BEYOND WHITE NOISE: INTEGRATING SPECTRAL GAUSSIAN PROCESS INTO DIFFUSION MODELS FOR WIRELESS SIGNAL RECOGNITION PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Wireless signal recognition (WSR) aims to infer attributes of passively received wireless signals, and is widely used in radio system. Existing WSR models can learn dependencies among discrete wireless signal samples implicitly, but generally do not make the signal correlation structure an explicit target or prior during representation learning. Consequently, these dependencies must be discovered from the training data and architectural bias alone, without a direct covariance-aware learning objective. Therefore, we propose Spectral Gaussian Process Diffusion (SGP-Diff), a self-supervised framework that replaces isotropic white noise with length-scale-conditioned spectral GP noise that couples neighboring frequency bins. The model predicts this structured noise with a covariance-normalized denoising objective. A linear second-order analysis shows how the optimal denoiser depends jointly on the data and noise covariances. Across four datasets covering three WSR tasks, SGP-Diff obtains the best reported accuracy among the compared methods and exceeds the strongest baseline by 1.83% on average. Further analysis results have validated the effectiveness and practical efficiency of the designed module.

1 INTRODUCTION

Wireless signals, as a crucial carrier of information, are widely present in modern communication systems Vinciarelli et al. (2008). Wireless Signal Recognition (WSR) Li et al. (2019) aims to identify key attributes of wireless signals under passive reception conditions, such as modulation type Meng et al. (2018) and wireless technology Bitar et al. (2017), thus providing essential prior references for subsequent refined processing. This passive and blind signal sensing capability is of great significance in scenarios such as spectrum monitoring Kulin et al. (2018) and anomalous signal detection O'Reilly et al. (2014). In recent years, the rapid development of deep learning has brought significant breakthroughs to WSR Wang et al. (2020). The research paradigm gradually shifted from manual feature engineering to end-to-end representation learning Li et al. (2019). In particular, self-supervised pre-training enables models to learn general representations from large-scale unlabeled signals and to be efficiently adapted to downstream tasks through fine-tuning with few labeled samples. This positions WSR as one of the most important tasks in 6G intelligent communication Alsharif et al. (2020).

Real-world wireless signals are continuous analog waveforms that are converted into discrete signal sequences by an analog-to-digital converter Walden (1999). Bandlimiting, pulse shaping, modulation, and channel dynamics induce structured dependencies among discrete samples, which inherently encode discriminative information. Recent WSR models, including the supervised IQFormer Shao et al. (2024) and the self-supervised pretraining methods SpectrumFM Zhou et al. (2025) and TS-DDAE Liu et al. (2026b), may learn inter-sample dependencies implicitly. However, their learning objectives generally do not make the signal correlation structure an explicit target or prior, forcing the model to implicitly approximate these dependencies from finite data without the guidance of physical regularization. This may limit models' deep understanding of signal characteristics.

While diffusion-based representation learning offers a principled mechanism to inject structured priors through the perturbation distribution Ho et al. (2020), existing diffusion-based signal pretraining methods such as TS-DDAE employ isotropic Gaussian noise, whose covariance does not encode

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.

relationships across sampling positions. To address this, we propose replacing white noise with Gaussian Process (GP) noise McHutchon & Rasmussen (2011) to introduce an explicit kernel-defined covariance, thereby transforming the denoising task from independent point-wise recovery to structured correlation recovery under correlated noise. Furthermore, we find that time-domain modeling is suboptimal for wireless signals. Applying temporal GP noise is difficult to capture the spectral locality of bandlimited signals due to complex oscillatory autocorrelations and asymptotic spectral independence. Therefore, in our work, we apply the discrete Fourier transform (DFT) to wireless signals and construct GP noise in the spectral domain to explicitly match the local smoothness of bandlimited spectra.

Based on these observations, we propose Spectral Gaussian Process Diffusion (SGP-Diff). It comprises a forward process driven by spectral GP noise and a corresponding reverse denoising objective. A linear second-order analysis characterizes how the optimal denoiser depends jointly on the data and structured-noise covariances. Experiments on four datasets covering three WSR tasks show that SGP-Diff improves the average accuracy over the strongest baseline by 1.83%.

The main contributions of this work are summarized as follows.

- We identify perturbation covariance as an underexplored design dimension in wireless signal diffusion pretraining and replace isotropic white noise with a kernel-defined spectral GP.
- We propose length-scale-conditioned denoising with a covariance-normalized objective and analyze its covariance-dependent behavior using a linear second-order model.
- We evaluate SGP-Diff on four datasets spanning three WSR tasks. The results characterize the accuracy, data efficiency, design choices, and computational cost of SGP-Diff.

2 Preliminaries and Related Work

2.1 Wireless Signals

In a typical communication system, a receiver samples a continuous complex baseband waveform $s(t)$ using an analog-to-digital converter, producing a discrete in-phase and quadrature (IQ) sequence $x[i] = s(iT_s)$ for $i = 0, 1, \ldots, L-1$, where T_s is the sampling period and L is the sequence length. The received sequence can be written as $x = I + jQ$, where $j = \sqrt{-1}$ and $I, Q \in \mathbb{R}^L$ denote its in-phase and quadrature components. In practice, we stack these components to obtain a two-channel real representation $\mathbf{x} \in \mathbb{R}^{2\times L}$. To expose its spectral-domain structure, we compute the discrete Fourier transform (DFT) to obtain the spectrum X,

$$X[n] = \sum_{i=0}^{L-1} x[i] \exp\left(-j\frac{2\pi ni}{L}\right), \qquad n = 0, 1, \ldots, L-1. \tag{1}$$

Physical front-end constraints and pulse-shaping filters at the transmitter generally render wireless signals bandlimited Aygur et al. (2025), meaning their spectral energy is confined within a finite bandwidth. Although individual spectra need not be perfectly smooth, they typically exhibit local smoothness within the occupied band and rapid decay outside it. Similarly to the temporal IQ sequence, we stack the real and imaginary parts of X to form a two-channel representation $\mathbf{X} \in \mathbb{R}^{2\times L}$.

Given an IQ sequence, the deep-learning-based WSR learns a mapping $f_\theta : \mathbb{R}^{2\times L} \to \{1, 2, \ldots, \mathcal{C}\}$, where $\mathcal{C}$ is the number of classes. Supervised WSR methods typically employ architectures such as convolutional neural networks and Transformers to extract discriminative features from raw IQ sequences Li et al. (2021); Vaswani et al. (2017), with representative models including AMC_Net Zhang et al. (2023), MCNet Huynh-The et al. (2020), and MSNet Zhang et al. (2021). More recently, self-supervised methods such as SpectrumFM Zhou et al. (2025) have learned transferable signal representations through masked reconstruction before adaptation to downstream tasks. Neural encoders can capture dependencies among IQ samples implicitly. However, existing signal pretraining objectives generally do not use a kernel-defined perturbation covariance to make these dependencies part of the pretext task.

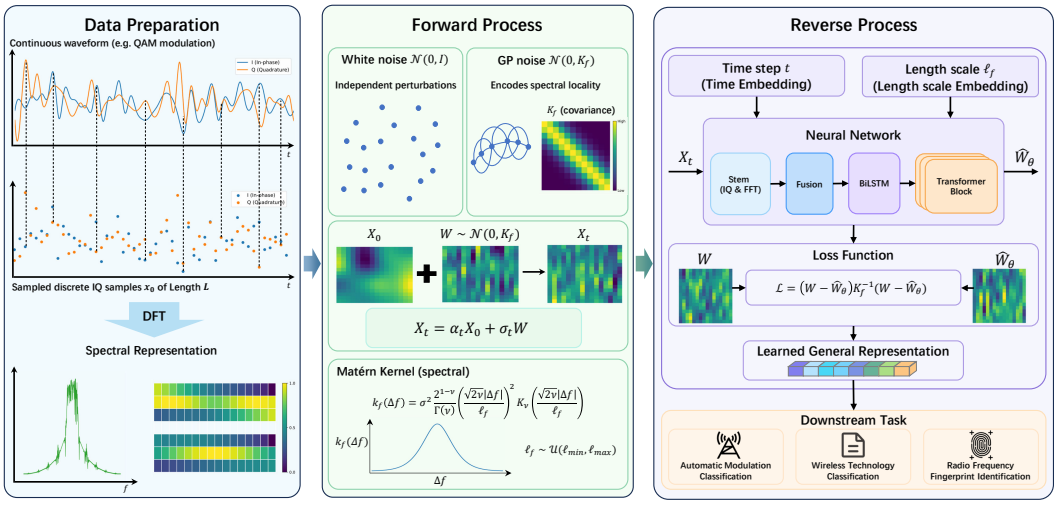


Figure 1: Overview of SGP-Diff. (1) An IQ sequence is transformed into the spectral domain. (2) GP noise with covariance K_f correlates perturbations across frequency bins. (3) A network conditioned on t and ℓ_f predicts the noise, and its intermediate features support downstream WSR.

2.2 Diffusion Models

Diffusion models define a forward process that progressively perturbs data and a learned reverse process that removes the perturbation. In a denoising diffusion probabilistic model (DDPM) Ho et al. (2020), a noisy observation at step t can be written as

$$z_t = \alpha_t z_0 + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \qquad \alpha_t^2 + \sigma_t^2 = 1, \tag{2}$$

where z_0 is a clean sample. A neural network ϵ_θ, often based on a U-Net Ronneberger et al. (2015), is trained to predict the injected noise using an objective of the form

$$\mathcal{L}_{\text{MSE}}(\theta) = \mathbb{E}_{z_0, \epsilon, t} \left[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2 \right]. \tag{3}$$

Diffusion models have subsequently been extended through alternative sampling procedures Song et al. (2020) and latent-space formulations Rombach et al. (2022). More directly related to our setting, Biloš et al. (2023) introduce GP noise for continuous temporal data, showing how a kernel-defined covariance can replace the standard isotropic prior.

Beyond generation, diffusion denoising can be used as a self-supervised representation-learning objective. DDAE Xiang et al. (2023), for example, extracts intermediate denoising features for image recognition. Diffusion models have also been introduced to wireless signal applications. RF-Diffusion Chi et al. (2024) performs time-frequency diffusion for signal generation, TS-DDAE Liu et al. (2026b) applies diffusion-based pretraining to WSR, and PWC-Diff Liu et al. (2026a) incorporates channel-related priors for signal recognition. However, these methods mainly use pointwise independent Gaussian noise. SGP-Diff instead studies perturbation covariance as an explicit part of the representation-learning objective.

3 Method

3.1 Overview

SGP-Diff leverages perturbation covariance as an explicit pretraining objective. As illustrated in Figure 1, an IQ sequence is transformed into the spectral domain and perturbed with GP noise whose covariance couples neighboring frequency bins. The network receives synchronized temporal and spectral views and predicts the injected noise while conditioned on the diffusion step t and GP length scale ℓ_f. Intermediate denoising features are subsequently adapted to downstream WSR tasks. Sections 3.2, Section 3.3 give the formulation, and Appendix C.1 details the architecture.

3.2 SPECTRAL GP FORWARD PROCESS

Standard diffusion uses isotropic white noise, whose covariance does not contain cross-sample structure. A stationary temporal GP is also unsuitable for directly prescribing correlations between neighboring frequency bins. Let K_τ be its length-L temporal covariance and let F be the complex unitary DFT matrix. The corresponding spectral covariance is $\widetilde{K}_f = FK_\tau F^H$. For a stationary process with absolutely summable autocovariance, Appendix A establishes

$$(\widetilde{K}_f)_{km} \longrightarrow 0 \quad (k \neq m), \qquad (\widetilde{K}_f)_{kk} \longrightarrow S(\omega_k), \tag{4}$$

for distinct non-conjugate frequency bins as L grows. A temporal GP can control frequency-dependent marginal noise power through $S(\omega)$, but its limiting spectral covariance is diagonal. It cannot directly maintain the nonzero off-diagonal terms required to couple distinct bins. We therefore define K_f in the spectral domain, where the desired cross-bin covariance is present at finite sequence length.

Given a two-channel IQ representation $\mathbf{x}_0 \in \mathbb{R}^{2\times L}$, let $\mathbf{X}_0 \in \mathbb{R}^{2\times L}$ denote the stacked real and imaginary parts of its DFT. For each component $c \in \{\Re, \Im\}$, we draw a zero-mean spectral GP

$$W^{(c)} \sim \mathcal{N}(0,\, K_f)\,, \quad c \in \{\Re, \Im\}, \qquad (K_f)_{mn} = k_f\big(\tfrac{m-n}{L}\big)\,, \tag{5}$$

where $(m-n)/L$ is the normalized lag between the ordered frequency bins and k_f is a stationary kernel. The kernel specifies the covariance of the injected perturbation. The real and imaginary components are sampled independently but share the same K_f.

We generate GP samples through a Cholesky factorization. Let $K_f = L_f L_f^\top$ and draw $\epsilon^{(c)} \sim \mathcal{N}(0, I)$. Then we can obtain

$$W^{(c)} = L_f \epsilon^{(c)}. \tag{6}$$

Setting $K_f = I$ gives $L_f = I$ and recovers pointwise independent Gaussian noise. In SGP-Diff, the off-diagonal entries of K_f determine how the perturbations at different bins co-vary. The factorization belongs to the forward perturbation process because K_f depends on the sampled kernel parameters rather than on the denoising-network parameters.

We use the Matérn family because it provides an explicit smoothness parameter and is less restrictive than the infinitely differentiable radial basis function (RBF) kernel. Specifically,

$$k_f(\Delta f) = \sigma_{\mathrm{GP}}^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\,|\Delta f|}{\ell_f} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}\,|\Delta f|}{\ell_f} \right), \tag{7}$$

where σ_{GP} is the marginal standard deviation, $\ell_f > 0$ controls the correlation range, ν controls sample-path smoothness, and K_ν is the modified Bessel function of the second kind. We set $\nu = 5/2$, which gives finitely differentiable perturbations. Small ℓ_f values produce short-range correlations, while large values couple a wider spectral neighborhood. During training, we sample $\ell_f \sim \mathcal{U}(\ell_{\min}, \ell_{\max})$ for each example. This changes the off-diagonal structure and eigenspectrum of K_f. Providing ℓ_f as a condition identifies the current perturbation covariance and allows one encoder to solve a family of correlation-dependent denoising tasks.

At diffusion time $t \in [0, 1]$, the noisy spectrum is

$$\mathbf{X}_t = \alpha_t \mathbf{X}_0 + \sigma_t \mathbf{W}, \tag{8}$$

where the operation is applied to both components and $\alpha_t^2 + \sigma_t^2 = 1$. We use coefficients from a continuous-time variance-preserving schedule with $\beta(t) = \beta_{\min} + t(\beta_{\max} - \beta_{\min})$:

$$\alpha_t = \exp\left[-\frac{1}{2}\beta_{\min} t - \frac{1}{4}(\beta_{\max} - \beta_{\min}) t^2 \right], \qquad \sigma_t = \sqrt{1 - \alpha_t^2}. \tag{9}$$

Here σ_{GP} controls the marginal scale of $\mathbf{W}$, while σ_t controls the amount of perturbation introduced at diffusion time t. We then apply the inverse DFT $\mathbf{x}_t = \mathrm{IDFT}(\mathbf{X}_t)$ to obtain a synchronized temporal view of the same corrupted signal. The temporal and spectral inputs therefore contain one shared GP realization rather than two independently perturbed views. For each training sequence, t selects the corruption strength and ℓ_f selects the cross-bin correlation range. Supplying both values to the network specifies the forward distribution that produces the current input.

3.3 COVARIANCE-AWARE DENOISING

From both views and the two scalar conditions, the network predicts spectral noise,

$$\widehat{\mathbf{W}}_\theta = \varepsilon_\theta(\mathbf{x}_t, \mathbf{X}_t, t, \ell_f)\,. \tag{10}$$

Errors measured by an unweighted MSE can be dominated by high-variance directions of K_f. We therefore normalize the prediction error in whitened coordinates. Let $K_f = L_f L_f^\top$ be a Cholesky factorization and write $W^{(c)} = L_f \epsilon^{(c)}$, where $\epsilon^{(c)} \sim \mathcal{N}(0, I)$. Measuring the MSE between $L_f^{-1} W^{(c)}$ and $L_f^{-1} \widehat{W}_\theta^{(c)}$ gives

$$\mathcal{L}_{\text{diff}} = \mathbb{E}\left[\sum_{c \in \{\Re, \Im\}} \left(W^{(c)} - \widehat{W}_\theta^{(c)}\right)^\top K_f^{-1} \left(W^{(c)} - \widehat{W}_\theta^{(c)}\right) \right]. \tag{11}$$

When $K_f = I$, Equation 11 reduces to the standard MSE loss, confirming consistency with the traditional white-noise diffusion learning objective. To see how the weighting acts, let $K_f = U_f \Lambda_f U_f^\top$ be the eigenvalue decomposition and define the residual $r^{(c)} = W^{(c)} - \widehat{W}_\theta^{(c)}$. Then

$$\left(r^{(c)}\right)^\top K_f^{-1}\, r^{(c)} = \sum_k \frac{\left(\mathbf{u}_k^\top r^{(c)}\right)^2}{\lambda_{f,k}}, \tag{12}$$

where $\mathbf{u}_k$ is the k-th column of U_f and $\lambda_{f,k}$ is the corresponding eigenvalue. The loss therefore measures prediction error in standardized covariance directions. Each direction is scaled by the inverse of its perturbation variance, preventing high-variance directions from dominating the gradient update. This reweighting changes the relative contribution of finite-capacity prediction errors during training. Importantly, for a fixed positive-definite K_f, it does not alter the population-optimal predictor, which remains the conditional mean $\mathbb{E}[W \mid X_t]$.

We next characterize how the structured perturbation enters the denoising rule. Let $\mathbf{z}_0 \in \mathbb{R}^L$ denote one zero-mean spectral component with covariance Σ_0. Its perturbed version is

$$\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \mathbf{w}, \qquad \mathbf{w} \sim \mathcal{N}(0, K_f), \tag{13}$$

where $\mathbf{w}$ is independent of $\mathbf{z}_0$. The relevant second-order quantities are

$$\mathrm{Cov}(\mathbf{z}_t) = \alpha_t^2 \Sigma_0 + \sigma_t^2 K_f, \qquad \mathrm{Cov}(\mathbf{w}, \mathbf{z}_t) = \sigma_t K_f. \tag{14}$$

The covariance of the observation contains both the unknown signal structure and the prescribed perturbation structure. More importantly, the target-observation cross-covariance is proportional to K_f. The forward covariance therefore enters the statistical relation that the denoiser must learn, rather than affecting only the marginal distribution of its input.

Proposition 1 (best linear noise predictor). For fixed t and K_f, assume that x_0 has finite second moments and that both K_f and $C_t = \alpha_t^2 \Sigma_0 + \sigma_t^2 K_f$ are positive definite. Among predictors of the form $B_t x_t$, the minimizer B_t^* is identical for the ordinary noise-prediction MSE and the K_f^{-1}-weighted objective in Equation 11. It is

$$B_t^* = \sigma_t K_f \left(\alpha_t^2 \Sigma_0 + \sigma_t^2 K_f\right)^{-1}. \tag{15}$$

Proof. For a positive-definite weighting matrix Q, define

$$J(B) = \mathbb{E}\big[(\mathbf{w} - B\mathbf{z}_t)^\top Q (\mathbf{w} - B\mathbf{z}_t)\big]\,. \tag{16}$$

Setting the matrix derivative of $J(B)$ with respect to B to zero yields the normal equation

$$Q\left(B\,\mathrm{Cov}(\mathbf{z}_t) - \mathrm{Cov}(\mathbf{w}, \mathbf{z}_t)\right) = 0. \tag{17}$$

Since Q is invertible, the minimizer is independent of the fixed weighting. Independence of $\mathbf{z}_0$ and $\mathbf{w}$ gives $\mathrm{Cov}(\mathbf{w}, \mathbf{z}_t) = \sigma_t K_f$, while Equation 14 gives $\mathrm{Cov}(\mathbf{z}_t) = C_t$. Substituting these terms yields Equation 15. □

At the bin level,

$$\mathrm{Cov}(w_m, z_{t,n}) = \sigma_t (K_f)_{mn}. \tag{18}$$

Whenever $(K_f)_{mn} \neq 0$, the observation at bin n carries information about the target noise at bin m. The optimal mapping in Equation 15 also depends on Σ_0, so estimating the perturbation requires separating its prescribed covariance from the dependencies of the clean signal. In general, B_t^* is non-diagonal. A predictor that processes each bin independently cannot attain this optimum, while a network that aggregates neighboring and long-range spectral information can represent the required cross-bin mapping.

For interpretation, suppose that Σ_0 and K_f share eigenvectors. Let $\lambda_{0,k}$ and $\lambda_{f,k}$ be their eigenvalues along direction k. Equation 15 becomes

$$\widehat{w}_k = \frac{\sigma_t \lambda_{f,k}}{\alpha_t^2 \lambda_{0,k} + \sigma_t^2 \lambda_{f,k}} z_{t,k}. \tag{19}$$

The gain depends on the signal variance and the perturbation variance in that direction. White noise makes $\lambda_{f,k}$ constant across directions. Spectral GP noise makes it direction dependent, while sampling ℓ_f changes the directional profile across training examples.

The shared-eigenvector assumption is needed only for the scalar form in Equation 19. The matrix result applies to general Σ_0 and K_f. Each sampled ℓ_f defines a different covariance and consequently a different optimal matrix B_t^*. Length-scale conditioning identifies the corresponding denoising rule. The shared encoder is therefore optimized across a family of prediction problems in which cross-bin information is required by the target itself. This establishes the covariance-aware learning objective that motivates SGP-Diff, while Section 4.4 measures its effect on downstream accuracy.

During pretraining, all parameters are optimized using Equation 11. For downstream adaptation, we fix t to the value reported in Appendix D, extract intermediate denoising features, apply global average pooling, and train a task-specific classifier with cross-entropy loss. The same feature-extraction pipeline is used at inference. Further architectural and optimization details are provided in Appendices C.1 and C.2.

4 Experiments

We evaluate SGP-Diff on multiple datasets and WSR tasks. The experiments address four questions: **Q1**: How does SGP-Diff compare with supervised and self-supervised baselines (Section 4.2)? **Q2**: How data-efficient are its pretrained representations under limited-label fine-tuning (Section 4.3)? **Q3**: How does each design choice affect performance (Section 4.4)? **Q4**: What computational overhead is introduced by structured GP perturbations (Section 4.5)?

4.1 Experimental Settings

SGP-Diff is implemented in PyTorch Paszke et al. (2019), and Optuna Akiba et al. (2019) is used for hyperparameter search. The architecture and optimization settings are described in Appendix C, and the selected hyperparameters are listed in Appendix D.

Datasets. We use four datasets covering three tasks: RML2016.10A O'shea & West (2016) and HisarMod2019 Tekbıyık et al. (2020) for automatic modulation classification (AMC), TechRec Fontaine & Shahid (2023) for wireless technology classification (WTC), and LoRa-RFFI Shen et al. (2022) for radio-frequency fingerprint identification (RFFI). RML2016.10A, HisarMod2019, and TechRec contain multiple classes evaluated under several signal-to-noise ratio (SNR) conditions. LoRa-RFFI contains signals from 30 devices and does not provide SNR annotations. We apply the traditional 80/20 training-evaluation split following the setting of TS-DDAE for all of datasets. Dataset statistics and pre-processing details are provided in Appendix B.1.

Baselines. We compare SGP-Diff against 12 baselines, encompassing 10 supervised models, including AMC_Net Zhang et al. (2023), CGDNN Njoku et al. (2021), CNN2 O'Shea et al. (2018), GRU2 Hong et al. (2017), MCNet Huynh-The et al. (2020), MSNet Zhang et al. (2021), ResNet Liu et al. (2017), Transformer Vaswani et al. (2017), VGG O'Shea et al. (2018), and IQFormer Shao et al. (2024), as well as 2 self-supervised pre-training methods, SpectrumFM Zhou et al. (2025) and TS-DDAE Liu et al. (2026b). For a fair comparison, supervised models are trained end-to-end with

standard cross-entropy loss, while the self-supervised counterparts are first pretrained using their respective objectives and subsequently fine-tuned under the identical downstream protocol.

Table 1: The main classification results (%). For the three datasets with SNR annotations, "Average" denotes accuracy averaged across SNRs and "Best" denotes the highest accuracy over SNRs. LoRa-RFFI reports overall accuracy. Bold and underlined entries indicate the best and second-best results.

Model	RML2016.10A		HisarMod2019		TechRec		LoRa-RFFI
	Average	Best	Average	Best	Average	Best	Overall
AMC_Net	60.28	89.95	46.28	52.94	88.20	98.70	10.81
CGDNN	47.51	70.55	35.77	42.35	70.31	81.19	28.25
CNN2	52.34	76.50	74.45	98.24	56.76	64.26	18.00
GRU2	58.11	86.14	70.67	97.81	37.79	39.09	21.58
MCNet	54.11	85.82	74.62	98.88	73.45	90.65	25.15
MSNet	60.48	90.41	65.06	85.45	90.38	99.11	20.77
ResNet	47.66	78.91	53.15	72.72	72.59	89.21	8.38
Transformer	56.94	85.50	71.91	93.90	87.63	96.05	17.16
VGG	50.40	78.55	45.88	54.99	81.38	93.46	13.04
IQFormer	<u>63.76</u>	<u>93.45</u>	68.21	87.38	<u>90.98</u>	98.72	22.32
SpectrumFM	62.85	93.32	68.51	84.93	88.33	97.39	34.56
TS-DDAE	63.62	93.41	<u>75.10</u>	<u>99.24</u>	90.11	<u>99.53</u>	<u>41.93</u>
SGP-Diff	**63.88**	**93.72**	**78.45**	**99.99**	**91.89**	**99.72**	**44.86**

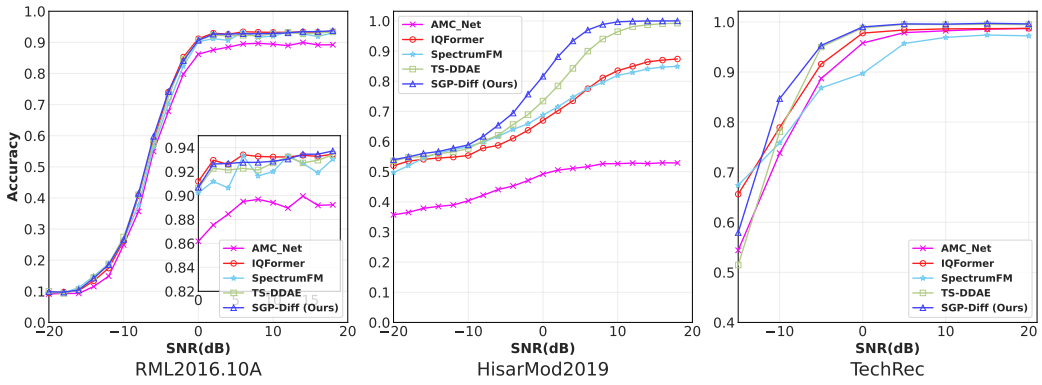


Figure 2: Visualization of performance under different SNR conditions (left: RML2016.10A, center: HisarMod2019, right: TechRec).

4.2 Main Results

The main results of SGP-Diff are presented in Table 1, with supplementary analyses in Appendix E.

Overall performance. Table 1 shows that SGP-Diff obtains the highest primary accuracy metric on all four datasets. Relative to the strongest baseline for each dataset, SGP-Diff achieves gains of 0.12%, 3.25%, 0.91%, and 2.93%, yielding an average improvement of 1.83%. Compared specifically to the self-supervised TS-DDAE baseline, the mean gain across the four datasets is 2.08%. The improvement is modest on RML2016.10A, where several methods are close, and larger on HisarMod2019 and LoRa-RFFI. This pattern supports the use of covariance-structured perturbations across datasets with different modulation, propagation, and device characteristics.

Performance across SNRs. Figure 2 compares SGP-Diff with four representative baselines, including AMC_Net, IQFormer, SpectrumFM, and TS-DDAE, on the three datasets with SNR annotations. At extremely low SNRs, all methods are limited because the received samples contain little observable signal information. In the low SNR range, particularly from approximately -10dB to 0dB, SGP-Diff generally maintains higher accuracy than the compared methods. At positive SNRs, its performance remains competitive and often approaches the dataset-specific upper range reported in Table 1. The trend complements the aggregate results by showing where the accuracy gains occur across channel conditions. Confusion matrices and feature visualizations are provided in Appendix E.

Behavior across tasks. The magnitude of the gain varies with the recognition target and dataset. RML2016.10A contains short synthetic sequences and several baselines already attain similar high-SNR accuracy, leaving limited room for improvement. HisarMod2019 contains more modulation classes and varied propagation effects, and shows the clearest AMC gain. TechRec has only three technology classes and strong baseline performance, so its improvement is comparatively small. LoRa-RFFI instead predicts the transmitter identity, and the gain on this dataset shows that the learned representation transfers beyond modulation and technology labels. The results therefore cover three distinct label semantics while showing that the magnitude of the benefit remains dataset dependent.

Table 2: Few-shot classification accuracy (%) of TS-DDAE and SGP-Diff.

Model	RML2016.10A		HisarMod2019		TechRec	
	Average	Best	Average	Best	Average	Best
TS-DDAE (1%)	37.83	59.23	20.04	24.76	79.86	92.43
SGP-Diff (1%)	42.17	62.95	34.62	41.85	84.40	95.38
TS-DDAE (5%)	47.06	72.31	37.56	45.75	87.23	97.50
SGP-Diff (5%)	51.70	76.27	48.01	55.16	88.11	97.58
TS-DDAE (10%)	50.69	75.72	34.82	39.13	87.71	98.72
SGP-Diff (10%)	53.28	79.00	49.30	55.49	89.73	98.84
TS-DDAE (20%)	53.98	80.91	56.04	63.98	89.00	98.82
SGP-Diff (20%)	58.62	88.64	58.59	72.68	90.12	99.03

4.3 Few-Shot Learning Capability

We compare SGP-Diff with TS-DDAE using 1%, 5%, 10%, and 20% of the labeled fine-tuning data, sampled uniformly across classes and SNR conditions. As shown in Table 2, SGP-Diff achieves higher point estimates for every dataset and labeling ratio. The gains are especially large on HisarMod2019 at the 1%, 5%, and 10% ratios, while the differences on TechRec become smaller as more labels are available. These results indicate that the representations learned with spectral GP perturbations transfer effectively when downstream supervision is scarce. A plausible explanation is that pretraining exposes the encoder to covariance-dependent denoising problems before task-specific labels are introduced, reducing the amount of labeled data needed to discover useful spectral dependencies during fine-tuning. Because the table reports point estimates, we interpret these results as evidence of improved data efficiency rather than as a claim of statistical significance.

Table 3: Ablation experimental results (%) regarding SGP-Diff.

Model	RML2016.10A		HisarMod2019		TechRec	
	Average	Best	Average	Best	Average	Best
SGP-Diff	63.88	93.72	78.45	99.99	91.89	99.72
w/o pretrain	62.64	92.05	69.78	89.22	91.73	99.59
w/ Gaussian noise	63.58	93.23	78.12	99.77	91.47	99.49
w/ temporal GP	62.84	93.09	78.37	99.96	91.35	99.41
w/ MSE loss	63.18	93.00	78.11	99.79	91.60	99.53
w/ RBF kernel	63.72	93.41	69.27	87.73	91.88	99.59

4.4 Ablation Study

We evaluate five variants on RML2016.10A, HisarMod2019, and TechRec: (1) training the downstream network without self-supervised pretraining (*w/o pretrain*); (2) replacing the spectral GP with isotropic Gaussian noise (*w/ Gaussian noise*); (3) applying GP noise in the temporal domain (*w/ temporal GP*); (4) replacing Equation 11 with an unweighted MSE (*w/ MSE loss*); and (5) replacing the Matérn kernel with an RBF kernel (*w/ RBF kernel*). The results are presented in Table 3.

Removing pretraining produces the largest decrease on HisarMod2019, while the smaller change on TechRec shows that the contribution of pretraining varies by dataset. Replacing the spectral GP with isotropic noise reduces average accuracy by 0.30%-0.42% across the three datasets, isolating the contribution of perturbation covariance under the shared architecture. The temporal-GP variant is also consistently lower, which supports constructing covariance directly in the domain where cross-bin relationships are required. The covariance-normalized objective improves average accuracy by 0.29%-0.70% relative to unweighted MSE, matching its role in balancing optimization across covariance directions. Finally, the Matérn and RBF kernels are close on RML2016.10A and TechRec but differ substantially on HisarMod2019. Matérn therefore provides the more consistent result across the evaluated benchmarks. Together, the variants attribute the final performance to self-supervised pretraining, spectral covariance, covariance normalization, and kernel choice.

4.5 Efficiency Analysis

Table 4: The efficiency comparison on TechRec.

Model	Parameters	Pretraining time/epoch	Evaluation time/epoch	Peak memory
SpectrumFM	7.98M	3,386s	105.3s	55,859MB
TS-DDAE	10.3M	712.8s	67.31s	33,461MB
SGP-Diff	8.53M	1,022s	69.78s	23,301MB

Table 4 compares the three self-supervised WSR methods on TechRec. The reported training and evaluation times are totals for one pass over the corresponding split. SGP-Diff is faster than SpectrumFM in both stages and uses less peak memory in the evaluated implementations. Compared with TS-DDAE, its pretraining time is higher because sampling correlated noise requires covariance factorization. Downstream adaptation and evaluation also construct GP-perturbed inputs according to the fixed feature-extraction protocol. However, the covariance matrix and its factor are used only in the forward input construction and are not differentiated during classifier optimization.

Direct Cholesky factorization of $K_f \in \mathbb{R}^{L\times L}$ requires $\mathcal{O}(L^3)$ time and $\mathcal{O}(L^2)$ storage. For $L = 1024$, one float32 covariance matrix and its factor occupy approximately 8 MB. Since K_f and L_f depend only on the sampled kernel parameters rather than on θ, they remain outside the gradient graph and introduce no backward pass through the factorization. The implementation performs the factorizations in parallel on the GPU. On TechRec, SGP-Diff requires 69.78 seconds per evaluation epoch, compared with 67.31 seconds for TS-DDAE. This small measured difference indicates that the forward-only covariance construction adds limited evaluation overhead at sequence length 1024, even though its Cholesky factorization contributes more noticeably to pretraining time. For substantially longer sequences, the stationary Toeplitz structure could support structured solvers or circulant approximations in place of direct factorization.

5 Conclusion

We introduce SGP-Diff, a self-supervised framework that replaces isotropic white noise with kernel-defined spectral GP perturbations. It couples neighboring frequency bins through a Matérn covariance, and trains the denoiser with a covariance-normalized objective. The resulting pretext task covers a controlled family of correlation ranges rather than one fixed perturbation pattern. A linear analysis clarifies the joint role of data and noise covariances. Across four datasets spanning typical WSR tasks, SGP-Diff obtains the best reported accuracy among the compared methods for each dataset. Other experimental results further support its practicality and the soundness of its architectural design.

AI USE STATEMENT

In this work, we used generative AI tools to assist with proof writing, translation, software-code editing, and language editing. We did not use generative AI tools for other tasks that require disclosure, and the remaining disclosure categories are not applicable to this work. We reviewed and corrected all AI-assisted code and text. We take responsibility for the final content, including all claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This research strictly adheres to data usage regulations. All experiments are based on public datasets, with a commitment not to process any private information. However, WSR can support beneficial applications (e.g., spectrum monitoring, anomaly detection) but may also enable surveillance or signal intelligence. Deployment should comply with local laws and be restricted in contexts that infringe on privacy or civil liberties. We encourage the academic community to work together to enhance the understanding and implementation of responsible AI.

REPRODUCIBILITY STATEMENT

Section 4.1 and Appendix D describe the experimental setup and hyperparameters. The supplementary materials include a demonstration of the code repository and the required external libraries. The source code and training scripts will be released upon acceptance and maintained thereafter.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Mohammed H Alsharif, Anabi Hilary Kelechi, Mahmoud A Albreem, Shehzad Ashraf Chaudhry, M Sultan Zia, and Sunghwan Kim. Sixth generation (6g) wireless networks: Vision, research activities, challenges and potential solutions. *Symmetry*, 12(4):676, 2020.

Michael Aygur, Sithamparanathan Kandeepan, Andrea Giorgetti, Akram Al-Hourani, Edward Arbon, and Mark Bowyer. Narrowband interference mitigation techniques: A survey. *IEEE Communications Surveys & Tutorials*, 27(6):3455–3482, 2025.

Marin Biloš, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, and Stephan Günnemann. Modeling temporal data as continuous functions with stochastic process diffusion. In *international conference on machine learning*, pp. 2452–2470. PMLR, 2023.

Naim Bitar, Siraj Muhammad, and Hazem H Refai. Wireless technology identification using deep convolutional neural networks. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6. IEEE, 2017.

Guoxuan Chi, Zheng Yang, Chenshu Wu, Jingao Xu, Yuchong Gao, Yunhao Liu, and Tony Xiao Han. Rf-diffusion: Radio signal generation via time-frequency diffusion. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pp. 77–92, 2024.

Jaron Fontaine and Adnan Shahid. Iq signals captured from lte, wifi, and dvb-t, 2023. URL `https://dx.doi.org/10.21227/5g7m-gf53`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Dehua Hong, Zilong Zhang, and Xiaodong Xu. Automatic modulation classification using recurrent neural networks. In *2017 3rd IEEE international conference on computer and communications (ICCC)*, pp. 695–700. IEEE, 2017.

Thien Huynh-The, Cam-Hao Hua, Quoc-Viet Pham, and Dong-Seong Kim. Mcnet: An efficient cnn architecture for robust automatic modulation classification. *IEEE Communications Letters*, 24(4): 811–815, 2020.

Merima Kulin, Tarik Kazaz, Ingrid Moerman, and Eli De Poorter. End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications. *IEEE access*, 6:18484–18501, 2018.

Xiaofan Li, Fangwei Dong, Sha Zhang, and Weibin Guo. A survey on deep learning techniques in wireless signal recognition. *Wireless Communications and Mobile Computing*, 2019(1):5629572, 2019.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019, 2021.

Xiaoyu Liu, Diyu Yang, and Aly El Gamal. Deep neural network architectures for modulation classification. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 915–919. IEEE, 2017.

Yaoqi Liu, Jin Wang, Chunchen Wang, Hui Wang, and Chuan Shi. From denoising to de-channeling: Integrating physical channel priors into diffusion models for radio signal understanding. In *Forty-third International Conference on Machine Learning*, 2026a.

Yaoqi Liu, Jin Wang, Hui Wang, and Chuan Shi. Ts-ddae: A novel temporal-spectral denoising diffusion autoencoder for wireless signal recognition model pre-training. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Andrew McHutchon and Carl Rasmussen. Gaussian process training with input noise. *Advances in neural information processing systems*, 24, 2011.

Fan Meng, Peng Chen, Lenan Wu, and Xianbin Wang. Automatic modulation classification: A deep learning enabled approach. *IEEE Transactions on Vehicular Technology*, 67(11):10760–10772, 2018.

Judith Nkechinyere Njoku, Manuel Eugenio Morocho-Cayamcela, and Wansu Lim. Cgdnet: Efficient hybrid deep learning model for robust automatic modulation recognition. *IEEE Networking Letters*, 3(2):47–51, 2021.

Colin O'Reilly, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar. Anomaly detection in wireless sensor networks in a non-stationary environment. *IEEE Communications Surveys & Tutorials*, 16(3):1413–1432, 2014.

Timothy J O'shea and Nathan West. Radio machine learning dataset generation with gnu radio. In *Proceedings of the GNU radio conference*, volume 1, 2016.

Timothy James O'Shea, Tamoghna Roy, and T Charles Clancy. Over-the-air deep learning based radio signal classification. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):168–179, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Mingyuan Shao, Dingzhao Li, Shaohua Hong, Jie Qi, and Haixin Sun. Iqformer: A novel transformer-based model with multi-modality fusion for automatic modulation recognition. *IEEE Transactions on Cognitive Communications and Networking*, 11(3):1623–1634, 2024.

Guanxiong Shen, Junqing Zhang, Alan Marshall, and Joseph R Cavallaro. Towards scalable and channel-robust radio frequency fingerprint identification for lora. *IEEE Transactions on Information Forensics and Security*, 17:774–787, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Kürşat Tekbıyık, Ali Rıza Ekti, Ali Görçin, Güneş Karabulut Kurt, and Cihat Keçeci. Robust and fast automatic modulation classification with cnn under multipath fading channels. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–6. IEEE, 2020.

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

Alessandro Vinciarelli, Maja Pantic, Hervé Bourlard, and Alex Pentland. Social signals, their function, and automatic analysis: a survey. In *Proceedings of the 10th international conference on Multimodal interfaces*, pp. 61–68, 2008.

Robert H Walden. Analog-to-digital converter survey and analysis. *IEEE Journal on selected areas in communications*, 17(4):539–550, 1999.

Jun Wang, Rong Li, Jian Wang, Qi-fan Zhang, Yi-qun Ge, and Wu-xian Shi. Artificial intelligence and wireless communications. *Frontiers of Information Technology & Electronic Engineering*, 21 (10):1413–1425, 2020.

Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15756–15766. IEEE, 2023.

Hao Zhang, Fuhui Zhou, Qihui Wu, Wei Wu, and Rose Qingyang Hu. A novel automatic modulation classification scheme based on multi-scale networks. *IEEE Transactions on Cognitive Communications and Networking*, 8(1):97–110, 2021.

Jiawei Zhang, Tiantian Wang, Zhixi Feng, and Shuyuan Yang. Amc-net: An effective network for automatic modulation classification. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Fuhui Zhou, Chunyu Liu, Hao Zhang, Wei Wu, Qihui Wu, Tony QS Quek, and Chan-Byoung Chae. Spectrumfm: A foundation model for intelligent spectrum management. *IEEE Journal on Selected Areas in Communications*, 2025.

A Frequency-Domain Covariance of Stationary Temporal GP Noise

This section makes precise the statement in Section 3.2 that the DFT approximately diagonalizes the covariance of a stationary temporal process. Let $\{w_t\}_{t\in\mathbb{Z}}$ be a zero-mean stationary Gaussian process with autocovariance $\gamma(\tau) = \mathrm{Cov}(w_{t+\tau}, w_t)$ satisfying $\sum_{\tau=-\infty}^{\infty} |\gamma(\tau)| < \infty$. For a length-$L$ segment $\mathbf{w} = [w_0, \ldots, w_{L-1}]^\top$, define the unitary DFT

$$\hat{w}_k = \frac{1}{\sqrt{L}} \sum_{t=0}^{L-1} w_t \, e^{-i2\pi kt/L}, \qquad \omega_k = \frac{2\pi k}{L}. \tag{20}$$

The covariance between two coefficients is

$$\mathrm{Cov}(\hat{w}_k, \overline{\hat{w}_m}) = \frac{1}{L} \sum_{t=0}^{L-1} \sum_{s=0}^{L-1} \gamma(t-s) e^{-i\omega_k t} e^{i\omega_m s}. \tag{21}$$

Setting $\tau = t - s$ rewrites Equation 21 as

$$\mathrm{Cov}(\hat{w}_k, \overline{\hat{w}_m}) = \frac{1}{L} \sum_{\tau=-(L-1)}^{L-1} \gamma(\tau) e^{-i\omega_k \tau} \sum_{s\in\mathcal{S}_\tau} e^{-i(\omega_k - \omega_m)s}, \tag{22}$$

where $\mathcal{S}_\tau$ contains the indices for which both s and $s+\tau$ lie in $\{0,\dots,L-1\}$ and hence has cardinality $L-|\tau|$.

Off-diagonal terms. Consider sequences of Fourier bins k_L and m_L whose frequencies approach distinct limits modulo 2π. The inner sum is a truncated geometric series. For each fixed τ, its value divided by L converges to zero. The contribution of large $|\tau|$ is controlled by the absolute summability of γ. Therefore,

$$\mathrm{Cov}(\hat{w}_{k_L}, \overline{\hat{w}_{m_L}}) \longrightarrow 0. \tag{23}$$

Thus, the spectral covariance becomes asymptotically diagonal for distinct frequency bins.

Diagonal terms. For $k=m$, the inner sum contains $L-|\tau|$ unit terms, giving

$$\mathrm{Var}(\hat{w}_k) = \sum_{\tau=-(L-1)}^{L-1} \left(1-\frac{|\tau|}{L}\right)\gamma(\tau)e^{-i\omega_k\tau}. \tag{24}$$

If $\omega_{k_L} \to \omega$, dominated convergence yields

$$\mathrm{Var}(\hat{w}_{k_L}) \longrightarrow S(\omega) = \sum_{\tau=-\infty}^{\infty} \gamma(\tau)e^{-i\omega\tau}, \tag{25}$$

the power spectral density of the temporal process. Therefore, a stationary temporal GP produces an approximately diagonal spectral covariance whose diagonal entries can vary with frequency. It should not be described as spectrally white unless $S(\omega)$ is constant.

Because a linear transform of a Gaussian vector is Gaussian, independence of complex coefficients follows when both their covariance and pseudo-covariance vanish. The same geometric-sum argument applies to $\mathbb{E}[\hat{w}_k\hat{w}_m]$ for distinct non-conjugate frequency bins. However, for a real-valued temporal GP, $\hat{w}_{L-k} = \overline{\hat{w}_k}$, conjugate pairs are deterministically related and must not be counted as independent. This qualification is relevant because the temporal-GP ablation applies real GP draws to the real-valued IQ channels.

Implication for SGP-Diff. A stationary temporal GP shapes the marginal variance of each frequency through its power spectral density, while its off-diagonal spectral covariance vanishes asymptotically. SGP-Diff instead specifies K_f in the spectral basis, which preserves the intended cross-bin covariance at finite L. This result provides the domain-selection argument used in Section 3.2, and the temporal-GP ablation in Section 4.4 evaluates its empirical consequence.

Table 5: Statistics of the four evaluation datasets. A dash indicates that SNR is unavailable.

Dataset	RML2016.10A	HisarMod2019	TechRec	LoRa-RFFI
Number of samples	220,000	780,000	202,762	600,000
Sequence length	128	1,024	1,024	1,024
Number of classes	11	26	3	30
Minimum SNR (dB)	-20	-20	-15	-
Maximum SNR (dB)	18	18	20	-
SNR interval (dB)	2	2	5	-
Channel	AWGN	AWGN + multipath fading	Real-world propagation	Real-world propagation

B Details of Datasets and Tasks

B.1 Datasets

Table 5 summarizes the four datasets. RML2016.10A and HisarMod2019 are used for AMC, and their modulation classes are listed in Table 6. RML2016.10A contains 11 modulation types at SNRs from -20dB to 18dB, while HisarMod2019 contains 26 types under more varied channel conditions. TechRec contains LTE, WiFi, and DVB-T recordings for WTC. Following the preprocessing protocol used by TS-DDAE, we divide the recordings into segments of length 1,024, retain the technology

label of each source recording, and add AWGN to construct the reported SNR conditions. LoRa-RFFI contains recordings of 30 commercial LoRa devices. We divide each raw data of length 8,192 into segments of length 1,024 and retain the predefined device labels. This dataset does not provide SNR annotations.

For all datasets, we use 80% of the samples for training and 20% for evaluation, consistent with Section 4.1. The datasets span synthetic AWGN, simulated multipath, and real-world propagation conditions. The Matérn kernel specifies the covariance of the injected perturbation and is not estimated from class labels or channel statistics. The clean examples retain their original modulation, propagation, and hardware characteristics.

B.2 Tasks

We evaluate three WSR tasks with different label semantics: modulation type, wireless technology, and transmitter identity. The detailed descriptions are introduced below.

Automatic Modulation Classification (AMC). A transmitter maps information to a waveform using a modulation scheme. AMC infers that scheme from a passively received signal without transmitter-side metadata.

Wireless Technology Classification (WTC). Communication technologies differ in protocol-dependent framing and waveform structure. WTC identifies the technology associated with a received signal, and TechRec uses LTE, WiFi, and DVB-T as its three classes.

Radio Frequency Fingerprint Identification (RFFI). Manufacturing tolerances and analog front-end imperfections introduce device-dependent characteristics even among transmitters using the same protocol. RFFI predicts the physical transmitter identity from these characteristics.

Table 6: Modulation classes in RML2016.10A and HisarMod2019.

Dataset	Modulation type
RML2016.10A	8PSK, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, AM-DSB, AM-SSB, WBFM
HisarMod2019	BPSK, QPSK, 8PSK, 16PSK, 32PSK, 64PSK, 4QAM, 8QAM, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, 2FSK, 4FSK, 8FSK, 16FSK, 4PAM, 8PAM, 16PAM, AM-DSB, AM-DSB-SC, AM-USB, AM-LSB, FM, PM

C Implementation Details

This section describes the dual-domain denoising network and the optimization protocol used in the experiments.

C.1 Neural Network Architecture

The denoising network ε_θ follows the general signal-encoder design used by IQFormer and TS-DDAE, but is adapted to process synchronized temporal and spectral views and to condition explicitly on both diffusion time t and GP length scale ℓ_f.

Feature extraction and fusion. Given the temporal input $\mathbf{x}_t \in \mathbb{R}^{B\times 2\times L}$ and its spectral counterpart $\mathbf{X}_t \in \mathbb{R}^{B\times 2\times L}$, two separate stem networks extract local features:

$$\mathbf{h}_{\text{time}} = \text{Stem}_{\text{IQ}}(\mathbf{x}_t), \qquad \mathbf{h}_{\text{freq}} = \text{Stem}_{\text{FFT}}(\mathbf{X}_t), \tag{26}$$

where both stems use one-dimensional convolutions and batch normalization. The outputs are concatenated along the channel dimension and passed through a convolutional fusion module with GELU activations:

$$\mathbf{h}_{\text{fused}} = \text{Fusion}([\mathbf{h}_{\text{time}}; \mathbf{h}_{\text{freq}}])\,. \tag{27}$$

The fused features are then processed by a bidirectional long short-term memory (BiLSTM) layer, yielding $\mathbf{h}^{(0)} \in \mathbb{R}^{B\times L\times C}$ for the stacked encoder layers.

Condition embedding. Sinusoidal embeddings followed by multilayer perceptrons map the two scalar conditions to C-dimensional vectors:

$$\mathbf{c}_t = \mathrm{MLP}_t(\mathrm{Sinusoidal}(t)), \qquad \mathbf{c}_\ell = \mathrm{MLP}_\ell(\mathrm{Sinusoidal}(\ell_f)). \tag{28}$$

The joint condition is $\mathbf{c} = \mathbf{c}_t + \mathbf{c}_\ell$ and is broadcast across sequence positions when injected into the feature tensor. Providing ℓ_f is necessary because the forward process randomizes the perturbation correlation range for each training example.

Main denoising network. The core contains N stacked encoder layers. For layer i, the forward computation is

$$\begin{aligned}
\widetilde{\mathbf{h}} &= \mathbf{h}^{(i-1)} + \mathrm{Linear}(\mathbf{c}),\\
\mathbf{h}_1 &= \widetilde{\mathbf{h}} + \mathrm{FFN}_1(\widetilde{\mathbf{h}}),\\
\mathbf{h}_2 &= \mathrm{Norm}(\mathbf{h}_1) + \mathrm{Attention}(\mathrm{Norm}(\mathbf{h}_1)),\\
\mathbf{h}_3 &= \mathrm{Norm}(\mathbf{h}_2) + \mathrm{DepthwiseConv}(\mathrm{Norm}(\mathbf{h}_2), \mathbf{c}),\\
\mathbf{h}^{(i)} &= \mathrm{Norm}(\mathbf{h}_3) + \mathrm{FFN}_2(\mathrm{Norm}(\mathbf{h}_3)),
\end{aligned} \tag{29}$$

where each FFN uses GELU activations.

The attention module uses multi-head self-attention with rotary position embeddings (RoPE). For head j,

$$\mathrm{head}_j = \mathrm{SDPA}(\mathrm{RoPE}(\mathbf{Q}_j), \mathrm{RoPE}(\mathbf{K}_j), \mathbf{V}_j)\,, \tag{30}$$

where SDPA denotes scaled dot-product attention.

The depthwise-convolution module captures local structure while injecting the shared condition. A pointwise convolution first generates content and gate tensors. The condition is then added before a depthwise convolution and a second gate:

$$\begin{aligned}
\mathbf{z}_{\mathrm{out}}, \mathbf{z}_{\mathrm{gate}} &= \mathrm{Chunk}(\mathrm{Conv}_{\mathrm{pw1}}(\mathbf{x}))\,,\\
\mathbf{z}_{\mathrm{gated}} &= \mathbf{z}_{\mathrm{out}} \odot \mathrm{sigmoid}(\mathbf{z}_{\mathrm{gate}}) + \mathbf{c},\\
\mathbf{z}_{\mathrm{dw}} &= \mathrm{BN}(\mathrm{Conv}_{\mathrm{dw}}(\mathbf{z}_{\mathrm{gated}}))\,,\\
\mathbf{y} &= \mathrm{Conv}_{\mathrm{pw2}}(\mathbf{z}_{\mathrm{dw}} \odot \mathrm{sigmoid}(\mathbf{z}_{\mathrm{dw}}))\,.
\end{aligned} \tag{31}$$

During pretraining, a final normalization and linear projection map the encoder features to the two-channel estimate $\widehat{\mathbf{W}}_\theta$ optimized by Equation 11. During downstream adaptation, the selected encoder feature tensor is globally averaged and passed to a task-specific classification head. The same temporal-spectral fusion and condition-embedding modules are retained in both stages.

C.2 Training Details

Unless otherwise stated, modules use their default PyTorch initialization. During pretraining, each example independently samples $t \sim \mathcal{U}(0, 1)$ and $\ell_f \sim \mathcal{U}(\ell_{\min}, \ell_{\max})$, then constructs $\mathbf{X}_t$ and $\mathbf{x}_t$ from the same GP realization. The network is optimized using Equation 11. We use AdamW with learning rate 10^{-4} and weight decay 0.01. A ReduceLROnPlateau scheduler monitors the pretraining loss with reduction factor 0.9, patience 25, and minimum learning rate 10^{-12}. Early stopping uses patience 1,000.

Fine-tuning also uses AdamW with weight decay 0.01, but follows a different learning-rate schedule. The learning rate begins at 10^{-6} with a two-epoch linear warmup and then follows cosine annealing with $T_{\max} = 15$ and minimum learning rate 10^{-8}. During fine-tuning and evaluation, the diffusion time is fixed to t_{ft} in Table 7, while the temporal and spectral inputs are constructed with the same GP perturbation pipeline used by the encoder. The covariance matrix and its Cholesky factor are excluded from the gradient graph. The classifier is optimized with cross-entropy, and model selection uses validation loss. Pretraining and fine-tuning are performed on an NVIDIA A100 GPU.

D HYPERPARAMETERS

Table 7 lists the default SGP-Diff configuration. These values are shared across datasets unless a sensitivity experiment or ablation explicitly varies the corresponding component. In particular, σ_{GP} is the marginal standard deviation in Equation 7, whereas σ_t is determined by the diffusion schedule and varies with t.

Table 7: Default hyperparameters used for SGP-Diff.

Parameter	Description	Value
$\ell_{\min}$	Minimum sampled spectral length scale	0.026
$\ell_{\max}$	Maximum sampled spectral length scale	0.216
σ_{GP}	GP marginal standard deviation	1.74
ν	Matérn smoothness parameter	$5/2$
t_{ft}	Diffusion time used for downstream feature extraction	2.79×10^{-3}
$\beta_{\min}$	Initial endpoint of the linear VP schedule	2.5×10^{-3}
$\beta_{\max}$	Final endpoint of the linear VP schedule	0.286
N	Number of encoder layers	4
C	Encoder hidden dimension	256
H	Number of attention heads	16

E ADDITIONAL EXPERIMENTS

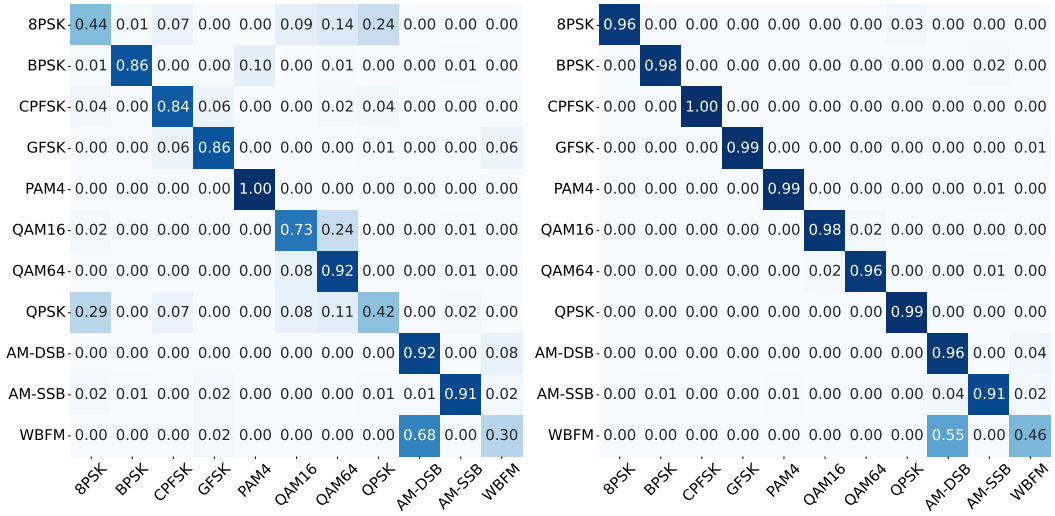


Figure 3: Normalized confusion matrices on RML2016.10A (left: -4dB; right: 2dB).

E.1 SUPPLEMENTARY VISUALIZATIONS

Figure 3 shows normalized confusion matrices at -4dB and 2dB. At -4dB, PAM4, GFSK, and CPFSK obtain class-wise accuracies of 100%, 86%, and 84%, respectively. The most visible confusions occur between related phase or amplitude modulations, including QPSK versus 8PSK and QAM16 versus QAM64, and between WBFM and AM-DSB. At 2dB, almost all digital modulation classes exceed 96% accuracy. These class-level results complement the aggregate SNR trends in Figure 2.

Figure 4 presents t-SNE projections of the learned features at -20dB, -6dB, and 12dB. The classes overlap heavily at -20dB, begin to form distinguishable groups at -6dB, and show stronger within-class concentration and between-class separation at 12dB. The projections provide a qualitative view

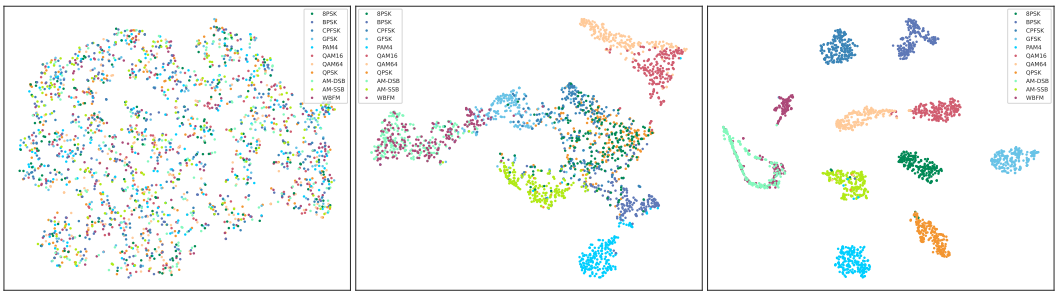


Figure 4: t-SNE projections of SGP-Diff features on RML2016.10A (left: -20dB; center: -6dB; right: 12dB).

of how the representation geometry changes with SNR, complementing the quantitative classification results.

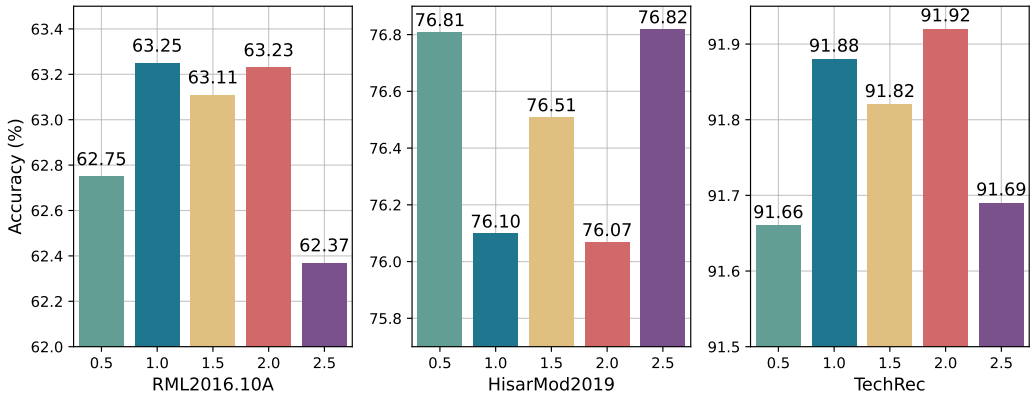


Figure 5: Accuracy for GP marginal standard deviation $\sigma_{\mathrm{GP}} \in [0.5, 2.5]$ (left to right: RML2016.10A, HisarMod2019, and TechRec).

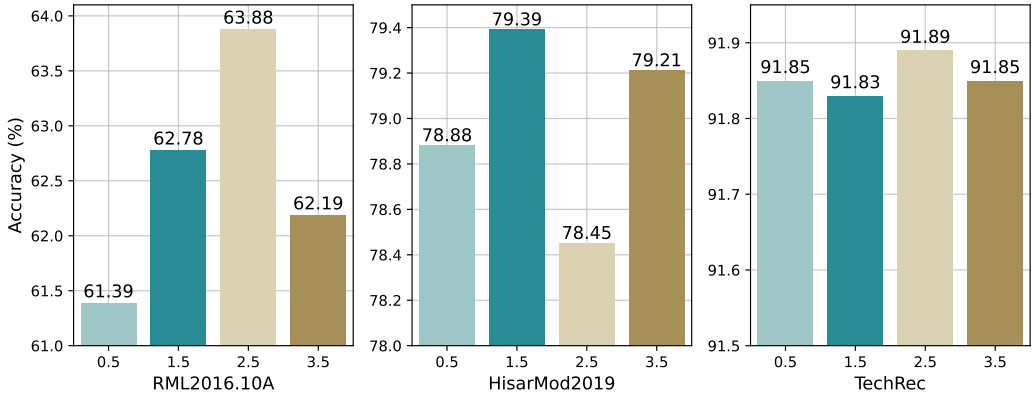


Figure 6: Accuracy for kernel smoothness $\nu \in \{0.5, 1.5, 2.5, 3.5\}$ (left to right: RML2016.10A, HisarMod2019, and TechRec).

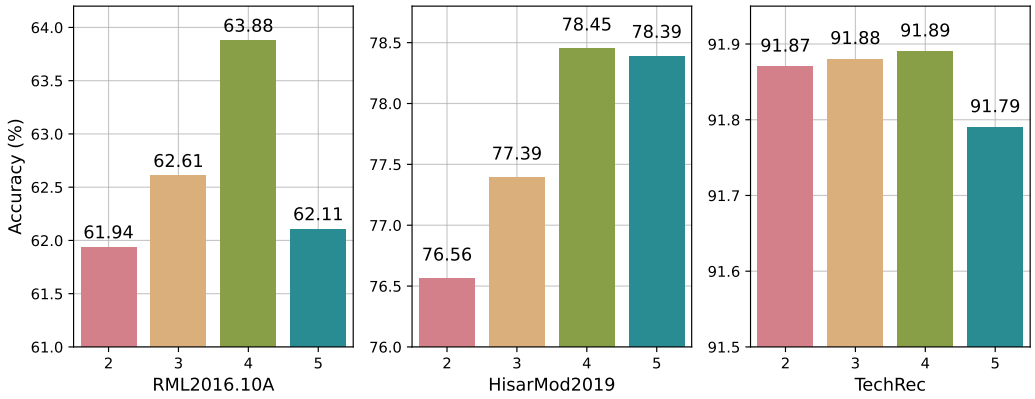


Figure 7: Accuracy for encoder depths from two to five layers (left to right: RML2016.10A, HisarMod2019, and TechRec).

E.2 Sensitivity to GP Marginal Scale

We vary the Matérn marginal standard deviation $\sigma_{\mathrm{GP}} \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$ while retaining the diffusion schedule. As defined in Equation 7, σ_{GP}^2 is the kernel variance, whereas σ_t is the time-dependent multiplier in the forward process. Figure 5 shows fluctuations below one percentage point over the tested values. The absolute accuracies come from independent runs and may differ slightly from Table 1. The experiment is intended to assess the relative sensitivity to this parameter. The default $\sigma_{\mathrm{GP}} = 1.74$ lies within the observed stable region.

E.3 Effect of Matérn Kernel Smoothness

We present the experiments with multiple kernel smoothness choices. The results can be seen in Figure 6. Notably, $\nu = 2.5$ achieves the optimal or near-optimal results on both RML2016.10A and TechRec. This empirical finding aligns perfectly with the physical reality that typical wireless signals, shaped by standard pulse-shaping filters, exhibit second-order smooth spectral envelopes. Interestingly, on the more challenging HisarMod2019 dataset, we observe slight, non-monotonic variations where $\nu = 1.5$ and $\nu = 3.5$ marginally outperform $\nu = 2.5$. This phenomenon suggests an intriguing hypothesis: the optimal spectral smoothness may intrinsically depend on the specific modulation composition and channel conditions. Building on this insight, an exciting direction for future research is to develop a data-driven adaptive mechanism that dynamically selects or mixes Matérn kernels with varying ν values during the diffusion process. Such an adaptive prior would allow the model to seamlessly match the heterogeneous physical characteristics of diverse and unseen wireless signals, further pushing the boundaries of physics-informed representation learning.

E.4 Effect of Encoder Depth

Figure 7 compares encoder depths from two to five layers. Accuracy increases from two to four layers on all three datasets and decreases slightly at five layers. We therefore use four layers as the default in Table 7. This depth provides the best empirical trade-off between model capacity and downstream accuracy in the evaluated setting.