source Only	–	0.486	0.440	0.684	0.529	0.646	0.529	0.509	0.477	0.352	0.391	0.548	0.498	0.244	0.325	0.579	0.483
UDA-TSF	DANN-RNN	JMLR16	0.439	0.338	0.443	0.437	0.649	0.410	0.445	0.458	0.324	0.365	0.405	0.421	0.202	0.279	0.410	0.411
UDA-TSF	AdaRNN	CIKM21	0.442	0.331	0.429	0.443	0.645	0.417	0.424	0.454	0.320	0.363	0.399	0.413	0.199	0.278	0.389	0.407
UDA-TSF	DAF	ICML22	0.398	0.337	0.421	0.428	0.602	0.404	0.414	0.383	0.319	0.362	0.398	0.414	0.196	0.276	0.386	0.399
UDA-TSF	RAINCOAT	ICML23	0.449	0.372	0.509	0.465	0.651	0.413	0.468	0.472	0.341	0.378	0.443	0.446	0.217	0.293	0.536	0.464
UDA-TSF	CauDiTS	ICML24	0.413	0.336	0.415	0.421	0.604	0.408	0.416	0.386	0.321	0.361	0.392	0.410	0.193	0.275	0.384	0.397
UDA-TSF	ACON	NeurIPS24	0.445	0.343	0.466	0.451	0.656	0.415	0.446	0.456	0.334	0.375	0.435	0.441	0.212	0.289	0.501	0.448
UDA-TSF	TransPL	ICML25	0.421	0.339	0.424	0.420	0.590	0.408	0.418	0.421	0.323	0.366	0.408	0.417	0.204	0.282	0.396	0.412
UDA-TSF	EDEN	NeurIPS25	0.406	0.335	0.420	0.432	0.598	0.406	0.419	0.388	0.327	0.364	0.420	0.428	0.205	0.281	0.462	0.436
SFDA	MAPU	KDD23	0.449	0.395	0.543	0.491	0.642	0.416	0.461	0.486	0.348	0.380	0.449	0.451	0.221	0.297	0.547	0.468
SFDA	SF(DA)^2	ICLR24	0.421	0.361	0.518	0.477	0.629	0.423	0.433	0.475	0.338	0.372	0.469	0.463	0.209	0.287	0.509	0.455
SFDA	TERSE	KDD25	0.445	0.378	0.529	0.483	0.648	0.410	0.451	0.474	0.331	0.380	0.459	0.459	0.211	0.296	0.525	0.449
SFDA	ProDe	ICLR25	0.431	0.356	0.498	0.468	0.631	0.419	0.456	0.463	0.318	0.368	0.394	0.412	0.213	0.294	0.390	0.401
SFDA	DUET	NeurIPS25	0.441	0.371	0.532	0.480	0.636	0.421	0.467	0.472	0.346	0.379	0.458	0.443	0.216	0.295	0.493	0.439
	Ours	–	**0.378**	**0.322**	**0.382**	**0.370**	**0.570**	**0.399**	**0.405**	**0.379**	**0.311**	**0.359**	**0.385**	**0.405**	**0.186**	**0.272**	**0.369**	**0.392**

Note: Lower is better. Darker and lighter blue backgrounds denote the best and second-best results, respectively.

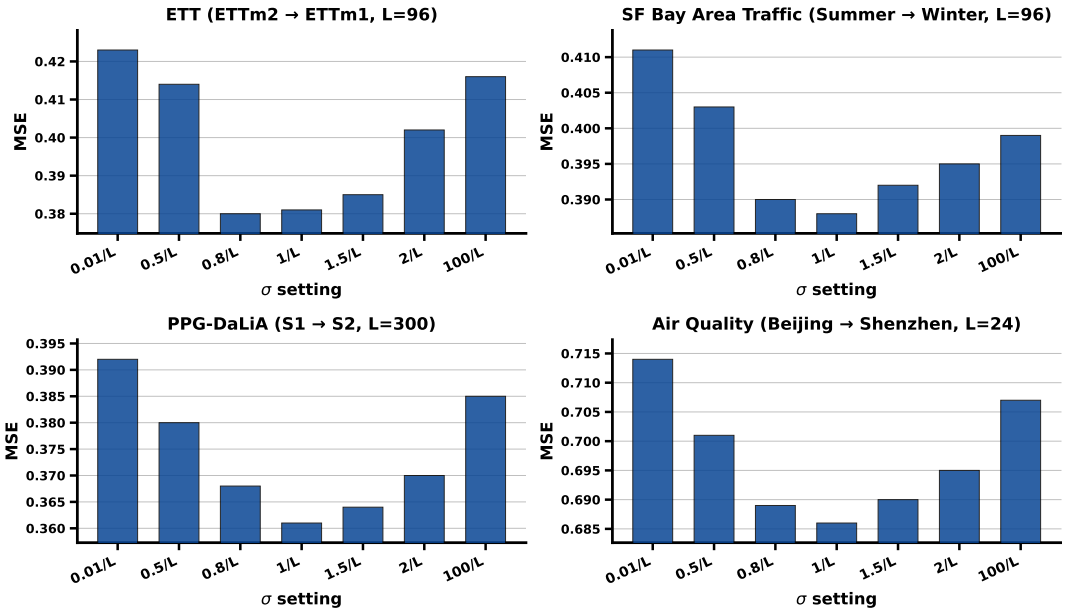


Figure 5: Sensitivity analysis of the Gaussian kernel bandwidth σ in SEA pattern computation where L represents the forecasting horizon. Results are reported on four cross-domain scenarios from four datasets.

E.2 Effect of the Gaussian Kernel Bandwidth σ

Given a short sequence of length $L/2$ and a long sequence of length L, SEA characterizes how spectral energy is allocated from the coarse frequency grid to the finer one. As defined in Eq. 9, we introduce a Gaussian kernel weight to emphasize neighboring frequency bins while suppressing distant ones:

$$w(i,j) = \exp\left(-\frac{\left(f_i^{(s)} - f_j^{(l)}\right)^2}{2\sigma^2}\right), \tag{33}$$

where σ controls the locality and smoothness of cross-scale energy allocation. Fig. 5 reports the sensitivity analysis of σ. We summarize the main observations as follows.

Too small σ. When σ is too small, the kernel becomes overly sharp and approaches a discrete bin-wise correspondence. Such a rigid mapping ignores the natural frequency refinement from length $L/2$ to length L, where energy from one coarse frequency bin should be distributed to multiple neighboring fine bins.

Too large σ. When σ is too large, the kernel becomes overly flat and assigns similar weights to distant frequency bins. This weakens the locality of energy allocation and allows spectral energy to diffuse to non-neighboring frequencies, leading to less meaningful cross-scale correspondence.

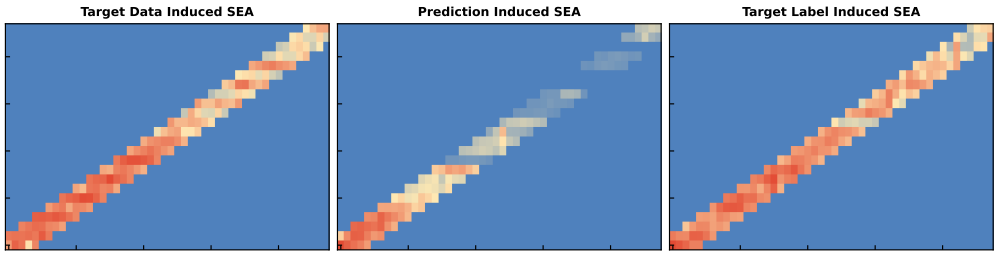
Moderate σ near the $1/L$ scale. The frequency resolution of the long sequence is $\Delta f = 1/L$, which provides a natural scale for local spectral spreading. When σ is close to this scale, such as $\sigma \in [0.8/L, 1.5/L]$, the kernel preserves locality while avoiding degenerate bin-wise matching.

Based on these observations, we set $\sigma = 1/L$ in all experiments.

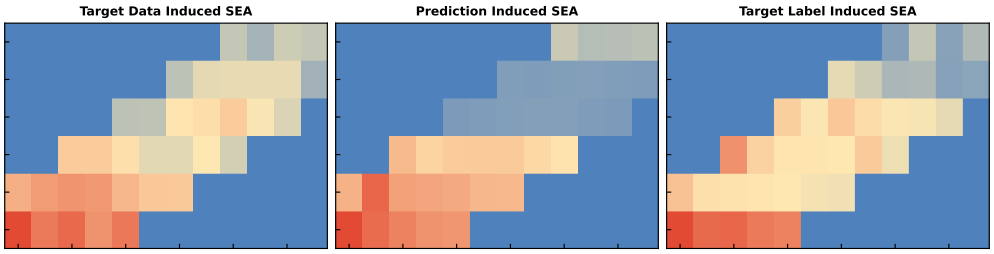
F More Visualizations

F.1 Additional SEA Pattern Visualizations

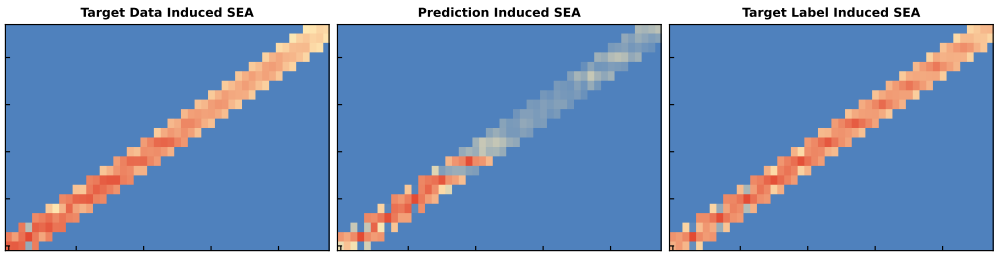
To further verify the consistency of SEA patterns within the target domain, Fig. 6 provides additional visualizations across four representative datasets. For each cross-domain scenario, we compare the input-induced, prediction-induced, and target label-induced SEA patterns. The results show that $\mathbf{SEA}(\mathbf{x}^{(1)}, \mathbf{x})$ remains highly consistent with $\mathbf{SEA}(\mathbf{y}^{(1)}, \mathbf{y})$ across different domain definitions, including stations, cities, seasons, and subjects.



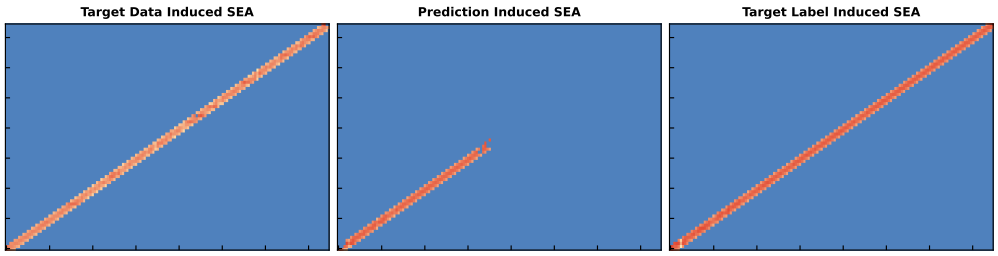
(a) ETT dataset, ETTm1→ETTm2.



(b) Air Quality dataset, Guangzhou→Tianjin.



(c) Traffic dataset, Spring→Autumn.

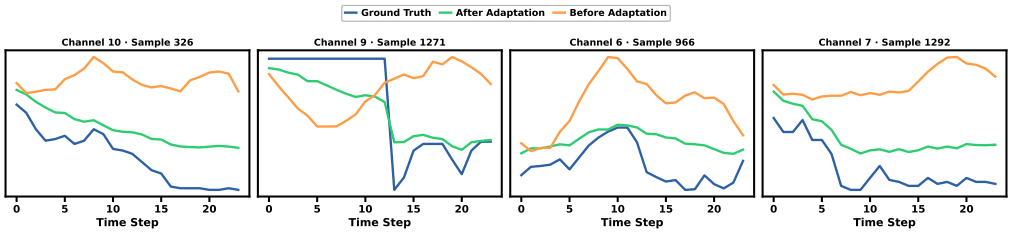


(d) PPG-DaLiA dataset, S1→S2.

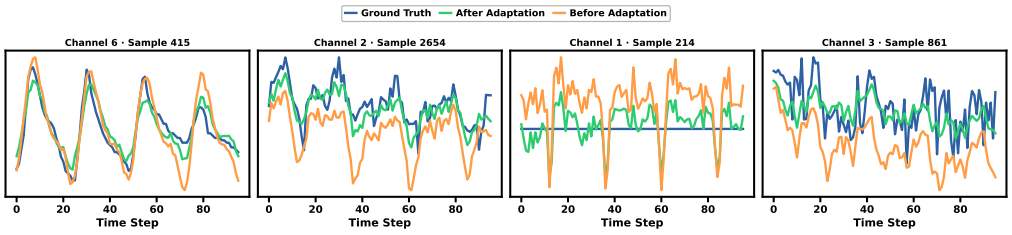
Figure 6: Additional visualizations of three SEA patterns across four datasets. The four panels use different forecasting horizon $L = 96$, 24, 96, and 300, respectively. Across different datasets and adaptation scenarios, the target data-induced SEA pattern remains highly consistent with the target label-induced SEA pattern, while the prediction-induced SEA pattern can exhibit visible deviations due to source-model bias under target-domain shifts.

F.2 Additional Forecasting Visualizations

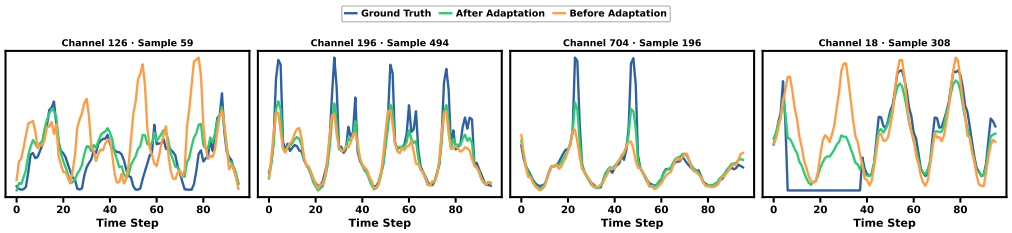
To further illustrate the effectiveness of the proposed method, we provide additional forecasting visualizations across four representative cross-domain scenarios in Fig. 7. These examples compare the predictions before and after adaptation, and qualitatively show that our method better captures the target-domain temporal dynamics and improves the alignment with the ground-truth forecasting curves.



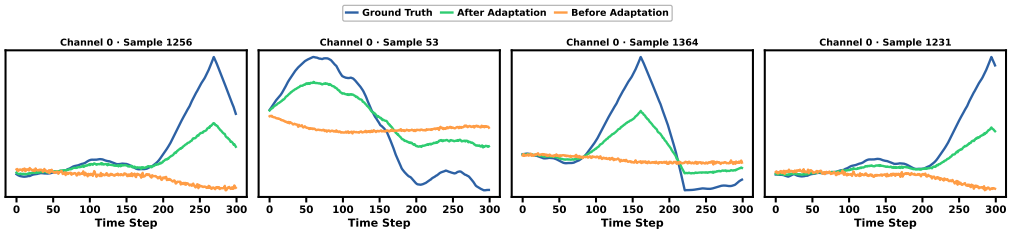
(a) Air Quality, Guangzhou→Tianjin.



(b) ETT, ETTh1→ETTh2.



(c) Traffic, Spring→Autumn.



(d) PPG-DaLiA, S1→S2.

Figure 7: Additional visualization results for source-free adaptation across four cross-domain forecasting scenarios. The results compare the prediction quality before and after adaptation and show that the proposed method improves the alignment between predicted and ground-truth temporal patterns.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See the abstract part.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: See the Limitation and Extension part of the Conclusion section.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: See Section 3.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See section 4.1 and Appendix C

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The Source code will be released after acceptance. We also provide a core code of the adaptation process in the supplementary materials.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 4.1 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Section 4.4 and Appendix C.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

    Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See Appendix C.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.