SPECTRAL AND GRADIENT ALIGNMENT FOR TIME-SERIES ADVERSARIAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks (DNNs) are vulnerable to small adversarial perturbations, raising concerns in high-stakes domains such as healthcare. Adversarial training (AT) is a standard defense but can reduce clean-test performance. For time series, norm-bounded perturbations can concentrate energy in frequencies weakly represented in the input, providing a poor training prior. We propose an adaptive filtering method that regularizes adversarial perturbations toward each input's empirical spectral distribution without application-specific frequency cutoffs. This distribution provides a data-dependent prior rather than a model of sensor bandwidth or all plausible acquisition noise. Across nine datasets spanning five health-related applications, our method improves clean-test performance over baselines while maintaining competitive robustness.

1 INTRODUCTION

Deep learning models are used in various health-related time series tasks, including cardiovascular health Al-Zaiti et al. (2023), activity recognition (Wang et al., 2019), and sleep analysis (Sun et al., 2019). These applications provide insights into individuals' health (Sandvik et al., 1995). Deep learning has achieved clinician-level performance on several tasks involving large-scale temporal data (Hannun et al., 2019; Ganglberger et al., 2024; Bayoumy et al., 2021). This advancement is expected to pave the way for broader adoption across domains by enabling faster, more cost-effective, and accessible solutions (Rajpurkar et al., 2022; Tran et al., 2019).

However, the performance of deep learning models degrades when data is perturbed by small noise (Szegedy et al., 2014). Even small Gaussian noise can lead to changes in model predictions (Gilmer et al., 2019), leading to misclassifications. This vulnerability raises concerns about the reliability and trustworthiness of the models and poses an obstacle to their widespread adoption in healthcare applications (Feeny et al., 2020; Chen et al., 2022a).

Adversarial training (AT) is a standard approach for improving robustness by augmenting data with adversarial perturbations (Goodfellow et al., 2015; Madry et al., 2018). However, it can reduce accuracy on clean test data (Yang et al., 2020b; Wang et al., 2022), creating a tradeoff that is problematic in healthcare, where performance on real data is critical. The structure of these perturbations is an important consideration, since norm constraints limit their magnitude without controlling their spectral content. Conventional attacks can introduce physiologically implausible artifacts (Han et al., 2020; Pialla et al., 2022). Prior work further shows that the frequency bands emphasized during AT affect the clean–robustness tradeoff (Maiya et al., 2023).

We propose an adaptive filtering method for generating adversarial perturbations with spectral alignment to each input. Our theoretical analysis identifies the phase that minimizes angular deviation for a given magnitude response, avoiding expensive optimization procedures for phase selection. Across applications, our method improves clean-test generalization over prior approaches and remains competitive under the evaluated adversarial attacks. Our main contributions are as follows.

- We propose an efficient adaptive filtering method for generating adversarial perturbations regularized toward each input's spectral distribution without requiring hand-specified application-specific frequency cutoffs.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- We provide theoretical and empirical analysis showing that the proposed method reduces perturbation energy in harmonics weakly represented in the input and avoids the extreme bandwidth expansion induced by prior methods.
- We evaluate the method across five health-related applications, showing improved clean-test generalization and adversarial robustness, and conduct an assessment with two physicians that provides evidence of the physiological plausibility of the perturbed signals.

2 METHOD

2.1 NOTATION

Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$ denote a dataset, where $\boldsymbol{x}_i \in \mathbb{R}^{K \times L}$ is a time series with K channels and length L, and $y_i \in \{1, \dots, C\}$ is its label. We use lowercase letters (x) for scalars, bold lowercase letters ($\boldsymbol{x}$) for vectors or sequences, and uppercase letters (X) for random variables. A neural network with parameters θ is denoted by $f_\theta(\cdot)$. We define spectra of a sequence over the frequency bins Ω_L. We write $\mathcal{F}(\boldsymbol{x}) = \widehat{\boldsymbol{x}}(\omega) \in \mathbb{C}$ for the Fourier coefficient of $\boldsymbol{x}$ at frequency bin $\omega \in \Omega_L$. We use $\|\cdot\|$ for a generic norm, including ℓ_2 and ℓ_∞. Additional notation is provided in Appendix A.1.

2.2 OBJECTIVE

Our goal is to improve clean-test generalization while preserving robustness to perturbations that remain consistent with the observed signal. To formalize this motivation, we introduce an idealized perturbation set that restricts both norm and spectral deviation.

$$\mathcal{P}(\boldsymbol{x}) = \left\{ \delta \;\middle|\; \|\delta\| \le \epsilon,\; D_{\mathrm{KL}}\left(\frac{|\mathcal{F}(\delta)|}{\|\mathcal{F}(\delta)\|_1} \,\middle\|\, \frac{|\mathcal{F}(\boldsymbol{x})|}{\|\mathcal{F}(\boldsymbol{x})\|_1} \right) \le \kappa \right\}, \tag{1}$$

where κ controls the allowed deviation between normalized magnitude spectra. This set defines the following spectrally constrained robust objective.

$$\min_{\theta}\; \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\delta \in \mathcal{P}(\boldsymbol{x})} \mathcal{L}(f_\theta(\boldsymbol{x}+\delta), y) \right]. \tag{2}$$

Unlike a norm constraint alone, Equation 1 restricts the spectral distribution of the perturbation. Norm-bounded perturbations can introduce frequency components weakly represented in the input, changing the spectral structure of adversarial sample. The spectral restriction uses the observed input spectrum as a prior without assuming that such components are necessarily physically implausible or absent from acquisition noise.

The challenge is that spectral structure is input-dependent and cannot be captured by fixed filters across applications. We therefore introduce an adaptive filter $h_\psi(\cdot; \boldsymbol{x})$ that learns task-driven spectral weights while encouraging alignment with the input. We next describe how this filter is learned.

2.2.1 LEARNING THE SPECTRAL FILTER

Rather than imposing a fixed low-pass or band-pass filter, we learn spectral weights adaptively from each input. Let $q_{\boldsymbol{x}}(\omega) = |\widehat{\boldsymbol{x}}(\omega)| / \sum_{\nu \in \Omega_L} |\widehat{\boldsymbol{x}}(\nu)|$ denote the normalized magnitude spectrum of the input. We feed $q_{\boldsymbol{x}}$ to a filter network G_ψ, which outputs a magnitude response

$$|H_\psi(\omega; \boldsymbol{x})| = G_\psi(q_{\boldsymbol{x}})(\omega), \qquad |H_\psi(\omega; \boldsymbol{x})| \ge 0.$$

The filtered perturbation spectrum is $\widehat{\delta}_\psi(\omega) = \widehat{\delta}(\omega)\,|H_\psi(\omega; \boldsymbol{x})|$. We do not set the filter response directly from the input spectrum. Instead, the network receives the input spectrum as conditioning and learns task-driven weights across frequencies. To encourage spectral alignment, we penalize the divergence between the magnitude spectra of the perturbation and the input as in Equation 3.

$$\mathcal{L}_{\mathrm{spec}}(\psi) = D_{\mathrm{KL}}(p_\psi \,\|\, q_{\boldsymbol{x}}) \quad \text{where } p_\psi(\omega) = \frac{|\widehat{\delta}_\psi(\omega)|}{\sum_\nu |\widehat{\delta}_\psi(\nu)|}, \text{ and } \; q_{\boldsymbol{x}}(\omega) = \frac{|\widehat{\boldsymbol{x}}(\omega)|}{\sum_\nu |\widehat{\boldsymbol{x}}(\nu)|}. \tag{3}$$

The KL term penalizes deviations between these normalized magnitude spectra. The following proposition bounds differences in spectral mass over any frequency region.

Proposition 2.1 (Spectral mass control). *Let p_ψ and $q_{\boldsymbol{x}}$ be the normalized magnitude spectra defined in Equation 3. For any frequency region $A \subseteq \Omega_L$,*

$$\left| \sum_{\omega \in A} p_\psi(\omega) - \sum_{\omega \in A} q_{\boldsymbol{x}}(\omega) \right| \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p_\psi \| q_{\boldsymbol{x}})}. \tag{4}$$

Thus, if $D_{\mathrm{KL}}(p_\psi \| q_{\boldsymbol{x}}) \leq \kappa$, the spectral mass assigned to any frequency differs by at most $\sqrt{\kappa/2}$.

Proof. For probability distributions, the mass difference over any subset is bounded by their total variation distance. Pinsker's inequality, with natural logarithms, gives

$$\left| \sum_{\omega \in A} \big(p_\psi(\omega) - q_{\boldsymbol{x}}(\omega)\big) \right| \leq \frac{1}{2} \|p_\psi - q_{\boldsymbol{x}}\|_1 \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p_\psi \| q_{\boldsymbol{x}})}.$$

Substituting the upper bound κ yields the stated result. □

Proposition 2.1 bounds the difference between the normalized spectral mass of the perturbation and input over any frequency region. The penalty therefore encourages spectral alignment, while the learned input-conditioned filter uses downstream task supervision to determine the weights. Appendix E.3 provides empirical evidence for spectral concentration and effective support. We train the filter by minimizing the combined adversarial and spectral objective as in Equation 5.

$$\mathcal{L}_\psi = -\mathcal{L}_{\mathrm{CE}}\left(f_\theta(\boldsymbol{x} + \Delta_\psi), y\right) + \lambda_{\mathrm{KL}} D_{\mathrm{KL}}\left(p_\psi \,\|\, q_{\boldsymbol{x}}\right). \tag{5}$$

Here, Δ_ψ is the total perturbation produced by the iterative filtered updates under the norm bound. The negative cross-entropy term encourages perturbations that increase the classifier's loss, while the KL term penalizes differences between the normalized magnitude spectra of the filtered perturbation and input. The spectral penalty does not specify the filter's phase (Oppenheim et al., 1996). We choose the phase to keep each filtered update as aligned as possible with the original gradient direction.

2.2.2 Direction Under Constraints

The direction of the perturbation is central to worst-case error (Goodfellow et al., 2015) where reversing the direction decreases the loss rather than increasing it. Accordingly, after determining the magnitude response, we choose the phase response so that the filtered perturbation δ_ψ deviates as little as possible from the original perturbation δ in angle. The following theorem shows that, for a fixed magnitude response, the minimum-angle solution is obtained by a zero-phase filter.

Theorem 2.2 (Minimum-angle perturbation). *Let $\delta_\psi \neq 0$ be obtained by filtering a real perturbation $\delta \neq 0$ with a filter h_ψ whose magnitude response $|H_\psi(\omega)|$ is fixed and even. Then the angle between δ and δ_ψ is minimized by zero phase, i.e.,*

$$\phi_H^* \in \arg\min_{\phi_H} \arccos\left(\frac{\langle \delta, \delta_\psi \rangle}{\|\delta\|_2 \, \|\delta_\psi\|_2}\right), \qquad \phi_H^*(\omega) = 0.$$

where $\phi_H(\omega) = \angle H_\psi(\omega)$ and the minimization is over phases satisfying conjugate symmetry.

Proof. For fixed $|H_\psi(\omega; \boldsymbol{x})|$, the filtered norm is independent of phase. By Parseval's theorem, minimizing the angle therefore amounts to maximizing $\sum_{\omega \in \Omega_L} |\widehat{\delta}(\omega)|^2 |H_\psi(\omega; \boldsymbol{x})| \cos(\angle H_\psi(\omega; \boldsymbol{x}))$. Every weight is nonnegative, so zero phase maximizes each term. The fixed even magnitude response ensures that this choice preserves real-valued outputs. □

Figure 1 illustrates this result, and Appendix A provides the full proof. We therefore learn the magnitude response using Equation 5 and use a zero-phase filter throughout. Proposition 2.1 describes the spectral alignment encouraged by the KL penalty, while Theorem 2.2 justifies the phase choice.

Our filter h_ψ aligns the spectrum of perturbation with the input while learning a task-driven redistribution of mass. Standard attacks apply elementwise sign updates and clipping to an input box. The sign operation is discontinuous, while clipping is nonlinear; both can introduce broadband

harmonics (Oppenheim et al., 1996; Karras et al., 2021). To avoid these distortions, we omit the sign transform and clipping, and instead generate perturbations by filtering the gradient as in Equation 6.

$$g = \nabla_{\boldsymbol{x}} \mathcal{L}(f_\theta(\boldsymbol{x}), y), \qquad \widetilde{\delta} = g * h_\psi, \qquad \mathcal{F}(h_\psi)(\omega) = H_\psi(\omega) = |H_\psi(\omega)|. \tag{6}$$

Unlike images, temporal sequences do not share a universal input box, since input scaling varies across applications, including z-score (Demirel & Holz, 2023; Temko, 2017), min–max scaling, and in some cases no normalization at all (Kiyasseh et al., 2020). Moreover, different channels can have dynamic ranges, so the same clipping in normalized coordinates can correspond to different perturbation magnitudes in the signal space. Thus, a universal box-clipping is not naturally defined across tasks and may reflect preprocessing choices rather than constraints of the underlying measurement process.

Instead, we control perturbation magnitude relative to the input norm as in Equation 7.

$$\delta_\psi = \alpha(\|\boldsymbol{x}\|_2 / \|\widetilde{\delta}\|_2)\widetilde{\delta}, \tag{7}$$

where α controls the relative update strength before projection, while the final perturbation is constrained by ϵ. For $\widetilde{\delta} \neq 0$, this scaling gives $\|\delta_\psi\|_2 / \|\boldsymbol{x}\|_2 = \alpha$ before projection and preserves the update direction without applying sign or box clipping. When the update exceeds the perturbation budget, projection determines its magnitude.

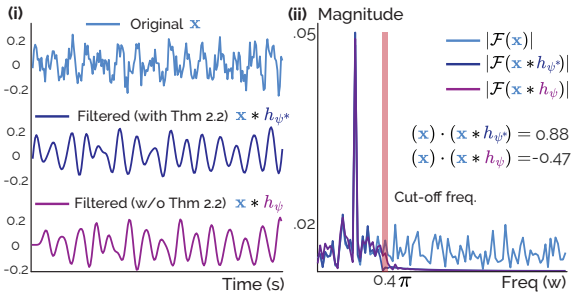


Figure 1: **(a)** A sequence and filtered versions obtained with zero and nonzero phase. **(b)** Their spectra and dot products with $\boldsymbol{x}$. The filtered vectors have the same magnitude spectrum, but their alignments differ.

Our method generates the augmented sample by iterative gradient ascent as in Equation 8.

$$\delta_\psi^t = \alpha\|\boldsymbol{x}\|_2 \frac{\nabla_{\boldsymbol{x}_{\text{adv}}^t} \mathcal{L}(f_\theta(\boldsymbol{x}_{\text{adv}}^t), y) * h_\psi}{\left\| \nabla_{\boldsymbol{x}_{\text{adv}}^t} \mathcal{L}(f_\theta(\boldsymbol{x}_{\text{adv}}^t), y) * h_\psi \right\|_2}, \quad \boldsymbol{x}_{\text{adv}}^{t+1} = \boldsymbol{x}_{\text{adv}}^t + \eta\, \delta_\psi^t, \quad \text{and} \quad \boldsymbol{x}_{\text{adv}}^0 = \boldsymbol{x}, \tag{8}$$

where η is the step size. The perturbation $\boldsymbol{x}_{\text{adv}}^T - \boldsymbol{x}$ is also rescaled whenever its ℓ_2 norm exceeds ϵ to satisfy the bound. The pseudocode of the overall method is provided in Appendix B.

2.2.3 Overall Mechanism

Adversarial training minimizes worst-case loss under norm-constrained perturbations (Madry et al., 2018). However, norm constraints control perturbation magnitude without specifying how it is distributed across frequencies. Moreover, frequency-domain analyses show that the spectral allocation of perturbations affects accuracy and robustness, with different effects across datasets (Maiya et al., 2023). These findings motivate using the observed input spectrum to guide perturbation generation.

For each sample $\boldsymbol{x}$, Equation 1 defines the perturbed-input set as

$$\mathcal{M}(\boldsymbol{x}) = \{\boldsymbol{x} + \delta \mid \delta \in \mathcal{P}(\boldsymbol{x})\}. \tag{9}$$

This set expresses the desired combination of bounded magnitude and spectral alignment. Our method enforces the norm bound by projection and encourages spectral alignment through a KL penalty rather than a prescribed threshold κ; Appendix E.2.1 and E.3 present evidence to motivate the choice.

Figure 2 illustrates this mechanism. Compared with prior on-manifold generation methods and heuristic low-frequency perturbation approaches, our formulation has three practical advantages:

1.The admissible set is defined in the input space, enabling direct analysis of perturbations, including the phase selection in Theorem 2.2.

2.Our approach does not require training a data-hungry generative model to constrain adversarial perturbations.

3.The constraint is estimated from each observed input, avoiding application-specific frequency cut-offs and specialized architectures.

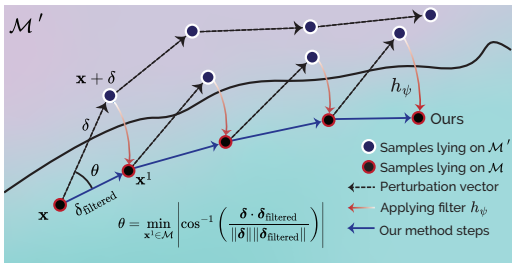


Figure 2: A norm-bounded gradient update can leave the idealized input-dependent admissible set $\mathcal{M}(\boldsymbol{x})$, while the learned filter h_ψ encourages spectral alignment with the observed input.

3 EXPERIMENTS

Datasets. We conducted experiments on nine datasets across five health-related tasks, including heart rate (HR) estimation from photoplethysmography (PPG), step counting and activity recognition using inertial measurements (IMUs), cardiovascular disease classification from electrocardiogram (ECG) and sleep stage classification from electroencephalography (EEG). We provide short descriptions of each dataset below, and further details can be found in Appendix C.1.

Heart rate We used IEEE Signal Processing Cup (IEEE SPC) (Zhang et al., 2015), and DaLia (Reiss et al., 2019) for HR prediction. We used the leave-one-session-out (LOSO) cross-validation.

Activity recognition We used UCIHAR (Anguita et al., 2012), and USC (Zhang & Sawchuk, 2012) for activity recognition from inertial measurement units. We evaluated the cross-person generalization performance of the models, that is, the model was evaluated on previously unseen subjects.

Cardiovascular disease (CVD) classification We used Chapman University, Shaoxing People's Hospital ECG (Zheng et al., 2020) and PhysioNet 2017 (Clifford et al., 2017; Goldberger et al., 2000). We selected the same four leads for the Chapman as in Alday et al. (2020). We split the data into training, validation, and testing according to the patient ID (each patient's recordings appear in only one set) using a 60, 20, 20 ratio as in Zheng et al. (2020).

Step counting We used Clemson (Mattfeld et al., 2017), which was released for pedometer evaluation. We conducted our experiments using wrist IMUs where labels are available through videos.

Sleep stage classification We used the Sleep-EDF, from PhysioBank (Goldberger et al., 2000), which includes whole-night PSGs, where we used the Fpz-Cz EEG channel with a sampling rate of 100 Hz.

Baselines. We compare our method against several approaches, including FGSM (Goodfellow et al., 2015) and PGD (Madry et al., 2018), as well as smoothing methods, like SGM (Pialla et al., 2022), SAP (Han et al., 2020), and LF-AP (Guo et al., 2020), which are designed to mitigate high-frequency harmonics. Additionally, we evaluate techniques that focus on clean accuracy, such as Frequency adversarial attack (FA) (Jia et al., 2022), f-VAT (Pan & Li, 2025), IFAP (Yoshida & Okuda, 2025), TRADES (Zhang et al., 2019) and DARG (Stutz et al., 2019). We also include F-SAT (Zhang et al., 2025), which applies frequency-selective AT. Finally, we include data augmentations. Details are given in Appendix C.3.

Implementation We employ ResNet (He et al., 2015) for time series (Hong et al., 2020). The filter h_ψ is a three-layer fully connected network with output $|\mathbf{H}_\psi(\omega)|$, applied independently to each channel for multivariate signals. We optimize both with Adam (Kingma & Ba, 2015); the main network uses categorical cross-entropy, while the filter additionally uses the KL term in Eq. 3 with unit weight. During each epoch, after the five epochs when h_ψ is used for stabilization, we randomly select a quarter of the batch to generate adversarial samples for all methods. The learning rate is halved if the validation loss does not improve for 15 epochs, and training is stopped after 90 such epochs. We select the model using validation loss. When spectral content varies over time, we can apply our method to shorter windows additional to the full sequence. This yields a localized variant, denoted as window. Hyperparameters are provided in Appendix C.4.

Table 1: Performance comparison of methods on HR estimation

Method	IEEE SPC12			IEEE SPC22			DaLiA		
	MAE↓	RMSE↓	ρ↑	MAE↓	RMSE↓	$\rho\uparrow$	MAE↓	RMSE↓	ρ↑
Standard training									
Only $\mathcal{D}_{\text{real}}$	5.27±0.21	11.47±0.69	83.86±1.31	5.76±0.07	10.70±0.18	81.16±0.43	5.53±0.067	12.47±0.17	85.65±0.20
Standard techniques									
Aug.	5.54±0.22	11.63±0.84	84.38±1.10	5.31±0.31	10.77±0.64	81.32±0.81	5.52±0.04	12.46±0.09	85.67±0.15
FGSM	5.59±0.22	12.03±0.74	83.82±0.65	5.45±0.11	10.57±0.22	82.70±0.35	5.43±0.08	12.26±0.10	86.04±0.07
PGD	5.97±0.67	12.71±1.15	81.25±2.20	6.19±0.41	11.71±0.65	80.79±1.11	5.62±0.03	12.45±0.12	85.50±0.05
Smoothing (low-pass)									
SAP	6.68±0.59	13.33±0.89	82.07±1.54	5.41±0.55	10.80±1.00	82.38±1.01	5.40±0.08	12.25±0.18	86.05±0.25
SGM	5.85±0.21	12.06±0.19	83.35±0.90	5.29±0.26	10.65±1.40	81.35±1.72	5.68±0.04	12.63±0.10	84.84±0.02
LF-AP	6.13±0.18	12.87±0.22	83.02±0.87	5.37±0.28	10.87±1.43	81.69±1.73	5.65±0.08	12.47±0.13	85.12±0.09
Clean accuracy-focused									
TRADES	6.18±0.50	12.50±0.66	82.22±0.61	5.98±0.35	11.43±0.91	80.70±1.42	5.54±0.10	12.43±0.16	85.52±0.30
f-VAT	5.35±0.22	11.13±0.90	84.70±1.30	8.83±0.88	17.42±2.33	64.90±8.50	5.91±0.51	13.11±0.14	84.91±0.19
FA	6.01±0.24	12.25±0.23	82.07±0.27	6.55±0.48	12.58±0.37	76.80±2.30	6.60±0.39	14.70±0.58	80.38±0.77
F-SAT	5.69±0.32	12.19±0.60	82.84±0.68	6.80±0.17	13.17±0.61	77.87±1.29	5.65±0.05	12.30±0.08	85.90±0.03
DARG	6.07±0.27	12.68±0.32	82.01±0.40	5.28±0.27	10.89±1.23	81.22±1.57	5.38±0.07	12.15±0.14	86.20±0.32
IFAP	5.71±0.30	11.62±0.31	83.70±0.39	5.04±0.15	15.26±0.53	74.30±1.10	5.55±0.05	12.68±0.07	84.40±0.10
Ours	**5.19**±0.10	**10.59**±0.50	**85.47**±0.60	**4.77**±0.98	**9.76**±1.50	**85.90**±1.79	**5.23**±0.09	**11.85**±0.25	**86.73**±0.50
Ours (window)	5.32±0.13	10.65±0.53	85.17±0.59	4.93±0.94	10.04±1.33	84.37±1.55	5.22±0.08	11.93±0.23	86.75±0.52

Table 2: Performance comparison of methods on Activity and Step datasets

Method	UCIHAR			USC			Clemson		
	Acc↑	W-F1↑	F1↑	Acc↑	W-F1↑	F1 ↑	MAPE↓	MAE↓	RMSE↓
Standard training									
Only $\mathcal{D}_{\text{real}}$	89.89±1.76	88.08±1.80	89.37±1.82	73.49±1.59	72.22±1.74	71.04±1.43	4.04±0.30	2.28±0.21	3.50±0.23
Standard techniques									
Aug.	89.60±0.95	88.22±1.08	89.30±1.13	73.65±1.47	72.10±1.73	71.31±1.37	4.60±0.60	2.58±0.32	3.93±0.45
FGSM	90.07±1.12	88.75±1.17	89.54±1.47	74.20±1.10	72.51±1.75	72.34±1.34	3.62±0.34	2.02±0.21	3.17±0.24
PGD	89.18±3.43	87.92±4.05	89.42±3.35	75.63±0.74	74.07±1.12	73.57±0.96	4.47±1.44	2.53±0.87	3.84±1.09
Smoothing (low-pass)									
SAP	91.79±0.95	90.91±0.96	92.01±0.81	76.09±0.81	74.82±1.03	73.79±1.12	4.34±0.35	2.44±0.19	3.70±0.21
SGM	88.87±3.41	87.00±4.83	88.14±4.43	70.70±1.06	69.20±0.78	68.53±1.29	4.47±0.42	2.53±0.26	3.73±0.23
LF-AP	86.18±2.23	86.65±0.21	87.12±1.98	72.65±0.86	72.18±1.65	73.16±1.65	4.12±0.43	2.68±0.13	3.75±0.03
Clean accuracy-focused									
TRADES	91.49±0.73	90.90±0.55	92.13±0.37	76.50±0.75	75.18±0.90	74.57±0.83	3.30±0.31	1.84±0.22	2.96±0.18
f-VAT	90.29±1.13	88.97±1.20	89.96±1.59	75.22±1.07	74.15±1.29	73.04±1.42	4.69±0.68	2.70±0.46	3.78±0.43
FA	87.68±0.57	87.71±1.08	86.21±0.45	73.56±0.62	72.95±0.81	73.17±0.91	3.80±0.47	2.77±0.21	3.77±0.21
F-SAT	88.89±1.93	86.94±2.53	88.86±2.11	75.96±0.85	74.42±1.26	73.76±0.71	4.25±0.57	2.43±0.39	3.56±0.41
DARG	89.60±0.95	88.22±1.08	89.30±1.13	74.65±2.21	73.37±2.43	72.66±2.06	4.62±0.18	2.62±0.10	3.87±0.15
IFAP	89.84±0.92	88.12±0.99	90.03±1.05	75.83±0.83	73.81±0.81	73.31±1.14	4.84±0.10	2.82±0.17	4.12±0.18
Ours	**92.89**±1.05	**92.09**±1.12	**93.14**±1.04	**77.49**±1.42	**76.18**±1.23	**75.58**±1.09	**3.08**±0.33	**1.70**±0.20	**2.91**±0.23
Ours (window)	92.90±0.93	91.98±1.03	93.17±1.00	77.52±1.13	76.20±1.10	75.63±1.04	3.36±0.27	1.83±0.16	3.13±0.20

4 RESULTS

Our method improves clean-test performance over baselines. The windowed variant is effective on EEG and ECG, where support can vary over time and is better captured within shorter segments. For the main comparison, we use the full-sequence variant as the default because it is more efficient while remaining consistently competitive. Tables show a clear limitation of fixed smoothing-based baselines on datasets containing higher-frequencies. For example, SAP, which uses a hand-crafted low-pass on perturbations (Han et al., 2020), performs poorly. Similarly, SGM, which enforces smoothness through a lasso penalty (Pialla et al., 2022), reduces clean performance in most datasets. These results suggest that imposing generic structure on perturbations, without adapting to the data, can remove task-relevant information and degrade generalization. In contrast, our method adapts the perturbation support, which leads to stronger and consistent performance. We assess significance across tasks using ranks, followed by Friedman test. Our method obtains the best average rank and is significantly better than the baselines. The results are given in Appendix D.1.

An interesting result is that standard FGSM and PGD occasionally outperform methods designed to reduce clean-data degradation, such as TRADES (Zhang et al., 2019) and DARG (Stutz et al., 2019). For example, DARG, which generates perturbations using a latent data manifold, performs worse on some smaller datasets. An explanation can be that accurately learning such a distribution is difficult in limited-data, especially in high dimensions. When the approximation is poor, the perturbations are less consistent with the observed signal, which can reduce clean-test performance.

Similarly, TRADES optimizes a regularized objective but relies on generic PGD perturbations, which may remain misaligned with the measured signal structure. Our method instead encourages alignment between the perturbation and input spectra through learned, input-conditioned filtering. It is also complementary to objective-level approaches such as TRADES and could be integrated into modified loss formulations to generate more plausible perturbations.

Table 3: Performance comparison of methods on CVD and Sleep datasets

Method	Chapman			PhysioNet 2017			Sleep-EDF		
	Acc↑	F1↑	AUC ↑	Acc↑	F1↑	AUC ↑	Acc↑	F1↑	κ ↑
Standard training									
Only $\mathcal{D}_{\text{real}}$	92.56±0.25	91.58±0.26	98.59±0.12	82.71±1.55	73.16±3.39	93.00±0.30	82.11±0.88	80.28±1.03	76.16±1.12
Standard techniques									
Aug.	92.43±0.05	91.43±0.08	98.66±0.13	83.79±0.38	**74.47**±2.25	93.55±0.22	81.75±1.43	79.81±0.48	75.60±1.75
FGSM	92.79±0.08	91.83±0.09	98.75±0.14	82.83±2.14	73.08±3.50	93.25±0.43	82.00±0.13	80.88±1.20	75.84±0.27
PGD	**93.28**±0.42	**92.36**±0.49	**98.83**±0.16	83.85±0.30	73.78±0.17	93.50±0.25	82.29±1.12	81.40±0.49	76.14±1.30
Smoothing (low-pass)									
SAP	92.41±0.43	91.43±0.45	98.64±0.16	83.57±0.75	72.99±2.01	93.27±0.20	78.33±0.86	77.14±0.94	70.82±1.06
SGM	92.94±0.09	92.02±0.14	98.60±0.18	83.10±0.45	73.90±1.51	93.34±0.09	80.75±1.67	79.71±2.32	74.37±2.03
LF-AP	91.87±0.25	91.18±0.59	98.02±1.03	82.87±0.48	73.16±2.25	92.65±0.34	79.32±1.53	76.32±2.01	72.66±1.18
Clean accuracy-focused									
TRADES	91.99±0.22	90.89±0.29	98.49±0.07	81.10±0.87	70.11±1.40	92.87±0.27	82.46±0.82	79.91±0.67	76.02±1.19
f-VAT	92.19±0.29	91.14±0.38	98.30±0.30	81.66±0.56	71.29±0.46	92.80±0.60	81.77±1.52	80.12±2.10	75.60±2.20
FA	90.10±0.25	89.81±0.23	97.81±0.09	80.82±0.91	68.33±1.73	91.38±0.21	81.30±0.81	79.13±0.88	75.73±1.21
F-SAT	91.57±0.34	90.43±0.31	98.49±0.20	82.36±2.28	73.37±3.00	93.11±1.05	77.11±4.31	77.02±2.28	69.25±0.55
DARG	92.20±0.43	91.15±0.45	98.59±0.01	83.58±1.30	73.89±3.45	93.15±0.79	82.20±1.80	80.43±1.30	76.04±2.10
IFAP	91.83±0.31	89.60±0.26	97.40±0.07	80.79±1.50	70.67±1.35	91.30±0.31	81.67±1.09	78.99±0.90	76.23±1.02
Ours	92.88±0.12	91.95±0.08	98.79±0.14	**84.10**±0.49	73.70±1.10	**93.57**±0.23	**83.02**±0.86	**81.53**±1.10	**77.50**±1.23
Ours (window)	92.87±0.23	92.10±0.10	98.80±0.15	84.83±0.40	74.10±1.03	93.62±0.22	83.10±0.73	81.60±0.99	77.52±1.44

Figure 3 compares the input and perturbation spectra. PGD spreads perturbation energy across regions containing little input spectral mass. Our method instead aligns the perturbation with the empirical distribution of the input. DARG shows similar behaviors as it aims for generating perturbations near the learned manifold. We further examine the role of spectral allocation by varying the perturbation energy outside the input's support in Appendix E.4. The results favor allocations concentrated within the input's support.

Other modalities: We also evaluate on audio and financial sequences to test generalization under different modalities. Details are reported in Appendix D.9 due to space constraints.

For robustness, we report two settings in Table 4 because attacks without spectral restrictions can introduce components weak or absent in the input spectra. We report tradeoff between clean accuracy and robustness in Figure 4.

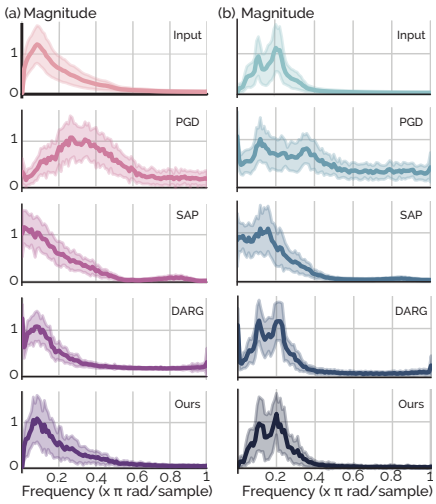


Figure 3: Qualitative comparison of input and perturbation spectra on (a) DaLiA and (b) IEEE SPC; more examples are in Appendix E.

Table 4: Performance under perturbations, i.e., PGD and Carlini & Wagner (2017), while f-PGD/CW denote frequency-constrained variants. Bold marks the strongest defense for dataset-evaluation pairs.

Dataset	Only $\mathcal{D}_{\text{real}}$					TRADES					DARG					Ours				
Def. / Att.	Clean	ℓ_2	f-PGD	CW	f-CW	Clean	ℓ_2	f-PGD	CW	f-CW	Clean	ℓ_2	f-PGD	CW	f-CW	Clean	ℓ_2	f-PGD	CW	f-CW
UCIHAR	89.9	74.6	80.3	68.9	80.0	91.5	71.2	77.8	67.5	80.1	89.6	60.8	73.7	**69.3**	82.4	**92.9**	**77.6**	**86.1**	68.4	**85.4**
Chapman	92.6	58.7	80.8	59.2	73.5	92.0	**75.0**	85.9	63.1	80.2	92.2	65.2	84.0	**64.7**	81.3	92.9	70.0	**87.7**	62.0	**83.6**
Physio 2017	82.7	59.5	55.8	60.6	64.8	81.1	**71.5**	68.4	**63.7**	69.3	83.6	64.2	68.9	61.3	72.5	**84.1**	66.1	**76.7**	62.5	**75.8**
DaLiA	85.7	60.3	73.4	67.8	77.4	85.5	72.1	70.2	74.6	78.3	86.2	68.4	75.9	70.6	80.2	**86.7**	**72.4**	**80.3**	**74.2**	**80.6**

Our primary robustness concerns perturbations within the input's spectral support, while traditional PGD and CW assess the norm-bounded threat model in Table 4. Appendix D.3 evaluates a white-box attack that uses the frozen filter to parameterize perturbations. To assess physiological plausibility under matched perturbation strength, we conducted a blinded expert assessment. Two physicians, each with more than 1,000 hours of experience reviewing ECG and PPG, labeled 2,400 perturbed signals as realistic or unrealistic. We selected 200 source signals from three datasets (600 in total) and generated four matched-strength perturbed versions. As shown in Table 5, our method received the highest realism ratings from both physicians across datasets. The annotation protocol and further details are provided in Appendix D.3.3.

Table 5: Expert (E1 and E2) assessment. Both denotes samples labeled realistic by both physicians, and Agreement denotes samples receiving the same label from physicians. Across samples, the physicians achieved 94.7% agreement with an overall Cohen's κ of 0.89.

Dataset	Method	Realistic (%)		Both (%)	Agree. (%)
		E1	E2		
DaLiA	PGD	20	18	17	96
	SAP	93	95	92	96
	DARG	94	92	91	96
	Ours	100	98	98	98
Chapman	PGD	0	0	0	100
	SAP	10	12	7	92
	DARG	90	84	80	86
	Ours	100	100	100	100
PhysioNet	PGD	4	0	0	96
	SAP	16	10	8	90
	DARG	90	80	78	86
	Ours	100	100	100	100

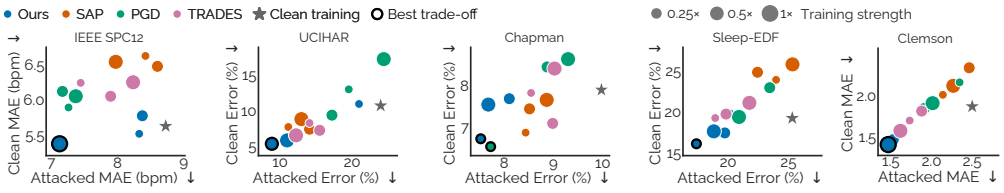


Figure 4: Clean–robust tradeoffs under ℓ_2-PGD. Marker size scales each method's reference training budget and step size; evaluation budgets are fixed within each dataset. Lower-left is better. Our method achieves a favorable balance between clean and attacked error across the datasets. More details and results on all datasets are provided in Appendix D.5.

Table 6: Ablation experiments. Parentheses show improvement (+) or degradation (-) relative to Ours.

(a) HR estimation task using PPG signals

Method	IEEE SPC12			IEEE SPC22			DaLiA$_{PPG}$		
	MAE↓	RMSE↓	ρ↑	MAE↓	RMSE↓	ρ↑	MAE↓	RMSE↓	ρ↑
Ours	**5.19**±0.10	**10.59**±0.50	**85.47**±0.60	**4.77**±0.98	**9.76**±1.50	**85.90**±1.79	**5.23**±0.09	**11.85**±0.25	**86.73**±0.50
Sign + Filter	6.08 (-0.89)	12.73 (-2.14)	82.78 (-2.69)	7.93 (-3.16)	17.15 (-7.39)	74.11 (-11.79)	5.64 (-0.41)	12.54 (-0.69)	85.16 (-1.57)
Sign only	6.37 (-1.18)	12.80 (-2.21)	81.59 (-3.88)	9.38 (-4.61)	19.69 (-9.93)	46.10 (-39.80)	7.93 (-2.70)	17.15 (-5.30)	74.11 (-12.62)
Input-filter	6.25 (-1.06)	12.65 (-2.06)	80.57 (-4.90)	8.25 (-3.48)	16.87 (-7.11)	55.17 (-30.73)	7.57 (-2.34)	16.38 (-4.53)	76.21 (-10.52)
Binary mask	6.92 (-1.73)	12.77 (-2.18)	82.15 (-3.32)	9.13 (-4.36)	17.03 (-7.27)	55.99 (-29.91)	8.03 (-2.80)	16.84 (-4.99)	76.52 (-10.21)
Raw gradient	5.76 (-0.57)	12.48 (-1.89)	82.34 (-3.13)	8.79 (-4.02)	22.04 (-12.28)	51.21 (-34.69)	5.30 (-0.07)	12.04 (-0.19)	81.50 (-5.23)
Global filter	5.80 (-0.61)	11.99 (-1.40)	83.71 (-1.76)	9.98 (-5.21)	23.24 (-13.48)	50.59 (-35.31)	7.90 (-2.67)	13.49 (-1.64)	80.21 (-6.52)
Without KL	5.66 (-0.47)	11.92 (-1.33)	82.72 (-2.75)	9.14 (-4.37)	22.42 (-12.66)	52.38 (-33.52)	6.03 (-0.80)	12.38 (-0.53)	77.45 (-9.28)
Without task loss	5.71 (-0.52)	11.48 (-0.89)	83.15 (-2.32)	9.44 (-4.67)	22.45 (-12.69)	53.11 (-32.79)	5.78 (-0.55)	12.10 (-0.25)	78.07 (-8.66)

(b) Activity and Step tasks using IMU signals

Method	UCIHAR			USC			Clemson		
	Acc↑	W-F1↑	F1↑	Acc↑	W-F1↑	F1↑	MAPE↓	MAE↓	RMSE↓
Ours	**92.89**±1.05	**92.09**±1.12	**93.14**±1.04	**77.49**±1.42	**76.18**±1.23	**75.58**±1.09	**3.08**±0.33	**1.70**±0.20	**2.91**±0.23
Sign + Filter	90.26 (-2.63)	88.98 (-3.11)	89.56 (-3.58)	74.77 (-2.72)	73.40 (-2.78)	73.17 (-2.41)	3.09 (-0.01)	1.73 (-0.03)	2.93 (-0.02)
Sign only	87.37 (-5.52)	85.58 (-6.51)	86.88 (-6.26)	74.09 (-3.40)	72.62 (-3.56)	71.87 (-3.71)	4.39 (-1.31)	2.49 (-0.79)	3.71 (-0.80)
Input-filter	88.12 (-4.77)	86.81 (-5.28)	87.10 (-6.04)	74.16 (-3.33)	72.84 (-3.34)	72.10 (-3.48)	3.97 (-0.89)	2.15 (-0.45)	3.55 (-0.64)
Binary mask	87.56 (-5.33)	86.13 (-5.96)	86.59 (-6.55)	73.68 (-3.81)	72.16 (-4.02)	71.99 (-3.59)	3.71 (-0.63)	2.20 (-0.50)	3.60 (-0.69)
Raw gradient	91.66 (-1.23)	90.64 (-1.45)	90.94 (-2.20)	72.99 (-4.50)	70.78 (-5.40)	69.51 (-6.07)	3.10 (-0.02)	1.88 (-0.18)	2.98 (-0.07)
Global filter	92.11 (-0.78)	91.21 (-0.88)	92.38 (-0.76)	70.35 (-7.14)	68.13 (-8.05)	68.98 (-6.60)	3.59 (-0.51)	1.76 (-0.06)	2.98 (-0.07)
Without KL	92.10 (-0.79)	91.08 (-1.01)	**93.14** (0.00)	70.93 (-6.56)	68.66 (-7.52)	67.45 (-8.13)	3.13 (-0.05)	1.75 (-0.05)	2.98 (-0.07)
Without task loss	92.66 (-0.23)	92.05 (-0.04)	93.07 (-0.07)	70.13 (-7.36)	71.12 (-5.06)	70.53 (-5.05)	5.06 (-1.98)	2.90 (-1.20)	3.95 (-1.04)

(c) CVD and Sleep tasks using ECG and EEG signals

Method	Chapman			PhysioNet			Sleep-EDF		
	Acc↑	F1↑	AUC↑	Acc↑	F1↑	AUC↑	Acc↑	F1↑	κ↑
Ours	**92.88**±0.12	**91.95**±0.08	**98.79**±0.14	84.10±0.49	73.70±1.10	93.57±0.23	**83.02**±0.86	**81.53**±1.10	**77.50**±1.23
Sign + Filter	92.61 (-0.27)	91.44 (-0.51)	98.68 (-0.11)	**84.43** (+0.33)	**74.44** (+0.74)	**93.72** (+0.15)	81.56 (-1.46)	79.91 (-1.62)	75.39 (-2.11)
Sign only	92.28 (-0.60)	91.25 (-0.70)	98.68 (-0.11)	83.98 (-0.12)	73.43 (-0.27)	93.46 (-0.11)	80.19 (-2.83)	78.15 (-3.38)	73.19 (-4.31)
Input-filter	92.10 (-0.78)	91.07 (-0.88)	98.22 (-0.57)	83.17 (-0.93)	73.20 (-0.50)	93.41 (-0.16)	80.03 (-2.99)	78.03 (-3.50)	72.47 (-5.03)
Binary mask	91.86 (-1.02)	90.34 (-1.61)	98.02 (-0.77)	82.79 (-1.31)	73.06 (-0.64)	93.13 (-0.44)	79.81 (-3.21)	77.52 (-4.01)	72.05 (-5.45)
Raw gradient	91.32 (-1.56)	90.04 (-1.91)	98.29 (-0.50)	84.06 (-0.04)	72.49 (-1.21)	93.20 (-0.37)	81.89 (-1.13)	81.50 (-0.03)	75.68 (-1.82)
Global filter	91.32 (-1.56)	90.02 (-1.93)	98.27 (-0.52)	82.19 (-1.91)	70.31 (-3.39)	92.59 (-0.98)	81.92 (-1.10)	81.24 (-0.29)	76.05 (-1.45)
Without KL	91.56 (-1.32)	90.29 (-1.66)	98.36 (-0.43)	84.05 (-0.05)	73.26 (-0.44)	93.50 (-0.07)	80.22 (-2.80)	81.01 (-0.52)	73.58 (-3.92)
Without task loss	92.19 (-0.69)	91.63 (-0.32)	98.43 (-0.36)	83.95 (-0.15)	73.45 (-0.25)	92.99 (-0.58)	75.57 (-7.45)	75.38 (-6.15)	67.12 (-10.38)

4.1 Ablations

We ablate the gradient update, filtering, input conditioning, and objective. To assess the gradient update, we apply the elementwise sign transform with and without the filter as in Equation 10.

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \alpha\, \mathcal{T}\big(\nabla_{\mathbf{x}} \mathcal{L}(f_\theta(\mathbf{x}^t), y)\big)\,, \qquad \mathcal{T}(\mathbf{g}) = \begin{cases} \operatorname{sign}(\mathbf{g}) * h_\psi, & \text{sign + filter,} \\ \operatorname{sign}(\mathbf{g}), & \text{sign only.} \end{cases} \tag{10}$$

The *sign + filter* variant filters the elementwise gradient sign, while *sign only* omits filtering. We also replace the learned filter response with the input magnitude spectrum or a binary spectral mask. The raw-gradient control retains gradient normalization without filtering. The global-filter control conditions on the training-set mean spectrum for every sample. Two further variants remove either the KL penalty or the adversarial task-loss term from filter training. We tune α separately for each variant and rescale perturbations exceeding the common ℓ_2 budget. Applying $\operatorname{sign}(\cdot)$ alters the gradient direction. Thus, the *sign + filter* variant no longer retains the minimum-angle guarantee of Theorem 2.2. The *sign + filter* variant remains competitive and is slightly better on a few ECG metrics, but these gains are small and inconsistent across tasks. Overall, the learned filter performs best on most metrics.

Table 6 supports the benefits of input-conditioned spectral weighting. Removing the KL penalty and the filter's task-loss term reduces performance. These results support combining spectral alignment with task-driven filter learning, although the contribution of each component varies across datasets. Figure 5 illustrates that PGD spreads energy across the spectrum, including regions that are absent in the input. In contrast, the filter concentrates the perturbation in the effective region as the input while avoiding the coarse behavior of the binary mask.

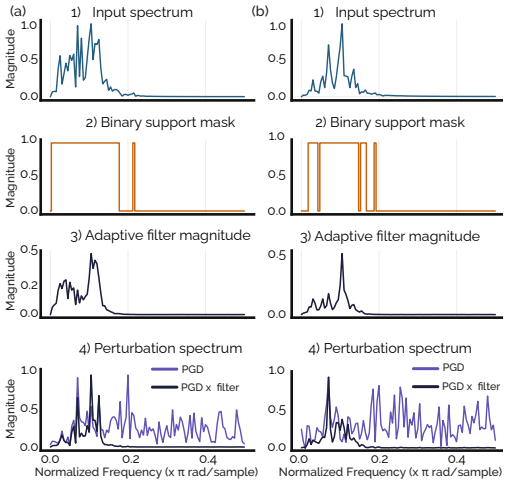


Figure 5: Filter ablations with two samples.

Additional robustness results are provided in Appendix D.3, computational overhead comparisons in Appendix F, and an analysis of delayed adversarial generation in Appendix D.8. Cross-dataset experiments support the reuse of frozen filters across datasets (Appendix D.4), while controlled experiments examine spectral energy allocation at matched perturbation norms (Appendix E.4).

4.2 Discussion of Results

A well-known perspective of Ilyas et al. (2019) is that adversarial examples can arise from non-robust but predictive features. Our results suggest that the perturbation model itself is important to this interpretation. When perturbations are generated without respecting the characteristics of the signal, they can introduce weakly represented harmonics. In this case, the resulting adversarial examples are less consistent with the data and are less likely to reflect plausible variations. Our findings therefore align with studies suggesting that adversarial examples need not correspond to meaningful real-world features and can depend on how the perturbations are constructed (Li et al., 2023).

5 Related Work

The robustness literature is broad (Yuan et al., 2019). We focus on the lines of work most relevant to methods that seek robustness without degrading clean-data performance.

On-manifold: Several methods generate adversarial examples by learning a manifold with a generative model and then perturbing the latent vectors (Stutz et al., 2019; Xue et al., 2023). Examples include GAN or VAE-based approaches that approximate the data distribution (Stutz et al., 2019; Larsen et al., 2016; Karras et al., 2018; Lin et al., 2020). These methods are conceptually aligned with the aim of producing realistic samples, but their effectiveness depends on how well the generative model captures the data distribution. In some settings, this approximation and prior can be inaccurate (Connor et al., 2021), which may lead to perturbations that are misaligned with the signal. In contrast, our method operates in the signal space and avoids fitting a separate generative model.

Smoothing-based perturbations: A second line of work aims to improve plausibility by smoothing perturbations to suppress spikes and high-frequency artifacts (Pialla et al., 2022; Guo et al., 2020). For example, prior work applies Gaussian or low-pass filtering with fixed hyperparameters (Han et al., 2020), while SGM (Pialla et al., 2022) promotes smoothness through regularization. These approaches reduce perturbation bandwidth, but they do so in a largely task-agnostic manner and suppress informative high-frequencies. As a result, fixed smoothing can be overly restrictive across heterogeneous domains. Our method differs by adapting perturbation support to the observed input rather than imposing a fixed low-pass prior.

Frequency-domain view: Prior work studies the frequency structure of adversarial perturbations (Yin et al., 2019; Luo et al., 2022; Chen et al., 2023), while F-SAT modifies STFT magnitudes within selected bands (Zhang et al., 2025). FA (Jia et al., 2022) uses prescribed input-dependent frequency weighting, and IFAP (Yoshida & Okuda, 2025) constrains perturbations through an input-derived spectral envelope. Our method learns input-conditioned frequency weights through an adversarial task objective and soft spectral alignment regularization. It also minimizes angular deviation from the original gradient for a fixed magnitude response. Our contribution combines learned spectral weighting with this minimum-angle phase choice.

6 Conclusion

We introduced an input-conditioned spectral filter that learns the frequency allocation of adversarial perturbations through task loss and spectral regularization. Across multiple tasks, our method improved clean-test performance relative to the baselines, with component ablations supporting the value of learned spectral weighting and frozen-filter transfer supporting its reuse across datasets. Beyond predictive performance and spectral metrics, we evaluated waveform realism through independent assessments by two physicians of 2,400 perturbed signals. This assessment provides evidence about perceived physiological plausibility that complements our experiments. Together, these evaluations connect learned perturbation design with generalization and physician-rated realism.

AI USE STATEMENT

In this work, we used generative AI tools for rephrasing text and for assisting with the preparation of figure-plotting scripts, such as adjusting font sizes and other presentation settings. We did not use generative AI tools to develop the core methodological implementation, including scripts for adversarial perturbation generation, which were implemented manually following the pseudocode and descriptions provided in the paper.

After completing the codebase, we first manually verified the figure-plotting scripts and the main implementation. We subsequently used generative AI tools as an additional code-review aid to identify potential bugs and to check that the implementation was consistent with the methodological description. All AI-assisted text and code were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on crafting perturbations to improve generalization for healthcare-related, human-generated temporal data. Our method encourages adversarial perturbations to align with the spectral characteristics of the observed input. Thus, we enhance the reliability of learning models in scenarios where accurate and robust predictions are crucial.

The potential societal impact of this work is noteworthy, as improving robustness in healthcare applications contributes to safer and more trustworthy decision-making systems. However, like any advancement in adversarial methods, there is a dual-use risk; the techniques developed could also inform the generation of more potent adversarial attacks. It is therefore essential that this research is responsibly applied, emphasizing its use in enhancing security and robustness in critical applications.

We believe the positive societal implications outweigh the risks, particularly given the potential to improve diagnostic tools and patient outcomes in healthcare.

REPRODUCIBILITY STATEMENT

We provide the pseudocode and a PyTorch implementation of the perturbation generator in Appendix B. Appendix C documents dataset preprocessing and partitions, task-specific evaluation metrics, baseline configurations, predictor and filter architectures, and training hyperparameters, including validation-based model selection. Appendix D describes the statistical analyses, robustness evaluations, ablations, and blinded expert-assessment protocol. The spectral-analysis procedures and controlled energy-allocation experiments are detailed in Appendix E. Hardware and training resources are reported in Appendix C, with computational complexity and perturbation-generation timings in Appendix F.

REFERENCES

Sajjad Abdoli, Patrick Cardinal, and Alessandro Lameiras Koerich. End-to-end environmental sound classification using a 1d convolutional neural network. *Expert Systems with Applications*, 136: 252–263, 2019. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2019.06.040. URL https://www.sciencedirect.com/science/article/pii/S0957417419304403.

Salah S. Al-Zaiti, Christian Martin-Gill, Jessica K. Zègre-Hemsey, Zeineb Bouzid, Ziad Faramand, Mohammad O. Alrawashdeh, Richard E. Gregg, Stephanie Helman, Nathan T. Riek, Karina Kraevsky-Phillips, Gilles Clermont, Murat Akcakaya, Susan M. Sereika, Peter Van Dam, Stephen W. Smith, Yochai Birnbaum, Samir Saba, Ervin Sejdic, and Clifton W. Callaway. Machine learning for ECG diagnosis and risk stratification of occlusion myocardial infarction. *Nature Medicine*, 29(7):1804–1813, July 2023. ISSN 1546-170X. doi: 10.1038/s41591-023-02396-3. URL https://www.nature.com/articles/s41591-023-02396-3.

Erick A Perez Alday, Annie Gu, Amit J Shah, Chad Robichaux, An-Kwok Ian Wong, Chengyu Liu, Feifei Liu, Ali Bahrami Rad, Andoni Elola, Salman Seyedi, Qiao Li, Ashish Sharma, Gari D Clifford, and Matthew A Reyna. Classification of 12-lead ecgs: the physionet/computing in cardiology challenge 2020. *Physiological Measurement*, 41(12):124003, dec 2020. doi: 10.1088/1361-6579/abc960. URL https://dx.doi.org/10.1088/1361-6579/abc960.

Gokhan Altan, Yakup Kutlu, Yusuf Garbi, Adnan Özhan Pekmezci, and Serkan Nural. Multimedia respiratory database (respiratorydatabase@tr): Auscultation sounds and chest x-rays. *ArXiv*, 2017.

D. Anguita, Alessandro Ghio, L. Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International Workshop on Ambient Assisted Living and Home Care*, 2012. URL https://api.semanticscholar.org/CorpusID:13178535.

Karim Bayoumy, Mohammed Gaber, Abdallah Elshafeey, Omar Mhaimeed, Elizabeth H. Dineen, Francoise A. Marvel, Seth S. Martin, Evan D. Muse, Mintu P. Turakhia, Khaldoun G. Tarakji, and Mohamed B. Elshazly. Smart wearable devices in cardiovascular care: where we are and how to move forward. *Nature Reviews Cardiology*, 18(8):581–599, August 2021. ISSN 1759-5010. doi: 10.1038/s41569-021-00522-7. URL https://www.nature.com/articles/s41569-021-00522-7.

Taha Belkhouja, Yan Yan, and Janardhan Rao Doppa. Dynamic time warping based adversarial framework for time-series domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. doi: 10.1109/TPAMI.2022.3224754.

Dwaipayan Biswas, Luke Everson, Muqing Liu, Madhuri Panwar, Bram-Ernst Verhoef, Shrishail Patki, Chris H. Kim, Amit Acharyya, Chris Van Hoof, Mario Konijnenburg, and Nick Van Helleputte. Cornet: Deep learning framework for ppg-based heart rate estimation and biometric identification in ambulant environment. *IEEE Transactions on Biomedical Circuits and Systems*, 2019.

Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks . In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, Los Alamitos, CA, USA, May 2017. IEEE Computer Society. doi: 10.1109/SP.2017.49. URL https://doi.ieeecomputersociety.org/10.1109/SP.2017.49.

Fang Chen, Jian Wang, Han Liu, Wentao Kong, Zhe Zhao, Longfei Ma, Hongen Liao, and Daoqiang Zhang. Frequency constraint-based adversarial attack on deep neural networks for medical image classification. *Computers in Biology and Medicine*, 164:107248, 2023. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2023.107248. URL https://www.sciencedirect.com/science/article/pii/S0010482523007138.

Xun Chen, Chang Li, Aiping Liu, Martin J. McKeown, Ruobing Qian, and Z. Jane Wang. Toward open-world electroencephalogram decoding via deep learning: A comprehensive survey. *IEEE Signal Processing Magazine*, 39(2):117–134, 2022a. doi: 10.1109/MSP.2021.3134629.

Yiting Chen, Qibing Ren, and Junchi Yan. Rethinking and improving robustness of convolutional neural networks: a shapley value-based approach in frequency domain. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=rQ1cNbi07Vq.

Gari D Clifford, Chengyu Liu, Benjamin Moody, Li-wei H. Lehman, Ikaro Silva, Qiao Li, A E Johnson, and Roger G. Mark. Af classification from a short single lead ecg recording: The physionet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)*, pp. 1–4, 2017. doi: 10.22489/CinC.2017.065-469.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 1960. URL https://api.semanticscholar.org/CorpusID:15926286.

Marissa Connor, Gregory Canal, and Christopher Rozell. Variational autoencoder with learned latent structure. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2359–2367. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/connor21a.html.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Berken Utku Demirel and Christian Holz. Finding order in chaos: A novel data augmentation method for time series in contrastive learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=dbVRDk2wt7.

Berken Utku Demirel and Christian Holz. Shifting the paradigm: A diffeomorphism between time series data manifolds for achieving shift-invariancy in deep learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=nibeaHUEJx.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 2352–2359. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/324. URL https://doi.org/10.24963/ijcai.2021/324. Main Track.

Albert K. Feeny, Mina K. Chung, Anant Madabhushi, Zachi I. Attia, Maja Cikes, Marjan Firouznia, Paul A. Friedman, Matthew M. Kalscheur, Suraj Kapa, Sanjiv M. Narayan, Peter A. Noseworthy, Rod S. Passman, Marco V. Perez, Nicholas S. Peters, Jonathan P. Piccini, Khaldoun G. Tarakji, Suma A. Thomas, Natalia A. Trayanova, Mintu P. Turakhia, and Paul J. Wang. Artificial intelligence and machine learning in arrhythmias and cardiac electrophysiology. *Circulation: Arrhythmia and Electrophysiology*, 13(8):e007952, 2020. doi: 10.1161/CIRCEP.119.007952. URL https://www.ahajournals.org/doi/abs/10.1161/CIRCEP.119.007952.

Riccardo Femiano, Charlotte Werner, Matthias Wilhelm, and Prisca Eser. Validation of open-source step-counting algorithms for wrist-worn tri-axial accelerometers in cardiovascular patients. *Gait & Posture*, 92:206–211, 2022. ISSN 0966-6362. doi: https://doi.org/10.1016/j.gaitpost.2021.11.035. URL https://www.sciencedirect.com/science/article/pii/S0966636221006196.

Wolfgang Ganglberger, Samaneh Nasiri, Haoqi Sun, Soriul Kim, Chol Shin, M Brandon Westover, and Robert J Thomas. Refining sleep staging accuracy: transfer learning coupled with scorability models. *Sleep*, 47(11):zsae202, 08 2024. ISSN 0161-8105. doi: 10.1093/sleep/zsae202. URL https://doi.org/10.1093/sleep/zsae202.

Justin Gilmer, Nicolas Ford, Nicholas Carlini, and Ekin Cubuk. Adversarial examples are a natural consequence of test error in noise. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2280–2289. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/gilmer19a.html.

Ary L. Goldberger, Luis A. N. 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. *Circulation*, 101(23):e215–e220, 2000. doi: 10.1161/01.CIR.101.23.e215. URL https://www.ahajournals.org/doi/abs/10.1161/01.CIR.101.23.e215.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. Low frequency adversarial perturbation. In Ryan P. Adams and Vibhav Gogate (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 1127–1137. PMLR, 22–25 Jul 2020. URL https://proceedings.mlr.press/v115/guo20a.html.

Xintian Han, Yuxuan Hu, Luca Foschini, Larry Chinitz, Lior Jankelson, and Rajesh Ranganath. Deep learning models for electrocardiograms are susceptible to adversarial attack. *Nature Medicine*, 26 (3):360–363, March 2020. ISSN 1546-170X. doi: 10.1038/s41591-020-0791-x. URL https://www.nature.com/articles/s41591-020-0791-x.

Awni Y. Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H. Tison, Codie Bourn, Mintu P. Turakhia, and Andrew Y. Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25(1):65–69, January 2019. ISSN 1546-170X. doi: 10.1038/s41591-018-0268-3. URL https://www.nature.com/articles/s41591-018-0268-3.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Shenda Hong, Yanbo Xu, Alind Khare, Satria Priambada, Kevin Maher, Alaa Aljiffry, Jimeng Sun, and Alexey Tumanov. Holmes: Health online model ensemble serving for deep learning models in intensive care units. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1614–1624, 2020.

Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5, 2004.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2015. URL https://proceedings.mlr.press/v37/ioffe15.html.

Shuai Jia, Chao Ma, Taiping Yao, Bangjie Yin, Shouhong Ding, and Xiaokang Yang. Exploring frequency adversarial attacks for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4103–4112, June 2022.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021.

Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Dani Kiyasseh, Tingting Zhu, and David A. Clifton. Clocs: Contrastive learning of cardiac signals across space, time, and patients. In *International Conference on Machine Learning*, 2020.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/larsen16.html.

Ang Li, Yifei Wang, Yiwen Guo, and Yisen Wang. Adversarial examples are not real features. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=hSkEcIFi3o.

Hanzhe LI, Jiaran Zhou, Yuezun Li, Baoyuan Wu, Bin Li, and Junyu Dong. Freqblender: Enhancing deepfake detection by blending frequency knowledge. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=otZPBS0un6.

Wei-An Lin, Chun Pong Lau, Alexander Levine, Rama Chellappa, and Soheil Feizi. Dual manifold adversarial robustness: Defense against lp and non-lp adversarial attacks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3487–3498. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/23937b42f9273974570fb5a56a6652ee-Paper.pdf.

Cheng Luo, Qinliang Lin, Weicheng Xie, Bizhu Wu, Jinheng Xie, and Linlin Shen. Frequency-driven imperceptible adversarial attack on semantic similarity. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15294–15303, 2022. doi: 10.1109/CVPR52688.2022.01488.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Shishira R. Maiya, Max Ehrlich, Vatsal Agarwal, Ser-Nam Lim, Tom Goldstein, and Abhinav Shrivastava. Unifying the harmonic analysis of adversarial attacks and robustness. In *BMVC*, pp. 620–621, 2023. URL http://proceedings.bmvc2023.org/620/.

Ryan Mattfeld, Elliot Jesch, and Adam Hoover. A new dataset for evaluating pedometer performance. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2017.

Manuel Meier, Berken Utku Demirel, and Christian Holz. WildPPG: A real-world PPG dataset of long continuous recordings. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=MojU63gze2.

Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals & Systems (2nd Ed.)*. Prentice-Hall, Inc., USA, 1996. ISBN 0138147574.

Guillermo Ortiz-Jimenez, Apostolos Modas, Seyed-Mohsen Moosavi, and Pascal Frossard. Hold me tight! influence of discriminative features on deep network boundaries. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2935–2946. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1ea97de85eb634d580161c603422437f-Paper.pdf.

Qingyi Pan and Yicheng Li. Functional virtual adversarial training for semi-supervised time series classification. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 125989–126020. Curran Associates, Inc., 2025. doi: 10.52202/085713-4200. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b75c000c500bf778568bf2c8f732b0fd-Paper-Conference.pdf.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Gautier Pialla, Hassan Ismail Fawaz, Maxime Devanne, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller, Christoph Bergmeir, Daniel Schmidt, Geoffrey Webb, and Germain Forestier. Smooth perturbations for time series adversarial attacks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 485–496. Springer, 2022.

Hangwei Qian, Sinno Jialin Pan, and Chunyan Miao. Latent independent excitation for generalizable sensor-based cross-person activity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11921–11929, May 2021. doi: 10.1609/aaai.v35i13.17416. URL https://ojs.aaai.org/index.php/AAAI/article/view/17416.

Hangwei Qian, Tian Tian, and Chunyan Miao. What makes good contrastive learning on small-scale wearable-based tasks? In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 3761–3771, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539134. URL https://doi.org/10.1145/3534678.3539134.

Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J. Topol. AI in health and medicine. *Nature Medicine*, 28(1):31–38, January 2022. ISSN 1546-170X. doi: 10.1038/s41591-021-01614-0. URL https://www.nature.com/articles/s41591-021-01614-0.

Attila Reiss, Ina Indlekofer, Philip Schmidt, and Kristof Van Laerhoven. Deep ppg: Large-scale heart rate estimation with convolutional neural networks. *Sensors*, 19(14), 2019. ISSN 1424-8220. doi: 10.3390/s19143079. URL https://www.mdpi.com/1424-8220/19/14/3079.

Leandro Giacomini Rocha, Dwaipayan Biswas, Bram-Ernst Verhoef, Sergio Bampi, Chris Van Hoof, Mario Konijnenburg, Marian Verhelst, and Nick Van Helleputte. Binary cornet: Accelerator for hr estimation from wrist-ppg. *IEEE Transactions on Biomedical Circuits and Systems*, 14(4):715–726, 2020. doi: 10.1109/TBCAS.2020.3001675.

Leiv Sandvik, Jan Erikssen, Myrvin Ellestad, Gunnar Erikssen, Erik Thaulow, Reidar Mundal, and Kaare Rodahl. Heart rate increase and maximal heart rate during exercise as predictors of cardiovascular mortality: a 16-year follow-up study of 1960 healthy men. *Coronary Artery Disease*, 6(8):667, August 1995. ISSN 0954-6928. URL https://journals.lww.com/coronary-artery/abstract/1995/08000/heart_rate_increase_and_maximal_heart_rate_during.12.aspx.

Ron A Shapira Weber, Matan Eyal, Nicki Skafte, Oren Shriki, and Oren Freifeld. Diffeomorphic temporal alignment nets. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/db98dc0dbafde48e8f74c0de001d35e4-Paper.pdf.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2019.

Ajay Subramanian, Elena Sizikova, Najib J. Majaj, and Denis G. Pelli. Spatial-frequency channels, shape bias, and adversarial robustness. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KvPwXVcslY.

Haoqi Sun, Wolfgang Ganglberger, Ezhil Panneerselvam, Michael J Leone, Syed A Quadri, Balaji Goparaju, Ryan A Tesh, Oluwaseun Akeju, Robert J Thomas, and M Brandon Westover. Sleep staging from electrocardiography and respiration with deep learning. *Sleep*, 43(7):zsz306, 12 2019. ISSN 0161-8105. doi: 10.1093/sleep/zsz306. URL https://doi.org/10.1093/sleep/zsz306.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Andriy Temko. Accurate heart rate monitoring during physical exercises using ppg. *IEEE Transactions on Biomedical Engineering*, 64(9):2016–2024, 2017. doi: 10.1109/TBME.2017.2676243.

Viet-Thi Tran, Carolina Riveros, and Philippe Ravaud. Patients' views of wearable devices and AI in healthcare: findings from the ComPaRe e-cohort. *npj Digital Medicine*, 2(1):1–8, June 2019. ISSN 2398-6352. doi: 10.1038/s41746-019-0132-y. URL https://www.nature.com/articles/s41746-019-0132-y.

Haotao Wang, Aston Zhang, Shuai Zheng, Xingjian Shi, Mu Li, and Zhangyang Wang. Removing batch normalization boosts adversarial training. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23433–23445. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wang22ap.html.

Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119:3–11, 2019. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2018.02.010. URL https://www.sciencedirect.com/science/article/pii/S016786551830045X. Deep Learning for Pattern Recognition.

Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1994. doi: 10.1089/cmb.1994.1.337. URL https://doi.org/10.1089/cmb.1994.1.337. PMID: 8790475.

Haotian Xue, Alexandre Araujo, Bin Hu, and Yongxin Chen. Diffusion-based adversarial sample generation for improved stealthiness and controllability. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=JTwxylP6U9.

Runze Yang, Jian Song, Baoqi Huang, Wuyungerile Li, and Guodong Qi. An Energy-Efficient Step-Counting Algorithm for Smartphones. *The Computer Journal*, 65(3):689–700, 08 2020a. ISSN 0010-4620. doi: 10.1093/comjnl/bxaa096. URL https://doi.org/10.1093/comjnl/bxaa096.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8588–8601. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/61d77652c97ef636343742fc3dcf3ba9-Paper.pdf.

Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D. Cubuk, and Justin Gilmer. *A fourier perspective on model robustness in computer vision*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Masatomo Yoshida and Masahiro Okuda. Ifap: Input-frequency adaptive adversarial perturbation via full-spectrum envelope constraint for spectral fidelity. *IEEE Access*, 13:217504–217518, 2025. doi: 10.1109/ACCESS.2025.3648201.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824, 2019. doi: 10.1109/TNNLS.2018.2886017.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019.

Mi Zhang and Alexander A. Sawchuk. Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pp. 1036–1043, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312240. doi: 10.1145/2370216.2370438. URL https://doi.org/10.1145/2370216.2370438.

Yuwei Zhang, Tong Xia, Jing Han, Yu Wu, Georgios Rizos, Yang Liu, Mohammed Mosuily, Jagmohan Chauhan, and Cecilia Mascolo. Towards open respiratory acoustic foundation models: Pretraining and benchmarking. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=vXnGXRbOfb.

Zhilin Zhang, Zhouyue Pi, and Benyuan Liu. Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. *IEEE Transactions on Biomedical Engineering*, 62(2):522–531, 2015. doi: 10.1109/TBME.2014.2359372.

Zirui Zhang, Wei Hao, Aroon Sankoh, William Lin, Emanuel Mendiola-Ortiz, Junfeng Yang, and Chengzhi Mao. I can hear you: Selective robust training for deepfake audio detection. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=2GcR9bO620.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1BLjgZCb.

Jianwei Zheng, Jianming Zhang, Sidy Danioko, Hai Yao, Hangyuan Guo, and Cyril Rakovski. A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. *Scientific Data*, 7(1):48, February 2020. ISSN 2052-4463. doi: 10.1038/s41597-020-0386-x. URL https://www.nature.com/articles/s41597-020-0386-x.

APPENDIX OVERVIEW

A Theoretical Analysis **19**

A.1 Notation and preliminaries 19

A.2 Proofs 20

A.2.1 Spectral mass control 20

A.2.2 Minimum angle perturbation 21

B Algorithmic Implementation of Our Method **22**

C Experiments **24**

C.1 Datasets 24

C.2 Metrics 25

C.3 Baselines 26

C.4 Implementation details 29

C.4.1 Architectures 29

C.4.2 Hyperparameters 30

D Additional Results and Details Regarding Robustness **31**

D.1 Significance analysis 31

D.2 Robustness of hyperparameters 32

D.3 Robustness to perturbations 35

D.3.1 Robustness with other techniques 35

D.3.2 Filter-parameterized white-box evaluation 36

D.3.3 Expert assessment 37

D.4 Cross-dataset filter transfer 40

D.5 Clean–Robust performance trade-off 41

D.6 Attack convergence and per-example union 42

D.7 Improvements in performance across different networks 43

D.7.1 Adaptive filtering network 43

D.8 Investigating the effect of epoch delays on adversarial generation 44

D.9 Additional modalities 45

E Spectral Analysis of Perturbations **46**

E.1 Visual evidence 46

E.2 Quantitative evidence 48

E.3 Empirical spectral concentration 50

E.4 Controlled spectral energy allocation 53

F Computational Overhead of Techniques **55**

G Expanded Review of Related Work **56**

A Theoretical Analysis

A.1 Notation and frequency-domain representation

Let $\boldsymbol{x} \in \mathbb{R}^{K \times L}$ denote a multichannel sequence of length L. For notational simplicity, the derivations are written for a single channel; the multichannel case follows by applying the same analysis to the vectorized sequence. We write $\widehat{\boldsymbol{x}}(\omega) = \mathcal{F}(\boldsymbol{x})$ for the Fourier transform of $\boldsymbol{x}$, where ω denotes angular frequency. Likewise, for a perturbation δ and filter h_ψ, we write

$$\widehat{\delta}(\omega) = \mathcal{F}(\delta), \qquad H_\psi(\omega) = \mathcal{F}(h_\psi). \tag{11}$$

The filtered perturbation is defined as in Equation 12, where $*$ denotes convolution.

$$\delta_\psi = \delta * h_\psi, \qquad \widehat{\delta}_\psi(\omega) = \mathcal{F}(\delta_\psi) = \widehat{\delta}(\omega)\, H_\psi(\omega), \tag{12}$$

We use the polar representation

$$\widehat{\boldsymbol{x}}(\omega) = |\widehat{\boldsymbol{x}}(\omega)|\, e^{j\phi_{\boldsymbol{x}}(\omega)}, \qquad H_\psi(\omega) = |H_\psi(\omega)|\, e^{j\phi_{H_\psi}(\omega)}, \qquad \widehat{\delta}(\omega) = |\widehat{\delta}(\omega)|\, e^{j\phi_\delta(\omega)}. \tag{13}$$

Accordingly,

$$\widehat{\delta}_\psi(\omega) = |\widehat{\delta}(\omega)|\, |H_\psi(\omega)|\, e^{j(\phi_\delta(\omega) + \phi_{H_\psi}(\omega))}. \tag{14}$$

The notation used throughout the analysis is summarized in Table 7.

A.1.1 Notation list

Table 7: Notation used in the main text and theoretical analysis.

Notation	Description
$\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^M$	Training dataset.
$\boldsymbol{x} \in \mathbb{R}^{K \times L}$	Input multichannel time series with K channels and length L.
$y \in \{1, \dots, C\}$	Label associated with $\boldsymbol{x}$.
$f_\theta(\cdot)$	Predictor parameterized by θ.
$\mathcal{L}(f_\theta(\boldsymbol{x}), y)$	Loss function for input-label pair $(\boldsymbol{x}, y)$.
δ	Additive perturbation applied to $\boldsymbol{x}$.
h_ψ	Adaptive filter parameterized by ψ.
$\delta_\psi = \delta * h_\psi$	Filtered perturbation.
$\mathcal{F}(\cdot)$	Fourier transform operator.
$\widehat{\boldsymbol{x}}(\omega) = \mathcal{F}(\boldsymbol{x})$	Fourier transform of the input.
$\widehat{\delta}(\omega) = \mathcal{F}(\delta)$	Fourier transform of the perturbation.
$H_\psi(\omega) = \mathcal{F}(h_\psi)$	Frequency response of the adaptive filter.
$\widehat{\delta}_\psi(\omega) = \widehat{\delta}(\omega) H_\psi(\omega)$	Fourier transform of the filtered perturbation.
$\lvert\widehat{\boldsymbol{x}}(\omega)\rvert$	Magnitude spectrum of the input.
$\phi_{\boldsymbol{x}}(\omega)$	Phase spectrum of the input.
$\phi_\delta(\omega)$	Phase spectrum of the perturbation.
$\phi_{H_\psi}(\omega)$	Phase response of the filter.
$\mathrm{supp}(\widehat{\boldsymbol{x}})$	Empirical spectral support of the input.
$\mathcal{P}(\boldsymbol{x})$	Support-constrained perturbation set defined in Eq. 1.
ϵ	Perturbation budget.
α	Perturbation-strength parameter in the update rule.
η	Step size for iterative adversarial updates.
$\mathcal{M}$	Spectrally admissible set.

A.2 PROOFS

A.2.1 PROOF FOR PROPOSITION 2.1

We restate Proposition 2.1 for completeness.

Proposition A.1 (Spectral mass control)**.** *Let p_ψ and $q_{\boldsymbol{x}}$ be the normalized magnitude spectra defined in Equation 3. For any frequency region $A \subseteq \Omega_L$,*

$$\left| \sum_{\omega \in A} p_\psi(\omega) - \sum_{\omega \in A} q_{\boldsymbol{x}}(\omega) \right| \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p_\psi \| q_{\boldsymbol{x}})}. \tag{15}$$

Thus, if $D_{\mathrm{KL}}(p_\psi \| q_{\boldsymbol{x}}) \leq \kappa$, the spectral mass assigned to any frequency differs by at most $\sqrt{\kappa/2}$.

Proof. Both normalized spectra are nonnegative and sum to one. Their KL divergence is

$$D_{\mathrm{KL}}(p_\psi \| q_{\boldsymbol{x}}) = \sum_{\omega \in \Omega_L} p_\psi(\omega) \log \frac{p_\psi(\omega)}{q_{\boldsymbol{x}}(\omega)}, \tag{16}$$

where logarithms are natural. We use the conventions that bins with $p_\psi(\omega) = 0$ contribute zero, whereas a bin with $p_\psi(\omega) > 0$ and $q_{\boldsymbol{x}}(\omega) = 0$ makes the divergence infinite. If the divergence is infinite, the bound is immediate. We therefore consider the finite case.

First, we bound the difference in spectral mass over an arbitrary region. Because both distributions sum to one, we can write the following,

$$\sum_{\omega \in \Omega_L} \big(p_\psi(\omega) - q_{\boldsymbol{x}}(\omega) \big) = 0.$$

Thus, the sum of the positive differences equals the absolute sum of the negative differences. Each is equal to $(1/2) \sum_{\omega \in \Omega_L} |p_\psi(\omega) - q_{\boldsymbol{x}}(\omega)|$. The signed sum over any subset A cannot exceed the total positive difference or fall below the total negative difference. It follows that

$$\left| \sum_{\omega \in A} \big(p_\psi(\omega) - q_{\boldsymbol{x}}(\omega) \big) \right| \leq \frac{1}{2} \sum_{\omega \in \Omega_L} |p_\psi(\omega) - q_{\boldsymbol{x}}(\omega)|. \tag{17}$$

The right-hand side is the total variation distance between the two distributions. Pinsker's inequality bounds this distance by

$$\frac{1}{2} \sum_{\omega \in \Omega_L} |p_\psi(\omega) - q_{\boldsymbol{x}}(\omega)| \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p_\psi \| q_{\boldsymbol{x}})}. \tag{18}$$

Combining Equations 17 and 18 proves Equation 15. If the divergence is at most κ, substituting this upper bound gives $\sqrt{\kappa/2}$.

The argument also applies when a positive constant is added to each magnitude bin before normalization. In that case, the bound concerns the resulting smoothed distributions. □

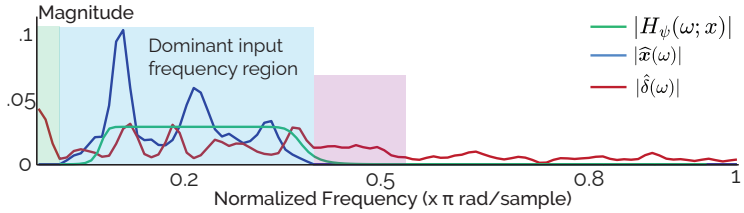


Figure 6: Illustration of the motivation for our method. The input magnitude spectrum (blue) concentrates in a limited region, while the unfiltered perturbation (red) also contains components weakly represented in the input. The filter magnitude response (green) illustrates frequency weighting.

A.2.2 PROOF OF THEOREM 2.2

Theorem A.2 (Minimum-angle phase selection)**.** *Let $\delta_\psi \neq 0$ be obtained by filtering a real perturbation $\delta \neq 0$ with a filter h_ψ whose magnitude response $|H_\psi(\omega)|$ is fixed and even for $\omega \in \Omega_L$. Then the angle between δ and δ_ψ is minimized by zero phase, i.e.,*

$$\phi^*_{H_\psi} \in \arg\min_{\phi_{H_\psi}} \arccos\left(\frac{\langle \delta, \delta_\psi \rangle}{\|\delta\|_2 \, \|\delta_\psi\|_2}\right), \qquad \phi^*_{H_\psi}(\omega) = 0,$$

where $\phi_{H_\psi}(\omega) = \angle H_\psi(\omega)$ and the minimization is over phases satisfying conjugate symmetry.

Proof. Let $\delta_\psi = h_\psi * \delta$, so that in the frequency domain

$$\widehat{\delta}_\psi(\omega) = \widehat{\delta}(\omega)\, H_\psi(\omega).$$

The angle between δ and δ_ψ is

$$\theta = \arccos\left(\frac{\langle \delta, \delta_\psi \rangle}{\|\delta\|_2 \, \|\delta_\psi\|_2}\right).$$

Thus, minimizing θ is equivalent to maximizing the normalized inner product.

By Parseval's theorem,

$$\langle \delta, \delta_\psi \rangle = c_L \sum_{\omega \in \Omega_L} \widehat{\delta}(\omega)\, \widehat{\delta}_\psi(\omega)^* = c_L \sum_{\omega \in \Omega_L} |\widehat{\delta}(\omega)|^2 H_\psi(\omega)^*,$$

where $c_L > 0$ is the transform normalization constant and the sum is over the full DFT frequency grid. Writing

$$H_\psi(\omega) = |H_\psi(\omega)|\, e^{j\phi_{H_\psi}(\omega)}$$

and using the real-valued time-domain inner product gives

$$\langle \delta, \delta_\psi \rangle = c_L \sum_{\omega \in \Omega_L} |\widehat{\delta}(\omega)|^2 |H_\psi(\omega)| \cos\big(\phi_{H_\psi}(\omega)\big).$$

Moreover,

$$\|\delta_\psi\|_2^2 = c_L \sum_{\omega \in \Omega_L} |\widehat{\delta}(\omega)|^2 |H_\psi(\omega)|^2,$$

so the filtered norm is independent of phase for a fixed magnitude response. Hence maximizing the inner product minimizes the angle.

Every weight in the inner-product sum is nonnegative. Its maximum is therefore attained when

$$\phi_{H_\psi}(\omega) = 0 \pmod{2\pi} \quad \text{wherever} \quad |\widehat{\delta}(\omega)|\, |H_\psi(\omega)| > 0.$$

At the remaining bins, the phase does not affect the angle and may vary subject to conjugate symmetry. Choosing zero phase at every bin gives

$$H_\psi(\omega) = |H_\psi(\omega)|.$$

Because the fixed magnitude response is even, this choice satisfies $H_\psi(-\omega) = H_\psi(\omega)^*$ and produces a real-valued filtered perturbation. Thus, zero phase is a minimum-angle solution, with any minimizing response having zero phase modulo 2π on the effective support. □

B Algorithmic Implementation of Our Method

We present the pseudocode and PyTorch (Paszke et al., 2019) implementation of our method. Algorithm 1 takes a sample $\mathbf{x}$ and a learned filter as input and returns the generated example. For the windowed variant, we follow the same procedure, where $\mathbf{x}$ is partitioned into four non-overlapping windows, processed independently, and then stitched back together.

Algorithm 1 Multi-step spectrally regularized perturbation

Input: $\boldsymbol{x}$, y, h_ψ, f_θ, T, α, η, ϵ; **Output:** $\boldsymbol{x}_{\text{adv}}$

$\boldsymbol{x}_{\text{adv}}^0 = \boldsymbol{x}$ ▷ initialize at clean input

for $t = 0, \ldots, T-1$ **do**

$\delta^t = \nabla_{\boldsymbol{x}_{\text{adv}}^t} \mathcal{L}(f_\theta(\boldsymbol{x}_{\text{adv}}^t), y)$ ▷ obtain perturbation direction

$|\widehat{\delta}^t(\omega)| e^{j\phi_{\widehat{\delta}^t}(\omega)} = \mathcal{F}(\delta^t),\ H_\psi(\omega) = \mathcal{F}(h_\psi)$ ▷ FFT of perturbation and filter

$\widehat{\delta}_\psi^t(\omega) = |\widehat{\delta}^t(\omega)|\, |H_\psi(\omega)|\, e^{j\phi_{\widehat{\delta}^t}(\omega)}$ ▷ apply filter magnitude, keep phase

$\delta_\psi^t = \mathcal{F}^{-1}\left(\widehat{\delta}_\psi^t(\omega)\right)$ ▷ back to time domain

$\boldsymbol{x}_{\text{adv}}^{t+1} = \boldsymbol{x}_{\text{adv}}^t + \eta\alpha\|\boldsymbol{x}\|_2 \frac{\delta_\psi^t}{\|\delta_\psi^t\|_2}$ ▷ add normalized filtered perturbation

$\boldsymbol{x}_{\text{adv}}^{t+1} = \boldsymbol{x} + \min\left(1, \frac{\epsilon}{\|\boldsymbol{x}_{\text{adv}}^{t+1} - \boldsymbol{x}\|_2}\right)(\boldsymbol{x}_{\text{adv}}^{t+1} - \boldsymbol{x})$ ▷ Norm bound

end for

return $\boldsymbol{x}_{\text{adv}}^T$

Algorithm 1 applies the filtered perturbation update for T iterations. At each step, the gradient is computed at the current adversarial iterate $\boldsymbol{x}_{\text{adv}}^t$, transformed into the frequency domain, filtered by the learned magnitude response $|H_\psi(\omega)|$, and transformed back to the time domain. The perturbation direction is normalized and scaled by the fixed clean-input norm $\|\boldsymbol{x}\|_2$, so the update magnitude remains tied to the original signal energy rather than changing with the adversarial iterate. After each update, the accumulated perturbation is rescaled when necessary to satisfy the ℓ_2 budget ϵ.

We also provide the PyTorch implementation of our method where the zero phase filter is realized in the frequency domain instead of forward-backward filtering in time domain.

```python
def constrain_perturb(samples, labels, model, zpfilter, steps, epsilon):
    norm_samples = samples.flatten(1).norm(p=2, dim=1) + 1e-10
    adv_samples = samples.clone().detach().requires_grad_(True)
    samplesFFT_mag = torch.abs(torch.fft.rfft(samples, dim=-1))
    samplesFFT_norm_mag = samplesFFT_mag / torch.sum(samplesFFT_mag, dim
        =-1, keepdim=True).clamp_min(1e-10)
    # Calculate the input dependent frequency support of the filter
    filterFFT_mag = zpfilter(samplesFFT_norm_mag)

    for _ in range(steps):
        logits = model(adv_samples)
        loss = nn.CrossEntropyLoss()(logits, labels)
        delta = torch.autograd.grad(loss, adv_samples, create_graph=True)
            [0]

        # Obtain the spectrum of the gradient vector
        deltaFFT = torch.fft.rfft(delta, dim=-1)
        delta_filteredFFT_mag = torch.abs(deltaFFT) * filterFFT_mag
        z = torch.polar(delta_filteredFFT_mag, torch.angle(deltaFFT))
        filtered_delta = torch.fft.irfft(z, n=samples.size(-1), dim=-1)

        # Normalize perturbations and update adversarial samples
        norm_factor = norm_samples / (filtered_delta.flatten(1).norm(p=2,
             dim=1) + 1e-10)
        alpha = 0.01 + 0.02 * torch.rand(samples.size(0), device=samples.
            device)
        adv_samples = adv_samples + (alpha * norm_factor).view(-1, *([1]
            * (samples.ndim - 1))) * filtered_delta

        # Project perturbation back to the epsilon norm ball
        perturbation = adv_samples - samples
        perturbation_norm = perturbation.flatten(1).norm(p=2, dim=1).
            clamp_min(1e-10)
        scale = torch.clamp(epsilon / perturbation_norm, max=1.0)
        adv_samples = samples + scale.view(-1, *([1] * (samples.ndim - 1)
            )) * perturbation

    # Compute KL using the final accumulated perturbation
    perturbationFFT_mag = torch.abs(torch.fft.rfft(adv_samples - samples,
         dim=-1))
    samplesFFT_norm_mag = (samplesFFT_mag + 1e-8) / (samplesFFT_mag + 1e
        -8).sum(dim=-1, keepdim=True)
    perturbationFFT_norm_mag = (perturbationFFT_mag + 1e-8) / (
        perturbationFFT_mag + 1e-8).sum(dim=-1, keepdim=True)

    kl_loss = F.kl_div(torch.log(samplesFFT_norm_mag),
        perturbationFFT_norm_mag, reduction='batchmean')
    return adv_samples, kl_loss
```

Implementation 1: PyTorch implementation of the proposed algorithm

The filter parameters ψ are optimized using

$$\mathcal{L}_\psi = -\mathcal{L}_{\mathrm{CE}}\left(f_\theta(\boldsymbol{x} + \Delta_\psi), y\right) + \lambda_{\mathrm{KL}} D_{\mathrm{KL}}\left(p_\psi \,\|\, q_{\boldsymbol{x}}\right),$$

where $\Delta_\psi = \boldsymbol{x}_{\mathrm{adv}} - \boldsymbol{x}$ is the final perturbation, rescaled when necessary to satisfy the ℓ_2 bound, and p_ψ is its normalized magnitude spectrum. The negative cross-entropy term encourages perturbations that increase the classifier's loss, while the KL term encourages spectral alignment. Gradients propagate through all attack updates, including the gradient computations, while classifier parameters remain fixed during filter optimization. The classifier parameters θ are updated separately using only cross-entropy on detached adversarial samples.

C EXPERIMENTS

Here, we give a detailed description of datasets, architectures, metrics, and training details for our experiments. We performed our experiments on NVIDIA GeForce RTX 4090 GPUs, involving training with three random seeds for all datasets, totaling approximately 500 GPU hours including the ablation experiments. We report the mean and standard deviation over three runs.

C.1 DATASETS

In this section, we give details about the datasets used in our experiments Overall, we use five different time series tasks (*heart rate prediction*, *cardiovascular disease classification*, *activity recognition*, *step counting*, *sleep stage classification*) from four sensor modalities (*photoplethysmography*, *electrocardiogram*, *inertial measurement units*, *electroencephalography*). When deciding datasets for our experiments, we put significant emphasis on selecting time series signals that carry valuable information about individuals' mental and physical health. Therefore, we focus on human-generated physiological and wearable sensor signals.

C.1.1 HEART RATE PREDICTION

IEEE SPC Overall, the IEEE SPC dataset has 22 recordings of 22 subjects, ages ranging from 18 to 58 performing three different activities (Rocha et al., 2020). Each recording has sampled data from three accelerometer signals and two PPG signals along with the ECG data with a sampling frequency of 125 Hz. All these recordings were recorded from the wearable device placed on the wrist of each individual. All recordings were captured with two 2-channel PPGs with green LEDs, a tri-axial accelerometer, and a chest ECG for the ground-truth HR estimation. We averaged the two channels of PPG for prediction. Throughout our experiments, we used PPG channels without integrating any inertial measurements.

DaLiA The PPG dataset for motion compensation and heart rate estimation in Daily Life Activities (DaLiA) was recorded from 15 subjects (8 females, 7 males, mean age of 30.6 years old), where each recording was approximately two hours long. PPG signals were recorded while subjects went through different daily life activities, for instance sitting, walking, driving, cycling, working, and so on. PPG signals were recorded at a sampling rate of 64 Hz.

We standardize all PPG datasets as follows, same as the previous works (Biswas et al., 2019; Meier et al., 2024). Initially, a fourth-order Butterworth bandpass filter with a frequency range of 0.5–4 Hz is applied to PPG signals. Subsequently, a sliding window of 8 seconds with 2-second shifts is employed for segmentation, followed by z-score normalization of each segment. Lastly, the signal is resampled to a frequency of 25 Hz for each segment. The validation subject for early stopping and hyperparameter selection is chosen from the training subjects.

C.1.2 HUMAN ACTIVITY RECOGNITION

UCIHAR Human activity recognition using a smartphone's dataset (UCIHAR) (Anguita et al., 2012) is collected by 30 subjects within the age range of 16 to 48 years old performing six daily living activities with a waist-mounted smartphone. Six activities include walking, sitting, lying, standing, walking upstairs, and walking downstairs. Data is captured by 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50 Hz. We used the same pre-processing technique as in (Qian et al., 2021) such that the input contains nine channels with 128 features (it is sampled in a sliding window of 2.56 seconds and 50% overlap, resulting in 128 features for each window). Windows are normalized to a mean of zero and unit standard deviation before feeding it to the models. The experiments are conducted with a leave-one-domain-out strategy using the first ten subjects where one of the first five subjects is chosen to be the unseen target (Qian et al., 2022) while the validation subject is randomly chosen among the rest.

USC USC human activity dataset (USC-HAD) Zhang & Sawchuk (2012) is composed of 14 subjects (7 male, 7 female, aged 21 to 49 years old with a mean of 30.1) executing 12 activities with a sensor on the front right hip. The data dimension is six (3-axis accelerometer, 3-axis gyroscope) and the sample rate is 100 Hz. 12 activities include walking forward, walking left, walking right, walking

upstairs, walking downstairs, running forward, jumping up, sitting, standing, sleeping, elevator up, and elevator down. We used the pre-processing technique with a smaller window size such that the input contains six channels with 100 features (it is sampled in a sliding window of 1 second and 50% overlap, resulting in 100 features for each window). We used the same subject splitting as in Demirel & Holz (2023). The same normalization is also applied to windows before feeding to models.

C.1.3 CARDIOVASCULAR DISEASE (CVD) CLASSIFICATION

Chapman Chapman University, Shaoxing People's Hospital (Chapman) ECG dataset which provides 12-lead ECG with 10-second recordings sampled at 500 Hz. The recordings are downsampled to 100 Hz, resulting in each ECG frame consisting of 1000 samples. The labeling setup follows the same approach as in Zheng et al. (2020) with four classes: atrial fibrillation, GSVT, sinus bradycardia, and sinus rhythm. The ECG frames are normalized to have a mean of 0 and scaled to have a standard deviation of 1. We split the dataset to 60/20–20% for training/validation and testing as suggested in Zheng et al. (2020). We chose leads I, II, III, and V2 during our experiments.

PhysioNet 2017 The 2017 PhysioNet/CinC Challenge aims to classify, from 8,528 single-lead ECG recordings, whether the recording shows normal sinus rhythm, atrial fibrillation (AF), an alternative rhythm, or is too noisy to be classified, i.e., four classes. We normalize the signals to have zero mean and unit standard deviation. Additionally, we zero-pad the shorter recordings to ensure they have the same length. We split the dataset into training, validation, and test sets according to the patients using a 60, 20, 20 configuration.

C.1.4 STEP COUNTING

The Clemson dataset has 30 participants (15 males, 15 females). Each participant wore three Shimmer3 sensors. We used the IMU sensor readings from non-dominant wrists to predict step count where each sensor recorded accelerometer and gyroscope data at 15 Hz. We calculated the total magnitude of the accelerometer and fed it to the model as a pre-processing without any filtering. We used window lengths of 32 seconds without an overlap in the regular walking setting. We conducted 10-fold cross-validation, with each fold consisting of 3 subjects for testing and validation.

Following Demirel & Holz (2025), we discretize the targets for HR estimation and step counting, formulating both tasks as classification problems trained with categorical cross-entropy.

C.1.5 SLEEP STAGE CLASSIFICATION

We used the Sleep-EDF dataset which has five classes: wake (W), three different non-rapid eye movements (N1, N2, N3), and rapid eye movement (REM). The dataset includes whole-night PSG sleep recordings, where we used a single EEG channel (i.e., Fpz-Cz) with a sampling rate of 100 Hz. We employed the identical data split as presented in the paper (Eldele et al., 2021), accessible online, without applying any additional pre-processing steps.

C.2 METRICS

We used the common evaluation metrics in the literature for each task. Specifically, we used mean absolute error (MAE), root mean square error (RMSE), and Pearson correlation coefficient (ρ) for *heart rate prediction*. We used accuracy (Acc), macro-F1 score (F1) for activity recognition, and an additional area under the receiver operating characteristic curve (AUC) for cardiovascular disease classification (Kiyasseh et al., 2020). We used the mean absolute percentage error (MAPE) for step counting (Yang et al., 2020a; Femiano et al., 2022). For sleep stage classification, we used accuracy, macro, and weighted F1 scores (F1, W-F1), along with Cohen's Kappa coefficient (κ) (Cohen, 1960).

C.3 BASELINES

In this section, we give a detailed explanation of each baseline we compared our method to.

C.3.1 AUGMENTATIONS

We used an augmentation module tailored to each data modality, selecting augmentations adopted from previous works (Kiyasseh et al., 2020). For inertial measurement unit and electrocardiogram signals, augmentations such as jittering, scaling, and time flipping were applied. Jittering involved adding Gaussian noise with a standard deviation of 0.01 to simulate minor variations. Scaling was implemented by multiplying the signal with a factor sampled from a normal distribution $(1 + \mathcal{N}(0, 0.1))$ to represent changes in amplitude. For electroencephalography, we applied augmentations including jitter, time flipping, and negating. For photoplethysmography signals, we additionally employed block-wise shuffling, which randomly reorders the sequence of data points to introduce variability, as demonstrated to be effective in previous work Demirel & Holz (2023).

C.3.2 FGSM

The Fast Gradient Sign Method (FGSM) Goodfellow et al. (2015) is a single-step attack that generates adversarial samples by linearizing the loss function and perturbing the input data along the gradient's direction. FGSM is a foundational method for evaluating robustness against adversarial examples, serving as a baseline for simpler perturbation techniques. In our experiments, we have employed both ℓ_2 and ℓ_∞-norm perturbations with different budgets and pick the best one for each application. For FGSM, PGD and TRADES, we followed the Torch implementations similar to Kim (2020).

C.3.3 PGD

Projected Gradient Descent (PGD) Madry et al. (2018) is an iterative adversarial attack that refines FGSM by taking multiple gradient-sign steps and projecting the perturbation back into an ℓ_∞ ball. Given an input x and label y, PGD updates $x^{k+1} = \Pi_{B_\infty(x,\epsilon)}\big(x^k + \alpha\,\mathrm{sign}(\nabla_x \mathcal{L}(f_\theta(x^k), y))\big)$, where ϵ is the perturbation budget and α is the step size. Unlike image settings, our signals are not restricted to $[0, 1]$, so we do not apply pixel-range clipping by default; we only clip to dataset-specific bounds when such bounds are well-defined.

C.3.4 SGM

Smooth Gradient Method (SGM) Pialla et al. (2022) is a smoothing technique to reduce the impact of high-frequency adversarial perturbations. SGM achieves this by incorporating a lasso regularization, as in Equation 19, on perturbations (shown as r in the referred work) to filter out the high-frequency components. We follow the original implementation in `https://github.com/Gpialla/SmoothPerturbationsTSAA/blob/main/attacks/sgm.py` by searching for ϵ for each task. We set the number of iterations to 50 or 100 and smoothness hyperparameter to 0.1.

$$\min_r \Big\{ -\mu\, D_{\mathrm{KL}}\big(f(x), f(x+r)\big) + \alpha\|r\|_2^2 + \lambda \sum_{i=1}^{T-1} \|r_i - r_{i+1}\|_1 \Big\} \tag{19}$$

C.3.5 SAP

Smooth Adversarial Perturbations (SAP) Han et al. (2020) applies a filter whose weights are determined by a Gaussian kernel. Following the original hyperparameter settings, we let s be {5, 7, 11, 15, 19} and σ be {1.0, 3.0, 5.0, 7.0, 10.0}. We follow the official implementation from the authors in `https://github.com/XintianHan/ADV_ECG`.

The smoothing process is applied iteratively during the perturbation generation phase, refining the adversarial perturbations through convolution with the Gaussian kernels each time. While SAP effectively balances perturbation smoothness and adversarial strength, its implementation involves multiple nested loops for perturbation generation, and smoothing operations with filters. This iterative and multi-kernel approach introduces significant computational overhead, particularly when handling large datasets or models with high-dimensional inputs.

The method employs filtering, where the weights of the filter v, are determined using a Gaussian kernel. For a Gaussian kernel of size $2K+1$ and standard deviation σ, the weights are defined as:

$$v = \exp\left(-\frac{(m-K-1)^2}{2\sigma^2}\right) / \sum_{i=1}^{2K+1} \exp\left(-\frac{(i-K-1)^2}{2\sigma^2}\right).$$

In the limiting cases, as $\sigma \to \infty$, the convolution with the Gaussian kernel approaches a simple average (smoothing). Conversely, as $\sigma \to 0$, the Gaussian kernel approximates a Dirac delta function, prioritizing the central value of the kernel. Although these parameters can be tailored to specific time series tasks using prior knowledge, the search space for identifying optimal filter parameters is extensive; therefore, we follow the original implementation.

C.3.6 LF-AP

Low-frequency adversarial perturbations (LF-AP) (Guo et al., 2020) restrict adversarial optimization to a low-frequency DCT subspace. We adapt the gradient-based variant to the temporal domain, optimizing the retained DCT coefficients using gradients of the adversarial objective and reconstructing temporal perturbations through the inverse DCT.

The frequency r controls the fraction of retained DCT coefficients. We performed a grid search over r to decide the lower half of the frequency range for each task.

This implementation uses a fixed low-frequency subspace across inputs. Our method instead learns input-conditioned frequency weighting, allowing adversarial updates to adapt to each signal's spectral characteristics without a predefined low-frequency cutoff.

C.3.7 F-VAT

Functional Virtual Adversarial Training (f-VAT) Pan & Li (2025) constructs virtual adversarial perturbations in a functional space rather than using the standard Euclidean input norm. Since there is no official repository of this work, we have implemented the algorithm as follows. For a multichannel time series $\mathbf{x} \in \mathbb{R}^{T\times C}$, we use the discrete negative-Sobolev norm

$$\|\mathbf{r}\|_{H^{-s}}^2 = \sum_{c=1}^{C}\sum_{k=1}^{T}(1+\lambda_k)^{-s}\,|\widehat{r}_{k,c}|^2, \qquad \lambda_k = 2-2\cos\left(\frac{\pi k}{T+1}\right), \tag{20}$$

where $\widehat{\mathbf{r}}$ denotes the orthonormal type-I discrete sine transform (DST-I) of the perturbation along the temporal dimension. Thus, $s \geq 0$ controls the functional geometry of the adversary, with larger values reducing the penalty on high-frequency perturbations relative to low-frequency perturbations.

For each input, f-VAT seeks a perturbation that maximally changes the predictive distribution while satisfying $\|\mathbf{r}\|_{H^{-s}} \leq \epsilon$. We use the squared ℓ_2 distance between categorical prediction vectors as the local distributional smoothness (LDS) term.

$$\mathcal{L}_{\mathrm{LDS}}(\mathbf{x},\mathbf{r};\theta) = \|\mathrm{softmax}(f_\theta(\mathbf{x})) - \mathrm{softmax}(f_\theta(\mathbf{x}+\mathbf{r}))\|_2^2, \tag{21}$$

$$\mathcal{L}_{\text{f-VAT}}(\theta) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\mathrm{train}}}\left[\mathcal{L}_{\mathrm{CE}}(f_\theta(\mathbf{x}),y) + \lambda_{\mathrm{fVAT}}\mathcal{L}_{\mathrm{LDS}}(\mathbf{x},\mathbf{r}_{\mathrm{adv}};\theta)\right]. \tag{22}$$

The adversary is initialized from uniform noise and normalized to the H^{-s} sphere. We use a Sobolev-gradient adaptation in which the Euclidean gradient is preconditioned by $(I+\mathbf{L})^s$ before normalization, where $\mathbf{L}$ is the discrete negative Laplacian diagonalized by the DST-I. The update is

$$\mathbf{r} \leftarrow \Pi_{\|\cdot\|_{H^{-s}}=\epsilon}\left[\mathbf{r} + \eta(I+\mathbf{L})^s\nabla_{\mathbf{r}}\mathcal{L}_{\mathrm{LDS}}(\mathbf{x},\mathbf{r};\theta)\right], \qquad \eta = 1.0, \tag{23}$$

where $\Pi_{\|\cdot\|_{H^{-s}}=\epsilon}(\mathbf{v}) = \epsilon\mathbf{v}/\|\mathbf{v}\|_{H^{-s}}$ denotes radial normalization for nonzero $\mathbf{v}$. The preconditioner converts the Euclidean gradient into the gradient associated with the H^{-s} inner product. This is an implementation adaptation of the gradient-ascent step in Algorithm 1 of Pan & Li (2025), which specifies ascent followed by Sobolev normalization without an explicit preconditioner. Euclidean gradient ascent with an appropriate projection can also optimize the same constrained objective.

The Sobolev norm is computed jointly over all temporal locations and input channels; no channel is perturbed independently. We use an orthonormal DST-I implementation and introduce no additional f-VAT-specific amplitude normalization beyond the preprocessing already used by each dataset. The adversary is generated with the network in evaluation mode, with BatchNorm running statistics fixed and dropout disabled during inner optimization.

C.3.8 TRADES

TRADES (TRadeoff-inspired Adversarial DEfense via Surrogate-loss) Zhang et al. (2019) balances the trade-off between accuracy and robustness by incorporating a regularization term into the training objective. This regularizer penalizes the discrepancy between clean and adversarial predictions, aiming to improve robustness without excessively degrading clean accuracy. The proposed regularizer is used with a hyper-parameter term β as in Equation 24.

$$\mathcal{L}_{\text{TRADES}}(\theta; \mathbf{x}, y) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big[\mathcal{L}_{\text{CE}}(f_\theta(\mathbf{x}), y) + \beta \cdot \sup_{\mathbf{x}'\in\mathcal{B}_\epsilon(\mathbf{x})} \mathcal{L}_{\text{KL}}(f_\theta(\mathbf{x}) \| f_\theta(\mathbf{x}'))\Big] \tag{24}$$

During our experiments, we searched for the β values from the set $\{4, 5, 6\}$.

C.3.9 FA

The frequency adversarial attack (FA) Jia et al. (2022) optimizes an additive perturbation P in the DCT domain using gradients of the adversarial loss. An input-dependent weighting matrix M adjusts the perturbation across frequency bands. The original formulation constrains the DCT-domain difference between the original and adversarial examples through an ℓ_p norm. Following its blockwise design, we adapt FA to time series by splitting each sequence into four non-overlapping segments. We use a per-timestamp perturbation budget of 0.2, analogous to the per-pixel bound. While FA adapts its frequency weighting to the input, it does not regularize the perturbation spectrum toward the true input spectrum. Our method introduces this alignment through a filter and spectral regularization.

C.3.10 F-SAT

Frequency-selective adversarial training (F-SAT) (Zhang et al., 2025) modifies STFT magnitudes within a predefined frequency band while retaining the original synthesis phase. We adapt it to multichannel time series by applying the STFT independently to each channel. Magnitude offsets are initialized randomly and optimized using sign-gradient ascent, with offsets outside the selected band fixed to zero. After each update, we enforce nonnegative magnitudes and, when necessary, uniformly contract the offsets to satisfy the reconstructed waveform perturbation budget. We add the difference between the modified and original inverse STFT reconstructions to the input, avoiding clean STFT–ISTFT round-trip error.

We select application-specific cutoffs through a dyadic grid search capped at the Nyquist frequency. Cutoffs, perturbation budgets, attack lengths, and step sizes are selected using validation performance. Unlike our input-conditioned spectral regularization, this adaptation modifies input STFT magnitudes within a band that remains fixed across examples. Therefore, it requires setting up heuristic cut-off frequencies while applying expensive search techniques.

C.3.11 DARG

Disentangling Adversarial Robustness and Generalization (DARG) Stutz et al. (2019) leverages manifold learning to address the drop in the accuracy. Authors used VAE-GANs Larsen et al. (2016) to approximate the data distribution, perturbing latent vectors to generate adversarial examples constrained to the learned manifold. For each task, we employed VAEs as in Demirel & Holz (2023). The discriminator architecture consists of a 3-layer convolutional network followed by a linear layer. VAE-GAN loss combines three components; reconstruction, class-based discriminator, and KL divergence loss for latent space regularization.

C.3.12 IFAP Adversarial Training

Input-Frequency Adaptive Adversarial Perturbation (IFAP) (Yoshida & Okuda, 2025) restricts spectral changes using an input-dependent envelope. We adapt it using an orthonormal one-dimensional DCT along the temporal axis of each channel. Starting from the clean input without random initialization, each iteration performs a sign-gradient ascent step followed by projection. We then clip the DCT coefficients of the resulting perturbation element-wise to $[-\eta|\mathcal{D}(\mathbf{x})|, \eta|\mathcal{D}(\mathbf{x})|]$, where $\mathcal{D}$ denotes the temporal DCT and η controls the envelope width. The clipped perturbation is transformed back to the time domain and projected again to enforce the norm budget. Unlike our learned spectral weighting, this baseline prescribes the spectral envelope from the input.

C.4 Implementation Details

C.4.1 Architectures

We use an eight-block ResNet1D classifier (He et al., 2015; Hong et al., 2020) for our method and all baselines in the main experiments. The network uses same-padded convolutions with kernel size 5, an initial width of 32 channels, and stride 2 in the first convolution of every second residual block. The channel width increases from 32 to 64 in the fifth block. Each residual block contains two convolutions, with batch normalization (Ioffe & Szegedy, 2015) and ReLU activations. Dropout (Srivastava et al., 2014) with probability 0.5 follows the internal block activations. Global average pooling produces a 64-dimensional representation before the final linear classifier. Table 8a reports the architecture and dataset-specific parameter counts, where C, T, and K denote the number of input channels, sequence length, and output classes.

We use a batch size of 64 and a learning rate of 0.001, except for step counting, where the learning rate is 0.003. The spectral mask network uses the same learning rate as the classifier.

Table 8: Architectures and trainable parameter counts. C, T, K, and F denote input channels, sequence length, output classes, and frequency bins, respectively.

(a) ResNet1D classifier

Layer	Output Size	Strides
Input	(C, T)	–
Conv1D	$(32, T)$	1
Residual Block 1	$(32, T)$	1, 1
Residual Block 2	$(32, \lceil T/2 \rceil)$	2, 1
Residual Block 3	$(32, \lceil T/2 \rceil)$	1, 1
Residual Block 4	$(32, \lceil T/4 \rceil)$	2, 1
Residual Block 5	$(64, \lceil T/4 \rceil)$	1, 1
Residual Block 6	$(64, \lceil T/8 \rceil)$	2, 1
Residual Block 7	$(64, \lceil T/8 \rceil)$	1, 1
Residual Block 8	$(64, \lceil T/16 \rceil)$	2, 1
Global Avg. Pooling	$(64,)$	–
Linear	$(K,)$	–
# Parameters for *dataset* (C, T, K)		
IEEE SPC12 $(1, 200, 180)$		208,884
IEEE SPC22 $(1, 200, 180)$		208,884
DaLiA $(1, 200, 190)$		209,534
UCIHAR $(9, 128, 6)$		198,854
USC $(6, 100, 12)$		198,764
Clemson $(1, 480, 48)$		200,304
Chapman $(4, 1000, 4)$		197,924
PhysioNet $(1, 6000, 4)$		197,444
Sleep-EDF $(1, 3000, 5)$		197,509

(b) Spectral mask network H_ψ

Layer	Mapping	Output Size
Input	–	(C, F)
Linear	$F \to 32$	$(C, 32)$
ReLU	–	$(C, 32)$
Linear	$32 \to 32$	$(C, 32)$
ReLU	–	$(C, 32)$
Linear	$32 \to F$	(C, F)
Sigmoid	elementwise	(C, F)
# Parameters for *dataset* (C, F)		
IEEE SPC12 $(1, 101)$		7,653
IEEE SPC22 $(1, 101)$		7,653
DaLiA $(1, 101)$		7,653
UCIHAR $(9, 65)$		5,313
USC $(6, 51)$		4,403
Clemson $(1, 241)$		16,753
Chapman $(4, 501)$		33,653
PhysioNet $(1, 3001)$		196,153
Sleep-EDF $(1, 1501)$		98,653

All classifier convolutions use kernel size 5. Each residual block contains two convolutions with strides listed in order. Classifier counts include biases and batch-normalization affine parameters. The mask network shares weights across channels and has $65F + 1088$ parameters, where $F = \lfloor T/2 \rfloor + 1$ for full sequences.

The spectral mask network H_ψ is a three-layer MLP with two 32-dimensional hidden layers and ReLU activations. A final sigmoid restricts the mask values to $(0, 1)$. The MLP operates independently on each channel's magnitude spectrum, with weights shared across channels. For a full sequence of length T, the one-sided real FFT produces $F = \lfloor T/2 \rfloor + 1$ frequency bins. The mask network therefore has $65F + 1088$ trainable parameters, including biases, independently of the number of channels. Table 8b reports its architecture and dataset-specific parameter counts.

C.4.2 HYPERPARAMETERS

In this section, we detail the hyperparameter settings for each method. During our experiments, we searched for ϵ within the ranges $\{0.2, 0.25, 0.3\}$ and $\{1, 1.5, 2, 2.5, 3\}$ for both ℓ_2 and ℓ_∞-norms. We also searched which perturbation norm performs best for each method, and unless otherwise specified, the infinity norm is used. The number of steps for generating perturbations was selected through search over $\{1, 3, 5, 10, 15, 20, 30\}$ for methods. Step sizes (α) were determined as a fraction of ϵ, specifically the ratios $\{1/3, 1/4\}$ and $0.1 \times \{1/3, 1/4\}$. We randomly sampled the α values within a short interval for each task, searching over a grid of multiples of 2 and 4, starting from 0.01.

If the final perturbation exceeds the budget, we uniformly rescale it to ϵ. For methods that have additional hyperparameters, we follow a grid search. For β value at TRADES, we searched over $\{4, 5, 6\}$. The parameters for filter in SAP are set to the original ones. We set the λ_{KL} term to 1 during our experiments. Hyperparameters are selected using clean validation performance as it is one of our main objectives. We assess robustness under the evaluation attacks and use clinician assessments to examine the plausibility of the generated examples. As evident from the tables,

Table 9: Hyperparameters for methods in HR prediction across datasets

Method	*IEEE SPC12*	*IEEE SPC22*	*DaLiA*
FGSM	$\epsilon = 0.25$, $\alpha = 0.25/3$	$\epsilon = 0.3$, $\alpha = 0.3/3$	$\epsilon = 0.3$, $\alpha = 0.3/3$
PGD	$\epsilon = 0.3$, $\alpha = 0.3/4$, ℓ_2, steps=10	$\epsilon = 1$, $\alpha = 1/4$, ℓ_2, steps=15	$\epsilon = 0.3$, $\alpha = 0.3/3$, ℓ_2, steps=10
SGM	$\epsilon = 0.3$, $\alpha = 0.01$, iterations = 100	$\epsilon = 1$, $\alpha = 0.01$, iterations = 100	$\epsilon = 0.3$, $\alpha = 0.01$, iterations = 100
SAP	$\epsilon = 1$, steps 10	$\epsilon = 0.3$, steps 5	$\epsilon = 0.3$, steps 5
LF-AP	$r = 8$ Hz, $\epsilon = 0.3$, steps 20	$r = 8$ Hz, $\epsilon = 0.3$, steps 20	$r = 8$ Hz, $\epsilon = 0.3$, steps 20
f-VAT	$\epsilon_{H^{-s}} = 1$, $\eta = 1$, steps=10, $\lambda_{\text{fVAT}} = 1$, $s = 1$	$\epsilon_{H^{-s}} = 1$, $\eta = 1$, steps=10, $\lambda_{\text{fVAT}} = 1$, $s = 1$	$\epsilon_{H^{-s}} = 1$, $\eta = 1$, steps=10, $\lambda_{\text{fVAT}} = 1$, $s = 1$
F-SAT	$\epsilon = 0.3$, $\alpha = 0.075$, steps=20, $B_f = [1, 6.5]$ Hz	$\epsilon = 0.3$, $\alpha = 0.075$, steps=20, $B_f = [1, 6.5]$ Hz	$\epsilon = 0.3$, $\alpha = 0.075$, steps=20, $B_f = [1, 6.5]$ Hz
TRADES	$\epsilon = 0.2$, $\alpha = 0.2/4$, $\beta = 4$	$\epsilon = 0.2$, $\alpha = 0.2/4$, $\beta = 4$	$\epsilon = 0.2$, $\beta = 4$
DARG	$\alpha = 0.3/3$, steps 5	$\alpha = 0.3/3$, steps 5	$\alpha = 0.2/3$ steps 3
IFAP	$\epsilon = 0.3$, $\alpha = 0.075$, steps=20, $\eta = 0.3$	$\epsilon = 0.3$, $\alpha = 0.075$, steps=20, $\eta = 0.3$	$\epsilon = 0.3$, $\alpha = 0.075$, steps=20, $\eta = 0.3$
Ours	$\epsilon = 0.3$, $\alpha \sim U(0.01, 0.03)$, steps 20	$\epsilon = 1$, $\alpha \sim U(0.01, 0.03)$, steps 20	$\epsilon = 0.3$, $\alpha \sim U(0.01, 0.03)$, steps 20

Table 10: Hyperparameters for methods in tasks regarding IMU

Method	*UCIHAR*	*USC*	*Clemson*
FGSM	$\epsilon = 1$, $\alpha = 1/3$	$\epsilon = 0.3$, $\alpha = 0.3/3$	$\epsilon = 1$, $\alpha = 1/3$
PGD	$\epsilon = 2$, $\alpha = 2/3$, ℓ_2, steps=20	$\epsilon = 0.3$, $\alpha = 0.3/3$, steps=15	$\epsilon = 0.3$, $\alpha = 0.4/3$, steps=10
SGM	$\epsilon = 0.3$, $\alpha = 0.01$, iterations = 100	$\epsilon = 0.25$, $\alpha = 0.01$, iterations = 100	$\epsilon = 0.3$, $\alpha = 0.01$, iterations = 50
SAP	$\epsilon = 0.3$, steps 10	$\epsilon = 0.2$, steps 5	$\epsilon = 1$, steps 5
LF-AP	$r = 25$ Hz, $\epsilon = 0.2$, steps 10	$r = 50$ Hz, $\epsilon = 0.3$, steps 20	$r = 7$ Hz, $\epsilon = 0.3$, steps 20
f-VAT	$\epsilon_{H^{-s}} = 1$, $\eta = 1$, steps=10, $\lambda_{\text{fVAT}} = 1$, $s = 1$	$\epsilon_{H^{-s}} = 1$, $\eta = 1$, steps=10, $\lambda_{\text{fVAT}} = 1$, $s = 1$	$\epsilon_{H^{-s}} = 1$, $\eta = 1$, steps=5, $\lambda_{\text{fVAT}} = 1$, $s = 1$
F-SAT	$\epsilon = 0.3$, $\alpha = 0.075$, steps=30, $B_f = [1, 18]$ Hz	$\epsilon = 0.3$, $\alpha = 0.075$, steps=30, $B_f = [1, 18]$ Hz	$\epsilon = 0.3$, $\alpha = 0.075$, steps=30, $B_f = [1, 5.5]$ Hz
TRADES	$\epsilon = 0.2$, $\alpha = 0.2/3$, $\beta = 4$	$\epsilon = 0.25$, $\alpha = 0.25/3$, $\beta = 5$	$\epsilon = 0.3$, $\alpha = 0.3/3$, $\beta = 4$
DARG	$\alpha = 0.3/3$, steps 5	$\alpha = 0.25/3$, steps 5	$\alpha = 0.3/3$, steps 10
IFAP	$\epsilon = 1$, $\alpha = 0.075$, steps=20, $\eta = 0.3$	$\epsilon = 0.3$, $\alpha = 0.075$, steps=10, $\eta = 0.3$	$\epsilon = 0.3$, $\alpha = 0.075$, steps=5, $\eta = 0.3$
Ours	$\epsilon = 1$, $\alpha \sim U(0.03, 0.06)$, steps 10	$\epsilon = 0.3$, $\alpha \sim U(0.03, 0.06)$, steps 10	$\epsilon = 0.3$, $\alpha \sim U(0.03, 0.06)$, steps 5

Table 11: Hyperparameters for methods across tasks

Method	*Chapman*	*PhysioNet*	*Sleep*
FGSM	$\epsilon = 0.3$, $\alpha = 0.3/3$	$\epsilon = 0.3$, $\alpha = 0.3/3$	$\epsilon = 0.3$, $\alpha = 0.3/3$
PGD	$\epsilon = 0.3$, $\alpha = 0.3/3$, steps=10	$\epsilon = 0.15$, $\alpha = 0.15/3$, steps=10	$\epsilon = 0.2$, $\alpha = 0.2/3$, ℓ_2, steps=15
SGM	$\epsilon = 0.2$, $\alpha = 0.01$, iterations = 100	$\epsilon = 0.2$, $\alpha = 0.01$, iterations = 50	$\epsilon = 2$, $\alpha = 0.01$, iterations = 100
SAP	$\epsilon = 0.3$, steps 1	$\epsilon = 0.3$, steps 1	$\epsilon = 0.3$, steps 1
LF-AP	$r = 50$ Hz, $\epsilon = 0.3$, steps 20	$r = 50$ Hz, $\epsilon = 0.3$, steps 20	$r = 50$ Hz, $\epsilon = 0.3$, steps 20
f-VAT	$\epsilon_{H^{-s}} = 1$, $\eta = 1$, steps=1, $\lambda_{\text{fVAT}} = 1$, $s = 3$	$\epsilon_{H^{-s}} = 1$, $\eta = 1$, steps=1, $\lambda_{\text{fVAT}} = 1$, $s = 3$	$\epsilon_{H^{-s}} = 1$, $\eta = 1$, steps=1, $\lambda_{\text{fVAT}} = 1$, $s = 3$
F-SAT	$\epsilon = 0.3$, $\alpha = 0.075$, steps=10, $B_f = [1, 18]$ Hz	$\epsilon = 0.3$, $\alpha = 0.075$, steps=10, $B_f = [1, 20]$ Hz	$\epsilon = 0.3$, $\alpha = 0.075$, steps=15, $B_f = [1, 20]$ Hz
TRADES	$\epsilon = 0.25$, $\alpha = 0.25/3$, $\beta = 5$	$\epsilon = 0.25$, $\alpha = 0.25/3$, $\beta = 5$	$\epsilon = 0.25$, $\alpha = 0.25/3$, $\beta = 5$
DARG	$\alpha = 0.3/3$, steps 3	$\alpha = 0.3/3$, steps 1	$\alpha = 0.3/3$, steps 1
IFAP	$\epsilon = 0.3$, $\alpha = 0.075$, steps=1, $\eta = 0.3$	$\epsilon = 0.3$, $\alpha = 0.075$, steps=1, $\eta = 0.3$	$\epsilon = 0.3$, $\alpha = 0.075$, steps=1, $\eta = 0.3$
Ours	$\epsilon = 0.3$, $\alpha \sim U(0.01, 0.03)$, steps 1	$\epsilon = 0.3$, $\alpha \sim U(0.01, 0.03)$, steps 1	$\epsilon = 0.3$, $\alpha \sim U(0.01, 0.03)$, steps 1

the coarse validation search identifies effective settings while sensitivity varies by dataset. For the windowed variant, we use four equal, non-overlapping windows across all tasks. We apply Hann tapering to the perturbations before concatenation to reduce boundary discontinuities. The same norm budget ϵ is enforced on the complete concatenated perturbation as in the full-sequence variant.

One interesting result from the search is the relatively high step size observed for most tasks, compared to typical values like 0.01. This finding is interesting, as temporal data often lacks universal box constraints, yet the ratio tends to remain consistent (e.g., a 2/255 step size for an 8/255 perturbation in 8-bit representation with [0–1] norm).

The parameter α specifies the update magnitude relative to the input norm, as defined in Equation 7, rather than an absolute step size. For the one-step configurations on Chapman, PhysioNet, and Sleep-EDF, $\alpha\|\boldsymbol{x}\|_2$ exceeds $\epsilon = 0.3$, so projection sets the final perturbation norm to this budget while preserving the learned filtered direction. Consequently, different values of α that reach the budget produce the same update from the clean input. These configurations therefore use ϵ to control perturbation magnitude and the filter to determine its spectral allocation. In the next section, we also showed further evidence for the robustness of our method to small changes in hyperparameters.

D Additional Results and Details Regarding Robustness

D.1 Significance Analysis

Individual baselines can perform well on particular datasets or metrics, but their performance varies considerably across tasks. For example, PGD is competitive on some ECG metrics in Table 3, while performing less consistently on the activity-recognition and step-counting tasks in Table 2. We therefore evaluate whether a method performs consistently across the complete benchmark rather than relying on isolated results.

We compare the methods over 27 dataset–metric settings spanning heart-rate estimation, activity recognition, step counting, ECG classification, and sleep staging. Each reported score is first averaged over three paired random seeds. Methods are then ranked independently within each setting, with rank one assigned to the best method and ties assigned their average rank. Consequently, the experimental units in this analysis are the 27 dataset–metric settings; the three seeds reduce estimation variance but are not treated as additional independent observations. We apply the Friedman omnibus test followed by the Nemenyi post-hoc test at $\alpha = 0.05$. We include only the default full-sequence configuration of our method because the windowed variant is a secondary, task-dependent configuration.

As shown in Figure 7, our method obtains the best average rank of 1.33, substantially ahead of the second-ranked method, FGSM, with an average rank of 5.31. The Friedman test rejects the null hypothesis of equivalent performance among the methods. These results show that its advantage is consistent across task families and evaluation criteria rather than being driven by a single benchmark.

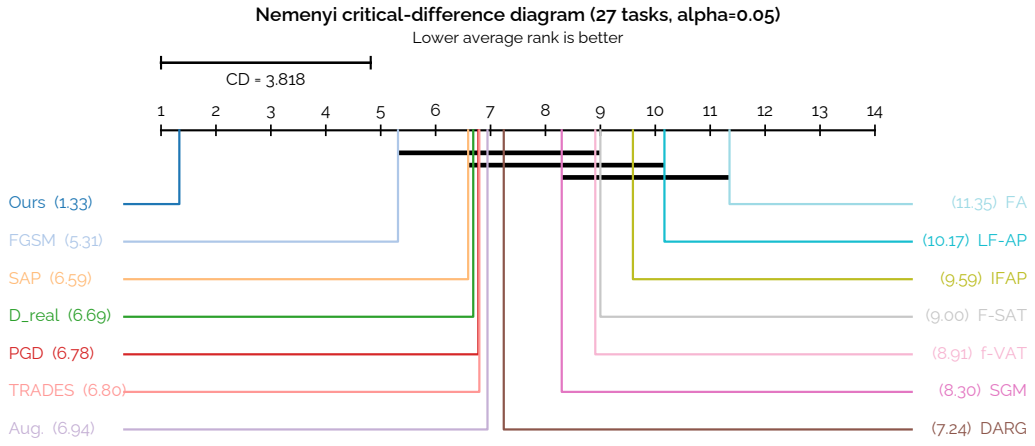


Figure 7: Critical-difference diagram over all 27 dataset–metric settings. Numbers denote average ranks, where lower is better. Horizontal bars connect methods that are not significantly different according to the Nemenyi post-hoc test at $\alpha = 0.05$. Our method achieves the best average rank and is significantly better than every competing method.

Because metrics obtained from the same dataset are statistically related, we also perform a conservative dataset-level analysis in which each dataset contributes one representative endpoint. The analysis uses Pearson correlation for the three heart-rate datasets, weighted F1 for activity recognition, RMSE for Clemson, AUC for Chapman and PhysioNet, and accuracy for Sleep-EDF. Scores are again averaged over the three seeds before ranking. Our method remains top-ranked, with an average rank of 1.11 compared with 4.89 for the next-ranked method. The Friedman test remains significant.

As a complementary comparison-versus-control analysis, we test whether our method consistently outperforms each baseline across the nine datasets. For every baseline, we record the number of datasets on which our method achieves better performance and apply an exact one-sided paired sign test. We control the family-wise error rate across the 13 comparisons using Holm's correction. All improvements remain statistically significant after correction. This analysis complements the Nemenyi test and provides further evidence that the improvement of our method is consistent across datasets rather than being driven by a small number of favorable tasks.

D.2 ROBUSTNESS OF HYPERPARAMETERS

In this section, we give the detailed results of our method when it is evaluated with different hyperparameters to evaluate its robustness. It is important to note that hyperparameter optimization was performed using the validation sets. Although both the perturbation magnitude and the number of steps are consistently set for our method, we further investigate the impact of varying the steps, as shown in the tables below. For each dataset, we also report the optimal performance achieved by fine-tuning both the perturbation magnitude and the number of steps on a finer grid, with single-step increments (1 by 1) for the number of steps and 0.01 increments for the magnitude. Since this finer grid search is exhaustive, we report the initial search we performed, which considers 4–5 parameters only, in the main results.

Tables 12, 13, and 14 summarize the results. For methods with ablation studies on specific parameters, the tables include three values for each dataset in a consistent order. To simplify comparisons across methods with numerous results, we highlight the performance of standard techniques under varying hyperparameter configurations.

Table 12: Performance comparison of ours with other methods for HR estimation

Method	IEEE SPC12			IEEE SPC22			DaLiA		
	MAE↓	RMSE↓	ρ↑	MAE↓	RMSE↓	ρ ↑	MAE↓	RMSE↓	ρ↑
Standard training									
Only $\mathcal{D}_{\text{real}}$	5.27±0.21	11.47±0.69	83.86±1.31	5.76±0.07	10.70±0.18	81.16±0.43	5.53±0.067	12.47±0.17	85.65±0.20
Standard techniques									
Aug.	5.54±0.22	11.63±0.84	84.38±1.10	5.31±0.31	10.77±0.64	81.32±0.81	5.52±0.04	12.46±0.09	85.67±0.15
FGSM (orig. ϵ) (.25 — .3 — .3)	5.59±0.22	12.03±0.74	83.82±0.65	5.45±0.11	10.57±0.22	82.70±0.35	5.43±0.08	12.26±0.10	86.04±0.07
FGSM (ablation) (.3 — .25 — .25)	5.91±0.25	12.45±0.56	81.78±1.05	6.07±0.36	11.60±0.22	81.50±0.80	5.47±0.03	12.54±0.03	86.03±0.10
FGSM (ablation) (.2 — .2 — .2)	6.02±0.28	12.50±0.19	82.31±0.55	6.07±0.36	11.93±0.22	81.32±0.80	5.41±0.06	12.23±0.14	86.03±0.17
PGD (orig. ϵ) (.3 — 1 — .3)	5.97±0.67	12.71±1.15	81.25 ±2.20	6.19 ±0.41	11.71 ±0.65	80.79 ±1.11	5.62 ±0.03	12.45 ±0.12	85.50 ±0.05
PGD (ablation ϵ) (1 — .3 — 1)	6.20±0.46	13.04±0.52	81.44±0.23	9.53±0.51	16.17±1.18	72.10±2.20	5.94 ±0.04	12.91 ±0.12	84.30 ±0.22
PGD (ablation ϵ) (.2 — 2 — .2)	5.59±0.27	12.96±0.70	80.61±1.09	6.26±0.94	11.97±1.47	81.53±2.50	5.60±0.03	12.45±0.11	85.54±0.25
PGD (Ablation step) (20 — 20 — 20) & (ϵ=.3—1—.3)	5.63±0.21	13.04±0.80	79.33±1.01	6.40±0.97	12.57±1.31	80.11±2.33	5.67±0.04	12.53±0.09	85.43±0.21
Smoothing (low-pass)									
SAP	6.68±0.59	13.33±0.89	82.07±1.54	5.41±0.55	10.80±1.00	82.38±1.01	5.40±0.08	12.25±0.18	86.05±0.25
SGM	5.85±0.21	12.06±0.19	83.35±0.90	5.29±0.26	10.65±1.40	81.35±1.72	5.68±0.04	12.63±0.10	84.84±0.02
LF-AP	6.13±0.18	12.87±0.22	83.02±0.87	5.37±0.28	10.87±1.43	81.69±1.73	5.65±0.08	12.47±0.13	85.12±0.09
Clean accuracy-focused									
TRADES	6.18±0.50	12.50±0.66	82.22±0.61	5.98±0.35	11.43±0.91	80.70±1.42	5.54±0.10	12.43±0.16	85.52±0.30
f-VAT ($s = 1$)	5.35±0.22	11.13±0.90	84.70±1.30	8.83±0.88	17.42±2.33	64.90±8.50	5.91±0.51	13.11±0.14	84.91±0.19
f-VAT ($s = 2$)	5.49±0.05	11.15±0.32	85.20±0.40	9.31±0.99	18.60±2.94	57.80±9.60	5.95±0.56	13.25±0.11	84.87±0.15
f-VAT ($s = 3$)	5.49±0.02	11.40±0.52	83.80±1.10	9.51±0.53	19.03±1.72	57.50±8.70	5.98±0.47	13.05±0.12	84.84±0.16
FA	6.01±0.24	12.25±0.23	82.07±0.27	6.55±0.48	12.58±0.37	76.80±2.30	6.60±0.39	14.70±0.58	80.38±0.77
F-SAT	5.69±0.32	12.19±0.60	82.84±0.68	6.80±0.17	13.17±0.61	77.87±1.29	5.65±0.05	12.30±0.08	85.90±0.03
DARG	6.07±0.27	12.68±0.32	82.01±0.40	5.28±0.27	10.89±1.23	81.22±1.57	5.38±0.07	12.15±0.14	86.20±0.32
IFAP	5.71±0.30	11.62±0.31	83.70±0.39	5.04±0.15	15.26±0.53	74.30±1.10	5.55±0.05	12.68±0.07	84.40±0.10
Ours (orig. steps) (20 — 20 — 20)	**5.19**±0.10	**10.59**±0.50	**85.47**±0.60	4.77±0.98	**9.76**±1.50	**85.90**±1.79	5.23±0.09	**11.85**±0.25	**86.73**±0.50
Ours (ablation) (30 — 30 — 30)	5.21±0.15	10.88±0.56	85.46±0.61	**4.71**±0.19	9.82±0.37	85.89±0.35	**5.22**±0.01	11.89±0.08	86.62±0.36
Ours (ablation) (10 — 10 — 10)	5.28±0.18	10.97±0.40	85.02±0.83	4.82±0.60	9.90±1.19	84.55±1.90	5.43±0.06	12.25±0.10	86.10±0.20
Ours (ablation) (optimal*)	4.90±0.50	10.30±0.53	87.10±0.70	4.63±0.45	9.19±1.15	85.92±1.20	5.20±0.06	11.50±0.25	86.83±0.17

As shown in Table 13, specific hyperparameters yield the best performance for certain datasets. For example, five attack iterations in USC outperforms the other parameters. However, to ensure a fair and realistic setup, we conducted a task-based hyperparameter search using a relatively coarse grid scheme.

Table 13: Performance comparison of ours with other methods using different hyperparameters for Activity and Step

Method	UCIHAR			USC			Clemson		
	Acc↑	W-F1↑	F1↑	Acc↑	W-F1↑	F1 ↑	MAPE↓	MAE↓	RMSE↓
Standard training									
Only $\mathcal{D}_{\text{real}}$	89.89±1.76	88.08±1.80	89.37±1.82	73.49±1.59	72.22±1.74	71.04±1.43	4.04±0.30	2.28±0.21	3.50±0.23
Standard techniques									
Aug.	89.60±0.95	88.22±1.08	89.30±1.13	73.65±1.47	72.10±1.73	71.31±1.37	4.60±0.60	2.58±0.32	3.93±0.45
FGSM (orig. ϵ) (1 — .3 — 1)	90.07±1.12	88.75±1.17	89.54±1.47	74.20±1.10	72.51±1.75	72.34±1.34	3.62±0.34	2.02±0.21	3.17±0.24
FGSM (ablation) (.3 — 3 — .2)	89.90±2.45	88.55±2.76	89.09±3.10	72.98±1.07	70.99±1.28	71.10±1.42	4.43±0.72	2.50±0.44	3.67±0.44
FGSM (ablation) (2 — .2 — 2)	90.01±1.35	88.17±1.64	89.54±1.62	74.61±0.64	72.83±0.71	72.48±0.61	4.06±0.53	2.27±0.32	3.48±0.32
PGD (orig. ϵ) (2 — .3 — .3)	89.18±3.43	87.92±4.05	89.42±3.35	75.63±0.74	74.07±1.12	73.57±0.96	4.47±1.44	2.53±0.87	3.84±1.09
PGD (orig. ϵ) (.2 — .1 — .2)	88.22±0.96	87.32±1.37	88.19±1.14	73.99±1.42	72.54±1.55	71.93±1.12	4.59±1.03	2.60±0.60	3.95±0.87
PGD (orig. ϵ) (1 — 1 — 1)	88.30±1.61	86.30±1.18	87.93±1.28	51.82±2.53	48.21±2.33	45.35±1.18	4.20±0.39	2.37±0.23	3.63±0.36
PGD (Ablation step) (10 — 10 — 5) & (ϵ=1—.3—.3)	88.12±1.53	86.30±1.10	86.35±1.13	50.16±2.21	47.19±2.01	44.30±1.13	4.84±0.31	2.43±0.19	3.80±0.35
Smoothing (low-pass)									
SAP	91.79±0.95	90.91±0.96	92.01±0.81	76.09±0.81	74.82±1.03	73.79±1.12	4.34±0.35	2.44±0.19	3.70±0.21
SGM	88.87±3.41	87.00±4.83	88.14±4.43	70.70±1.06	69.20±0.78	68.53±1.29	4.47±0.42	2.53±0.26	3.73±0.23
LF-AP	86.18±2.23	86.65±0.21	87.12±1.98	72.65±0.86	72.18±1.65	73.16±1.65	4.12±0.43	2.68±0.13	3.75±0.03
Clean accuracy-focused									
TRADES	91.49±0.73	90.90±0.55	92.13±0.37	76.50±0.75	75.18±0.90	74.57±0.83	3.30±0.31	1.84±0.22	2.96±0.18
f-VAT ($s = 1$)	90.29±1.13	88.97±1.20	89.96±1.59	75.22±1.07	74.15±1.29	73.04±1.42	4.69±0.68	2.70±0.46	3.78±0.43
f-VAT ($s = 2$)	90.06±1.84	88.75±2.14	89.52±2.37	75.57±2.20	74.25±2.97	73.07±2.12	4.92±0.67	2.84±0.44	3.91±0.45
f-VAT ($s = 3$)	88.70±2.01	87.20±2.39	88.28±2.39	75.80±2.97	74.40±3.71	73.32±2.67	4.83±0.78	2.79±0.51	3.85±0.51
FA	87.68±0.57	87.71±1.08	86.21±0.45	73.56±0.62	72.95±0.81	73.17±0.91	3.80±0.47	2.77±0.21	3.77±0.21
F-SAT	88.89±1.93	86.94±2.53	88.86±2.11	75.96±0.85	74.42±1.26	73.76±0.71	4.25±0.57	2.43±0.39	3.56±0.41
DARG	89.60±0.95	88.22±1.08	89.30±1.13	74.65±2.21	73.37±2.43	72.66±2.06	4.62±0.18	2.62±0.10	3.87±0.15
IFAP	89.84±0.92	88.12±0.99	90.03±1.05	75.83±0.83	73.81±0.81	73.31±1.14	4.84±0.10	2.82±0.17	4.12±0.18
Ours (orig. steps) (10 — 10 — 5)	**92.89**±1.05	**92.09**±1.12	**93.14**±1.04	77.49±1.42	76.18±1.23	75.58±1.09	**3.08**±0.33	**1.70**±0.20	**2.91**±0.23
Ours (ablation) (5 — 5 — 3)	91.70±1.55	91.10±1.67	92.15±1.57	**78.10**±0.45	**77.03**±0.30	**76.01**±0.18	3.37±0.75	1.87±0.44	3.11±0.45
Ours (ablation) (15 — 15 — 10)	89.10±1.05	87.40±1.95	87.43±1.51	75.80±1.45	74.33±1.57	74.45±1.23	3.18±0.53	1.76±0.32	3.01±0.30
Ours (ablation) (optimal*)	92.91±1.03	92.13±0.98	93.98±1.02	79.33±0.33	78.12±0.76	76.10±0.57	3.07±0.25	1.75±0.30	2.80±0.20

Table 14: Performance comparison of ours with other methods using different hyperparameters for CVD and Sleep

Method	Chapman			PhysioNet 2017			Sleep-EDF		
	Acc↑	F1↑	AUC ↑	Acc↑	F1↑	AUC ↑	Acc↑	F1↑	κ ↑
Standard training									
Only $\mathcal{D}_{\text{real}}$	92.56±0.25	91.58±0.26	98.59±0.12	82.71±1.55	73.16±3.39	93.00±0.30	82.11±0.88	80.28±1.03	76.16±1.12
Standard techniques									
Aug.	92.43±0.05	91.43±0.08	98.66±0.13	83.79±0.38	**74.47**±2.25	93.55±0.22	81.75±1.43	79.81±0.48	75.60±1.75
FGSM (orig. ϵ) (.3 — .3 — .3)	92.79±0.08	91.83±0.09	98.75±0.14	82.83±2.14	73.08±3.50	93.25±0.43	82.00±0.13	80.88±1.20	75.84±0.27
FGSM (Ablations) (.2 — .2 — .2)	92.92±0.45	91.96±0.54	98.70±0.16	82.13±1.07	71.03±2.43	92.68±0.43	81.59±0.83	80.16±0.55	75.10±1.10
FGSM (Ablations) (1 — 1 — 1)	92.95±0.03	91.95±0.05	98.74±0.10	82.77±0.86	71.99±2.52	93.47±0.41	78.90±3.52	78.65±4.33	71.86±4.98
PGD (orig. ϵ) (.3 — .15 — .2)	**93.28**±0.42	**92.36**±0.49	**98.83**±0.16	83.85±0.30	73.78±0.17	93.50±0.25	82.29±1.12	81.40±0.49	76.14±1.30
PGD (Ablation ϵ) (.2 — .1 — .1)	93.00±0.32	92.61±0.20	98.82±0.20	83.48±0.25	72.80±0.79	93.79±0.20	73.71±2.45	72.35±3.32	64.32±3.53
PGD (Ablation ϵ) (1 — 1 — 1)	86.27±2.79	84.63±2.82	96.50±1.02	80.25±1.40	63.67±2.20	90.24±0.57	79.59±1.66	78.82±2.15	72.55±2.21
PGD (Ablation step) (1 — 1 — 1) & (ϵ=0.3)	92.70±0.05	91.80±0.08	98.71±0.12	82.79±2.04	73.05±2.46	93.16±0.39	82.04±0.11	80.39±1.03	75.36±0.21
Smoothing (low-pass)									
SAP	92.41±0.43	91.43±0.45	98.64±0.16	83.57±0.75	72.99±2.01	93.27±0.20	78.33±0.86	77.14±0.94	70.82±1.06
SGM	92.94±0.09	92.02±0.14	98.60±0.18	83.10±0.45	73.90±1.51	93.34±0.09	80.75±1.67	79.71±2.32	74.37±2.03
LF-AP	91.87±0.25	91.18±0.59	98.02±1.03	82.87±0.48	73.16±2.25	92.65±0.34	79.32±1.53	76.32±2.01	72.66±1.18
Clean accuracy-focused									
TRADES	91.99±0.22	90.89±0.29	98.49±0.07	81.10±0.87	70.11±1.40	92.87±0.27	82.46±0.82	79.91±0.67	76.02±1.19
f-VAT ($s = 3$)	92.19±0.29	91.14±0.38	98.30±0.30	81.66±0.56	71.29±0.46	92.80±0.60	81.77±1.52	80.12±2.10	75.60±2.20
f-VAT ($s = 1$)	91.31±0.97	90.15±1.15	98.30±0.50	82.07±0.30	72.13±1.19	92.90±0.50	80.19±2.04	80.08±2.09	74.60±2.80
f-VAT ($s = 2$)	91.82±0.48	90.76±0.53	98.40±0.30	82.23±0.41	72.31±1.80	92.80±0.80	80.76±2.09	81.29±2.77	74.30±2.90
FA	90.10±0.25	89.81±0.23	97.81±0.09	80.82±0.91	68.33±1.73	91.38±0.21	81.30±0.81	79.13±0.88	75.73±1.21
F-SAT	91.57±0.34	90.43±0.31	98.49±0.20	82.36±2.28	73.37±3.00	93.11±1.05	77.11±4.31	77.02±2.28	69.25±0.55
DARG	92.20±0.43	91.15±0.45	98.59±0.01	83.58±1.30	73.89±3.45	93.15±0.79	82.20±1.80	80.43±1.30	76.04±2.10
IFAP	91.83±0.31	89.60±0.26	97.40±0.07	80.79±1.50	70.67±1.35	91.30±0.31	81.67±1.09	78.99±0.90	76.23±1.02
Ours (orig. steps) (1 — 1 — 1)	92.88±0.12	91.95±0.08	98.79±0.14	**84.10**±0.49	73.70±1.10	**93.57**±0.23	**83.02**±0.86	**81.53**±1.10	**77.50**±1.23
Ours (ablation) (3 — 3 — 3)	92.56±0.43	91.56±0.55	98.67±0.15	83.10±2.13	72.98±2.34	93.01±0.50	82.47±1.20	81.50±1.13	76.49±1.21
Ours (ablation) (5 — 5 — 5)	91.78±0.48	91.10±0.57	98.46±0.20	83.26±0.80	73.04±1.65	93.64±0.10	82.43±1.40	81.50±1.48	74.80±2.56
Ours (ablation) (optimal*)	92.60±0.37	91.88±0.43	98.74±0.12	84.15±0.43	73.87±1.10	93.69±0.16	83.09±1.34	81.64±1.21	78.02±1.68

Overall, the tables demonstrate that the methods exhibit comparative robustness to hyperparameter changes. However, it is important to note that, in the absence of an early stopping during adversarial example generation, increasing the step size can lead to performance degradation in certain cases. This highlights the balance between optimizing hyperparameters and maintaining model performance Zhang et al. (2019). Nevertheless, our results indicate that a straightforward grid search is sufficient to identify effective hyperparameters.

We also did an additional analysis (the last row of PGD ablations) by matching PGD-ℓ_2 and our method in budgets, step sizes, and update counts across all datasets. Our method outperforms PGD in all datasets under this setting. This comparison supports the benefit of spectral filtering beyond perturbation-budget selection.

D.3 Robustness

Our robustness evaluation also includes additive band-limited noise perturbations applied to the original samples. To generate these perturbations, we sample random noise and filter it with a 4th-order Butterworth bandpass filter whose cut-off frequencies are estimated from the mean spectrum of each dataset, as given in Table 16. We evaluate two noise levels with standard deviations of 0.2 and 0.3, corresponding to signal-to-noise ratios (SNRs) of approximately 13.98 dB and 10.46 dB, respectively. As shown in Figure 8, increasing the noise level to 0.3 causes a 7–12% drop for standard training, whereas our method degrades by only 3–5%.

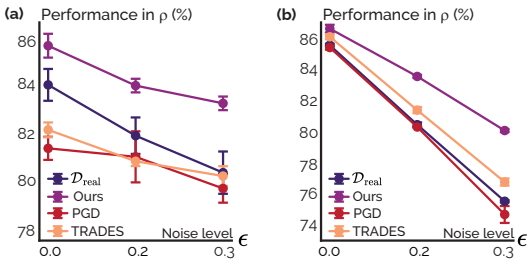


Figure 8: Performance comparison of models across noise levels, trained using three perturbation generation methods, in **(a)** SPC12 and **(b)** DaLiA.

D.3.1 Evaluating robustness with other techniques

We also examine training with unconstrained adversarial perturbations such as Carlini & Wagner (2017) and AutoAttack (Croce & Hein, 2020). These methods target broad norm-bounded threat sets and, in our experiments, often reduce clean accuracy even when early stopping is selected on a held-out validation set. This is consistent with our empirical finding that unconstrained attacks can introduce strong harmonic components that are not aligned with the measured signal structure.

Accordingly, we do not treat CW and AutoAttack as the primary robustness setting for the proposed method. Instead, we report CW as supplementary reference points, while emphasizing support-constrained perturbations in the main text. This is also consistent with prior work showing that methods designed for structured or on-manifold perturbations need not provide robustness to fully unconstrained off-manifold adversaries (Stutz et al., 2019; Lin et al., 2020).

Table 15: Robustness across datasets under norm and harmonic-constrained threats. Each defense forms a column group, and subcolumns correspond to clean performance and attacked performance under unconstrained and support-constrained threats. PGD-ℓ_2 and CW are harmonic-unconstrained attacks, while f-PGD and f-CW denote constrained variants using the effective support of inputs. Bold marks the best defense for each dataset–attack pair, respecting the metric direction of that dataset. We report correlation and mean absolute error for heart-rate estimation and step counting, respectively, and, accuracy for all remaining classification tasks.

Dataset	Only $\mathcal{D}_{\text{real}}$					TRADES					DARG					Ours					Ours (window)				
Def. / Att.	Clean	ℓ_2	f-PGD	CW	f-CW	Clean	ℓ_2	f-PGD	CW	f-CW	Clean	ℓ_2	f-PGD	CW	f-CW	Clean	ℓ_2	f-PGD	CW	f-CW	Clean	ℓ_2	f-PGD	CW	f-CW
UCIHAR	89.9	74.6	80.3	68.9	80.0	91.5	71.2	77.8	67.5	80.1	89.6	60.8	73.7	**69.3**	82.4	**92.9**	**77.6**	**86.1**	68.4	**85.4**	**92.9**	74.0	84.7	68.3	84.9
USC	73.5	49.8	55.7	**60.2**	59.7	76.5	53.1	63.0	56.5	65.6	74.7	55.0	59.6	56.7	63.7	**77.5**	58.2	**71.7**	56.2	70.8	**77.5**	**58.7**	70.2	56.9	**71.5**
Clemson	2.3	3.8	3.4	3.3	3.0	1.8	**2.7**	2.5	2.9	2.6	2.6	3.4	3.1	3.0	2.9	**1.7**	2.8	2.6	**2.5**	2.3	1.8	2.9	**2.4**	2.8	**2.1**
Chapman	92.6	58.7	80.8	59.2	73.5	92.0	**75.0**	85.9	63.1	80.2	92.2	65.2	84.0	64.7	81.3	92.9	70.0	**87.7**	62.0	**83.6**	**92.9**	70.7	76.1	**66.5**	80.7
Physio 2017	82.7	59.5	55.8	60.6	64.8	81.1	**71.5**	68.4	**63.7**	69.3	83.6	64.2	68.9	61.3	72.5	84.1	66.1	**76.7**	62.5	**75.8**	**84.8**	66.4	75.1	63.1	75.1
Sleep-EDF	82.1	62.6	64.4	63.8	69.2	82.5	**69.3**	70.4	64.9	74.2	82.2	64.2	71.8	**65.6**	75.6	83.0	67.5	**77.2**	64.2	77.4	**83.1**	68.1	76.2	64.7	**77.9**
IEEE SPC12	83.9	62.2	65.0	60.5	71.3	82.2	63.5	67.7	66.9	71.6	82.0	63.8	**70.1**	67.3	68.2	**85.5**	**64.3**	69.3	**68.2**	72.7	85.2	63.8	68.7	66.6	**74.3**
IEEE SPC22	81.2	57.7	61.5	**62.3**	64.9	80.7	58.1	64.3	60.3	**68.0**	81.2	55.7	63.1	59.6	64.0	**85.9**	58.5	**66.1**	62.1	67.2	84.4	**58.6**	66.0	62.1	67.1
DaLiA	85.7	60.3	73.4	67.8	77.4	85.5	72.1	70.2	**74.6**	78.3	86.2	68.4	75.9	70.6	80.2	86.7	**72.4**	**80.3**	74.2	80.6	**86.8**	70.1	79.5	74.3	**80.7**

To provide a broader robustness evaluation, we report performance under unconstrained norm-bounded perturbations and support-constrained perturbations, which are intended to remain within the observable signal support. All attacks are untargeted and evaluated on the full test set. We report the results in Table 15.

PGD-ℓ_2 computes worst-case perturbations under an ℓ_2 budget, $\|\delta\|_2 \leq \epsilon$, using K projected gradient steps, without restricting spectral content. Its constrained counterpart, f-PGD, uses the same optimization procedure and perturbation budget but applies the projection $\delta \leftarrow P_F(\delta)$ at each step so that perturbations remain within a dataset-specific passband. For completeness, we also report

unconstrained and support-constrained CW attacks, where the constrained variant applies the same projection P_F. We evaluate the untargeted CW-ℓ_2 attack using 1k Adam iterations with learning rate 0.005, 5 binary-search steps for the loss coefficient without restarts. We estimate each passband from the mean training-set spectrum using a 99% energy. The resulting intervals are listed in Table 16.

The estimated passbands vary across datasets. For IEEE SPC12 and SPC22, the estimated passband covers approximately one-third of the available frequency range. Before bandpass preprocessing, the corresponding 99% energy interval is $[0.1,\ 5.250]$ Hz. Thus, the spectral restriction reflects the measured training spectra rather than imposing a uniformly narrow low-pass band. These dataset-level passbands define the evaluation threat sets; they are distinct from the input-dependent KL alignment used to train our filter.

Table 16: Dataset-specific 99% energy passbands used by f-PGD and f-CW, estimated from the mean training-set power spectrum.

Dataset	Passband (Hz)	Nyquist (Hz)
IEEE SPC12	[0.25, 4.375]	12.5
IEEE SPC22	[0.25, 4.375]	12.5
DaLiA	[0.125, 5.625]	12.5
UCIHAR	[0, 11.719]	25
USC-HAD	[0, 22.5]	50
Clemson	[0.031, 6.375]	7.5
PhysioNet 2017	[0.05, 90.65]	150
Chapman	[0.1, 35.7]	50
Sleep-EDF	[0.05, 44.967]	50

Our method performs strongest overall under support-constrained threats, while remaining competitive under unconstrained attacks. This is consistent with the objective of the defense where we aim to improve robustness to perturbations that remain within the effective bandwidth, rather than to arbitrary ℓ_p-bounded distortions with unrestricted spectral content.

By set inclusion, Proposition D.1 shows that restricting the perturbation set cannot increase the worst-case risk of any fixed classifier.

Proposition D.1 (Monotonicity under threat-set restriction)**.** *Let $f_\theta\colon \mathbb{R}^d \to \mathcal{Y}$ be a classifier, and define the natural and robust error over a perturbation set $\mathcal{S} \subseteq \mathbb{R}^d$*

$$\mathcal{R}_{\mathrm{rob}}(f_\theta;\mathcal{S}) = \Pr_{(X,Y)\sim\mathcal{D}}\Big(\exists\,\delta\in\mathcal{S}:\ f_\theta(X+\delta)\neq Y\Big) \quad \textit{and} \quad \mathcal{R}_{\mathrm{nat}}(f_\theta) = \Pr_{(X,Y)\sim\mathcal{D}}\big(f_\theta(X)\neq Y\big), \tag{25}$$

Let $\mathcal{S}$ be an ℓ_p-ball of radius ϵ, and define the frequency set $\mathcal{S}_F = \{\delta\in\mathcal{S} : \delta = P_F(\delta)\}$. Then

$$\mathcal{R}_{\mathrm{rob}}(f_\theta;\mathcal{S}_F)\ \le\ \mathcal{R}_{\mathrm{rob}}(f_\theta;\mathcal{S}). \tag{26}$$

Proof. Since $\mathcal{S}_F \subseteq \mathcal{S}$, for every (x,y) we have

$$\mathbf{1}\{\exists\,\delta\in\mathcal{S}_F:\ f_\theta(x+\delta)\neq y\} \le \mathbf{1}\{\exists\,\delta\in\mathcal{S}:\ f_\theta(x+\delta)\neq y\}\,. \tag{27}$$

Taking expectation over $(X,Y)\sim\mathcal{D}$ yields $\mathcal{R}_{\mathrm{rob}}(f_\theta;\mathcal{S}_F) \le \mathcal{R}_{\mathrm{rob}}(f_\theta;\mathcal{S})$. □

This proposition does not claim improved robustness under arbitrary norm-bounded attacks; rather, it formalizes that our evaluation targets a restricted but motivated threat model.

D.3.2 Filter-parameterized white-box evaluation

We evaluate a white-box attack that differentiates through the learned filter during perturbation generation. For each predictor, the attacker optimizes a variable $\boldsymbol{z}$ using that predictor's loss gradients. We use the filter trained by our method for the predictors, including TRADES and DARG, and keep its parameters fixed. For an input $\boldsymbol{x}$, the applied perturbation is given in Equation 28.

$$\delta_\psi(\boldsymbol{z};\boldsymbol{x}) = \min\{\epsilon, \alpha\|\boldsymbol{x}\|_2\}\,\frac{\boldsymbol{z} * h_\psi(\cdot;\boldsymbol{x})}{\max\{\|\boldsymbol{z} * h_\psi(\cdot;\boldsymbol{x})\|_2, \tau\}}, \tag{28}$$

where $\tau > 0$ prevents division by zero. This scaling ensures $\|\delta_\psi(\boldsymbol{z};\boldsymbol{x})\|_2 \le \epsilon$. We therefore impose no additional norm constraint on the raw variable $\boldsymbol{z}$ and update it by gradient ascent as in below.

$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t + \eta\nabla_{\boldsymbol{z}}\mathcal{L}\big(f_\theta\big(\boldsymbol{x}+\delta_\psi(\boldsymbol{z}^t;\boldsymbol{x})\big), y\big)\,. \tag{29}$$

Gradients propagate through the fixed filter and perturbation normalization. The norm budget is enforced only on the perturbation applied to the input. A-PGD evaluates optimization through the learned filter rather than enforcing a hard spectral constraint. It is initialized as $\boldsymbol{z}^0 \sim \mathcal{N}(0, I)$ and

Table 17: Filter-parameterized white-box evaluation against the proposed method. ℓ_2 denotes the standard PGD evaluation, f-PGD denotes the support-constrained baseline evaluation, and A-PGD denotes the white-box evaluation that differentiates through the learned filter. Bold marks the strongest defense for each dataset-evaluation pair, respecting the metric direction of that dataset.

Dataset	Only $\mathcal{D}_{\text{real}}$				TRADES				DARG				Ours				Ours (window)			
Def. / Att.	Clean	ℓ_2	f-PGD	A-PGD	Clean	ℓ_2	f-PGD	A-PGD	Clean	ℓ_2	f-PGD	A-PGD	Clean	ℓ_2	f-PGD	A-PGD	Clean	ℓ_2	f-PGD	A-PGD
UCIHAR	89.9	74.6	80.3	60.2	91.5	71.2	77.8	70.3	89.6	60.8	73.7	60.5	**92.9**	**77.6**	**86.1**	**70.5**	**92.9**	74.0	84.7	65.6
USC	73.5	49.8	55.7	50.3	76.5	53.1	63.0	52.7	74.7	55.0	59.6	54.9	**77.5**	58.2	**71.7**	60.1	**77.5**	**58.7**	70.2	**60.6**
Clemson	2.3	3.8	3.4	3.6	1.8	**2.7**	2.5	2.6	2.6	3.4	3.1	2.6	**1.7**	2.8	2.6	**2.1**	1.8	2.9	**2.4**	2.3
Chapman	92.6	58.7	80.8	62.7	92.0	**75.0**	85.9	74.8	92.2	65.2	84.0	68.3	92.9	70.0	**87.7**	75.0	**92.9**	70.7	76.1	**83.5**
Physio 2017	82.7	59.5	55.8	65.0	81.1	**71.5**	68.4	73.2	83.6	64.2	68.9	77.5	84.1	66.1	**76.7**	80.6	**84.8**	66.4	75.1	**81.5**
Sleep-EDF	82.1	62.6	64.4	65.7	82.5	**69.3**	70.4	68.7	82.2	64.2	71.8	69.3	83.0	67.5	**77.2**	**73.7**	**83.1**	68.1	76.2	72.4
IEEE SPC12	83.9	62.2	65.0	61.1	82.2	63.5	67.7	62.6	82.0	63.8	**70.1**	63.6	**85.5**	**64.3**	69.3	67.5	85.2	63.8	68.7	**78.1**
IEEE SPC22	81.2	57.7	61.5	67.8	80.7	58.1	64.3	60.9	81.2	55.7	63.1	59.7	**85.9**	58.5	**66.1**	**68.8**	84.4	**58.6**	66.0	68.5
DaLiA	85.7	60.3	73.4	59.7	85.5	72.1	70.2	69.6	86.2	68.4	75.9	69.3	86.7	**72.4**	**80.3**	**75.1**	**86.8**	70.1	79.5	74.4

Table 18: Robustness evaluation protocol used for Tables 15 and 17. Left: attack optimization and candidate-selection settings. Right: dataset-specific radius caps ϵ_d.

(a) Attack protocol

Attack	Radius	Step / opt.	Iter.	Init. / restarts	Objective / selection	Constraint
PGD-ℓ_2	$r_d(\mathbf{x})$	0.1	50	Gaussian / 2	CE; max loss	Joint ℓ_2
f-PGD	$r_d(\mathbf{x})$	0.1	50	Gaussian / 2	CE; max loss	Joint $\ell_2 + P_F$
CW-ℓ_2	$r_d(\mathbf{x})$	Adam, lr $= .005$	1000	5 binary search / 0	CW loss; feasible cand.	Joint ℓ_2
f-CW	$r_d(\mathbf{x})$	Adam, lr $= .005$	1000	5 binary search / 0	CW loss; feasible cand.	Joint $\ell_2 + P_F$
A-PGD	$r_d(\mathbf{x})$	0.1	50	Single / 0	CE through h_ψ	Filter parameterization

Attacks are untargeted and evaluated on the test set. The norm is computed over channels and time points. For dataset d, the evaluation radius is $r_d(\mathbf{x}) = \min\{\epsilon_d, \|\mathbf{x}\|_2\}$. PGD/f-PGD use the same optimization and radius. For each example, the candidate with the largest task loss is retained. For CW, the perturbation is required to satisfy the same radius. A-PGD differentiates through the filter without imposing a hard spectral constraint.

(b) Radius cap

Dataset	ϵ_d
IEEE SPC12	0.3
IEEE SPC22	1
DaLiA	0.3
UCIHAR	1
USC-HAD	0.3
Clemson	1
PhysioNet 2017	0.3
Chapman	0.3
Sleep-EDF	0.3

uses a single initialization without restarts. For a strictly positive filter response and unrestricted $\boldsymbol{z}$, the filtering operator is invertible, so the parameterization can represent any perturbation within the effective ℓ_2 radius.

For datasets, we use the same evaluation budget across defenses. The parameters ϵ and α are shared across defenses with the same effective radius. For comparisons across attack formulations, we apply the same sample-dependent radius to every attack. These evaluation settings are fixed independently of the method-specific configurations.

Table 17 shows that our method retains its advantage when the attacker differentiates through the learned filter and optimizes perturbations against the evaluated predictor. Either our full-sequence or windowed variant achieves the best A-PGD result on every dataset. Thus, the advantage persists when the attacker explicitly accounts for the filtering operation.

D.3.3 Expert Assessment

We conducted a blinded assessment of PPG and ECG signals with two physicians with backgrounds in cardiology, each with more than 1,000 hours of experience reviewing ECG and PPG recordings. The physicians were informed about the study objective but were unaware of which method generated each perturbation. They did not see previous ratings, agreement statistics, running totals, or method-level results during the assessment.

We sampled 200 source signals from the held-out test set of each dataset. For each source signal, we generated four perturbed versions using PGD, SAP, DARG, and our method at matched perturbation strength. These baselines represent gradient-based, smoothing-based, and generative approaches, respectively. The assessment therefore comprised 2,400 original–perturbed pairs, each rated indepen-

dently by both physicians. Each pair was displayed using identical time and amplitude scales, with method names removed and presentation order randomized independently for each physician.

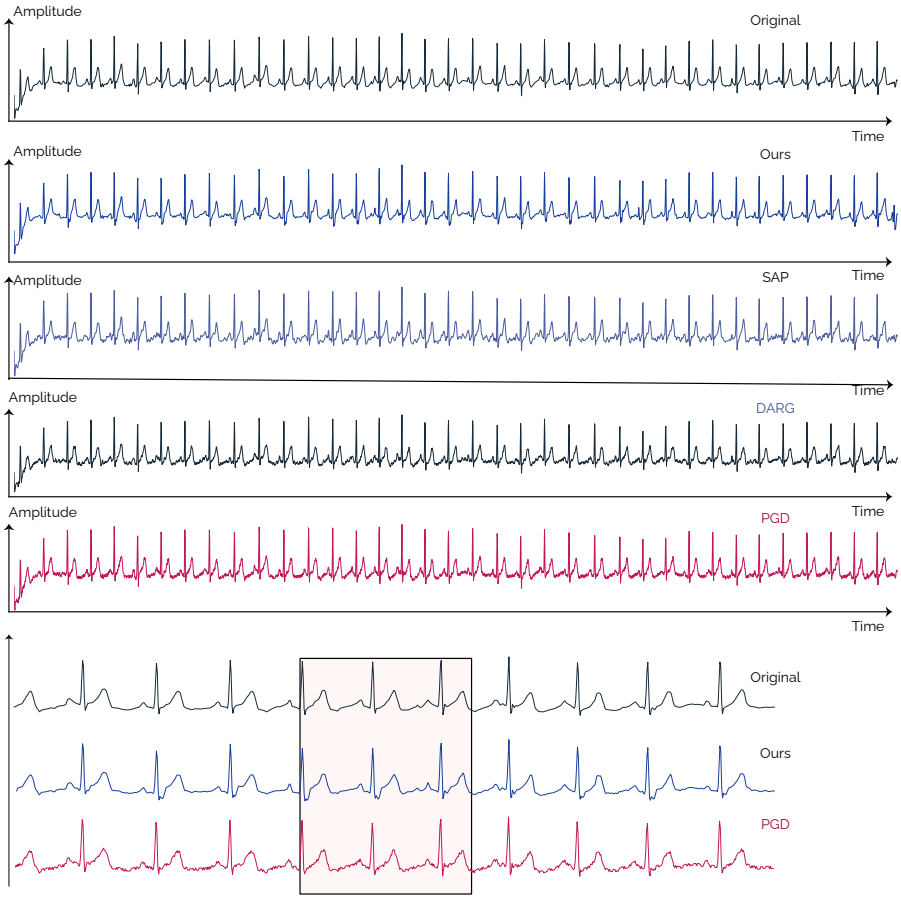
The physicians labeled each perturbed signal as realistic or unrealistic according to whether it could plausibly occur as a measurement of the same underlying physiological signal. This judgment considered both preservation of physiological structure and consistency with plausible acquisition artifacts. No feedback was provided after a rating was submitted.

Our method received the highest proportion of realistic ratings from both physicians on all datasets. On PhysioNet, both physicians labeled every signal perturbed by our method as realistic. The paired-source design, matched perturbation strength, and blinded presentation support interpreting these results as evidence of physician-rated physiological plausibility across PPG and ECG signals.

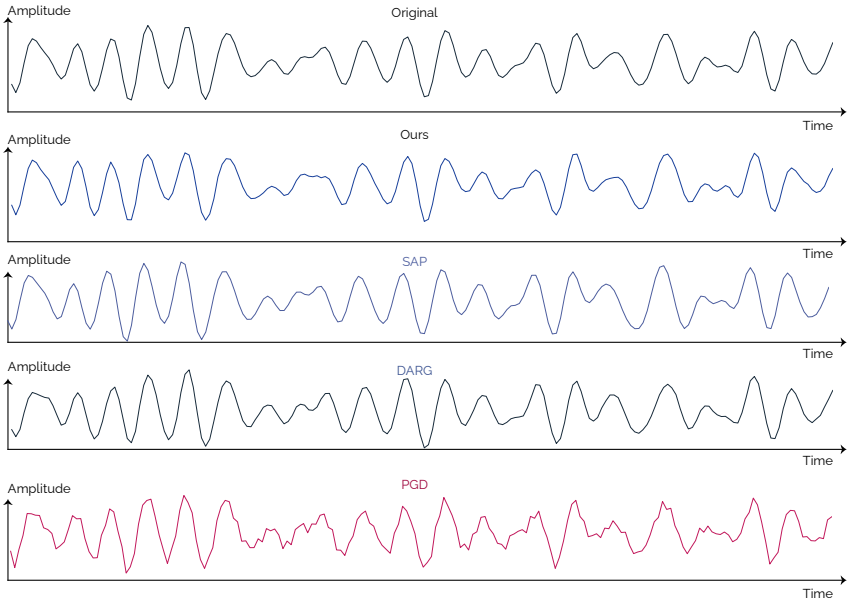
Expert-Annotated Examples Figures 9b and 9a show perturbations from DaLiA and PhysioNet 2017. Each example contains the matched-strength perturbation. Method identities were hidden during the assessment. The clinicians considered whether the perturbation preserved the underlying physiological structure and whether the introduced variation was consistent with the corresponding acquisition setting. Consensus-realistic examples retain the main waveform morphology, whereas consensus-unrealistic examples contain changes that the clinicians considered incompatible with the original signal or acquisition process. The figures provide qualitative context for the aggregate realism rates.

Figure 9 shows matched-strength examples from PhysioNet 2017 and DaLiA. All perturbations are matched by the total per-sample ℓ_2 norm, but their temporal distributions differ. Our method preserves the main morphology and introduces variations around the observed waveform. PGD distributes dense high-frequency changes across the sequence. The enlarged region at the beginning of the recording further shows that our method preserves local morphology without removing waveform details. For DaLiA, our method similarly follows the PPG structure.

One interesting result from this is that SAP, which aims to smooth high frequency perturbations using hand-designed low-pass filters, keep waveforms similar to sawtooth. We also observed this in different datasets in Appendix E where smoothing methods kept some portions of the high frequencies. We conjecture that this is caused by the hand-crafted hyperparameters which requires tuning. Our method addresses this drawback by learning the filter adaptively from input data distribution.



(a) PhysioNet 2017 ECG.



(b) DaLiA PPG.

Figure 9: Matched-ℓ_2 perturbations from two datasets. **(a)** PhysioNet 2017 ECG and **(b)** DaLiA PPG. In each panel, the original signal and perturbations generated by our method, SAP, DARG, and PGD are shown at matched ℓ_2 strength. Our method preserves the main waveform morphology more closely. DARG remains comparable on the PPG but degrades on the ECG.

D.4 CROSS-DATASET FILTER TRANSFER

We evaluate whether the learned spectral filter transfers across datasets, including within-modality transfer and an ECG-to-EEG transfer. We resample the source dataset for filter training before transferring to have a matched number of frequency bins. Table 19 compares transferred filters with the original variants and the best baselines selected by average rank across the metrics. This comparison assesses whether the learned spectral weighting captures structure shared across datasets.

Table 19: Cross-dataset filter transfer for HR estimation.

Method	IEEE SPC12			IEEE SPC22			DaLiA		
	MAE↓	RMSE↓	ρ↑	MAE↓	RMSE↓	ρ↑	MAE↓	RMSE↓	ρ↑
Reference baselines									
FGSM	5.59 ± 0.22	12.03 ± 0.74	83.82 ± 0.65	5.45 ± 0.11	10.57 ± 0.22	82.70 ± 0.35	5.43 ± 0.08	12.26 ± 0.10	86.04 ± 0.07
Aug.	5.54 ± 0.22	11.63 ± 0.84	84.38 ± 1.10	5.31 ± 0.31	10.77 ± 0.64	81.32 ± 0.81	5.52 ± 0.04	12.46 ± 0.09	85.67 ± 0.15
DARG	6.07 ± 0.27	12.68 ± 0.32	82.01 ± 0.40	5.28 ± 0.27	10.89 ± 1.23	81.22 ± 1.57	5.38 ± 0.07	12.15 ± 0.14	86.20 ± 0.32
SAP	6.68 ± 0.59	13.33 ± 0.89	82.07 ± 1.54	5.41 ± 0.55	10.80 ± 1.00	82.38 ± 1.01	5.40 ± 0.08	12.25 ± 0.18	86.05 ± 0.25
Dataset-specific filters									
Ours	5.19 ± 0.10	10.59 ± 0.50	85.47 ± 0.60	4.77 ± 0.98	9.76 ± 1.50	85.90 ± 1.79	5.23 ± 0.09	11.85 ± 0.25	86.73 ± 0.50
Ours (window)	5.32 ± 0.13	10.65 ± 0.53	85.17 ± 0.59	4.93 ± 0.94	10.04 ± 1.33	84.37 ± 1.55	5.22 ± 0.08	11.93 ± 0.23	86.75 ± 0.52
Transferred filters									
Filter source		DaLiA			DaLiA			IEEE SPC 22	
Ours (transfer)	5.21 ± 0.09	10.58 ± 0.45	85.45 ± 0.57	4.75 ± 0.93	9.74 ± 1.47	85.88 ± 1.75	5.25 ± 0.07	11.87 ± 0.24	86.70 ± 0.50

Table 20: Cross-dataset filter transfer for activity recognition and step counting.

Method	UCIHAR			USC			Clemson		
	Acc↑	W-F1↑	F1↑	Acc↑	W-F1↑	F1↑	MAPE↓	MAE↓	RMSE↓
Reference baselines									
TRADES	91.49 ± 0.73	90.90 ± 0.55	92.13 ± 0.37	76.50 ± 0.75	75.18 ± 0.90	74.57 ± 0.83	3.30 ± 0.31	1.84 ± 0.22	2.96 ± 0.18
SAP	91.79 ± 0.95	90.91 ± 0.96	92.01 ± 0.81	76.09 ± 0.81	74.82 ± 1.03	73.79 ± 1.12	4.34 ± 0.35	2.44 ± 0.19	3.70 ± 0.21
FGSM	90.07 ± 1.12	88.75 ± 1.17	89.54 ± 1.47	74.20 ± 1.10	72.51 ± 1.75	72.34 ± 1.34	3.62 ± 0.34	2.02 ± 0.21	3.17 ± 0.24
F-SAT	88.89 ± 1.93	86.94 ± 2.53	88.86 ± 2.11	75.96 ± 0.85	74.42 ± 1.26	73.76 ± 0.71	4.25 ± 0.57	2.43 ± 0.39	3.56 ± 0.41
Dataset-specific filters									
Ours	92.89 ± 1.05	92.09 ± 1.12	93.14 ± 1.04	77.49 ± 1.42	76.18 ± 1.23	75.58 ± 1.09	3.08 ± 0.33	1.70 ± 0.20	2.91 ± 0.23
Ours (window)	92.90 ± 0.93	91.98 ± 1.03	93.17 ± 1.00	77.52 ± 1.13	76.20 ± 1.10	75.63 ± 1.04	3.36 ± 0.27	1.83 ± 0.16	3.13 ± 0.20
Transferred filters									
Filter source		USC			UCIHAR			USC	
Ours (transfer)	93.16 ± 0.93	92.67 ± 1.04	93.32 ± 1.00	77.45 ± 1.32	75.86 ± 1.25	75.43 ± 1.12	3.12 ± 0.21	1.84 ± 0.23	3.04 ± 0.25

Table 21: Cross-dataset filter transfer for CVD and sleep classification.

Method	Chapman			PhysioNet 2017			Sleep-EDF		
	Acc↑	F1↑	AUC↑	Acc↑	F1↑	AUC↑	Acc↑	F1↑	κ↑
Reference baselines									
PGD	93.28 ± 0.42	92.36 ± 0.49	98.83 ± 0.16	83.85 ± 0.30	73.78 ± 0.17	93.50 ± 0.25	82.29 ± 1.12	81.40 ± 0.49	76.14 ± 1.30
Aug.	92.43 ± 0.05	91.43 ± 0.08	98.66 ± 0.13	83.79 ± 0.38	74.47 ± 2.25	93.55 ± 0.22	81.75 ± 1.43	79.81 ± 0.48	75.60 ± 1.75
FGSM	92.79 ± 0.08	91.83 ± 0.09	98.75 ± 0.14	82.83 ± 2.14	73.08 ± 3.50	93.25 ± 0.43	82.00 ± 0.13	80.88 ± 1.20	75.84 ± 0.27
DARG	92.20 ± 0.43	91.15 ± 0.45	98.59 ± 0.01	83.58 ± 1.30	73.89 ± 3.45	93.15 ± 0.79	82.20 ± 1.80	80.43 ± 1.30	76.04 ± 2.10
Dataset-specific filters									
Ours	92.88 ± 0.12	91.95 ± 0.08	98.79 ± 0.14	84.10 ± 0.49	73.70 ± 1.10	93.57 ± 0.23	83.02 ± 0.86	81.53 ± 1.10	77.50 ± 1.23
Ours (window)	92.87 ± 0.23	92.10 ± 0.10	98.80 ± 0.15	84.83 ± 0.40	74.10 ± 1.03	93.62 ± 0.22	83.10 ± 0.73	81.60 ± 0.99	77.52 ± 1.44
Transferred filters									
Filter source		PhysioNet 2017			Chapman			PhysioNet 2017	
Ours (transfer)	92.52 ± 0.11	91.78 ± 0.07	98.73 ± 0.12	84.08 ± 0.45	73.65 ± 1.03	93.54 ± 0.21	82.47 ± 0.84	81.48 ± 1.03	77.42 ± 1.19

Tables 19–21 show that transferred filters retain performance close to their dataset-specific counterparts. Although the filter parameters remain frozen, its frequency response still adapts to each target input through spectral conditioning, while the target predictor's loss gradients guide perturbation generation. These results suggest that the filter learns a transferable mapping from input spectra to perturbation weights. Transfer from PhysioNet ECG to Sleep-EDF EEG also remains competitive, suggesting that this mapping can extend beyond a single modality, although dataset-specific training yields higher sleep-classification accuracy.

D.5 CLEAN–ROBUST PERFORMANCE TRADE-OFF

We examine whether reducing the perturbation strength of existing training methods can recover the clean–robust performance achieved by our method. We compare SAP, PGD, TRADES, and our method across three training strengths. Every trained predictor is evaluated under the same fixed attack settings within each dataset. This separates changes in training strength from changes in evaluation difficulty. We keep the original preprocessing, training/validation partitions, and test splits.

We evaluate strength multipliers $s \in \{0.25, 0.5, 1\}$. Each multiplier scales the budget and update-size parameters of a method's reference configuration for the sweep. All remaining hyperparameters are fixed across experiments.

We attack each selected predictor using ordinary ℓ_2 PGD while using an absolute joint ℓ_2 radius of 0.3 across all channels and time points, step size 0.1, 50 iterations, and two random restarts. Each restart is initialized using a Gaussian direction and a radius sampled uniformly from $[0, 0.3]$. For each test example, we retain the candidate with the largest task loss across the two restarts and the clean input. We apply no input clipping or post-attack normalization.

Ordinary PGD projects each update onto the joint ℓ_2 ball. Initialization and subsequent updates are projected. Thus, the evaluation constraint is identical across methods for each test input and does not depend on any learned filter.

Each plotted point represents one training configuration, with clean error on the vertical axis and attacked error on the horizontal axis. Lower-left therefore indicates better performance in every panel. Color identifies the method, marker size identifies training strength, and the gray star denotes clean training.

We identify nondominated configurations separately for each dataset and attack across all evaluated methods, strengths, and the clean-training reference. A configuration is nondominated if no other evaluated configuration has no greater error on both axes and strictly lower error on at least one.

Figure 10 shows that our method contributes at least one nondominated configuration on datasets, achieving a favorable balance between clean and attacked error. All methods are evaluated under the same fixed attack, so these differences reflect the effectiveness of training. Increasing training strength alone does not consistently improve this balance and can worsen both metrics. The relevant comparison is therefore the obtained clean–robust performance across training strengths. Across the evaluated settings, our method consistently reaches the empirical frontier, showing that its gains extend beyond a single choice of perturbation strength.

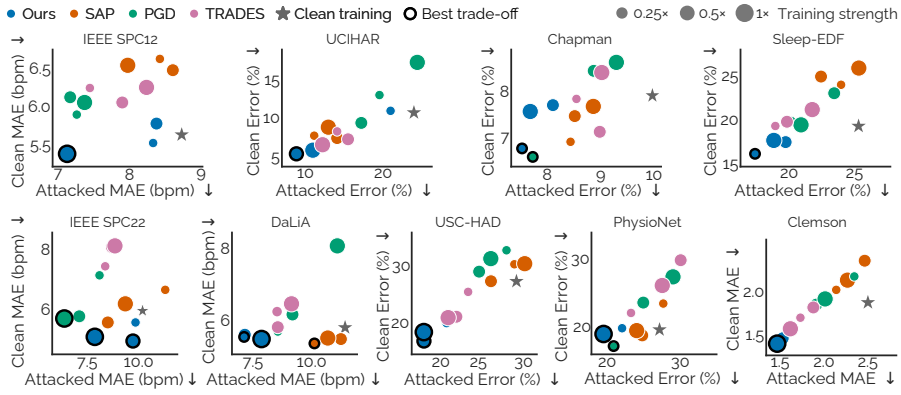


Figure 10: Clean–robust trade-offs under ordinary ℓ_2 PGD in all datasets. Lower-left is better, and black outlines mark nondominated configurations. All configurations are evaluated with the same attack budget, 50 steps, and two restarts.

D.6 Attack Convergence and Per-Example Union

We assess whether increasing attack optimization effort changes the measured robustness of validation-selected checkpoints and hyperparameters. We re-evaluate these points without retraining or changing the perturbation sets. Ordinary and input-support spectral PGD use the same absolute joint ℓ_2 radius ϵ across channels and time points, step size 0.1, and no input clipping or post-attack normalization. Spectral PGD retains the input-derived 99% energy-support construction used in the preceding evaluation. We vary the number of iterations over $\{10, 20, 50, 100\}$ and random restarts over $\{1, 2, 5\}$. Random initializations and candidate trajectories are nested, so increasing either budget retains all candidates from the corresponding smaller setting.

For classification, an attack succeeds when the predicted class differs from the ground-truth label. For each test example, a successful candidate takes priority over any correctly classified candidate; ties are resolved by higher cross-entropy. The union counts an example as misclassified if either attack family finds a successful candidate. Thus, union error is computed from per-example outcomes rather than averaging attack-level metrics.

For discretized regression, we generate candidates using both cross-entropy and an expected-bin absolute-error objective,

$$\mathcal{L}_{\text{soft-MAE}}(x, y_{\text{bin}}) = \left| \sum_k k \, \text{softmax}(f_\theta(x))_k - y_{\text{bin}} \right|, \tag{30}$$

where y_{bin} is the discretized target index. Candidate selection uses the original decoding rule and target units: for each example, we retain the candidate with the largest decoded absolute error across both objectives and attack families. This selection maximizes MAE and RMSE over the available per-example candidate pools. Pearson correlation is computed on the same selected predictions and reported descriptively; it is not optimized by this selection.

Every spectral candidate also satisfies the ordinary ℓ_2 constraint. The union therefore provides a stronger empirical evaluation within the same ordinary threat set, without enlarging its radius.

Increasing the attack budget from 50 iterations and two restarts to 100 iterations and five restarts changes every union result by less than 0.01 in its reported metric. Classification union errors remain unchanged at the reported precision. The intermediate settings separately confirm stability to additional iterations and restarts.

Together, these results show that the measured union robustness is stable under additional iterations and restarts for the tested attack objectives, step size, and perturbation radius. This provides evidence against insufficient iterations or restarts as the explanation for these results.

Table 22: Attack-convergence and per-example-union at joint ℓ_2 radius 0.3. $T \times R$ denotes PGD iterations and random restarts. Entries are classification error (%) or decoded MAE (steps for Clemson; bpm for IEEE SPC12). Higher values indicate more damaging attacks. The union remains stable as optimization effort increases.

		Per-example union				100×5	
Dataset	Checkpoint	50×2	100×2	50×5	100×5	Ordinary	Spectral
Clemson	Ours ($s = 1$)	2.80	2.80	2.80	2.80	2.80	2.57
IEEE SPC12	Ours ($s = 1$)	7.15	7.15	7.15	7.15	7.12	6.64
PhysioNet	Ours ($s = 0.25$)	22.03	22.03	22.03	22.03	22.03	21.32
Sleep-EDF	Ours ($s = 0.25$)	18.63	18.63	18.63	18.63	18.37	17.31
UCIHAR	Ours ($s = 0.5$)	8.83	8.83	8.83	8.83	8.83	8.20

D.7 IMPROVEMENTS IN PERFORMANCE ACROSS DIFFERENT NETWORKS

In this section, we assess the performance of our proposed method using a different network architecture. For the activity recognition and heart rate estimation tasks, we replaced the 1D ResNet with a Fully Convolutional Network (FCN). The results are presented in Tables 23 and 24.

Table 23: Performance comparison of methods on HR estimation

Method	IEEE SPC12			IEEE SPC22			DaLiA		
	MAE↓	RMSE↓	ρ↑	MAE↓	RMSE↓	ρ↑	MAE↓	RMSE↓	ρ↑
Standard training									
Only $\mathcal{D}_{\text{real}}$	5.33±0.43	11.68±0.83	83.14±1.57	5.96±0.12	11.97±0.24	80.13±0.41	5.42±0.067	12.96±0.23	85.13±0.23
Standard techniques									
Aug.	5.62±0.25	11.81±0.85	83.46±1.15	5.57±0.40	11.89±0.70	80.26±0.95	5.67±0.09	13.05±0.11	84.12±0.13
FGSM	5.67±0.25	12.58±0.58	82.99±0.36	5.46±0.12	10.61±0.25	81.57±0.46	5.44±0.12	12.96±0.13	85.87±0.10
PGD	6.10±0.83	13.90±1.23	80.11±2.43	6.25±0.53	11.96±0.83	79.67±1.28	5.81±0.06	12.85±0.17	84.13±0.10
Smoothing (low-pass)									
SAP	6.96±0.63	14.79±0.96	81.58±1.67	5.55±0.62	11.95±1.10	81.69±1.58	5.60±0.17	12.48±0.19	85.10±0.39
SGM	5.99±0.27	13.18±0.25	82.15±0.93	5.33±0.59	10.73±1.44	80.02±1.82	5.86±0.06	12.95±0.13	84.73±0.06
LF-AP	6.59±0.20	12.99±0.27	82.63±0.90	5.96±0.45	11.13±1.58	80.53±1.91	5.80±0.10	12.86±0.20	83.99±0.11
Clean accuracy-focused									
TRADES	6.43±0.58	13.10±0.72	83.19±0.60	6.05±0.42	12.03±0.99	79.18±1.56	5.94±0.13	13.06±0.20	84.03±0.34
FA	6.23±0.27	12.93±0.26	81.82±0.29	6.73±0.52	13.05±0.44	75.34±2.33	6.95±0.43	15.03±0.60	79.37±0.81
DARG	6.67±0.30	12.99±0.37	81.43±0.50	5.71±0.38	11.84±1.39	80.94±1.60	5.96±0.10	13.05±0.20	85.42±0.38
Ours	**5.21**±0.11	**10.72**±0.47	**85.37**±0.63	**4.96**±1.03	**9.84**±1.53	**85.40**±1.80	**5.33**±0.10	**12.12**±0.28	**85.93**±0.58

Table 24: Generalization performance on Activity and Step datasets

Method	UCIHAR			USC			Clemson		
	Acc↑	W-F1↑	F1↑	Acc↑	W-F1↑	F1 ↑	MAPE↓	MAE↓	RMSE↓
Standard training									
Only $\mathcal{D}_{\text{real}}$	91.07±0.47	90.50±0.34	90.95±0.79	72.92±1.09	71.94±1.02	70.88±0.97	4.27±0.26	2.41±0.14	3.65±0.20
Standard techniques									
Aug.	90.81±0.83	90.10±0.99	90.73±0.72	71.56±2.75	70.35±3.11	69.40±2.85	4.60±0.30	2.63±0.17	3.70±0.28
FGSM	91.05±2.92	90.07±3.06	90.96±3.04	71.87±0.62	70.85±0.79	69.86±0.31	4.20±0.19	2.35±0.12	3.64±0.06
PGD	90.90±0.70	89.20±1.32	90.62±0.99	73.44±1.03	72.72±1.42	72.19±1.15	4.00±0.10	2.20±0.09	3.47±0.09
Smoothing (low-pass)									
SAP	91.21±0.81	90.22±1.07	91.35±0.72	73.90±0.84	72.60±1.09	71.97±0.82	4.14±0.39	2.32±0.24	3.68±0.28
SGM	90.98±0.61	90.05±0.95	91.31±0.83	73.90±0.53	73.30±0.85	71.85±0.80	4.47±0.05	2.52±0.04	3.82±0.12
LF-AP	90.57±0.77	89.62±1.00	90.93±0.75	73.67±0.36	72.91±1.11	70.79±1.10	4.87±0.21	2.55±0.18	3.80±0.16
Clean accuracy-focused									
TRADES	90.70±0.86	89.67±0.89	90.85±0.65	70.29±0.11	68.85±0.30	68.21±0.55	4.05±0.05	2.27±0.04	3.53±0.02
FA	86.52±0.63	86.94±1.27	85.13±0.53	72.16±0.85	72.13±0.95	72.35±0.98	4.03±0.58	2.90±0.31	3.93±0.42
DARG	89.97±1.28	88.84±1.52	90.21±1.21	71.16±2.56	70.05±2.50	68.95±2.97	4.62±0.18	2.62±0.10	3.87±0.15
Ours	**92.43**±0.25	**91.73**±0.28	**92.89**±0.21	**77.02**±0.80	**76.22**±0.67	**75.17**±0.62	**3.81**±0.28	**2.12**±0.13	**3.42**±0.22

Our method demonstrates its effectiveness across different architectures. By replacing the 1D ResNet with an FCN, our method consistently improved model performance. These results highlight the robustness of our approach, achieving superior performance across architectures.

D.7.1 FILTER NETWORK

We study the sensitivity of our method to the architecture of the adaptive filter network. In the main experiments, the filter is parameterized by a three-layer MLP. Here, we vary the filter depth while keeping the backbone, optimizer, and adversarial training procedure fixed.

As shown in Table 25, the method is stable across moderately expressive filter networks. The two, three, and four-layer MLPs achieve similar performance, indicating that the adaptive filter does not require a large network. However, reducing the filter to a single linear layer degrades performance across all datasets. This suggests that nonlinear capacity is important for learning the mapping while aligning the perturbations with the input spectrum. Thus, while the method is not highly sensitive to MLP depth, a purely linear filter is too restrictive to model the desired adaptive frequency-support constraint.

Table 25: Ablation study on HR estimation. We compare the adaptive filter network using different levels of depth.

Variant	IEEE SPC12			IEEE SPC22			DaLiA		
	MAE↓	RMSE↓	ρ↑	MAE↓	RMSE↓	ρ↑	MAE↓	RMSE↓	ρ↑
Ours, 4-layer MLP	**5.18**±0.07	**10.57**±0.48	85.33±0.52	**4.75**±0.90	9.76±1.47	85.87±1.70	5.24±0.10	11.84±0.23	86.68±0.44
Ours, 3-layer MLP	5.19±0.10	10.59±0.50	**85.47**±0.60	4.77±0.98	9.76±1.50	**85.90**±1.79	5.23±0.09	11.85±0.25	**86.73**±0.50
Ours, 2-layer MLP	5.20±0.12	10.60±0.50	85.40±0.55	**4.75**±0.93	**9.74**±1.45	85.85±1.63	**5.22**±0.11	**11.83**±0.26	86.70±0.48
Ours, linear filter network	6.90±0.20	11.73±0.43	82.17±0.57	6.18±1.09	10.42±1.69	83.10±1.55	6.33±0.16	12.91±0.30	82.13±0.53

D.8 Investigating the effect of epoch delays on adversarial generation

We generated adversarial examples for every epoch using the randomly chosen quarter of the batch. In our approach, the parametrized filter h_ψ should be trained to filter perturbations for matching the distribution of the frequency support. Thus, we delayed adversarial generation until the filter parameters were updated five times, i.e., after five epochs. Given that our method introduces a unique delay in adversarial generation compared to the baselines, we investigated whether initiating adversarial generation after five epochs of model training impacts the performance of the baseline methods.

We conducted experiments in which all baseline methods delayed the generation of adversarial examples until after five epochs. However, we observed no significant change in the performance of the models; the variation across all methods and metrics was smaller than 0.001. This outcome is rather expected, as the initial epochs and the adversarial examples generated during these early stages have a negligible influence on the overall training process, given that the models continue to train and generate adversarial samples for a much larger number of epochs.

We also conducted the opposite experiment, where our model was trained from the initial epoch without delaying adversarial generation, even though the filter parameters had not yet fully converged. In this case, we observed a performance decrease of approximately 0.01—0.02 on average across the tasks. We believe these two experiments demonstrate that the small number of adversarial samples generated at the beginning of training, during the initial epochs, have minimal impact on the neural network's performance. This suggests that the model ultimately converges to the expected point as training progresses, regardless of whether adversarial generation is delayed a few epochs or initiated immediately.

D.9 ADDITIONAL MODALITIES

We additionally evaluate on audio and financial time series to test generalization at higher sampling rates and under different temporal dynamics. For audio, we use Respiratory@TR (lung sounds) and UrbanSound8K (environmental audio). For finance, we use FI-2010 limit order book data following the standard DeepLOB preprocessing and official train/test splits.

UrbanSound8K. UrbanSound8K consists of environmental audio clips with non-stationary and diverse acoustic conditions (e.g., varying backgrounds and overlapping sources) (Abdoli et al., 2019). While it is not a strictly human-centered dataset, it provides a complementary test for generalization in audio. Although spectrogram-based pipelines are also common for this task, we use raw waveforms to keep the input representation consistent with our main experiments. Following the protocol, we use the official 10-fold cross-validation split (Abdoli et al., 2019).

Respiratory@TR contains lung sounds recorded using two stethoscopes (Altan et al., 2017). Recordings were annotated by two pulmonologists using supporting clinical information (e.g., chest X-rays, pulmonary function tests, and auscultation). Following the protocol in Zhang et al. (2024), we use labels with five COPD severity levels (COPD0–COPD4).

FI-2010 is a limit order book benchmark derived from high-frequency order streams, where each time step is represented by 40 LOB features and labels indicate mid-price movement over multiple prediction horizons. We follow the standard DeepLOB-style preprocessing where we use the first 40 rows as input features. We construct samples using a sliding window of length $T{=}100$, yielding inputs in $\mathbb{R}^{100\times 40}$; the label is taken from horizon k as $y = Y_{t,k} - 1 \in \{0, 1, 2\}$ (default $k{=}4$). For evaluation, we use the original predefined split files: we form an $80/20$ train/validation split by time.

Table 26: Performance comparison of methods in additional modalities

Method	UrbanSound8K			Respiratory@TR			FI-2010		
	Acc↑	W-F1↑	F1↑	Acc↑	W-F1↑	F1 ↑	Acc↑	W-F1↑	F1↑
Standard training									
Only $\mathcal{D}_{\text{real}}$	56.79±0.27	56.91±0.25	55.89±0.21	25.21±5.60	21.21±5.98	57.01±3.62	78.40±0.12	78.47±0.17	78.40±0.20
Standard techniques									
Aug.	56.03±0.71	56.40±0.73	55.46±0.24	20.32±5.18	15.31±6.07	45.81±3.51	78.63±0.27	78.61±0.29	78.54±0.21
FGSM	57.16±0.45	57.96±0.28	57.15±0.32	26.32±3.11	25.28±3.03	55.89±3.45	76.10±0.10	76.01±0.08	75.89±0.12
PGD	57.54±1.22	58.43±1.17	56.62±0.90	24.45±6.13	20.61±6.03	44.78±6.40	73.89±0.22	73.40±0.16	73.22±0.26
Smoothing (low-pass)									
SAP	56.90±0.60	57.30±0.55	56.20±0.60	26.61±5.10	22.83±5.62	58.14±3.30	78.16±0.18	78.01±0.13	78.54±0.17
SGM	55.15±1.80	55.60±1.90	54.60±1.70	25.97±5.31	22.10±5.89	57.43±3.48	77.42±0.20	77.18±0.18	77.38±0.19
LF-AP	54.60±1.50	55.20±1.40	54.15±1.60	26.20±5.00	22.50±5.60	57.63±3.32	77.90±0.17	77.83±0.16	77.88±0.17
Clean accuracy-focused									
TRADES	55.79±0.92	57.18±0.85	55.12±0.79	27.80±4.80	24.10±5.20	58.60±3.10	78.68±0.32	78.64±0.23	78.57±0.20
FA	55.83±0.30	55.82±0.24	54.78±0.39	27.10±4.90	23.40±5.30	58.10±3.20	77.12±0.15	77.04±0.12	77.93±0.14
Ours	**60.02**±0.33	**60.87**±0.23	**59.22**±0.13	**33.32**±3.89	**32.74**±4.43	**60.95**±2.88	**80.07**±0.22	**79.67**±0.31	**79.92**±0.30

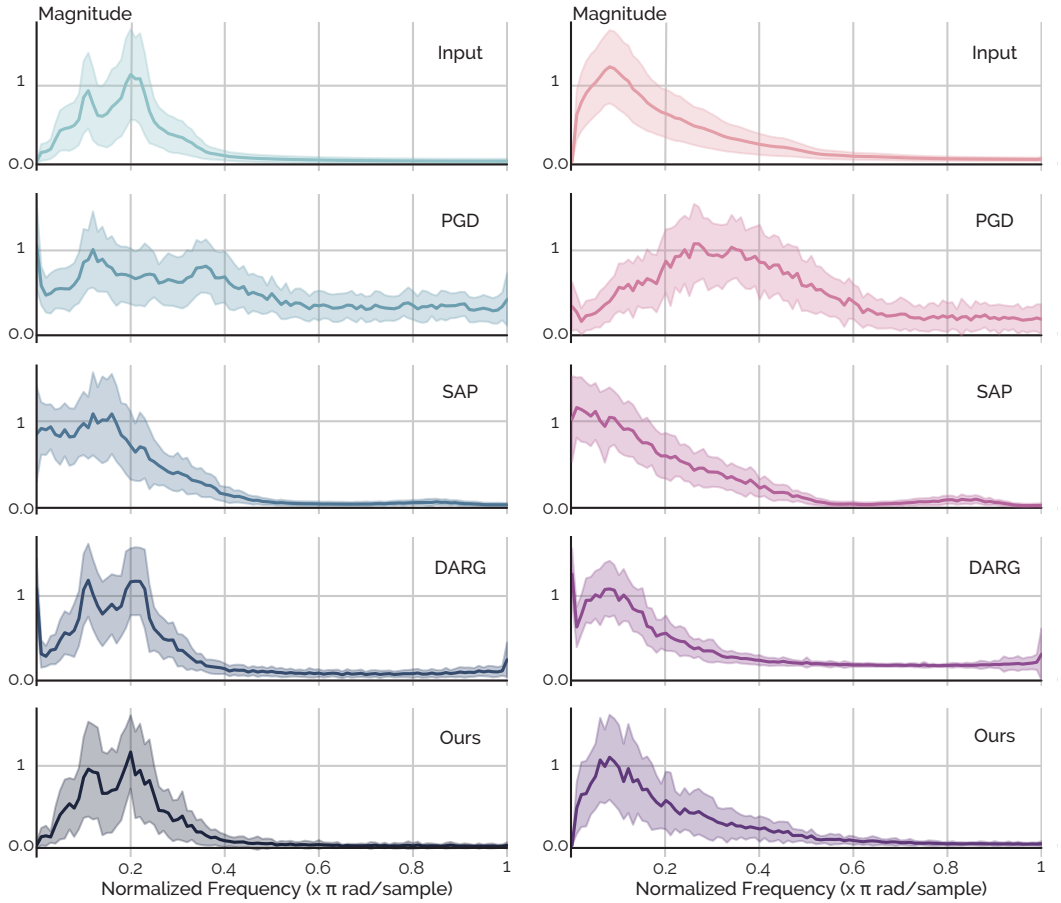
For audio, we apply the same update as in the main method, but in a short-time setting to handle non-stationary waveforms. Given an input $x \in \mathbb{R}^{T\times C}$, we split it into K (8 for UrbanSound8k and 10 for Respiratory) equal temporal chunks without overlap. Within each chunk, we run the same update to obtain perturbations. We then stitch the chunks back into a waveform, similar to *window* variant evaluation in the main results. Technically, this is the same proposed algorithm under a weaker stationarity assumption where the signal is treated as approximately stationary within each chunk rather than over the full sequence.

E SPECTRAL ANALYSIS OF PERTURBATIONS

In this section, we provide visual examples of the spectra of the input signals and the crafted perturbations across datasets, together with representative time-domain examples.

E.1 SPECTRA OF INPUTS AND PERTURBATIONS

We first compare the spectra of the input signals and the perturbations generated by different methods. For visualization, each sample is standardized before computing its Fourier transform. We then report the mean and standard deviation of the resulting magnitude spectra across the dataset. The spectral magnitude is shown on a linear scale, and frequency is reported in normalized radians per sample.

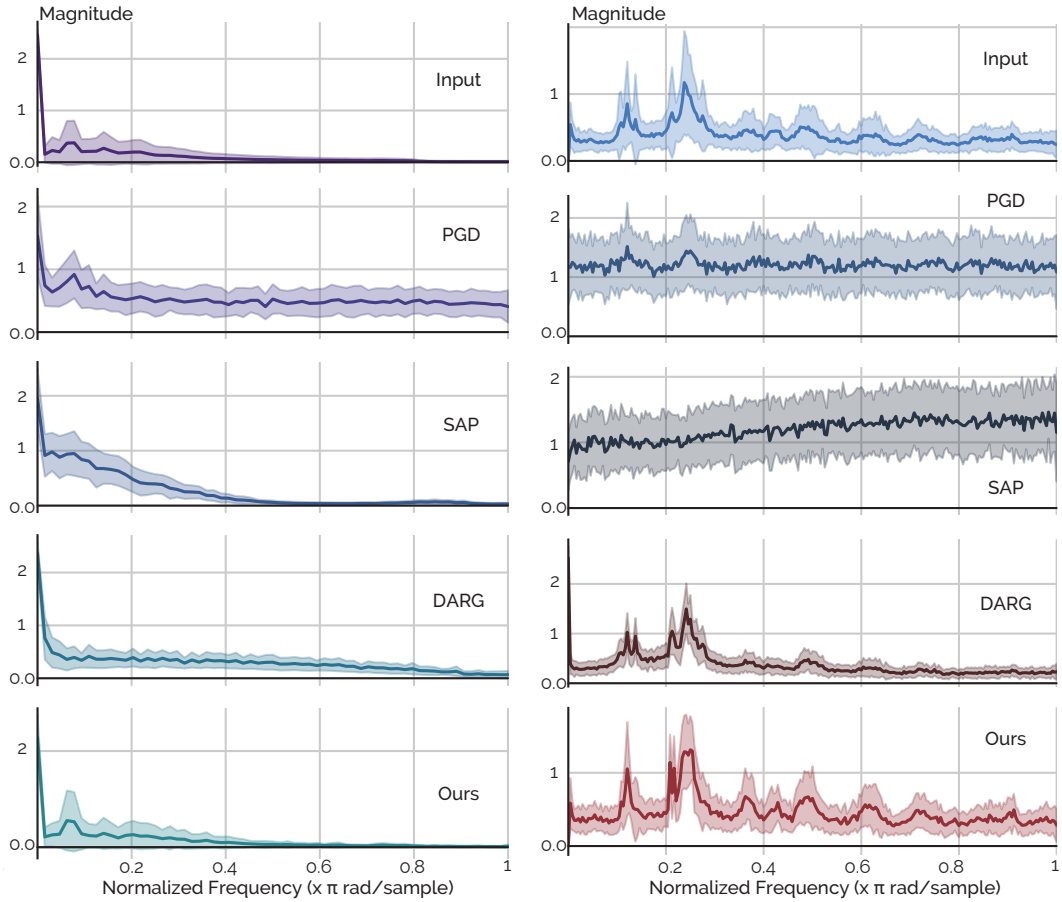


(a) Mean and standard deviation of the spectrum for inputs and perturbations in the IEEE SPC12 dataset.

(b) Mean and standard deviation of the spectrum for inputs and perturbations in the DaLiA dataset.

Figure 11: Frequency spectra of input signals and adversarial perturbations generated by different methods for two datasets in the heart-rate estimation task: (a) IEEE SPC12 and (b) DaLiA.

For multi-channel IMU tasks, such as activity recognition from accelerometer and gyroscope signals, we first computed the resultant magnitude across the three spatial axes and then applied the Fourier transform to the resulting sequence. The corresponding spectra are shown in Figure 12. For the Clemson, the input is already represented as a magnitude signal, so we directly computed its spectrum.



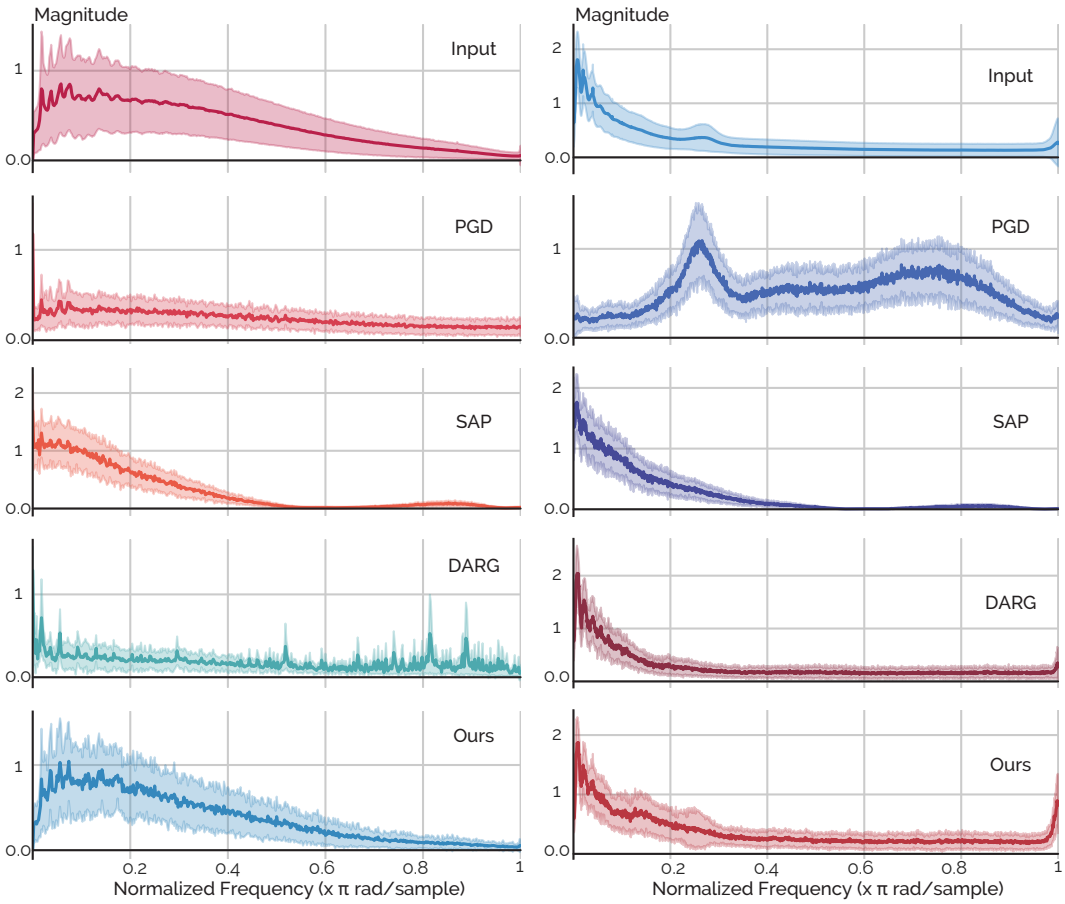
(a) Mean and standard deviation of the spectrum for inputs and perturbations in the UCIHAR dataset.

(b) Mean and standard deviation of the spectrum for inputs and perturbations in the Clemson dataset.

Figure 12: Frequency spectra of input signals and adversarial perturbations generated by different methods for IMU-based tasks: (a) UCIHAR for activity recognition and (b) Clemson for step counting.

One of the compared methods, SAP Han et al. (2020), uses a hand-crafted Gaussian filter with fixed kernel size and standard deviation. Although these parameters encourage smoothness, they do not adapt to the spectral characteristics of the input. As a result, performance can vary across datasets, and in some cases the filter does not sufficiently suppress irrelevant frequency regions. This behavior is visible in Figure 12b, where the perturbation spectrum remains broadly spread. Such behavior highlights the limitation of static filtering strategies when applied across heterogeneous tasks.

For ECG datasets with multiple leads, such as Chapman Zheng et al. (2020), we report the spectrum of lead II, as shown in Figure 13a.



(a) Mean and standard deviation of the spectrum for inputs and perturbations in the Chapman dataset.

(b) Mean and standard deviation of the spectrum for inputs and perturbations in the Sleep-EDF dataset.

Figure 13: Frequency spectra of input signals and adversarial perturbations generated by different methods for (a) cardiovascular disease classification on Chapman and (b) sleep-stage classification on Sleep-EDF.

The differences between the input spectra and the crafted perturbation spectra are shown in Figures 11, 12, and 13. These visualizations help explain the behavior of the different perturbation-generation methods. In particular, PGD often introduces spectral components that are weak or absent in the observed data. By contrast, our method keeps perturbations concentrated within the effective support of the input spectrum while avoiding introducing weakly represented harmonics into the perturbations.

This observation helps explain why unconstrained perturbations can reduce clean-data performance during adversarial training. When perturbations contain oscillations that are not supported by the data distribution, the model is encouraged to fit unrealistic spectral patterns.

E.2 Quantitative Spectral Alignment

Beyond the visual examples in Appendix E, we quantify spectral agreement between each clean sample $\boldsymbol{x}$ and its matched-strength perturbation δ. We compute their one-sided RFFT power spectra, average across channels, and normalize each spectrum to sum to one, obtaining $p_{\boldsymbol{x}}$ and p_{δ}.

We report four measures: (i) Jensen–Shannon divergence (JSD) between the normalized spectra; (ii) the fraction of perturbation energy outside the smallest set of frequency bins containing 99% of

Table 27: Spectral alignment across representative datasets. Outside-99 denotes the fraction of perturbation energy outside the smallest set of bins containing 99% of the input energy. Bold marks the closest agreement for each metric.

Dataset	Attack	JSD↓	Outside-99↓	HF excess→ 0	Peak shift (bins)↓
Clemson	AutoAttack	0.1216	0.0985	0.1510	75.4
	PGD-ℓ_∞	0.1221	0.0979	0.1572	77.4
	SAP	0.2388	0.1299	-0.2076	55.4
	SGM	0.1686	0.1179	0.3035	128.9
	TRADES	0.1122	0.0969	0.1265	61.7
	Ours	**0.0363**	**0.0000**	**0.0032**	**10.3**
Sleep-EDF	AutoAttack	0.4561	0.8372	0.2718	633.5
	PGD-ℓ_∞	0.4652	0.8483	0.2865	682.2
	SAP	0.2013	0.4428	-0.1005	27.4
	SGM	0.5812	0.9276	0.3562	1167.8
	TRADES	0.4451	0.8204	0.2465	559.5
	Ours	**0.0207**	**0.0000**	**0.0035**	**12.4**
UCIHAR	AutoAttack	0.0885	0.0262	0.1968	2.3
	PGD-ℓ_∞	0.0888	0.0254	0.1985	2.5
	SAP	0.0633	0.0026	-0.3560	2.9
	SGM	0.2524	0.0524	0.4535	54.5
	TRADES	0.0958	0.0249	0.1943	1.6
	Ours	**0.0238**	**0.0000**	**0.0040**	**0.5**
Chapman	AutoAttack	0.1294	0.3284	0.1555	34.3
	PGD-ℓ_∞	0.1356	0.3290	0.1560	34.1
	SAP	0.1378	0.1342	-0.0887	28.2
	SGM	0.2032	0.5116	0.3650	346.6
	TRADES	0.1648	0.3205	0.1743	28.6
	Ours	**0.0635**	**0.0000**	**-0.0051**	**8.7**

the input energy; (iii) high-frequency (HF) excess; and (iv) the absolute difference between their peak-frequency bins.

For a window of length T, the one-sided spectrum contains $F = \lfloor T/2 \rfloor + 1$ bins. We define the high-frequency region as the upper approximately 40% of these bins,

$$\mathcal{H} = \{\lfloor 0.6F \rfloor, \ldots, F-1\}.$$

HF excess measures the difference between perturbation and input energy fractions in this region:

$$\text{HF-excess} = \sum_{k \in \mathcal{H}} p_\delta(k) - \sum_{k \in \mathcal{H}} p_{\boldsymbol{x}}(k).$$

The bin-selection rule is fixed across datasets and methods and does not use training data. Its corresponding frequency range depends on the sampling rate and window length, with bin k corresponding to kf_s/T Hz.

Lower JSD, lower outside-region energy, and smaller peak shift indicate closer spectral agreement. HF excess near zero indicates similar high-frequency energy fractions; positive and negative values indicate greater and smaller fractions in the perturbation, respectively. Unlike the sample-dependent 99% energy region, $\mathcal{H}$ is a fixed upper-frequency band.

Table 27 summarizes the results. Across all datasets, our method produces the lowest JSD and peak-frequency shift, places negligible perturbation energy outside the 99% input region, and maintains high-frequency excess close to zero. The differences are particularly large on Sleep-EDF, where unconstrained and smoothing-based methods distribute substantial perturbation energy outside the dominant input frequencies. These results show that the learned filter aligns perturbations with the input-dependent spectral distribution without imposing a fixed low-pass structure.

This analysis examines how different perturbation methods redistribute spectral mass. We next investigate whether the input spectra themselves exhibit sufficient concentration to motivate an input-dependent spectral constraint. Although finite noisy signals are not sparse, concentration of most energy within a limited set of frequencies makes distributional spectral alignment meaningful in practice.

E.2.1 Spectral Deviation without a Prescribed Threshold

Equation 1 expresses spectral alignment through a threshold κ on the divergence between normalized magnitude spectra. This threshold specifies an admissible spectral deviation. Our formulation instead balances the adversarial task loss against a spectral-alignment penalty. Specifically, the filter minimizes

$$\mathcal{L}_{\text{filter}} = -\mathcal{L}_{\text{CE}} + \lambda_{\text{KL}} D_{\text{KL}}\big(p_{\Delta_\psi} \,\big\|\, q_{\boldsymbol{x}}\big)\,,$$

where p_{Δ_ψ} is computed from the final accumulated perturbation Δ_ψ. The task loss encourages loss-increasing perturbations, while the KL term discourages spectral deviation. Their interaction determines the learned spectral weights and the resulting divergence for each input.

Thus, λ_{KL} controls the strength of spectral regularization, whereas κ in Equation 1 specifies a hard admissibility threshold. We do not set a hard κ. In our sensitivity experiments, alternative penalty weights changed the achieved divergence but gave mixed downstream performance relative to $\lambda_{\text{KL}} = 1$. We therefore retain $\lambda_{\text{KL}} = 1$ as the common setting across applications and datasets, without dataset-specific tuning.

To relate the learned perturbations to κ in Equation 1, we measure their achieved divergence $D_{\text{KL}}(p_{\Delta_\psi} \| q_{\boldsymbol{x}})$. We use normalized magnitude spectra and the same numerical stabilization as the training objective.

On IEEE SPC12, approximately 95% of the 1,768 evaluated perturbations have divergence at most 0.361. Thus, choosing $\kappa = 0.361$ in Equation 1 would admit approximately 95% of these perturbations. On Sleep-EDF, the corresponding threshold is $\kappa = 1.382$ over 8,910 perturbations.

These results illustrate the spectral deviations produced by the same regularization weight $\lambda_{\text{KL}} = 1$ on two different datasets. A hard threshold κ would require specifying which deviations are admissible before learning. However, the observed spectrum is estimated from a finite, noisy window, and its small-magnitude components depend on noise, window length, and preprocessing. It therefore provides a useful spectral prior without directly defining a calibrated boundary between admissible and inadmissible perturbations. We use this prior through a soft penalty, allowing the task loss and spectral regularization to jointly determine the achieved deviation. This choice retains a common regularization weight across datasets without imposing a common divergence ceiling.

The remaining question is whether the observed spectra contain enough structure to make this prior informative. We address this by measuring spectral concentration.

E.3 Empirical Spectral Concentration

Finite and noisy signals typically have nonzero Fourier coefficients across many frequency bins. The practical question is therefore not whether the spectrum is exactly sparse, but whether its energy is concentrated in a limited subset of frequencies. We examine this property using the model inputs after the same preprocessing used in the main experiments.

For each input $\boldsymbol{x}$, we compute the normalized one-sided power spectrum $p_{\boldsymbol{x}}(\omega)$ and let N denote the number of one-sided frequency bins. We measure normalized spectral entropy as $-\sum_\omega p_{\boldsymbol{x}}(\omega) \log p_{\boldsymbol{x}}(\omega) / \log N$, where lower values indicate greater spectral concentration. We also report Hoyer sparsity (Hoyer, 2004), calculated from the magnitude spectrum as below.

$$\frac{\sqrt{N} - \|\boldsymbol{a}_{\boldsymbol{x}}\|_1 / \|\boldsymbol{a}_{\boldsymbol{x}}\|_2}{\sqrt{N} - 1}, \quad \text{where } \boldsymbol{a}_{\boldsymbol{x}} = |\,\text{rFFT}(\boldsymbol{x})| \tag{31}$$

which ranges from 0 for a uniform spectrum to 1 when only one bin has nonzero magnitude. The 99% support fraction is the smallest fraction of frequency bins whose combined energy reaches 99%, after ranking bins by energy. Finally, the 99% bandwidth is the narrowest contiguous frequency interval containing the same energy. Together, these metrics quantify spectral concentration without requiring any frequency bin to be exactly zero.

Table 28 shows strong spectral concentration in IEEE SPC. Sleep-EDF is more heterogeneous, but its median input still concentrates 99% of the spectral energy within 16.3% of the frequency bins. Its larger contiguous bandwidth indicates that the dominant frequencies can be distributed across separated spectral regions rather than confined to a single narrow band. This difference supports an input-dependent spectral constraint instead of a fixed low-pass or band-pass filter across modalities.

Table 28: Empirical spectral concentration of the preprocessed model inputs. Support denotes the fraction of frequency bins required to contain 99% of the spectral energy, while bandwidth is the narrowest contiguous interval containing the same energy.

IEEE SPC

Metric	Mean	Med.	Std.	$Q_{.05}$	$Q_{.25}$	$Q_{.75}$	$Q_{.95}$
Entropy	.508	.512	.084	.374	.450	.573	.632
Hoyer sparsity	.608	.607	.060	.518	.575	.653	.695
Support	.276	.257	.078	.208	.238	.287	.402
Bandwidth (Hz)	3.673	3.500	.914	3.000	3.250	3.875	5.038

Sleep-EDF

Metric	Mean	Med.	Std.	$Q_{.05}$	$Q_{.25}$	$Q_{.75}$	$Q_{.95}$
Entropy	.653	.663	.083	.538	.597	.707	.787
Hoyer sparsity	.542	.543	.115	.312	.466	.610	.690
Support	.218	.163	.148	.064	.116	.276	.524
Bandwidth (Hz)	30.48	28.18	14.12	10.46	17.33	45.39	49.34

These results do not assume that finite-sample spectra contain exact zeros. Rather, they show that the empirical spectra are strongly nonuniform, making distributional alignment meaningful even when every frequency bin contains some energy. Figure 14 provides representative time and frequency-domain examples.

These results show that the observed spectra exhibit a well-defined effective spectral concentration. Accordingly, our practical formulation does not rely on identifying a binary cutoff from finite signals. Instead, the KL objective aligns the perturbation distribution with the empirical input spectrum and limits perturbation mass in frequency regions receiving little input spectral mass. The expert assessment in Appendix D.3.3 separately evaluates whether this spectral alignment produces physiologically and acquisition-consistent perturbations.

Figure 14 complements the dataset-level statistics with examples from IEEE SPC and Sleep-EDF. The shaded region denotes the narrowest contiguous band containing 99% of the spectral energy. The Sleep-EDF examples in Figures 14(c)–(d) contain broader temporal dynamics, yet 99% of their energy remains within 7.87 Hz and 11.80 Hz. Although small coefficients remain outside these regions, the spectra are strongly concentrated. This illustrates why exact Fourier sparsity is unnecessary for our method. The KL term can align perturbations with the dominant empirical spectral distribution while suppressing mass in the low-energy tails.

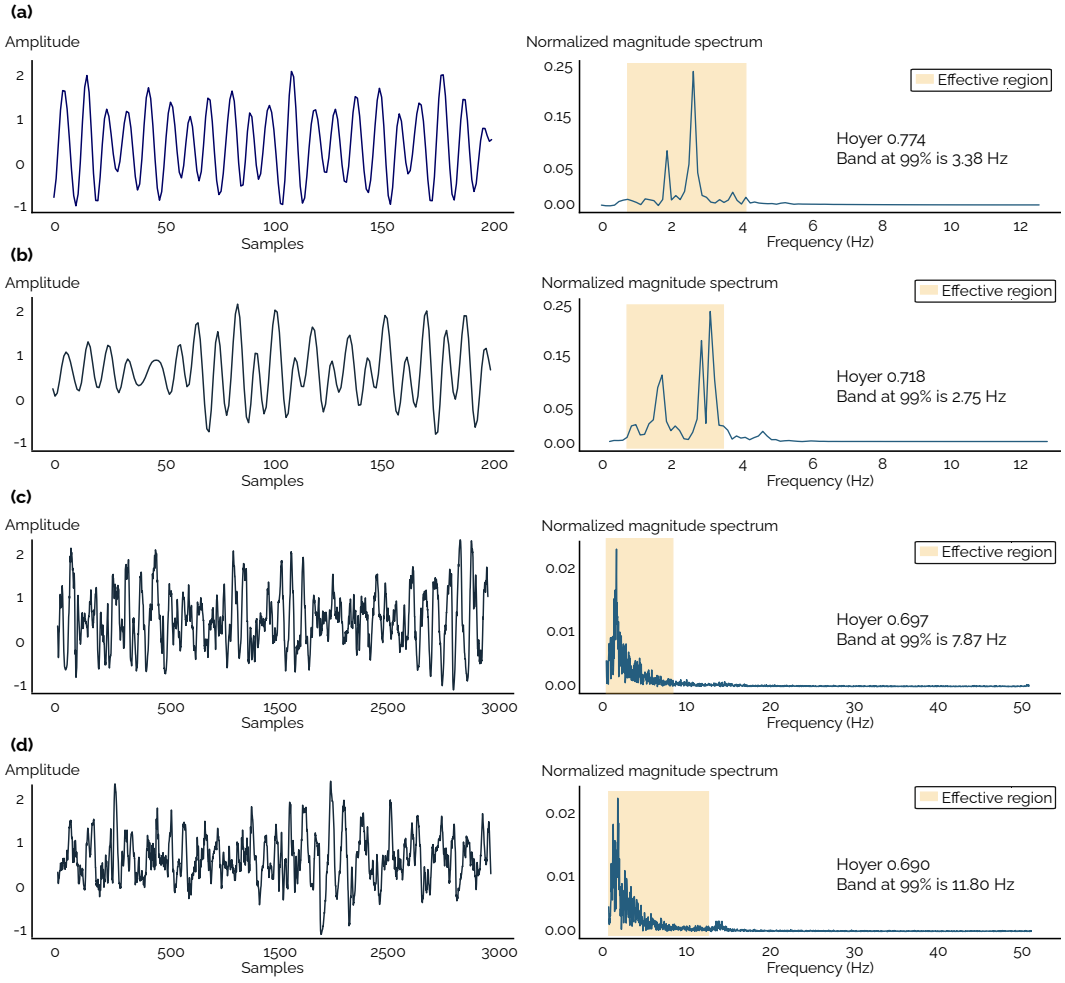


Figure 14: Examples of empirical spectral concentration using the original samples from **(a)–(b)** IEEE SPC and **(c)–(d)** Sleep-EDF. The shaded region is the narrowest contiguous frequency band containing 99% of the spectral energy. Most of the energy remains concentrated within a limited, input-dependent bandwidth.

E.4 Controlled Spectral Energy Allocation

We examine whether the frequency allocation of adversarial perturbations affects generalization. The preceding analysis establishes that the input spectra are concentrated, but does not establish whether placing perturbation energy outside their effective spectral support changes the benefit of adversarial training. We therefore vary the fraction of perturbation energy outside this support while keeping the remaining training procedure unchanged.

Effective spectral support. For each preprocessed training input $\boldsymbol{x}$, we define $S(\boldsymbol{x})$ as the smallest set of frequency bins containing at least 99% of its spectral energy. Bins are ranked by power, with ties resolved by frequency index. This is the same energy threshold used in Section E.3. The set may contain separated frequency regions and is not restricted to a contiguous band. It is computed from the unperturbed input and held fixed when constructing the corresponding perturbation. No test inputs or labels are used to select the support or perturbation parameters.

Controlled perturbations. We compute the gradient of the training loss with respect to the input, $\boldsymbol{g} = \nabla_{\boldsymbol{x}} \mathcal{L}(f_\theta(\boldsymbol{x}), y)$. We then separate this gradient into components inside and outside the effective support as in Equation 32.

$$\boldsymbol{g}_{\text{in}} = \mathcal{F}^{-1}\left[\mathbf{1}_{S(\boldsymbol{x})} \odot \mathcal{F}(\boldsymbol{g})\right], \quad \boldsymbol{g}_{\text{out}} = \mathcal{F}^{-1}\left[\mathbf{1}_{S(\boldsymbol{x})^c} \odot \mathcal{F}(\boldsymbol{g})\right]. \tag{32}$$

The transform is applied along the time axis. For multichannel inputs, support is determined separately for each channel, and the resulting components are normalized jointly across channels and time. The masks preserve conjugate symmetry so that the reconstructed perturbations are real valued. One-sided power calculations include the appropriate weights for conjugate frequency pairs.

For an outside-support energy fraction ρ, we construct a unit-norm perturbation direction as follows.

$$\boldsymbol{d}_\rho = \sqrt{1-\rho}\,\frac{\boldsymbol{g}_{\text{in}}}{\|\boldsymbol{g}_{\text{in}}\|_2} + \sqrt{\rho}\,\frac{\boldsymbol{g}_{\text{out}}}{\|\boldsymbol{g}_{\text{out}}\|_2}. \tag{33}$$

The two components occupy disjoint frequency bins and are orthogonal. Consequently, $\|\boldsymbol{d}_\rho\|_2 = 1$, and exactly a fraction ρ of its energy lies outside $S(\boldsymbol{x})$, up to numerical precision. We evaluate $\rho \in \{0, 0.25, 0.5, 0.75, 1\}$. Thus, $\rho = 0$ places all perturbation energy inside the effective support, whereas $\rho = 1$ places all energy outside it.

These controlled perturbations are constructed from the loss gradient without the learned filter. This allows us to examine spectral energy allocation without introducing differences in learned filter architecture or filter training. Perturbations are applied to the preprocessed model inputs. We do not subsequently filter, clip, or renormalize the perturbed inputs, as these operations could change the prescribed energy allocation. We verify both the perturbation norm and the realized outside-support energy fraction numerically.

Equal perturbation magnitude. We construct perturbations as $\boldsymbol{\delta}_\rho = \epsilon_{\boldsymbol{x}} \boldsymbol{d}_\rho$. For each input, the ℓ_2 budget $\epsilon_{\boldsymbol{x}}$ is the same across all values of ρ. This keeps perturbation magnitude fixed while changing its spectral allocation. The budget is determined with validation data.

Training and evaluation. We evaluate IEEE SPC, Chapman, PhysioNet, and Sleep-EDF. For each dataset, we train a separate model for each value of ρ. We use the backbone, preprocessing, data partitions, clean and perturbed loss weights, and checkpoint-selection criterion from the main experiments. We also keep the optimizer, training duration, batch size, and fraction of perturbed samples fixed. Within each seed, models start from the same initialization and use the same source-sample order. We repeat the experiment with five paired random seeds. For each model, perturbations are constructed using one input-gradient calculation at the unperturbed input. The gradient calculation follows the same procedure for every value of ρ.

Table 29 reports held-out performance as mean $\pm$ standard deviation across seeds. The comparison measures the effect of spectral energy allocation at matched perturbation magnitude. The results discussion considers both the mean differences and the variation across seeds.

Table 29 shows that allocating all or most perturbation energy inside the effective support gives the best mean performance across the evaluated physiological datasets. On PhysioNet, $\rho = 0$ gives the

Table 29: Held-out performance under controlled spectral energy allocation for physiological signals at equal ℓ_2 norm. The parameter ρ denotes the fraction of perturbation energy outside the 99% effective input support. Entries report mean $\pm$ standard deviation across seeds. Bold indicates the best mean for each metric, including ties at the reported precision.

Dataset	Metric	$\rho = 0$	$\rho = 0.25$	$\rho = 0.50$	$\rho = 0.75$	$\rho = 1$
IEEE SPC	RMSE (bpm) ↓	12.062 ± 0.480	$\mathbf{11.891 \pm 0.704}$	12.450 ± 1.135	12.509 ± 1.219	12.771 ± 1.456
	MAE (bpm) ↓	$\mathbf{5.475 \pm 0.239}$	5.578 ± 0.375	5.737 ± 0.469	5.690 ± 0.438	5.774 ± 0.607
	Pearson r ↑	$\mathbf{0.834 \pm 0.012}$	0.832 ± 0.017	0.824 ± 0.023	0.827 ± 0.018	0.819 ± 0.026
Chapman	Accuracy (%) ↑	$\mathbf{92.516 \pm 0.425}$	92.264 ± 0.432	92.233 ± 0.360	92.148 ± 0.098	92.030 ± 0.125
	AUROC (OVO) ↑	$\mathbf{0.986 \pm 0.002}$	0.986 ± 0.002	0.984 ± 0.002	0.981 ± 0.002	0.980 ± 0.002
	Macro F1 (%) ↑	$\mathbf{91.586 \pm 0.422}$	91.241 ± 0.485	91.214 ± 0.422	91.167 ± 0.196	91.140 ± 0.159
PhysioNet	Accuracy (%) ↑	$\mathbf{82.796 \pm 2.100}$	82.171 ± 2.286	82.425 ± 1.593	82.067 ± 1.030	82.002 ± 1.295
	AUROC (OVO) ↑	$\mathbf{0.932 \pm 0.007}$	0.929 ± 0.006	0.923 ± 0.003	0.920 ± 0.002	0.918 ± 0.003
	Macro F1 (%) ↑	$\mathbf{73.285 \pm 3.424}$	72.525 ± 3.953	72.316 ± 3.421	72.307 ± 2.831	72.179 ± 3.635
Sleep-EDF	Accuracy (%) ↑	77.134 ± 3.376	$\mathbf{77.396 \pm 2.393}$	76.839 ± 2.615	76.528 ± 2.849	75.331 ± 3.365
	Weighted F1 (%) ↑	77.400 ± 2.674	$\mathbf{77.574 \pm 1.972}$	75.915 ± 2.419	75.177 ± 1.872	74.983 ± 3.288
	Cohen's κ ↑	0.694 ± 0.043	$\mathbf{0.696 \pm 0.031}$	0.684 ± 0.036	0.680 ± 0.036	0.675 ± 0.035

highest mean values for all three metrics. Sleep-EDF favors $\rho = 0.25$, which places 75% of the perturbation energy inside the effective support.

These results favor concentrating perturbation energy within the effective input support. Although, they do not show a strictly monotonic relationship between outside-support energy and performance, there is a clear trend where the performance is higher for the concentrated perturbations. The physician assessment in Appendix D.3.3 provides separate evidence about the physiological plausibility of the evaluated perturbation methods.

F COMPUTATIONAL OVERHEAD

We compare the computational cost of adversarial sample generation for SGM, SAP, and our method. For an input with K channels and length L, let $C_{\nabla f}$ denote the cost of a predictor forward pass and the backward pass required to compute an input gradient. This cost depends on the predictor architecture and cannot generally be expressed using only the input size and number of output classes. The expressions below include predictor-gradient computations and the additional operations introduced by each method.

F.1 SGM

SGM iteratively optimizes the perturbation using a task loss and a fused Lasso penalty. Each iteration requires a predictor-gradient computation. Evaluating the fused Lasso penalty and updating the perturbation with Adam both require $O(KL)$ operations. For T_{SGM} iterations, the total cost is

$$O(T_{\mathrm{SGM}}\left[C_{\nabla f} + KL\right])$$

F.2 SAP

SAP smooths the perturbation using a collection of Gaussian kernels. For direct convolution, applying all kernels costs $O(KL\sum_{s\in\mathcal{S}} s)$, where $\mathcal{S}$ contains their lengths. In the implementation considered here, smoothing is repeated $T_{\mathrm{SAP}} = 2 \times \texttt{steps}$ times. Including the predictor-gradient evaluations used during perturbation generation gives

$$O\left(n_{\mathrm{SAP}} C_{\nabla f} + T_{\mathrm{SAP}} KL \sum_{s\in\mathcal{S}} s\right),$$

where n_{SAP} is the total number of predictor-gradient evaluations. This accounts separately for model evaluations and smoothing operations.

F.3 OURS

The input-conditioned filter response is computed once per sample and reused across all adversarial steps. Let C_ψ denote the cost of evaluating the filter network. Each step computes an input gradient and applies temporal FFTs and inverse FFTs across K channels, costing $O(KL\log L)$. Spectral multiplication, normalization, and norm rescaling require $O(KL)$ operations. For T_{ours} adversarial steps, perturbation generation therefore costs

$$O(C_\psi + T_{\mathrm{ours}}\left[C_{\nabla f} + KL\log L\right]).$$

Filter training adds the final adversarial task-loss evaluation and the spectral KL penalty. Computing the final accumulated perturbation's spectrum costs $O(KL\log L)$, followed by $O(KL)$ operations for normalization and KL evaluation.

Table 30 reports adversarial sample generation time on an NVIDIA GeForce RTX 4090 GPU, averaged over three runs. The benchmark uses 128 samples per batch from Dalia. The generation settings use 10 iterations for all methods. Our method has the lowest measured generation time among the methods.

Table 30: Adversarial sample generation time per batch on an NVIDIA GeForce RTX 4090. Values are mean ± standard deviation over three runs.

Metric	**Ours**	**SGM**	**SAP**
Time (s)	0.4612 ± 0.009	1.3165 ± 0.14	0.6026 ± 0.11

G Expanded Review of Related Work

In this section, we expand the related-work discussion, with emphasis on methods that aim to reduce unrealistic perturbations and better preserve the structure of observed time-series signals.

Dynamic time warping: Crafting adversarial perturbations for time series is challenging because simple additive distortions can easily alter waveform structure in unrealistic ways (Han et al., 2020; Pialla et al., 2022). Dynamic Time Warping (DTW) is often used to compare temporal signals through alignment rather than pointwise Euclidean distance, and has therefore been considered as a tool for generating similar temporal samples (Belkhouja et al., 2023). However, DTW has important limitations in our setting.

First, DTW is not designed to explicitly control spectral structure. Signals with substantial frequency differences can still yield relatively small DTW distance if they remain temporally alignable (Belkhouja et al., 2023). This is problematic in our setting, where preserving physically plausible frequency support is central.

Second, DTW-based methods are computationally expensive. Standard DTW requires dynamic programming over a cost matrix $C \in \mathbb{R}^{T\times T}$, leading to $O(T^2)$ time complexity (Wang & Jiang, 1994; Shapira Weber et al., 2019). This becomes costly when perturbations must be recomputed repeatedly during adversarial training.

In our preliminary experiments, a recent DTW-based adversarial method (Belkhouja et al., 2023) was substantially slower than the competing approaches and did not achieve competitive performance. For this reason, we did not include it in the main comparison.

Adversarial examples closer to real data: A major line of work seeks to generate adversarial examples that remain close to real data by approximating the data distribution with generative models and perturbing samples in the latent space (Stutz et al., 2019; Lin et al., 2020; Zhao et al., 2018). These methods are appealing because they attempt to remain on or near a learned data manifold, but they also have several limitations.

First, they typically require large datasets and expressive generative models in order to approximate the data distribution reliably (Lin et al., 2020). Their success therefore depends on both model capacity and the suitability of the chosen inductive bias.

Second, the geometry of the latent space is model-dependent and may not provide a reliable notion of semantic or physical proximity. As a result, controlling perturbation size in latent space does not always translate into controlled perturbations in the observed signal space.

Third, many generative methods impose a prior on the latent representation. When that prior is mismatched to the true data distribution, the learned representation can distort the underlying structure and degrade the quality of the generated adversarial examples (Connor et al., 2021).

Our method takes a different route. Rather than learning a full generative model of the data, we constrain perturbations in the observed signal domain by matching their spectral distribution with the input. This avoids reliance on latent priors and reduces modeling overhead.

Adversarial samples in the frequency domain The spectral structure of adversarial perturbations has been widely studied Yin et al. (2019); LI et al. (2024). Early work asked whether adversarial perturbations concentrate in high or low-frequency bands (Yin et al., 2019; Luo et al., 2022; Chen et al., 2023), while more recent studies suggest that their spectral behavior depends on the dataset and signal characteristics (Chen et al., 2022b; Ortiz-Jimenez et al., 2020; Maiya et al., 2023). Our empirical results are consistent with this view.

Several adversarial methods can expand effective frequency support, introducing spectral components that are weak or absent in the observed input (Subramanian et al., 2023). This motivates examining spectral alignment alongside attack strength, particularly when perturbations are used for training.

FA (Jia et al., 2022) optimizes perturbations under a DCT-domain norm constraint and uses input-dependent weighting to adjust updates across frequency bands. IFAP (Yoshida & Okuda, 2025) takes a more explicit approach to spectral alignment by clipping DCT-domain perturbation coefficients

within an envelope proportional to the input's spectral magnitude. Both methods use adversarial gradients, while their spectral weighting or bounds are prescribed rather than learned through a filter network.

Our method learns input-conditioned frequency weights and uses soft regularization to encourage alignment of filtered gradients with the input spectrum. We additionally address the effect of filtering on the update direction. For a given magnitude response, retaining the gradient's Fourier phase minimizes angular deviation from the original gradient. Our contribution combines learned spectral weighting with this explicit minimum-angle phase selection for adversarial training on time series.