Phase-driven Generalizable Representation Learning for Nonstationary Time Series Classification

Payal Mohapatra **Lixu Wang** **Qi Zhu**
Northwestern University
Evanston, IL, USA
{payal.mohapatra, qzhu}@northwestern.edu

Abstract

Pattern recognition is a fundamental task in continuous sensing applications, but real-world scenarios often experience distribution shifts that necessitate learning generalizable representations for such tasks. This challenge is exacerbated with time-series data, which also exhibit inherent *nonstationarity*—variations in statistical and spectral properties over time. In this work, we offer a fresh perspective on learning generalizable representations for time-series classification by considering the phase information of a signal as an approximate proxy for nonstationarity and propose a phase-driven generalizable representation learning framework for time-series classification, `PhASER`. It consists of three key elements: 1) *Hilbert transform-based augmentation*, which diversifies nonstationarity while preserving task-specific discriminatory semantics, 2) *separate magnitude-phase encoding*, viewing time-varying magnitude and phase as independent modalities, and 3) *phase-residual feature broadcasting*, integrating 2D phase features with a residual connection to the 1D signal representation, providing inherent regularization to improve distribution-invariant learning. Extensive evaluations on five datasets from sleep-stage classification, human activity recognition, and gesture recognition against 13 state-of-the-art baseline methods demonstrate that `PhASER` consistently outperforms the best baselines by an average of 5% and up to 11% in some cases. Additionally, the principles of `PhASER` can be broadly applied to enhance the generalizability of existing time-series representation learning models.

1 Introduction

Time-series data are crucial in many real-world applications, such as continuous monitoring for human activity recognition [26], gesture identification [37], and sleep tracking [18]. They often exhibit *non-stationarity*, where statistical and spectral properties evolve over time. Another challenge is domain shift, where the data-generating process changes due to sensor type, sub-population shifts, or environmental variations, degrading model performance on unseen distributions. Thus, developing methods for generalizable pattern recognition in nonstationary time-series classification is essential.

Most existing methods [40, 41, 11] address distribution shifts via domain adaptation, assuming accessible target domain samples, which may not always be feasible. Some works [6, 50] apply standard domain generalization (DG) algorithms [47, 42, 38] to temporally-varying time-series data, but performance gaps remain compared to visual data. Recent DG research for time series explores latent-domain characterization [33, 5], augmentation strategies [16, 27], preservation of non-stationarity dictionaries [29, 21], and spectral features [11, 52, 19]. Limitations persist: latent-domain methods rely on assumptions about latent domains; augmentation (shift, jittering, masking) may not generalize and can distort signals, e.g., physiological signals; spectral perturbations are often heavily parametric [48] and application-specific; methods preserving non-stationarity typically maintain the

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

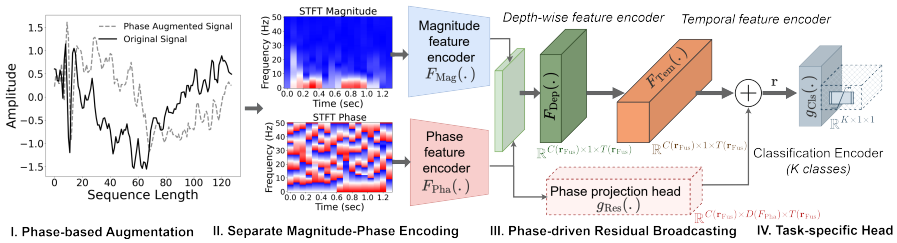

Figure 1: PhASER's components: (I) Hilbert transform-based phase augmentation, (II) separate encoding of time-varying phase and magnitude from STFT using F_{Mag} and F_{Pha}, (III) phase-residual broadcasting network including depth-wise feature encoder F_{Dep}, temporal encoder F_{Tem}, and incorporation of the phase-projection head output g_{Res} for broadcasting (with annotated intermediate feature map dimensions), and (IV) task-specific classification encoder g_{Cls}.

same input-output space, unsuitable for multivariate tasks; frequency-domain approaches overlook time-varying spectral responses; and many methods require domain labels, which are costly and intrusive [51, 2]. Achieving generalizable time-series classification without unseen data or domain labels remains a critical challenge.

We introduce Phase-Augmented Separate Encoding and Residual (PhASER), a framework for learning generalizable representations for *nonstationary* time-series classification without domain labels or explicit sub-domain characterization. We leverage phase information as a proxy for non-stationarity, since it captures local time shifts and time-localized frequency variations characteristic of nonstationary signals [24, 36]. PhASER's phase-anchored components include three key modules (Figure 1): (i) intra-instance phase-shift augmentation via the non-parametric Hilbert Transform (HT) [22] to diversify source data; (ii) separate encoding of time-varying magnitude and phase for richer time-frequency integration; and (iii) a broadcasting mechanism with a non-linear residual connection from the phase embedding to the backbone to learn generalizable [10, 34] task-specific features [12]. Evaluations on 5 datasets against 13 baselines demonstrate PhASER's superiority, including in challenging one-to-many domain transfer scenarios.

2 Approach

Given a dataset $\mathbf{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{M}$ of nonstationary time series from multiple unknown source domains $S = \{\mathcal{S}_i\}_{i=1}^{N_S}$, where each series $\mathbf{x}_i \in \mathbb{R}^{V \times T}$ can be decomposed as $\Pr_{\mathbf{x} \sim \mathcal{D}_{\mathbf{x}}}(\mathbf{x})(t) = \mu_t + \sigma_t \times z$ with time-varying statistics satisfying $\exists t, [\mu_t \neq \mu_{t+L}] \vee [\sigma_t \neq \sigma_{t+L}]$ for some $L \geq 1$, our objective is to train a model $F \circ g : \mathcal{X}_{\mathbf{S}} \to \mathcal{Y}_{\mathbf{S}}$ (feature extractor + classifier) that minimizes the expected loss on any unseen target domain, i.e., $\min \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\text{U}}}[\mathcal{L}(g(F(\mathbf{x})), y)]$. The time series samples are nonstationary, the source domains have distinct distributions, and the domain identities are unknown.

Hilbert Transform based Phase Augmentation. Non-task-specific non-stationarity degrades a model's generalization (Figure 3 in the Appendix), and the signal's phase encodes its non-stationarity (Figure 4 in the Appendix). Motivated by this, we introduce an intra-sample phase-augmentation technique that diversifies non-stationarity while preserving the original signal's discriminatory properties. Unlike standard augmentations, we shift a signal's phase while keeping its magnitude intact, providing an augmented view. For a real-valued time series $\mathbf{x} = \{x_0, \ldots, x_t, \ldots\} \in \mathbb{R}$ with $x_t = \mathbf{x}(t)$, its Hilbert Transform (HT) is $\widehat{\mathbf{x}}(t) = \text{HT}(\mathbf{x}(t)) = \int_{-\infty}^{\infty} \mathbf{x}(\tau) \frac{1}{\pi(t-\tau)} d\tau$, which in the frequency domain is $\widehat{\mathbf{x}}(t) = \mathcal{F}^{-1}\{-i \cdot \text{sgn}(\xi)\mathcal{F}\{\mathbf{x}(t)\}(\xi)\}$, where $\mathcal{F}, \mathcal{F}^{-1}$ are the Fourier transform and its inverse, and $\text{sgn}(\cdot)$ is the sign function. HT induces a phase shift of $-\pi/2$, generating an out-of-phase signal $\widehat{\mathbf{x}}$. Applying this across all feature dimensions, we merge the augmented set $\widehat{\mathbf{S}}$ with the original $\mathbf{S}$ to form $\mathbf{S}' = \widehat{\mathbf{S}} \cup \mathbf{S}$; thereafter, no distinction is made between samples from $\widehat{\mathbf{S}}$ or $\mathbf{S}$.

Magnitude-Phase Separate Encoding. After augmenting the source domain with phase-shifts via HT, we identify optimal ways to encode time series for generalization. Unlike most methods that separate time and frequency information, we treat *magnitude* and *phase* as distinct modalities. We adopt STFT instead of DFT, as STFT captures time-varying signals by applying DFT within a moving window. For a discrete time series $\mathbf{x}[n]$ sampled from $\mathbf{x}(t)$ with length N, STFT is $f_{\mathbf{x}}[n,k] = \sum_{m=n-(W-1)}^{n} w[n-m]\mathbf{x}[m]e^{i\xi_k m}$, with $\xi_k = 2\pi k/\Xi$ and $w[n]$ a Hanning window $w[n] = 0.5(1 - \cos\frac{2\pi n}{W-1})$. Window lengths $W_i = 2^{p_i} \leq \Xi, p_i \sim \mathcal{U} \in \mathbb{Z}_0^+, i \in [1, V]$ are randomly sampled

per feature. Magnitude and phase are computed as $\mathrm{Mag}(\mathbf{x}) = \sqrt{\mathrm{Re}(f_{\mathbf{x}}[n,k])^2 + \mathrm{Im}(f_{\mathbf{x}}[n,k])^2}$ and $\mathrm{Pha}(\mathbf{x}) = \arctan 2(\mathrm{Im}(f_{\mathbf{x}}[n,k]), \mathrm{Re}(f_{\mathbf{x}}[n,k]))$, forming inputs to separate encoders F_{Mag} and F_{Pha}. To enhance generalization, we apply sub-feature normalization [4] over B sub-feature spaces: $F_{\mathrm{Mag}}(\mathbf{x})_b := (F_{\mathrm{Mag}}(\mathbf{x})_b - \overline{F_{\mathrm{Mag}}(\mathbf{x})_b})/\sigma(F_{\mathrm{Mag}}(\mathbf{x})_b)$, and similarly for $F_{\mathrm{Pha}}(\mathbf{x})$. Finally, both embeddings are fused along the variate axis via 2D convolutions in F_{Fus}, yielding $\mathbf{r}_{\mathrm{Fus}} = F_{\mathrm{Fus}}(F_{\mathrm{Mag}}(\mathbf{x}), F_{\mathrm{Pha}}(\mathbf{x}))$ for downstream modules.

Intuition for treating phase and magnitude separately. Prior studies [11, 19] highlight the importance of spectral input for generalizable learning. To investigate optimal time-frequency representations, we perform a small-scale study on the WISDM HAR dataset [25], comparing four configurations: magnitude-only, phase-only, concatenated magnitude and phase, and separate encoders for magnitude and phase (Table 1). Results show that magnitude-only features are more discriminative than phase-only (0.81 vs. 0.62 accuracy), while phase alone still exceeds chance accuracy(0.17), indicating the presence of task-relevant but time-varying information. Concatenating magnitude and phase does not improve performance, whereas separate encoding with late fusion yields the best accuracy (0.85), suggesting that independently extracting high-level features from each modality both captures discriminative content and leverages phase as an approximate proxy for non-stationarity. This motivates our approach of separate magnitude and phase encoders in PhASER.

Table 1: Comparison of various time-frequency input configurations.

Input Modality	Accuracy
Only Magnitude (Mag)	0.81 ± 0.03
Only Phase (Pha)	0.62 ± 0.03
Mag-Pha Concatenate	0.73 ± 0.03
Mag-Pha Separate	0.85 ± 0.01

Phase-Residual Feature Broadcasting. We propose a phase-based broadcasting approach for domain-generalizable representation learning. Fused embeddings $\mathbf{r}_{\mathrm{Fus}}$ are first transformed by a depthwise feature encoder F_{Dep} into 1D feature maps $\mathbf{r}_{\mathrm{Dep}}$ along the temporal dimension: $\mathbb{R}^{C(\mathbf{r}_{\mathrm{Fus}})\times D(\mathbf{r}_{\mathrm{Fus}})\times T(\mathbf{r}_{\mathrm{Fus}})} \rightarrow \mathbb{R}^{C(\mathbf{r}_{\mathrm{Fus}})\times 1\times T(\mathbf{r}_{\mathrm{Fus}})}$, where $C(\cdot), D(\cdot), T(\cdot)$ denote channel, feature, and temporal dimensions. F_{Dep} uses convolution layers followed by average pooling to unify features at each temporal index. A sequence-to-sequence temporal encoder F_{Tem} is applied to $\mathbf{r}_{\mathrm{Dep}}$ to capture temporal dependencies; here, convolution layers are used, but other architectures are compatible (see Section B). This separation allows F_{Dep} to specialize in spectral attributes while F_{Tem} models global temporal structure. We introduce a non-linear projection g_{Res} of $F_{\mathrm{Pha}}(\mathbf{x})$ as a shortcut to match F_{Tem}'s output dimensions: $\mathbb{R}^{C(F_{\mathrm{Pha}}(\mathbf{x}))\times D(F_{\mathrm{Pha}}(\mathbf{x}))\times T(F_{\mathrm{Pha}}(\mathbf{x}))} \rightarrow \mathbb{R}^{C(\mathbf{r}_{\mathrm{Fus}})\times D(F_{\mathrm{Pha}}(\mathbf{x}))\times T(\mathbf{r}_{\mathrm{Fus}})}$. The final representation is obtained by broadcasting: $\mathbf{r} = F_{\mathrm{Tem}}(\mathbf{r}_{\mathrm{Dep}}) + g_{\mathrm{Res}}(F_{\mathrm{Pha}}(\mathbf{x}))$, preserving discriminatory characteristics under non-stationarity.

Intuition for phase-residual broadcasting. We conduct a controlled experiment comparing different residual broadcasting strategies: no residual, using magnitude (F_{Mag}), using phase (F_{Pha}), and using fused magnitude and phase (F_{Fus}). We evaluate in-domain (held-out source samples) and out-of-domain (target domain) performance on the Gesture Recognition (GR) dataset (Figure 2). As expected, the magnitude residual performs well in-domain but shows a larger drop in OOD accuracy, suggesting overfitting to non-task-specific, in-domain non-stationarity. In contrast, using a phase residual—especially after diversifying phase via the proposed augmentation—helps regularize the model against non-task-specific non-stationarity, improving generalization.

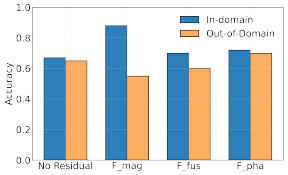

Figure 2: Comparison of generalization performance for different residual broadcasting features.

Finally, semantic distinction is optimized via Cross-Entropy Loss applied to a classification head g_{Cls}: $\mathcal{L}_{\mathrm{CE}} = \frac{1}{N_B}\sum_{i=1}^{N_B} \mathbf{y}_i \log g_{\mathrm{Cls}}(\mathbf{r}_i)$, where N_B is the mini-batch size and $\mathbf{y}_i$ is the one-hot label.

3 Experiments

We evaluate PhASER against 13 state-of-the-art methods, including a large foundation time-series model, on five datasets across three applications, using per-segment accuracy. Implementation details are in Section D.

Datasets. We evaluate on three applications: Human Activity Recognition (HAR), Sleep-Stage Classification (SSC), and Gesture Recognition (GR). For HAR, we use three benchmarks: ***WISDM*** [25] (36 users, 3 univariate channels), ***UCIHAR*** [3] (30 users, 9 channels), and ***HHAR*** [44] (9 users,

3 channels). All contain 6 activities with sequence length 128. For SSC, we use PhysioBank [8], single-channel EEG from 20 healthy individuals (sequence length 3000). For GR, we use EMG data [31] (8 channels, 6 gestures, sequence length 200) prepared as in Lu et al. [32]. HAR and SSC follow ADATime [40]. Dataset statistics are in Table 7, with class distributions (Figure D.1) and metric trends for WISDM (Figure D.1) motivating our choice of AUC and accuracy [40, 32].

Experimental Setup. Each dataset is partitioned into four non-overlapping cross-domain scenarios [33, 40]. We reserve 20% of training data for validation and report mean results over three trials in the main text; full statistics are in Section E.

Comparison Baselines. We compare against: domain generalization methods ERM, DANN [7], GroupDRO [42], RSC [14], and ANDMask [38] (DomainBed [9]); audio DG method BCResNet [20]; time-series representation learner MAPU [41]; deep classifier InceptionTime [15] [35]; time-series DG method Diversify [32]; foundation model Chronos [1]; forecasting models NSTrans [29] and Koopa [30]; and statistical normalization RevIN [21] integrated with ours (Ours+RevIN). Default setups are used with minimal modifications; details are in Sections D.2 and D.4.

Table 2: Average cross-person classification accuracy per dataset. **Best** in bold, second-best underlined.

Method	WISDM	HHAR	UCIHAR	Sleep-Stage	Gesture
ERM	0.53	0.47	0.70	0.47	0.54
GroupDRO	0.66	0.59	0.87	0.57	0.48
DANN	0.68	0.68	0.83	0.65	0.64
RSC	0.66	0.48	0.78	0.49	0.59
ANDMask	0.71	0.66	0.80	0.54	0.45
InceptionTime	0.81	0.80	0.88	0.76	0.70
BCResNet	0.79	0.70	0.80	0.80	0.64
NSTrans	0.40	0.24	0.42	0.39	0.33
Koopa	0.63	0.69	0.78	0.56	0.58
Ours+RevIN	0.85	0.85	0.94	0.80	0.76
MAPU	0.75	0.76	0.83	0.68	0.68
Diversify	0.82	0.77	0.89	0.73	0.75
Chronos	0.66	0.72	0.61	0.51	0.51
Ours	**0.85**	**0.87**	**0.95**	**0.82**	**0.76**

Table 3: Classification accuracy with Source 0–8 for one-person-to-another generalization on HHAR; **best** in bold, second-best underlined.

Method	0	1	2	3	4	5	6	7	8	Avg.
ERM	0.27	0.40	0.41	0.44	0.42	0.44	0.45	0.44	0.48	0.42
GroupDRO	0.33	0.53	0.38	0.48	0.47	0.51	0.47	0.48	0.49	0.46
DANN	0.32	0.44	0.42	0.45	0.42	0.48	0.49	0.45	0.51	0.44
RSC	0.27	0.45	0.38	0.45	0.40	0.47	0.50	0.44	0.53	0.43
ANDMask	0.34	0.50	0.37	0.43	0.46	0.51	0.46	0.47	0.52	0.45
InceptionTime	0.52	0.62	0.44	**0.69**	0.60	0.57	0.66	0.64	0.61	0.59
BCResNet	0.28	0.48	0.32	0.47	0.42	0.52	0.44	0.45	0.49	0.43
NSTrans	0.20	0.22	0.17	0.20	0.21	0.22	0.26	0.17	0.20	0.21
Koopa	0.32	0.42	0.37	0.40	0.42	0.45	0.35	0.43	0.48	0.40
Ours+RevIN	0.48	0.66	0.57	0.65	0.61	0.64	0.65	0.64	0.63	0.62
MAPU	0.39	0.57	0.35	0.52	0.49	0.54	0.49	0.50	0.52	0.49
Diversify	0.42	0.62	0.32	0.62	0.56	0.61	0.53	0.52	0.61	0.53
Chronos	0.32	0.23	0.26	0.25	0.27	0.23	0.21	0.24	0.25	0.25
Ours	**0.53**	**0.70**	**0.63**	0.66	**0.64**	**0.67**	**0.65**	**0.67**	**0.62**	**0.64**

Effectiveness of PhASER across Applications. We evaluate PhASER's generalization ability across multiple time-series tasks. For Human Activity Recognition (HAR), we consider cross-person generalization (training on $N_S > 1$ sources, testing on unseen targets) and one-person-to-another (training on a single person, testing on another). In the cross-person setting (Table 2), existing domain generalization methods underperform for time-series tasks [6, 32], whereas PhASER consistently outperforms the best baselines on WISDM, HHAR, and UCIHAR by 3%, 9%, and 6%, respectively. In the one-person-to-another setting (HHAR, Table 3), it surpasses Diversify by nearly 20% and InceptionTime by almost 8%. For EEG-based sleep-stage classification, a challenging cross-person task, PhASER achieves the best performance across five types, exceeding BCResNet by 2% and Diversify by nearly 11% (Table 2, left). Similarly, in Gesture Recognition, where bio-electronic signals are highly non-stationary, evaluations on 6 common classes in a cross-person setting (Table 2, right) show that PhASER again attains the best overall performance.

Ablation Study. We evaluate the contribution of PhASER's components on WISDM and GR (Table 4). Removing phase augmentation (row 2) drops performance by 11.6% on WISDM and 5.8% on GR, while separate magnitude-phase encoding (row 6 vs 5) is crucial, consistent with Table 1. Phase-residual broadcasting boosts performance by 4% (row 1 vs 5), confirming that phase serves as a proxy for non-stationarity. Removing both phase residual and separate encoding (rows 3–7) leads to average drops of 10.6% and 13.7%.

	Phase Augmentation	Separate Encoders	F_{Pha} Residual	Accuracy WISDM	Accuracy GR
1	✓	✓	✓	$0.86_{\pm 0.02}$	$0.70_{\pm 0.01}$
2	✗	✓	✓	$0.81_{\pm 0.01}$	$0.61_{\pm 0.01}$
3	✓	✓	✗(F_{Mag} Res.)	$0.82_{\pm 0.01}$	$0.55_{\pm 0.01}$
4	✓	✓	✗(F_{Fus} Res.)	$0.84_{\pm 0.01}$	$0.60_{\pm 0.01}$
5	✓	✓	✗	$0.82_{\pm 0.01}$	$0.65_{\pm 0.01}$
6	✓	✗(Mag Only)	✗	$0.73_{\pm 0.01}$	$0.59_{\pm 0.03}$
7	✓	✗(Mag Only)	✗(F_{Mag} Res.)	$0.83_{\pm 0.01}$	$0.66_{\pm 0.02}$
8	✓	✗(Mag-Pha Concat.)	✗	$0.73_{\pm 0.03}$	$0.61_{\pm 0.02}$

Table 4: Ablation of PhASER on WISDM and GR. ✓ = inclusion, ✗ = exclusion/modification.

4 Conclusion

We address generalization for nonstationary time-series classification using a phase-driven approach, without accessing source domain labels or samples from unseen distributions. Our approach applies phase-based augmentation, treats time-varying magnitude and phase as separate modalities, and incorporates a phase-derived residual connection. We support our design choices with rigorous theoretical and empirical evidence.

References

[1] Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

[2] Guangji Bai, Chen Ling, and Liang Zhao. Temporal domain generalization with drift-aware dynamic neural networks. *arXiv preprint arXiv:2205.10664*, 2022.

[3] Erhan Bulbul, Aydin Cetin, and Ibrahim Alper Dogru. Human activity recognition using smartphones. In *2018 2nd international symposium on multidisciplinary studies and innovative technologies (ismsit)*, pages 1–6. IEEE, 2018.

[4] Simyung Chang, Hyoungwoo Park, Janghoon Cho, Hyunsin Park, Sungrack Yun, and Kyuwoong Hwang. Subspectral normalization for neural audio data processing. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 850–854. IEEE, 2021.

[5] 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 30th ACM international conference on information & knowledge management*, pages 402–411, 2021.

[6] Jean-Christophe Gagnon-Audet, Kartik Ahuja, Mohammad-Javad Darvishi-Bayazi, Pooneh Mousavi, Guillaume Dumas, and Irina Rish. Woods: Benchmarks for out-of-distribution generalization in time series. *arXiv preprint arXiv:2203.09978*, 2022.

[7] 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*, 17(59):1–35, 2016.

[8] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

[9] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[10] Fengxiang He, Tongliang Liu, and Dacheng Tao. Why resnet works? residuals generalize. *IEEE transactions on neural networks and learning systems*, 31(12):5349–5362, 2020.

[11] Huan He, Owen Queen, Teddy Koker, Consuelo Cuevas, Theodoros Tsiligkaridis, and Marinka Zitnik. Domain adaptation for time series under feature and label shifts. *arXiv preprint arXiv:2302.03133*, 2023.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Norden Eh Huang. *Hilbert-Huang transform and its applications*, volume 16. World Scientific, 2014.

[14] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 124–140. Springer, 2020.

[15] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

[16] Brian Kenji Iwana and Seiichi Uchida. An empirical survey of data augmentation for time series classification with neural networks. *Plos one*, 16(7):e0254841, 2021.

[17] Ming Jin, Yifan Zhang, Wei Chen, Kexin Zhang, Yuxuan Liang, Bin Yang, Jindong Wang, Shirui Pan, and Qingsong Wen. Position paper: What can large language models tell us about time series analysis. *arXiv preprint arXiv:2402.02713*, 2024.

[18] Bob Kemp, Aeilko H Zwinderman, Bert Tuk, Hilbert AC Kamphuisen, and Josefien JL Oberye. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194, 2000.

[19] Byeonggeun Kim, Simyung Chang, Jinkyu Lee, and Dooyong Sung. Broadcasted residual learning for efficient keyword spotting. *arXiv preprint arXiv:2106.04140*, 2021.

[20] Byeonggeun Kim, Seunghan Yang, Jangho Kim, and Simyung Chang. Domain generalization on efficient acoustic scene classification using residual normalization. In *6th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2021.

[21] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.

[22] Frederick W King. *Hilbert Transforms: Volume 2*, volume 2. Cambridge University Press, 2009.

[23] Diederik P Kingma, J Adam Ba, and J Adam. A method for stochastic optimization. arxiv 2014. *arXiv preprint arXiv:1412.6980*, 106, 2020.

[24] R Klein, D Ingman, and S Braun. Non-stationary signals: Phase-energy approach—theory and simulations. *Mechanical Systems and Signal Processing*, 15(6):1061–1089, 2001.

[25] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[26] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2020.

[27] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8886–8895, 2021.

[28] Chunyu Liu, Yongpei Ma, Kavitha Kothur, Armin Nikpour, and Omid Kavehei. Biosignal copilot: Leveraging the power of llms in drafting reports for biomedical signals. *medRxiv*, pages 2023–06, 2023.

[29] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893, 2022.

[30] Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in Neural Information Processing Systems*, 36, 2024.

[31] Sergey Lobov, Nadia Krilova, Innokentiy Kastalskiy, Victor Kazantsev, and Valeri A Makarov. Latent factors limiting the performance of semg-interfaces. *Sensors*, 18(4):1122, 2018.

[32] Wang Lu, Jindong Wang, Xinwei Sun, Yiqiang Chen, and Xing Xie. Out-of-distribution representation learning for time series classification. In *The Eleventh International Conference on Learning Representations*, 2022.

[33] Wang Lu, Jindong Wang, Xinwei Sun, Yiqiang Chen, and Xing Xie. Out-of-distribution representation learning for time series classification. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=gUZWOE42l6Q.

[34] Pierre Marion, Yu-Han Wu, Michael E Sander, and Gérard Biau. Implicit regularization of deep residual networks towards neural odes. *arXiv preprint arXiv:2309.01213*, 2023.

[35] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, pages 1–74, 2024.

[36] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 2nd edition, 1999.

[37] Mehmet Akif Ozdemir, Deniz Hande Kisa, Onan Guren, Aytug Onan, and Aydin Akan. Emg based hand gesture recognition using deep learning. In *2020 Medical Technologies Congress (TIPTEKNO)*, pages 1–4. IEEE, 2020.

[38] Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

[39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[40] Mohamed Ragab, Emadeldeen Eldele, Wee Ling Tan, Chuan-Sheng Foo, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. Adatime: A benchmarking suite for domain adaptation on time series data. *ACM Transactions on Knowledge Discovery from Data*, 17(8):1–18, 2023.

[41] 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 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1989–1998, 2023.

[42] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[43] Said E Said and David A Dickey. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3):599–607, 1984.

[44] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pages 127–140, 2015.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[46] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[47] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

[48] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2021.

[49] Zhaohua Wu, James Bridges, Xuxin Chen, Koji Ide, Joshua Garland, Keith James, Satyajit Dash, Vasilis Z. Marmarelis, Vasilis Z. Marmarelis, Vasilis Z. Marmarelis, and Andre Longtin. Nonlinear phase interaction between nonstationary signals: A comparison study of methods based on hilbert-huang and fourier transforms. *Physical Review E*, 79(6):061924, June 2009. doi: 10.1103/PhysRevE.79.061924. URL https://doi.org/10.1103/PhysRevE.79.061924.

[50] Xuhai Xu, Han Zhang, Yasaman Sefidgar, Yiyi Ren, Xin Liu, Woosuk Seo, Jennifer Brown, Kevin Kuehn, Mike Merrill, Paula Nurius, et al. Globem dataset: Multi-year datasets for longitudinal human behavior modeling generalization. *Advances in Neural Information Processing Systems*, 35:24655–24692, 2022.

[51] Siyuan Yan, Chi Liu, Zhen Yu, Lie Ju, Dwarikanath Mahapatra, Brigid Betz-Stablein, Victoria Mar, Monika Janda, Peter Soyer, and Zongyuan Ge. Prompt-driven latent domain generalization for medical image classification. *arXiv preprint arXiv:2401.03002*, 2024.

[52] Ling Yang and Shenda Hong. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning*, pages 25038–25054. PMLR, 2022.

Appendix

A Motivation Cases

Now we pose the question: *What is the impact of non-stationarity of time series on a model's generalization ability ?* We conduct a simple empirical study on the HHAR dataset by varying the sequence length to synthesize increasing non-stationarity, measured by the ADF statistic (a higher ADF value indicates greater non-stationarity). More details of the ADF test are provided in Section B of the Appendix. We adopt a DG model BCResNet from Kim et al. [19] for time-series classification to explore the relationship between the degree of non-stationarity and the model's generalization ability to unseen distributions.

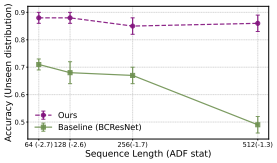

Figure 3: (**Nonstationarity impacts generalization**) Comparison between PhASER and BCResNet with increasing non-stationarity in HHAR dataset.

We intuitively justify our design choice by exploring a question: *Does shifting the phase response of a time series change its non-stationarity?* In Figure 4(a–c), we illustrate a stationary sinusoid and two non-stationary sinusoids all sharing the same frequency, as evidenced by their magnitude responses shown in Figure 4(d–f). However, the distinct phase responses of each signal reveal changes in the underlying dynamics. These phase variations occur as time-local oscillometric fluctuations arise, motivating the use of phase information as a proxy for capturing the underlying non-stationarity [49].

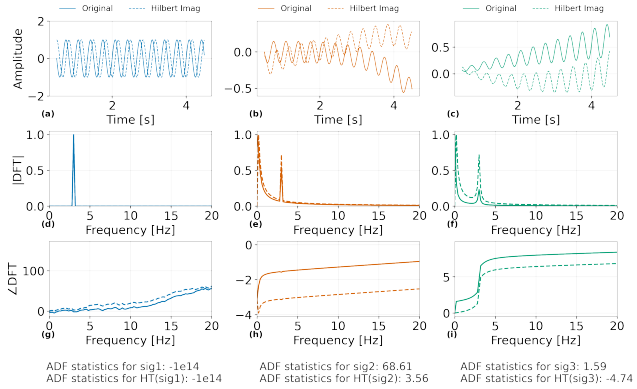

Figure 4: Phase as a proxy for non-stationarity. (a–c) Time-domain signals: (a) Stationary sinusoid; (b, c) Non-stationary sinusoids with the same base frequency. (d–f) Magnitude spectra (DFT) of the corresponding signals, showing similar frequency content. (g–i) Phase spectra, displaying distinct responses that reflect differences in underlying dynamics. ADF statistics below each column summarize the stationarity of the respective signals and show how it changes with the proposed Hilbert Transform (HT)-based augmentation.

B Additional Details on PhASER

Augmented Dickey Fuller (ADF) Test. This is a statistical tool to assess the non-stationarity of a given time-series signal. This test operates under a null hypothesis $\mathbb{H}_0$ where the signal has a *unit-root*. The existence of *unit-root* is a guarantee that the signal is non-stationary [43]. To reject $\mathbb{H}_0$, the statistic value of the ADF test should be less than the critical values associated with a significance level of 0.05 (denoted by p, the probability of observing such a test statistic under the null hypothesis). Throughout the paper, for multivariate time series, the average ADF statistics across all variates are reported. Besides, since this is a statistical tool to evaluate non-stationarity for each instance of time-series data, we provide an average of this number across a dataset to give the reader a view of the degree of non-stationarity.

Phase Augmentation. In this work, we are particularly interested in learning representations robust to temporal distribution shifts. Incorporating a phase shift in a signal is a less-studied augmentation technique. One of the main challenges is that real-world signals are not composed of a single frequency component and accurately estimating and controlling the shifting of the phase while

retaining the magnitude spectrum of a signal is difficult. To solve this, we leverage the analytic transformation of a signal using the Hilbert Transform. The key advantages of this technique are maintaining global temporal dependencies and magnitude spectrum, no exploration of design parameters and being extendible to non-stationary and periodic time series.

Lets walk through a simple example for a signal, $\mathbf{x}(t) = 2cos(w_0 t)$ which can be written in the polar coordinates as $\mathbf{x}(t) = e^{iw_0 t} + e^{-iw_0 t}$. Applying the HT conditions, $\mathrm{HT}(\mathbf{x}(t)) = 2sin(w_0 t)$. Essentially, HT shifts the signal by $\pi/2$ radians. We conduct this instance-level augmentation for each variate of the time series input. The aim is to diversify the phase representation. We use the *scipy* [46] library to implement this augmentation.

STFT Specifications. Non-stationary signals contain time-varying spectral properties. We use STFT to capture these magnitude and phase responses in both time and frequency domains. There are three main arguments to compute STFT - length of each segment (characterized by the window size and the ratio for overlap), the number of frequency bins, and the sampling rate. We use the scipy library to implement this operation and use a $k < 1$ as a multiplier to the length of the window W to give the segment length as $k \times W$ with no overlap between segments. The complete list of STFT specifications is given in Table 5. We also demonstrate a sensitivity analysis concerning the number of frequency bins and the segment length in Figure 5.

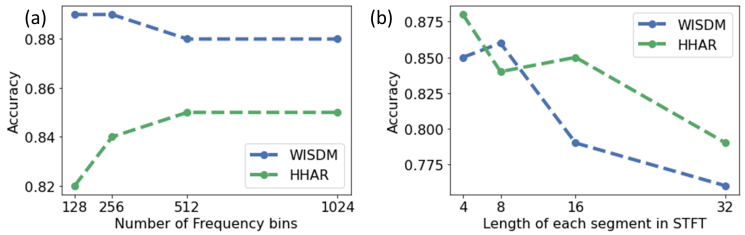

Figure 5: Illustration of the sensitivity of performance to the design choices of STFT by varying a) the number of frequency bins with a fixed segment length of 4 and b) by varying the segment lengths with a 1024 frequency bins.

Table 5: Arguments for STFT computation

Dataset	Sampling Rate	Sequence Length	STFT segment length	Number of frequency bins
WISDM	20 Hz	128	4	1024
HHAR	100 Hz	128	4	1024
UCIHAR	50 Hz	128	4	1024
SSC	100 Hz	3000	16	1024
GR	200 Hz	200	4	1024

Note: It is tempting to use an empirical mode transformation and then apply a Hilbert-Huang transformation to obtain an instantaneous phase and amplitude response in the case of non-stationary signals. It absolves us from a finite time-frequency resolution for the STFT spectra. However, our initial results indicate a high dependence on the choice of the number of intrinsic mode functions [13] for signal decomposition. Hence, for a generalizable approach, we choose STFT as the tool for the time-frequency spectrum.

Backbones for Temporal Encoder. The choice of temporal encoder, F_{Tem}, is not central to our design. Table 6 demonstrates the performance of `PhASER` under the identical settings for four cross-person settings using WISDM datasets using different backbones for F_{Tem}. For the convolution-based self-attention (second row in Table 6) we use three encoders to compute query (W_q), key (W_k), and value(V) matrices for $\mathbf{r}_{\text{Dep}}$ following the guidelines from Vaswani et al. [45]. Then we compute self-attention as, $A = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$, where d_k is the temporal dimension of $\mathbf{r}_{\text{Dep}}$. Subsequently, we use $\hat{\mathbf{r}}_{\text{Dep}} = \mathbf{r}_{\text{Dep}} + A$, as the input to F_{Tem}.

C Dataset Details

Past works [6, 40] have shown that the datasets used in our work suffer from a distribution shift across users and also within the same user temporally. This makes them suitable for evaluating the efficacy

Table 6: Results for 4 different cross-person settings for WISDM dataset.

Backbones for F_{Tem}	1	2	3	4
2D Convolution based	0.86	0.85	0.86	0.84
2D Convolution based with self-attention	0.88	0.83	0.84	0.81
Transformer	0.87	0.84	0.87	0.84

of our framework. In this section, we provide more details on the datasets. Table 7 summarizes the average ADF statistics of the datasets along with their variates and their number of classes and domains.

Table 7: Summary of the dataset attributes. Higher value of ADF stat indicates greater non-stationarity within a signal.

Category	**Dataset**	**Representative ADF-Statistic (mean across all variates)**	**Variates**	**Domains**	**Classes**
Human Activity recognition	UCIHAR	-2.58 (0.044)	9	31	6
Human Activity recognition	HHAR	-1.74 (0.062)	3	9	6
Human Activity recognition	WISDM	-0.78 (0.051)	3	36	6
Gesture Recognition	EMG	-33.14 (0.011)	8	36	6
Sleep Stage Classification	EEG	-3.7 (0.047)	1	20	5

WISDM [25]: It originally consists of 51 subjects performing 18 activities but we follow the ADATime [40] suite to utilize 36 subjects comprising of 6 activity classes given as walking, climbing upstairs, climbing downstairs, sitting, standing, and lying down. The dataset consists of 3-axis accelerometer measurements sampled at 20 Hz to predict the activity of each participant for a segment of 128-time steps. According to Ragab et al. [40], this is the most challenging dataset suffering from the highest degree of class imbalance.

HHAR [44]: To remain consistent with the existing AdaTime benchmark we leverage the Samsung Galaxy recordings of this dataset from 9 participants from a 3-axis accelerometer sampled at 100 Hz. The 6 activity classes, in this case, are - biking, sitting, standing, walking, climbing up the stairs, and climbing down the stairs.

UCIHAR [3]: This dataset is collected from 30 participants using 9-axis inertial motion unit using a waist-mounted cellular device sampled at 50 Hz. The six activity classes are the same as WISDM dataset.

SSC [8]: This is a single channel EEG dataset collected from 20 subjects to classify five sleep stages - wake, non-rapid eye movement stages - N1, N2, N3, and rapid-eye-movement.

GR [31]: For surface-EMG based gesture recognition we follow Lu et al. [33]'s preprocessing and use an 8-channel data recorded from 36 participants for six types of gestures sampled at 200 Hz. Note, that this is the least stationary dataset (see Table 7, yet `PhASER` performs as well as or better than the stat-of-the-art techniques as shown in Table 2 in the main paper.

D Implementation Details

All experiments are performed on an Ubuntu OS server equipped with NVIDIA TITAN RTX GPU cards using PyTorch framework. Every experiment is carried out with 3 different seeds (2711, 2712, 2713). During model training, we use Adam optimizer [23] with a learning rate from 1e-5 to 1e-3 and maximum number of epochs is set to 150 based on the suitability of each setting. We tune these optimization-related hyperparameters for each setting and save the best model checkpoint based on early exit based on the minimum value of the loss function achieved on the validation set.

D.1 Dataset Configuration

There is no standard benchmarking for domain generalization for time-series where the domain labels and target samples are inaccessible. We leverage past works of Ragab et al. [40], Lu et al. [33] for preprocessing steps. For each dataset, we use a cross-person setting in four scenarios. The details of the target domains chosen in each scenario are given in Table 8, the rest are used as source domains.

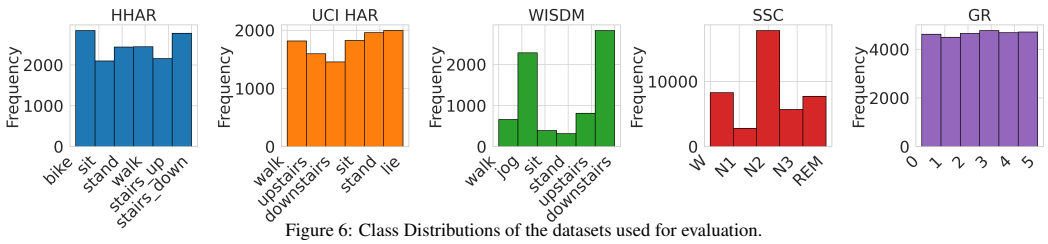

Figure 6: Class Distributions of the datasets used for evaluation.

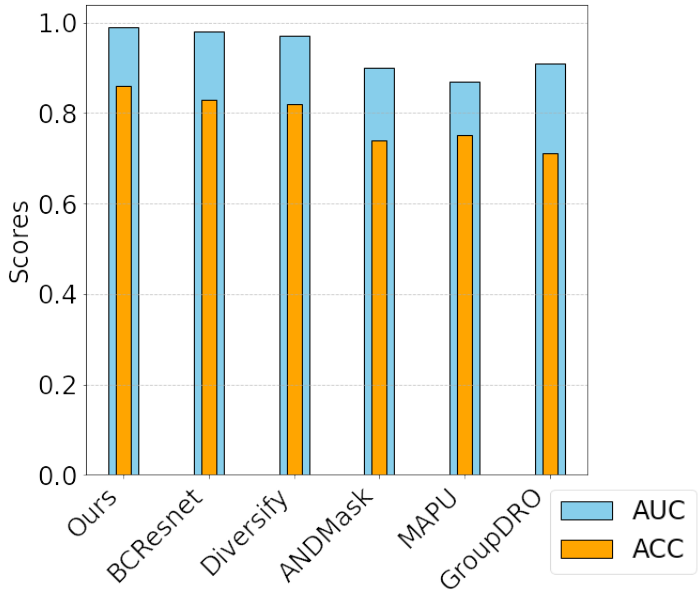

Figure 7: Illustration of additional performance metric, Area Under the ROC Curve (AUC), along with Accuracy—for Scenario 1 of the WISDM dataset, for the top-performing baselines. These metrics demonstrate consistency and justify our choice of accuracy as the primary evaluation metric.

Note for GR we use the same splits as Lu et al. [33]. Our method is not influenced by domain labels as we do not require them for our optimization.

Table 8: Target domain splits for 4 scenarios of each dataset.

Target Domains	Scenario 1	Scenario 2	Scenario 3	Scenario 4
WISDM	0-9	10-17	18-27	28-35
HHAR	0,1	2,3	4,5	6-8
UCIHAR	0-7	8-15	16-23	24-29
GR	0-8	9-17	18-26	27-35
SSC	0-5	5-9	10-14	15-20

Figure D.1 illustrates the class distribution for each dataset. Only the WISDM and Sleep Stage Classification (SSC) datasets exhibit notable imbalances among certain classes. To validate the consistency of our conclusions, we compare the Area Under the Curve (AUC) with the adopted accuracy metric in Figure D.1. Generally, past works [33, 6], utilizing these datasets have adopted accuracy as the primary performance metric, and we follow the same approach.

D.2 Baseline Methods

General Domain Generalization Methods. For all the standard domain generalization baselines we use conv2D layers for feature transformation of multivariate time series. It is worth mentioning that DANN is actually a domain adaptation study, which requires access to certain unlabeled target domain data. For cross-person generalization, the source domain consists of data from multiple people, in

which we divide the source domain data into two parts with equal size and view one of them as the target domain to leverage DANN for domain-invariant training. As for one-person-to-another cases, we randomly sample a small number of unlabeled instances from each target person and merge them into the target set that is needed for running DANN.

BCResNet. This is a competitive benchmark for several audio-scene recognition challenges and demonstrates many useful techniques for domain generalization. BCResNet originally required mel-frequency-cepstral-coefficients but it is not suitable for time-series, hence, we use standard STFT of the multivariate-time series as input in this case.

Non-Stationary Transformer and Koopa. These are forecasting baselines that particularly address non-stationarity in short-term time sequences, Non-stationary transformer (NSTrans) [29] and Koopa [30]. To adapt it to our setting we use the encoder part of NSTrans followed by a classification head composed of fully connected layers. We simply average the encoder's output from all time steps and feed it to this classifier head.

Ours+RevIN. Further, we demonstrate that statistical techniques like Reversible Instance Normalization (RevIN) [21] may be used as a plug-and-play module with our framework. One limitation of using RevIN is that the input and output dimensions of this module must have the same dimensions to de-normalize the instance in the feature space. This may limit the usability of the module, however, we find that applying this module around the fusion encoder specifying the same number of input and output channels in the 2D convolution layer is suitable. We do not observe any significant benefit of incorporating this module from the experiments, however, if an application can specifically benefit from such RevIN, `PhASER` framework can support it.

Diversify. The goal of this design is to characterize the latent domains and use a proxy-training schema to assign pseudo-domain labels to the samples to learn generalizable representations. It is an end-to-end version of the adaptive RNN [5] method which also proposes to identify sub-domains within a domain for generalization. It is interesting to note that for time-series generalizable representation viewing the non-stationarity or intra-domain shifts is crucial. Both diversify and `PhASER` address this problem from completely different approaches and demonstrate improvement over other standard methods or even domain adaptation methods that have the advantage of accessing samples from unseen distributions. While diversify aims to characterize latent distributions and uses a parametric setting, `PhASER` forces the model to learn domain-invariant features by anchoring the design to the phase which is intricately tied to non-stationarity. It also highlights that time-series domain generalization is a unique problem (compared to the more popular visual domain) and dedicated frameworks need to be designed in this case.

MAPU. MAPU is the state-of-the-art source-free domain adaptation study for time series, thus, in fact, it does not apply to the time-series domain generalizable learning problem. However, we still view it as an effective approach that can address distribution shifts and achieve domain-invariant learning. In our implementation, in addition to the source domain data, we still provide MAPU with the unlabeled target domain data for both cross-person generalization and one-person-to-another cases. The training procedure is identical to the default MAPU design, which is to pre-train the model on labeled source domain data and then conduct the training on unlabeled target domain data.

Chronos. Large foundation models are a sought-after approach in many domains and Chronos is one such most recent candidate for time-series. It is trained on 42 datasets and presents impressive zero-shot and few-shot abilities. Although it is largely targeted as a forecasting tool, the authors indicate its universal representation ability for a variety of tasks. Four variants of Chronos model checkpoints are available ranging from 20M to 70M parameters and embedding sizes from 256 to 1024. Based on pilot testing with scenario 1 on WISDM dataset (accuracies with a 1M parameter downstream model for the three variants: tiny-0.65, base-0.41, large-0.36), we find that the smallest version of the model, Chronos-tiny best suits our conservative dataset sizes for downstream fine-tuning. We use a few layers of 2D convolution layers with max-pooling to reduce the feature size which is dependent of the length of the sequence and then flatten and input to fully-connected layers as our downstream model.

Note: A few works [17, 28] use large language models directly to analyze raw time-series despite the obvious modality gap and can report comparable performance. However, our preliminary testing with ChatGPT [39] with in-context-learning by prompting similar to Jin et. al [17] using the HHAR

dataset does not provide satisfactory results and we do not pursue that direction. Instead, we use a domain-specific large foundation model like Chronos as a fair baseline.

Table 9: Complete set of results from three trials on each baseline for WISDM cross-person generalization setting.

Baselines	Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ERM	0.57	0.02	0.50	0.02	0.51	0.02	0.55	0.02
GroupDRO	0.71	0.06	0.67	0.06	0.60	0.07	0.67	0.04
DANN	0.71	0.02	0.65	0.01	0.65	0.06	0.70	0.03
RSC	0.69	0.05	0.71	0.07	0.64	0.10	0.61	0.11
ANDMask	0.74	0.01	0.73	0.03	0.69	0.06	0.69	0.03
InceptionTime	0.83	0.01	0.82	0.02	0.80	0.04	0.77	0.01
BCResNet	0.83	0.00	0.79	0.04	0.75	0.04	0.78	0.04
NSTrans	0.43	0.02	0.40	0.01	0.37	0.02	0.37	0.03
Koopa	0.63	0.02	0.61	0.04	0.72	0.03	0.57	0.01
MAPU	0.75	0.02	0.69	0.04	0.79	0.06	0.79	0.03
Diversify	0.82	0.01	0.82	0.01	0.84	0.01	0.81	0.01
Chronos	0.71	0.01	0.67	0.01	0.65	0.01	0.62	0.01
Ours + RevIN*	0.86	0.01	0.85	0.01	0.84	0	0.84	0.03
Ours	0.86	0.01	0.85	0.01	0.85	0.01	0.82	0.02

Table 10: Complete set of results from three trials on each baseline for HHAR cross-person generalization setting.

Baselines	Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ERM	0.49	0.05	0.46	0.01	0.45	0.02	0.47	0.03
GroupDRO	0.60	0.01	0.53	0.02	0.59	0.02	0.64	0.03
DANN	0.66	0.01	0.71	0.01	0.67	0.09	0.69	0.03
RSC	0.52	0.05	0.49	0.04	0.44	0.03	0.47	0.03
ANDMask	0.63	0.02	0.64	0.06	0.66	0.11	0.69	0.05
InceptionTime	0.77	0.04	0.80	0.01	0.82	0.03	0.83	0.01
BCResNet	0.66	0.05	0.70	0.06	0.75	0.04	0.68	0.04
NSTrans	0.21	0.02	0.22	0.03	0.27	0.04	0.28	0.02
Koopa	0.72	0.04	0.63	0.03	0.72	0.05	0.69	0.02
MAPU	0.73	0.02	0.72	0.03	0.81	0.01	0.78	0.03
Diversify	0.82	0.01	0.76	0.01	0.82	0.01	0.68	0.01
Chronos	0.73	0.04	0.75	0.03	0.73	0.01	0.66	0.12
Ours + RevIN*	0.82	0.05	0.82	0.02	0.92	0.04	0.85	0.03
Ours	0.83	0.02	0.83	0.02	0.94	0.03	0.88	0.02

D.3 Ablation Details of `PhASER`

For row 1 in Table 4, the modification to `PhASER` is straightforward by simply omitted the Hilbert transformation during data preprocessing. When the separate encoders are not used (rows 6 and 7 in Table 4), we only use F_{Mag} and connect the output of the sub-feature normalization block directly to the F_{Dep}. When the residual is removed entirely (rows 5 and 6 in Table 4), we cannot broadcast the 1D input to 2D anymore so we take the mean across all the temporal indices of $F_{\text{Tem}}(\mathbf{r}_{\text{Dep}})$ and flatten it to input to fully connected layers. Based on the dataset we choose a few fully connected layers truncating to the number of classes finally.

D.4 Computational Analyses

To assess the resource utilization of `PhASER` against other baselines, we offer two metrics - 1) Number of Multiply and Accumulate operations per sample (MACs) for approximate computational complexity at run-time and 2) Number of trainable parameters to determine the memory footprint. We compute these for the HHAR dataset in Table 15 (these metrics are dependent on input dimensions, hence different choices of dataset, sequence length, and modalities can yield different numbers).

Table 11: Complete set of results from three trials on each baseline for UCIHAR cross-person generalization setting.

Baselines	Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ERM	0.72	0.09	0.64	0.05	0.70	0.01	0.72	0.03
GroupDRO	0.91	0.02	0.84	0.01	0.89	0.04	0.85	0.07
DANN	0.84	0.02	0.79	0.01	0.81	0.02	0.86	0.03
RSC	0.82	0.13	0.73	0.07	0.74	0.03	0.81	0.06
ANDMask	0.86	0.08	0.80	0.06	0.76	0.13	0.78	0.09
InceptionTime	0.91	0.03	0.82	0.07	0.88	0.02	0.91	0.04
BCResNet	0.81	0.02	0.77	0.02	0.78	0.02	0.83	0.02
NSTrans	0.35	0.02	0.35	0.01	0.51	0.02	0.47	0.01
Koopa	0.81	0.02	0.72	0.05	0.81	0.06	0.77	0.03
MAPU	0.85	0.03	0.80	0.01	0.85	0.02	0.82	0.03
Diversify	0.89	0.03	0.84	0.04	0.93	0.02	0.90	0.02
Chronos	0.56	0.05	0.57	0.01	0.50	0.02	0.82	0.13
Ours + RevIN*	0.96	0.01	0.90	0.01	0.93	0.03	0.97	0.01
Ours	0.96	0.01	0.91	0.01	0.95	0	0.97	0.01

Table 12: Complete set of results from three trials on each baseline for SSC cross-person generalization setting.

Baselines	Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ERM	0.50	0.05	0.46	0.04	0.49	0.02	0.45	0.03
GroupDRO	0.57	0.07	0.56	0.03	0.55	0.05	0.59	0.06
DANN	0.64	0.02	0.63	0.02	0.69	0.03	0.63	0.04
RSC	0.50	0.09	0.48	0.02	0.52	0.07	0.46	0.01
ANDMask	0.55	0.10	0.50	0.09	0.54	0.07	0.57	0.08
InceptionTime	0.74	0.04	0.78	0.03	0.72	0.05	0.80	0.02
BCResNet	0.79	0	0.82	0.01	0.79	0.01	0.81	0
NSTrans	0.43	0.02	0.37	0.04	0.42	0.06	0.35	0.03
Koopa	0.58	0.02	0.62	0.01	0.53	0.04	0.49	0.06
MAPU	0.69	0.01	0.68	0.01	0.65	0.03	0.69	0.02
Diversify	0.73	0.03	0.76	0.02	0.68	0.05	0.77	0.02
Chronos	0.53	0.04	0.47	0.04	0.47	0.01	0.57	0.03
Ours + RevIN*	0.82	0.01	0.79	0.02	0.78	0.01	0.81	0.01
Ours	0.85	0.01	0.80	0.01	0.79	0.01	0.83	0.01

Our computation cost is comparable to the other methods, achieving much better performance. We also determine the asymptotic time complexity of the `PhASER` modules in Table 16. For multi-layer neural network modules, the representative time complexity for one layer is provided (rows 3-7).

E Supplementary of Main Results

We conduct all experiments with three random seeds (2711, 2712, 2713), and present the error range in this section. Tables 9, 10 and 11 represent the mean and standard deviation corresponding to the main paper's Table 2 for the WISDM, HHAR and UCIHAR datasets respectively. Tables 12 and 13 are the complete representations of all the runs corresponding to Table 2 in the main paper for sleep stage classification and gesture recognition respectively. Table 14 corresponds to the Table 3 in the main paper for the complete performance statistics for one person to another generalization using HHAR dataset.

Table 13: Complete set of results from three trials on each baseline for GR cross-person generalization setting.

Baselines	Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ERM	0.45	0.02	0.58	0.03	0.57	0.03	0.54	0.04
GroupDRO	0.53	0.08	0.36	0.11	0.59	0.05	0.45	0.13
DANN	0.60	0.01	0.66	0.04	0.65	0.02	0.64	0.03
RSC	0.50	0.10	0.66	0.05	0.64	0.03	0.56	0.03
ANDMask	0.41	0.13	0.54	0.20	0.45	0.15	0.39	0.12
InceptionTime	0.68	0.07	0.70	0.09	0.72	0.03	0.69	0.02
BCResNet	0.62	0.06	0.67	0.09	0.65	0.05	0.61	0.07
NSTrans	0.31	0.01	0.34	0.01	0.34	0.01	0.32	0.02
Koopa	0.47	0.03	0.54	0.02	0.60	0.05	0.70	0.06
MAPU	0.64	0.02	0.69	0.03	0.71	0.01	0.68	0.04
Diversify	0.69	0.01	0.80	0.01	0.76	0.02	0.76	0.01
Chronos	0.49	0.01	0.54	0.03	0.51	0.05	0.48	0.02
Ours + RevIN*	0.68	0.03	0.81	0.04	0.77	0.03	0.76	0.02
Ours	0.70	0.02	0.82	0.02	0.77	0.04	0.75	0.01

Table 14: Complete set of results from three trials on each baseline for HHAR one-person-to-another setting.

Baselines	0		1		2		3		4		5		6		7		8	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ERM	0.27	0.01	0.40	0.05	0.41	0.05	0.44	0.05	0.42	0.08	0.44	0.01	0.45	0.04	0.44	0.04	0.48	0.02
GroupDRO	0.33	0.02	0.53	0.02	0.38	0.05	0.48	0.04	0.47	0.04	0.51	0.08	0.47	0.03	0.48	0.02	0.49	0.05
DANN	0.32	0.03	0.44	0.05	0.42	0.03	0.45	0.06	0.42	0.03	0.48	0.04	0.49	0.02	0.45	0.05	0.51	0.01
RSC	0.27	0.03	0.45	0.06	0.38	0.05	0.45	0.09	0.40	0.08	0.47	0.02	0.50	0.06	0.44	0.08	0.53	0.01
ANDMask	0.34	0.06	0.50	0.03	0.37	0.04	0.43	0.05	0.46	0.04	0.51	0.07	0.46	0.03	0.47	0.02	0.52	0.03
InceptionTime	0.52	0.05	0.62	0.02	0.44	0.03	0.69	0.04	0.60	0.09	0.57	0.05	0.66	0.03	0.64	0.01	0.61	0.01
BCResNet	0.28	0.03	0.48	0.08	0.32	0.04	0.47	0.03	0.42	0.06	0.52	0.05	0.44	0.02	0.45	0.02	0.49	0.06
NSTrans	0.20	0.01	0.22	0.02	0.17	0.02	0.20	0.01	0.21	0.01	0.22	0.01	0.26	0.07	0.17	0.05	0.20	0.01
Koopa	0.32	0.02	0.42	0.04	0.37	0.01	0.40	0.01	0.42	0.02	0.45	0.05	0.35	0.02	0.43	0.03	0.48	0.02
MAPU	0.39	0.05	0.57	0.05	0.35	0.06	0.52	0.03	0.49	0.04	0.54	0.02	0.49	0.01	0.50	0.06	0.52	0.04
Diversify	0.42	0.04	0.62	0.04	0.32	0.09	0.62	0.01	0.56	0.03	0.61	0.01	0.53	0.04	0.52	0.10	0.61	0.05
Chronos	0.32	0.03	0.23	0.05	0.26	0.04	0.25	0.03	0.27	0.09	0.23	0.08	0.24	0.06	0.21	0.08	0.24	0.05
Ours + RevIN*	0.48	0.02	0.66	0.08	0.57	0.05	0.65	0.03	0.61	0.04	0.64	0.05	0.65	0.06	0.64	0.01	0.63	0.03
Ours	0.53	0.04	0.70	0.03	0.63	0.01	0.66	0.03	0.64	0.06	0.67	0.01	0.65	0.03	0.67	0.04	0.62	0.02

Table 15: Model comparison based on MACs and number of trainable parameters.

Model	MACs ($\times 10^6$)	Trainable Parameters ($\times 10^3$)
ERM	19.5	98.1
GroupDRO	19.5	98.1
DANN	21.7	102.9
RSC	19.5	98.1
ANDMask	19.5	98.1
BCResNet	55.3	154.7
NSTrans	35.3	75.6
Koopa	32.7	118.7
MAPU	46.9	128.3
Diversify	35.7	922.9
Chronos	345.5	1049.8
Ours	48.6	81.4

Table 16: Complexity per module and input notation for each module.

	Module	Complexity
1	Hilbert augmentation (using Fast-Fourier transform)	$\mathcal{O}(V \cdot N \log N)$
2	Short-Term Fourier Transform	$\mathcal{O}(V \cdot N \cdot W \log W)$
3	Magnitude Encoder (F_{Mag}), Phase Encoder (F_{Pha}), Phase Projection Head (g_{Res}) - 2D Convolution Layers	$\mathcal{O}(k^2 \cdot N \cdot d \cdot c_{in} \cdot c_{out})$
4	Depthwise Feature Encoder (F_{Dep}) - 2D Convolution Layers with average pooling along feature axis	$\mathcal{O}(k^2 \cdot N \cdot d \cdot c_{in} \cdot c_{out}) + \mathcal{O}(d)$
5	Temporal Encoder (F_{Tem}) - (worst case backbone) Transformer Encoder	$\mathcal{O}(N \cdot d)$
6	Classification Encoder (g_{Cls}) - fully connected layers	$\mathcal{O}(d \cdot h)$