
Appendices: Score-based Source Separation with Applications to Digital Communication Signals

Contents

A Broader Impact	2
A.1 Regarding the Proposed α -RGS Method	2
A.2 Regarding Applications to RF Systems	2
B Additional Characterizations of α-RGS	2
B.1 Multivariate Normal Sources	3
B.2 Tweedie's Formula and Digital RF Sources	4
B.3 Gaussian Mixture Source Model	5
C Datasets	8
C.1 Terminology	8
C.2 Synthetic and Over-the-air RF Datasets	9
D Diffusion Models for RF Signals	10
D.1 RRC-QPSK	11
D.2 OFDM (BPSK and QPSK)	11
D.3 CommSignal2	12
E Classical RF Interference Mitigation Techniques	13
E.1 Matched Filtering	13
E.2 LMMSE Estimation	14
F BASIS for RF Source Separation	15
G Complete Results	18

A Broader Impact

A.1 Regarding the Proposed α -RGS Method

Contribution and Distinction from Existing Works. Our method is in the effort of i) using score-based/diffusion-based methods beyond sampling, and to solve inverse problems, and ii) developing source separation strategies with a focus on sources with underlying discrete properties, which may make it differ from existing works where the usual goal is to obtain perceptually close solutions, which is not a good criterion in the RF domain.

Scalable Approach to Source Separation with Pre-Trained Models. An attractive property of our method is that by using pre-trained models as independent priors, we minimize the need for expensive compute resources to retrain models to tackle different scenarios/different mixture settings.

Trade-off in Computation Time. While the training time is reduced, the inference computation time is as of now longer than other methods (e.g., end-to-end training with mixtures as inputs and SOIs as desired outputs), thus making it unsuitable for real-time operation. As we discuss about its potential use for applications, speeding up the algorithmic aspects of the operation is essential in furthering this method towards practical applications.

Ethical Considerations of our Work. The intersection between generative AI and RF signals is a nascent field and hence, their ethical implications have been less explored. Nevertheless, we believe that this is an area that requires special attention to address issues such as malicious parties learning and spoofing, among others. While these are not the focus and the intent of our work, we believe that works in security in this space are actively exploring such problems.

Privacy Considerations with Generative Models. Where privacy and sensitive information is concerned, our work solely focuses on synthetic and public datasets. No sensitive transmissions have been captured, and therefore learned, by our generative models.

A.2 Regarding Applications to RF Systems

Novelty, Relevance and Timeliness. To the best of our knowledge, this is the first work that uses score-based methods for physical layer transmission in digital communication systems. The interference mitigation problem has significant relevance to enabling reliable operations in next-generation wireless systems. With the growth of heterogeneous wireless networks, there is overcrowding and growing scarcity of wireless spectrum. This work explores interference mitigation solutions with the goal of enabling efficient sharing of the finite resources to allow growth of the wireless ecosystem in a sustainable way.

Proof-of-Concept Exploration. Our work demonstrates via a proof-of-concept what might be possible for non-orthogonal access¹ to explore opportunities for new generation of transmission strategies and communication protocols; particularly, the emphasis on the low SIR regime is in line with motivations in lower-power device operations.

Considerations arising from Demonstration via Computational Simulations. All our experiments are done via computational simulation and thus, we did not operate in RF bands that we do not have rights to access. Our synthetic simulations may differ from practice where unmodeled physical phenomenon may occur, thus necessitating natural extensions of our work to deal with such impediments.

B Additional Characterizations of α -RGS

We now “dissect” (14) in §4, and focus on a single term,

$$\mathcal{L}_s(\boldsymbol{\theta}) := -\mathbb{E}_{\mathbf{t}, z_s} [\log p_{\tilde{s}_t}(\tilde{s}_t(\boldsymbol{\theta}))], \quad (\text{B.1})$$

¹That is, neighboring communicating devices that operate of the same, shared physical resources (e.g., time and frequency).

where we recall the definition of the Gaussian smoothing model from §3.2,

$$\tilde{\mathbf{s}}_t(\boldsymbol{\theta}) = \sqrt{\alpha_t} \bar{\boldsymbol{\theta}} + \sqrt{1 - \alpha_t} \mathbf{z}_s, \quad \mathbf{z}_s \sim \mathcal{N}(0, \mathbf{I}_d). \quad (\text{B.2})$$

As detailed in §3.2.1, when estimating a discrete source \mathbf{s} , $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta} + \varepsilon_s$ for $\varepsilon \sim \mathcal{N}(0, \sigma_s^2 \mathbf{I})$, $\sigma_s \rightarrow 0$, is a continuous surrogate of the discrete estimate $\boldsymbol{\theta}$, useful for optimization via gradient descent. The above loss only depends on the prior of the source, and is in particular independent of the data. Therefore, this term serves (and can be viewed) as a regularizer in our inference optimization problem. Although analysis of (B.1) is (strictly speaking) not useful in order to make statements about (14), it is nevertheless informative and insightful to show that the local extrema of (B.1) approach the modes of the underlying source distribution P_s , by solving for the stationary points,

$$\mathbb{E}_{t, \mathbf{z}_s} [\nabla_{\boldsymbol{\theta}} \log p_{\tilde{\mathbf{s}}_t}(\gamma_t \boldsymbol{\theta} + \sigma_t \mathbf{z}_s)] = 0 \quad (\text{B.3})$$

where we have used Leibniz rule for differentiation under the integral and where we have defined the shorthand notation,

$$\gamma_t := \sqrt{\alpha_t} \quad \text{and} \quad \sigma_t := \sqrt{1 - \alpha_t}. \quad (\text{B.4})$$

Through our examples we will demonstrate the mode-approaching nature of the solutions to (B.3) thus establishing another interpretation of the asymptotic behavior of our method—two loss terms whose stationary points are modes of the corresponding source distributions which work together in unison to satisfy the constraints imposed by the observed mixture. We thereby end up, with high probability, in the correct mass-points of the underlying distributions.

B.1 Multivariate Normal Sources

We first start with the analysis of a multivariate normal source and show that the the *exact* mode is obtained as the local extremum of (B.1).

Proposition 1. *Let $\mathbf{s} \in \mathbb{R}^d$ be a multivariate normal source with mean $\boldsymbol{\mu}_s$ and covariance matrix $\boldsymbol{\Sigma}_s$. For the Gaussian smoothing model in (B.2), the score at timestep t is*

$$\nabla_{(\tilde{\mathbf{s}}_t(\mathbf{s}))} \log p_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{s}}_t(\mathbf{s})) = -\boldsymbol{\Sigma}_{\tilde{\mathbf{s}}_t}^{-1} (\gamma_t \mathbf{s} + \sigma_t \mathbf{z}_t - \boldsymbol{\mu}_{\tilde{\mathbf{s}}_t}), \quad (\text{B.5})$$

$$\boldsymbol{\mu}_{\tilde{\mathbf{s}}_t} := \gamma_t \boldsymbol{\mu}_s \quad \text{and} \quad \boldsymbol{\Sigma}_{\tilde{\mathbf{s}}_t} := \gamma_t^2 \boldsymbol{\Sigma}_s + \sigma_t^2 \mathbf{I}.$$

Then, the minimizer $\boldsymbol{\theta}^ = \arg \min_{\boldsymbol{\theta}} \mathcal{L}_s(\boldsymbol{\theta})$ is equal to $\boldsymbol{\mu}_s$, i.e., the mode of the source distribution.*

Proof. Since \mathbf{s} is a continuous source, we perform optimization over a continuous space with respect to $\boldsymbol{\theta} \in \mathbb{R}^d$.² Notice that,

$$\nabla_{\boldsymbol{\theta}} \log p_{\tilde{\mathbf{s}}_t}(\gamma_t \boldsymbol{\theta} + \sigma_t \mathbf{z}_s) = \gamma_t \nabla_{(\tilde{\mathbf{s}}_t(\boldsymbol{\theta}))} \log p_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{s}}_t(\boldsymbol{\theta})) \quad (\text{B.6a})$$

$$= \gamma_t \boldsymbol{\Sigma}_{\tilde{\mathbf{s}}_t}^{-1} (\gamma_t \boldsymbol{\theta} + \sigma_t \mathbf{z}_s - \boldsymbol{\mu}_{\tilde{\mathbf{s}}_t}) \quad (\text{B.6b})$$

Substituting (B.6b) in (B.3), we see that the minimizer must satisfy

$$\mathbb{E}_t [\boldsymbol{\Sigma}_{\tilde{\mathbf{s}}_t}^{-1} (\gamma_t \boldsymbol{\theta} - \boldsymbol{\mu}_{\tilde{\mathbf{s}}_t})] = 0, \quad (\text{B.7})$$

where we have used the fact that \mathbf{z}_s is a zero mean Gaussian realization. Thus, the minimizer is

$$\boldsymbol{\theta}^* = \left(\sum_{t=1}^T \gamma_t \boldsymbol{\Sigma}_{\tilde{\mathbf{s}}_t}^{-1} \right)^{-1} \left(\sum_{t=1}^T \boldsymbol{\Sigma}_{\tilde{\mathbf{s}}_t}^{-1} \boldsymbol{\mu}_{\tilde{\mathbf{s}}_t} \right), \quad (\text{B.8})$$

which can be simplified further as

$$\boldsymbol{\theta}^* = \left(\sum_{t=1}^T \gamma_t \boldsymbol{\Sigma}_{\tilde{\mathbf{s}}_t}^{-1} \right)^{-1} \left(\sum_{t=1}^T \gamma_t \boldsymbol{\Sigma}_{\tilde{\mathbf{s}}_t}^{-1} \right) \boldsymbol{\mu}_s = \boldsymbol{\mu}_s, \quad (\text{B.9})$$

where we have used (1) to substitute $\boldsymbol{\mu}_{\tilde{\mathbf{s}}_t} = \gamma_t \boldsymbol{\mu}_s$. \square

Thus, we have established that the local extremum of the (B.1) corresponds to the mode in the multivariate Gaussian case.

²In this continuous setting, a surrogate distribution is not needed and $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}$ in (B.2)

B.2 Tweedie's Formula and Digital RF Sources

We next analyze (B.1) in the context of RF source separation by studying sources defined by a digital constellation with symbols from the finite set \mathcal{A} . Before proceeding, we first state Tweedie's formula [1], a tool that will aid us in computing the scores of arbitrary distributions. For completeness of the exposition, we prove it for the case of scalar random variables below.

Lemma 1. (*Tweedie's formula*) *Let a be a random variable with an arbitrary distribution and let z_σ be a normal random variable with mean μ and variance σ^2 independent of a . The score of the random variable $x := a + z_\sigma$ is*

$$\nabla_x \log p_x(x) = \frac{\mu - x}{\sigma^2} + \frac{1}{\sigma^2} \mathbb{E}_{a|x} [a|x = a + z_\sigma]. \quad (\text{B.10})$$

Proof. Since x is a sum of two independent random variable, it follows a convolution distribution given by,

$$p_x(x) = (p_a * p_{z_\sigma})(x). \quad (\text{B.11})$$

Therefore,

$$\nabla_x \log(p_a * p_{z_\sigma})(x) = \frac{1}{(p_a * p_{z_\sigma})(x)} (p_a * \nabla_x p_{z_\sigma})(x) \quad (\text{B.12})$$

$$= \frac{1}{(p_a * p_{z_\sigma})(x)} \int p_a(t) \nabla_x p_{z_\sigma}(x - t) dt \quad (\text{B.13})$$

$$= \frac{1}{(p_a * p_{z_\sigma})(x)} \int p_a(t) p_{z_\sigma}(x - t) \frac{(\mu - x + t)}{\sigma^2} dt \quad (\text{B.14})$$

$$= \frac{\mu - x}{\sigma^2} + \frac{1}{\sigma^2} \int t p_{a|a+z_\sigma}(t|x) dt \quad (\text{B.15})$$

$$= \frac{\mu - x}{\sigma^2} + \frac{1}{\sigma^2} \mathbb{E}[a|x = a + z_\sigma]. \quad (\text{B.16})$$

where (B.14) follows from,

$$\nabla_x p_{z_\sigma}(x - t) = -p_{z_\sigma}(x - t) \frac{(\mu - x + t)}{\sigma^2}. \quad (\text{B.17})$$

□

Proposition 2. *Let s be a scalar random variable representing a symbol drawn uniformly from the finite constellation set $\mathcal{A} \subset \mathbb{C}$. For the Gaussian smoothing model in (B.2) where, at timestep t ,*

$$\tilde{s}_t(s) = \gamma_t s + \sigma_t z_t,$$

the score is

$$\nabla_{(\tilde{s}_t(s))} \log p_{\tilde{s}_t}(\tilde{s}_t(s)) = \frac{1}{\sigma_t^2} \left(-\tilde{s}_t(s) + \sum_{a \in \mathcal{A}} a \cdot \phi_t(a, \tilde{s}_t(s)) \right), \quad (\text{B.18})$$

where,

$$\phi_t(a, \tilde{s}_t(s)) := \frac{\exp\left\{-\frac{|\tilde{s}_t(s) - \gamma_t a|^2}{\sigma_t^2}\right\}}{\sum_{\bar{a} \in \mathcal{A}} \exp\left\{-\frac{|\tilde{s}_t(s) - \gamma_t \bar{a}|^2}{\sigma_t^2}\right\}}. \quad (\text{B.19})$$

Proof. We will use Lemma 1 to compute the score, for which the only quantity we need to compute is the conditional expectation. Let $s' := \gamma_t s$. Since s is discrete, $P_{s'}$ is also a uniform distribution over

the symbol set $\mathcal{A}_t := \{\gamma_t a \mid a \in \mathcal{A}\}$. By Bayes' rule,

$$\mathbb{P}[s' = \gamma_t a \mid s' + \sigma_t z_s = x] = \frac{\mathbb{P}[s' + \sigma_t z_s = x \mid s' = \gamma_t a] \mathbb{P}[s' = \gamma_t a]}{\sum_{\bar{a} \in \mathcal{A}} \mathbb{P}[s' + \sigma_t z_s = x \mid s' = \gamma_t \bar{a}] \mathbb{P}[s' = \gamma_t \bar{a}]} \quad (\text{B.20})$$

$$= \frac{\mathbb{P}\left[z_s = \frac{x - \gamma_t a}{\sigma_t}\right] \cdot \frac{1}{|\mathcal{A}|}}{\sum_{\bar{a} \in \mathcal{A}} \mathbb{P}\left[z_s = \frac{x - \gamma_t \bar{a}}{\sigma_t}\right] \cdot \frac{1}{|\mathcal{A}|}} \quad (\text{B.21})$$

$$= \frac{\exp\left\{-\frac{|x - \gamma_t a|^2}{\sigma_t^2}\right\}}{\sum_{\bar{a} \in \mathcal{A}} \exp\left\{-\frac{|x - \gamma_t \bar{a}|^2}{\sigma_t^2}\right\}} \quad (\text{B.22})$$

$$= \phi_t(a, x). \quad (\text{B.23})$$

Therefore, from Lemma 1,

$$\nabla_{(\tilde{s}_t(s))} \log p_{\tilde{s}_t}(\tilde{s}_t(s)) = \frac{1}{\sigma_t^2} \left(-\tilde{s}_t(s) + \sum_{a \in \mathcal{A}} a \cdot \phi_t(a, \tilde{s}_t(s)) \right)$$

□

As an illustrative example, consider a BPSK source with modes at -1 and $+1$. To solve (B.3) for θ , we require the score of the BPSK source at different non-zero levels of Gaussian smoothing. Note that since the source is discrete, the score is undefined if no smoothing is applied. Though this smoothened score can be computed using (B.18), the stationary points of (B.3) cannot be computed in closed-form. Hence, in order to study the behavior of the solutions, we simulate (B.1), as shown in Figure B.1. The colored curves plot the loss for a single noise level, i.e., when $T = 1$, while the black curve computes the loss across multiple levels of smoothing, by computing an average over the single noise level curves. It is evident from these curves, that while the minima might not lie at the modes -1 and $+1$ for the single noise level curves, by averaging across multiple noise levels, the solution(s) to (B.3) approach the mode(s) of the source distribution. As pointed out in §4, an important caveat is that this behavior is only observed for a good choice of the noise level range.³ Larger noise levels aid in moving between the modes and smaller noise levels help resolving the solution. Only using larger noise levels, for example, can smoothen out the landscape to the point that the modes are no longer discernible, thus resulting in erroneous estimates. We leverage the noise levels from pre-trained diffusion models that have learned the statistical structure sufficiently well, thus avoiding cumbersome tuning of the noise level range.

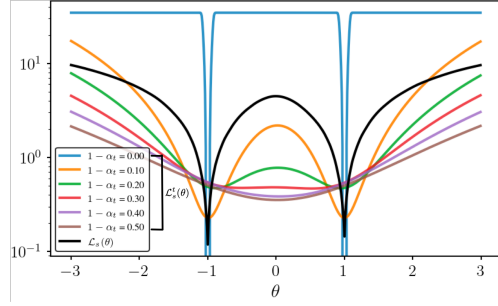


Figure B.1: Simulating (B.1) on a BPSK source. The loss at individual timestamps is visualized in addition to the total loss. The minima are at the modes of the BPSK source distribution, -1 and $+1$. Larger noise levels allow for exploration between modes and smaller noise levels sharpen the mode-seeking behavior.

B.3 Gaussian Mixture Source Model

Finally, we consider a scalar Gaussian mixture model (GMM) that can often model arbitrary complicated scalar distributions. We derive the score for a 2-component GMM and then generalize the result to an arbitrary K -component GMM.

Proposition 3. Consider a two-component scalar Gaussian mixture source,

$$p_s(s) = \lambda \mathcal{N}(s; \mu_1, \sigma_1^2) + (1 - \lambda) \mathcal{N}(s; \mu_2, \sigma_2^2), \quad \lambda \in (0, 1). \quad (\text{B.24})$$

³This only corresponds to choosing an appropriate lower bound on $1 - \alpha_T$ and should not be confused with tuning a noise schedule.

Under the Gaussian smoothing model in (B.2),

$$\begin{aligned} \nabla_{\tilde{s}_t(s)} \log p_{\tilde{s}_t}(\tilde{s}_t(s)) &= \frac{1}{Z(\tilde{s}_t(s))} \left[\lambda c_1(\tilde{s}_t(s)) \left(\frac{\gamma_t \mu_1 - \tilde{s}_t(s)}{\gamma_t^2 \sigma_1^2 + \sigma_t^2} \right) \right. \\ &\quad \left. + (1 - \lambda) c_2(\tilde{s}_t(s)) \left(\frac{\gamma_t \mu_2 - \tilde{s}_t(s)}{\gamma_t^2 \sigma_2^2 + \sigma_t^2} \right) \right], \end{aligned} \quad (\text{B.25})$$

where

$$c_1(\tilde{s}_t(s)) := \mathcal{N}(\tilde{s}_t(s); \gamma_t \mu_1, \gamma_t^2 \sigma_1^2 + \sigma_t^2), \quad (\text{B.26})$$

$$c_2(\tilde{s}_t(s)) := \mathcal{N}(\tilde{s}_t(s); \gamma_t \mu_2, \gamma_t^2 \sigma_2^2 + \sigma_t^2), \quad (\text{B.27})$$

$$Z(\tilde{s}_t(s)) := \lambda c_1(\tilde{s}_t(s)) + (1 - \lambda) c_2(\tilde{s}_t(s)). \quad (\text{B.28})$$

Proof. We need to compute the score of the following distribution,

$$p_{\tilde{s}_t}(\tilde{s}_t(s)) = p_{\gamma_t \mathbf{s} + \sigma_t \mathbf{z}}(\gamma_t \mathbf{s} + \sigma_t \mathbf{z}_s = \tilde{s}_t(s)), \quad (\text{B.29})$$

Our goal, to this end, is to leverage Lemma 1 to compute the score, thereby requiring an expression for the conditional expectation in (B.16). Since \mathbf{s} and \mathbf{z}_s are independent, (B.29) can be written as a convolution between two distributions,

$$p_{\tilde{s}_t}(\tilde{s}_t(s)) = (p_{\mathbf{s}'} * p_{\mathbf{z}_{\sigma_t}})(\mathbf{s}' + \mathbf{z}_{\sigma_t} = \tilde{s}_t(s)), \quad (\text{B.30})$$

where

$$\mathbf{s}' := \gamma_t \mathbf{s} \quad \text{and} \quad \mathbf{z}_{\sigma_t} := \sigma_t \mathbf{z}_s. \quad (\text{B.31})$$

In order to compute the conditional expectation, we first show that,

$$\begin{aligned} &p_{\mathbf{s}' + \mathbf{z}_{\sigma_t} | \mathbf{s}'}(\mathbf{s}' + \mathbf{z}_{\sigma_t} = \tilde{s}_t(s) | \mathbf{s}' = a') p_{\mathbf{s}'}(\mathbf{s}' = a') \\ &= \frac{1}{Z(\tilde{s}_t(s))} \left[\lambda c_1(\tilde{s}_t(s)) \mathcal{N} \left(a; \frac{\gamma_t^2 \sigma_1^2 \tilde{s}_t(s) + \gamma_t \sigma_t^2 \mu_1}{\gamma_t^2 \sigma_1^2 + \sigma_t^2}, \frac{\gamma_t^2 \sigma_1^2 \sigma_t^2}{\gamma_t^2 \sigma_1^2 + \sigma_t^2} \right) \right. \\ &\quad \left. + (1 - \lambda) c_2(\tilde{s}_t(s)) \mathcal{N} \left(a; \frac{\gamma_t^2 \sigma_2^2 \tilde{s}_t(s) + \gamma_t \sigma_t^2 \mu_2}{\gamma_t^2 \sigma_2^2 + \sigma_t^2}, \frac{\gamma_t^2 \sigma_2^2 \sigma_t^2}{\gamma_t^2 \sigma_2^2 + \sigma_t^2} \right) \right], \end{aligned} \quad (\text{B.32})$$

where

$$c_1(\tilde{s}_t(s)) := \mathcal{N}(\tilde{s}_t(s); \gamma_t \mu_1, \gamma_t^2 \sigma_1^2 + \sigma_t^2),$$

$$c_2(\tilde{s}_t(s)) := \mathcal{N}(\tilde{s}_t(s); \gamma_t \mu_2, \gamma_t^2 \sigma_2^2 + \sigma_t^2),$$

$$Z(\tilde{s}_t(s)) := \lambda c_1(\tilde{s}_t(s)) + (1 - \lambda) c_2(\tilde{s}_t(s)).$$

We start by deriving the density functions for the random variables in (B.31). Since the convolution between two Gaussians is a Gaussian,

$$p_{\mathbf{s}'}(a) = \lambda \mathcal{N}(a; \gamma_t \mu_1, \gamma_t^2 \sigma_1^2) + (1 - \lambda) \mathcal{N}(a; \gamma_t \mu_2, \gamma_t^2 \sigma_2^2), \quad (\text{B.33})$$

$$p_{\mathbf{z}_{\sigma_t}}(w) = \mathcal{N}(w; 0, \sigma_t^2), \quad (\text{B.34})$$

from which it follows, for example through the identities derived in [2, Sec 1], that

$$p_{\tilde{s}_t}(\tilde{s}_t(s)) = \lambda \mathcal{N}(\tilde{s}_t(s); \gamma_t \mu_1, \gamma_t^2 \sigma_1^2 + \sigma_t^2) + (1 - \lambda) \mathcal{N}(\tilde{s}_t(s); \gamma_t \mu_2, \gamma_t^2 \sigma_2^2 + \sigma_t^2). \quad (\text{B.35})$$

We conclude the derivation of the conditional distribution by using Bayes' rule,

$$p_{\mathbf{s}' | \mathbf{s}' + \mathbf{z}_{\sigma_t}}(\mathbf{s}' = a | \mathbf{s}' + \mathbf{z}_{\sigma_t} = \tilde{s}_t(s)) = \frac{p_{\mathbf{s}' + \mathbf{z}_{\sigma_t} | \mathbf{s}'}(\mathbf{s}' + \mathbf{z}_{\sigma_t} = \tilde{s}_t(s) | \mathbf{s}' = a) p_{\mathbf{s}'}(\mathbf{s}' = a)}{\int p_{\mathbf{s}' + \mathbf{z}_{\sigma_t} | \mathbf{s}'}(\mathbf{s}' + \mathbf{z}_{\sigma_t} = \tilde{s}_t(s) | \mathbf{s}' = a') p_{\mathbf{s}'}(\mathbf{s}' = a') da'}. \quad (\text{B.36})$$

whereby simplifying the terms in the numerator further, we have

$$p_{\mathbf{s}' + \mathbf{z}_{\sigma_t} | \mathbf{s}'}(\mathbf{s}' + \mathbf{z}_{\sigma_t} = \tilde{s}_t(s) | \mathbf{s}' = a) = p_{\mathbf{z}_{\sigma_t}}(\mathbf{z}_{\sigma_t} = \tilde{s}_t(s) - a) = \mathcal{N}(a; \tilde{s}_t(s), \sigma_t^2). \quad (\text{B.37})$$

Finally, by following the Gaussian product identities in [2, Sec 1], between the distribution in (B.37) and (B.33) we reach the result in (B.32).

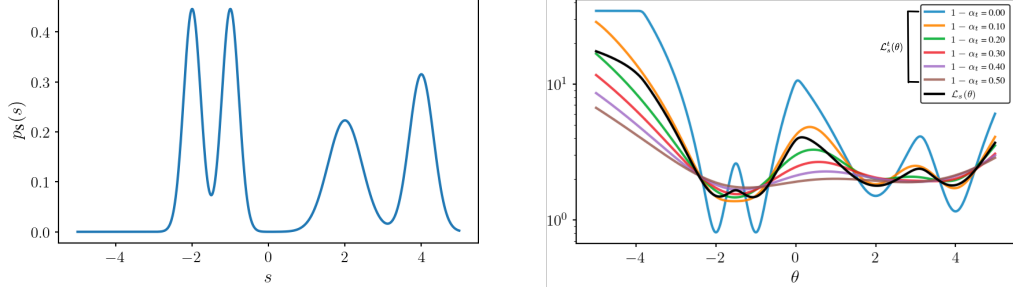


Figure B.2: **Left:** A GMM source with two equiprobable modes at -1 and -2 . Two smaller modes are present at $+2$ and $+4$. **Right:** The minima of (B.1) lie (approximately) at the modes of the source distribution (black curve). Colored curves correspond to (B.1) evaluated with $T = 1$.

Now we can compute the expectation as,

$$\mathbb{E}[s'|s' + z_{\sigma_t} = \tilde{s}_t(s)] = \frac{1}{Z(\tilde{s}_t(s))} \left[\lambda c_1(\tilde{s}_t(s)) \left(\frac{\gamma_t^2 \sigma_1^2 \tilde{s}_t(s) + \gamma_t \sigma_t^2 \mu_1}{\gamma_t^2 \sigma_1^2 + \sigma_t^2} \right) + (1 - \lambda) c_2(\tilde{s}_t(s)) \left(\frac{\gamma_t^2 \sigma_2^2 \tilde{s}_t(s) + \gamma_t \sigma_t^2 \mu_2}{\gamma_t^2 \sigma_2^2 + \sigma_t^2} \right) \right]. \quad (\text{B.38})$$

Using Lemma 1 we can express the score as,

$$\nabla_{\tilde{s}_t(s)} \log p_{\tilde{s}_t}(\tilde{s}_t(s)) = \frac{1}{Z(\tilde{s}_t(s))} \left[\lambda c_1(\tilde{s}_t(s)) \left(\frac{\gamma_t \mu_1 - \tilde{s}_t(s)}{\gamma_t^2 \sigma_1^2 + \sigma_t^2} \right) + (1 - \lambda) c_2(\tilde{s}_t(s)) \left(\frac{\gamma_t \mu_2 - \tilde{s}_t(s)}{\gamma_t^2 \sigma_2^2 + \sigma_t^2} \right) \right].$$

□

We can generalize the score to an arbitrary K -component GMM by exploiting symmetries in (B.25).

Proposition 4. Consider a K -component Gaussian mixture source,

$$p_S(s) = \sum_{i=1}^K \lambda_i \mathcal{N}(s; \mu_i, \sigma_i^2), \quad \sum_{i=1}^K \lambda_i = 1. \quad (\text{B.39})$$

Under the Gaussian smoothing model in §3.1 and using the shorthands in (B.4),

$$\nabla_{\tilde{s}_t(s)} \log p_{\tilde{s}_t}(\tilde{s}_t(s)) = \frac{1}{Z(\tilde{s}_t(s))} \sum_{i=1}^K \lambda_i c_i(\tilde{s}_t(s)) \left(\frac{\gamma_t \mu_i - \tilde{s}_t(s)}{\gamma_t^2 \sigma_i^2 + \sigma_t^2} \right), \quad (\text{B.40})$$

where

$$c_i(\tilde{s}_t(s)) := \mathcal{N}(\tilde{s}_t(s); \gamma_t \mu_i, \gamma_t^2 \sigma_i^2 + \sigma_t^2), \quad (\text{B.41})$$

$$Z(\tilde{s}_t(s)) := \sum_{i=1}^K \lambda_i c_i(\tilde{s}_t(s)). \quad (\text{B.42})$$

Similar to the BPSK case, since a closed-form expression for the solution to (B.3) cannot be obtained, we simulate (B.1) for a four-component GMM with two large equiprobable modes and two smaller modes. As shown in the colored curves on the right in Figure B.2, we again notice similar behavior where the modes are sharper at lower noise levels, with the sharpness decreasing at large noise levels. Hence, gradient-based methods can leverage the randomly chosen larger noise levels to move between modes and use the smaller noise levels to resolve the estimate. On average, the minima of the loss approach the modes of the original GMM source, as shown by the black curve on the right in Figure B.2.

C Datasets

In this section, we particularize the signal models introduced in §2.1 to RF signals. We first consider a single-carrier signal modulating, e.g., QPSK symbols, which can be expressed as,

$$s(t) = \sum_{p=-\infty}^{\infty} c_{p,\ell} g(t - pT_s). \quad (\text{C.1})$$

The second form corresponds to multi-carrier signals, such as OFDM signals, where multiple symbols are modulated in parallel,

$$b(t) = \sum_{p=-\infty}^{\infty} \sum_{\ell=0}^{L-1} c_{p,\ell} \exp \{j2\pi\ell t/L\}. \quad (\text{C.2})$$

Both types of signals can be grouped in a single family of signals represented as,

$$u(t) = \sum_{p=-\infty}^{\infty} \sum_{\ell=0}^{L-1} c_{p,\ell} g(t - pT_s, \ell) \exp \{j2\pi\ell t/L\}. \quad (\text{C.3})$$

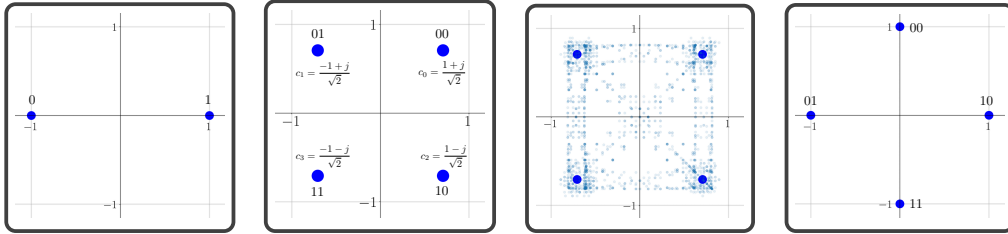
We use $s(t)$ to represent the signal of interest (SOI) and \mathbf{s} as the vector representation for a collection of N consecutive samples thereof. Similarly, the interference is represented by $b(t)$ or \mathbf{b} .

C.1 Terminology

Before we describe the construction of our datasets, we will provide some terminology that we use in our description. This terminology is meant to supplement the details already provided in §5.1 in the main paper.

Symbol constellation: A digital constellation describes the underlying scheme for mapping bits to symbols. Typically used digital constellations (the mapping between bits and symbols) include binary phase shift keying (BPSK) and quadrature phase shift keying (QPSK). Specific BPSK and QPSK examples are depicted in Figure C.1.

Pulse/Symbol shaping function: This corresponds to $g(\cdot, \cdot)$ in (C.3) and is meant to help limit the bandwidth of the signal. Each symbol $c_{p,\ell}$ is multiplied with a pulse shaping function with different time offsets and a main lobe width T_s , helping smoothen the signal and remove high frequencies. In this work, we will make use of the root-raised cosine (RRC) pulse shaping for our single-carrier QPSK SOI. On the other had, OFDM signals do not use an explicit pulse shaping function (as shown in (C.2)), i.e., $g(t, \cdot)$ corresponds to a rectangular function in (C.3).



(a) BPSK constellation. (b) QPSK constellation. (c) Effect of RRC filtering. (d) Effect of 45° phase.

Figure C.1: Discrete constellations for a BPSK and QPSK signal. Application of the RRC filter in the time domain leads to interpolation between constellation point as shown in (c). Phase offsets in the waveform domain can be corrected by looking at the rotated constellation as in (d).

Total number of subcarriers: This corresponds to L in (C.3) and specifically to OFDM signals of the form in (C.2). The subcarriers are complex exponentials of the form $\exp \{j2\pi\ell/L\}$, $\ell \in \{0, \dots, L-1\}$ with spacing $2\pi/L$ in the (angular) frequency domain, and are orthogonal to each

other. Note that the weighted summation with complex exponentials can be interpreted as computing the *inverse discrete Fourier transform* (IDFT) of the pulse-shaped symbols, for which the inverse fast Fourier transform (IFFT) is used during implementation. Thus, in an interference-free setting, decoding the symbols from $b(t)$ can be performed by computing the FFT of $b(t)$.

Starting point: When decoding an RF waveform to extract the underlying symbols, we need them to be *time-synchronized* to a specified frame of reference known as the starting point, τ . If synchronized correctly, this indicates the number of time samples to stream in before bit decoding can begin. If unsynchronized, this corresponds to a shift in time that results in “non-natural” looking decoded constellations at all other time shifts except for the correct time-shift τ . Here, decoding of $u(t)$ can be performed either via an FFT for OFDM, or sampling after matched filtering for a single-carrier signal.

C.2 Synthetic and Over-the-air RF Datasets

Below we describe the details relating to the RF signals used in our single-channel source separation (SCSS) experiments. To simulate representative types of RF signals, we generated single-carrier QPSK signals with RRC pulse shaping and OFDM signals using the NVIDIA Sionna library [3]. We also used over-the-air recordings corresponding to “CommSignal2” from the dataset in the RF challenge. We do not have explicit knowledge on CommSignal2 source model as it is not provided in the challenge. Hence, we are unable to explicitly describe the pulse shaping function of such a signal type—leaving learning of such properties to our data-driven diffusion models instead.

RRC-QPSK. We use QPSK signals with root raised cosine (RRC) pulse shaping [4, Sec 11.3] as the SOI across all our experiments. These signals are used in a variety of different protocols ranging from low-power radios to cellular signals and even for satellite communication. For our experiments, we generated a dataset of 100,000 RRC-QPSK waveforms with the following parameters:

- **Modulation scheme:** QPSK (2 bits per symbol)
- **Pulse shaping function:** RRC, roll-off factor 0.5, spanning ~ 8 symbols
- **Default starting point:** $\tau_s = 8$
- **QPSK-specific details:**
 - **Oversampling rate:** 16 [samples/symbol], using zero-padding between symbols.

OFDM (QPSK/BPSK). For the OFDM signals (specifically, cyclic prefix OFDM), we generated waveforms with the following parameters (so as to simulate an 802.11n WiFi waveform):

- **Modulation scheme:** Either QPSK (2 bits per symbol) or BPSK (1 bit per symbol)
- **Total number of subcarriers (FFT size):** 64
- **Symbol shaping:** None as per (C.2) or rectangular window as per (C.3)
- **Other notes:** Random time offset and random phase rotation for each time series realization
- **OFDM-specific details:**
 - **Number of non-zero subcarriers:** 56 (null at DC subcarrier ($\ell = 0$))
 - **Length of cyclic prefix:** 16 (25% of FFT size)
 - **Number of non-zero symbols:** 56 symbols per 80 samples (1792 symbols in a 2,560 window)
 - **Symbol length:** 80 (64 subcarriers + cyclic prefix of length 16)

In our experiments, we choose to use OFDM signals as one class of interference signals. This means that, given a new realization, we assume that the starting point of a new OFDM symbol (i.e., τ_b) is *unknown*. Further, we assume no access to knowledge about the parameters outlined above, such as the number of subcarriers, the FFT size and the length of cyclic prefix. Hence, we seek a data-driven approach where characteristics of this signal should be learned through data.

Table 1: Hyperparameters used for training our diffusion models.

	QPSK	OFDM (BPSK)	OFDM (QPSK)	CommSignal2
Number of residual blocks (R)	30	30	30	30
Dilation cycle length (m)	10	10	10	10
Residual channels (C)	64	128	256	128
Random shift + phase rotation	\times	\checkmark	\checkmark	\checkmark
Batch size	128	128	64	128
Learning rate	$5e-4$	$5e-4$	$5e-4$	$1e-4$
Early stopping iteration	360,000	220,000	340,000	90,000

CommSignal2. CommSignal2 is a dataset comprising over-the-air recordings of type of communication waveforms, provided in [5]. This dataset contains 150 frames each of length 43,560 samples. For purposes of this work, we use the first 100 frames in creating our training set for training the diffusion model, and we extract windows of relevant lengths (2560 samples) from it, to create a dataset of size $\sim 160,000$ samples. The last 50 frames are reserved for creating our test set. Similar to the OFDM case, we also perform random time offsets and phase rotations⁴ for each time series realization.

D Diffusion Models for RF Signals

We train diffusion models to learn the statistical structure inherent to digital RF signals. As detailed in Appendix C, these signals exhibit both continuous and discrete properties. To this end, we adopt an architecture based on an open-sourced implementation⁵ of the DiffWave [6] diffusion model, initially developed for speech synthesis.

Broadly speaking, as shown in [6, Figure 2, Appendix C], DiffWave employs R residual blocks with dilated convolutions, where the output of block $i - 1$ serves as an input to block i , $i \in \{0, \dots, R - 1\}$. The dilated convolutions assist in the learning of long range temporal and periodic structures. The dilations first start small and successively get larger, such that the dilation at block i is given by $2^{i \bmod m}$, where m is the dilation cycle length. For example, if the dilation periodicity is $m = 10$, then in block $i = 9$ the dilation is 512 and in block 10 the dilation is reset to 1. This allows the network to efficiently tradeoff between learning local and global temporal structures. All residual blocks use the same number of channels, C .

Modifications. Several changes have been made in order to facilitate training with RF signals. First, since we are dealing with complex-valued continuous waveforms, we train on two channel signals where the real and imaginary components of the RF signals are concatenated in the channel dimension. Second, while the open-sourced implementation uses an ℓ_1 loss for training, we train with an MSE (squared ℓ_2) loss to match the training objective in [7]. We monitor the validation MSE loss, and once the loss stops decreasing substantially, we choose to stop training early. Lastly, we had to increase the channel dimension C to learn more complicated RF signals, e.g., OFDM (QPSK). We train all our models on $2 \times$ NVIDIA 3090 GPUs. Additionally, during data loading, we perform random time shifts and phase rotations on the OFDM (BPSK), OFDM (QPSK) and CommSignal2 signals. Physically, these simulate transmission impairments in RF systems. The hyperparameters that we use for training each model are shown in Table 1.

Metrics. In the image domain, metrics such as the Fréchet Inception Distance (FID) [8] can be used to assess the quality of generative models. However, such perceptual metrics are less relevant to the digital communications domain. In the RF domain, the fidelity is measured by the ability to extract the underlying transmitted bits. Hence, in the context of our synthetic datasets, we probe generated samples and compare the estimated received symbols with the underlying constellation. However, for real world signals such as CommSignal2, for which we do not know the system parameters, we have no other means to assess the fidelity apart from looking at time-domain structure. This is another

⁴Namely, multiplication of the signal with a complex exponential with a random (imaginary) exponent.

⁵<https://github.com/lmnt-com/diffwave>

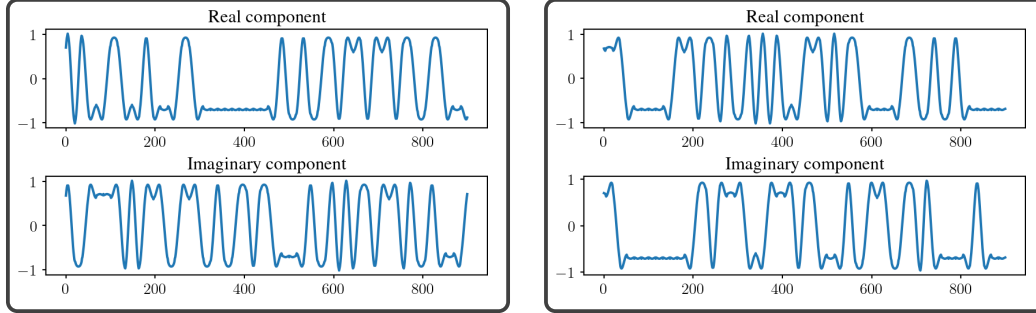


Figure D.1: **Left:** A ground truth RRC-QPSK time-domain waveform. **Right:** A sample generated by our trained RRC-QPSK diffusion model. Evidently, the generated waveform resembles the true one.

motivation for studying RF source separation, as it provides us with a framework to assess the quality of the learned statistical priors.

D.1 RRC-QPSK

We train a diffusion model on the RRC-QPSK dataset (see Appendix C.2). While a QPSK signal has four distinct constellation points, application of the RRC filter results in interpolation between points as shown in Figure C.1c. Figure D.1 shows an example of a ground truth RRC-QPSK waveform on the left. On the right we show a RRC-QPSK sample generated by our diffusion model. While it is hard to judge whether it has learned the underlying discrete structure, visually, the waveform seems to have characteristics similar to the ground truth.

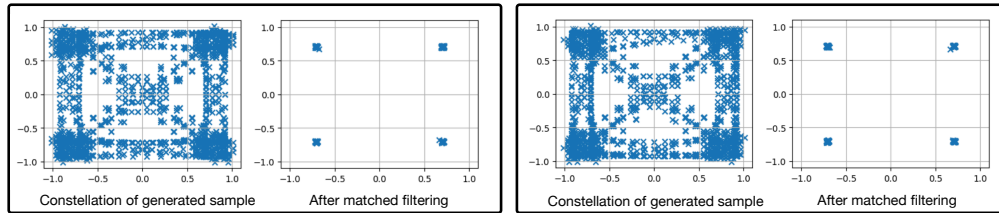


Figure D.2: Two samples generated from the diffusion model trained on RRC-QPSK samples. Within each box, the image on the left is underlying constellation of the realization generated by the diffusion model. Notice that the RRC filtering results in interpolation between the QPSK constellation points. After applying MF, the effects of pulse shaping are reversed and the original QPSK constellation is recovered.

In Figure D.2, we show that by probing generated samples for the underlying (interpolated) symbols, we do indeed recover the constellation for an RRC-QPSK signal. Note that we can do such probing since we know the parameters and signal generation model for an RRC-QPSK signal. Furthermore, by performing MF and removing the effects of pulse shaping we are able to recover the original QPSK constellation, thus demonstrating that the diffusion model has indeed successfully learned the discrete constellation along with the pulse shaping function from data samples.

D.2 OFDM (BPSK and QPSK)

We train two OFDM diffusion models, on the OFDM (BPSK) and the OFDM (QPSK) dataset respectively (see Appendix C.2). Figure D.3, shows an example of an OFDM (BPSK) signal from our dataset. On the left we plot the real and imaginary components of the waveform. These components visually look like Gaussian noise and it is not evident that there is actually inherent structure to these signals. As mentioned in Appendix C.2, the OFDM signals can be demodulated using oracle knowledge about the FFT size and the cyclic prefix. Additionally, since the signal undergoes a random time shift and phase rotation, the underlying (rotated) BPSK constellation will only be visible

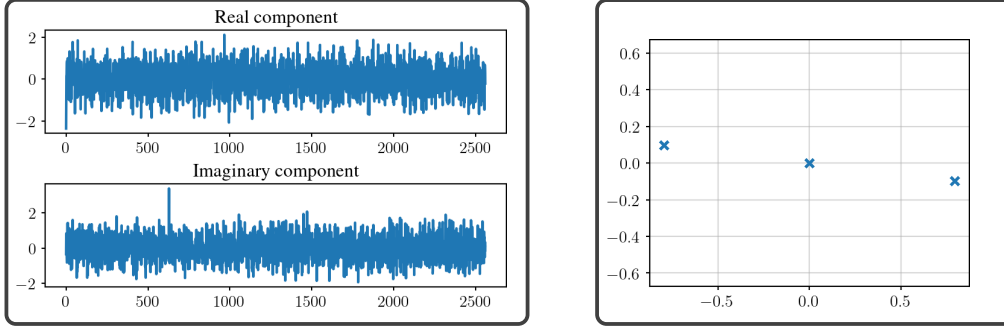


Figure D.3: **Left:** A complex-valued OFDM (BPSK) source augmented with a time shift of 8 samples and with a random phase rotation plotted in the time domain. In the time domain, the discrete structure is not discernible and the time-domain waveforms visually look like Gaussian noise. **Right:** Extracting the underlying (rotated) symbols from the waveform on the left by demodulating the OFDM signal using oracle knowledge about the FFT size, cyclic prefix, and time shift.

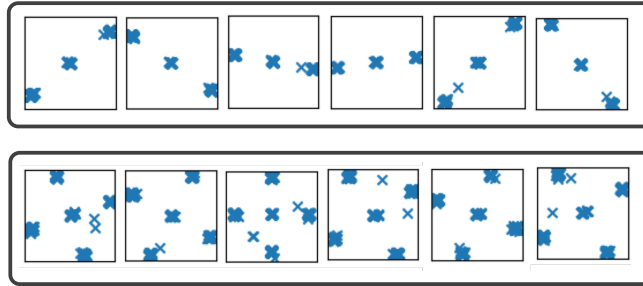


Figure D.4: **Top:** Probing six generated samples from the OFDM (BPSK) diffusion model recovers the underlying BPSK constellation after time synchronization. **Bottom:** Probing six generated samples from the OFDM (QPSK) diffusion model recovers the underlying BPSK constellation after time synchronization.

when the time shift has been compensated for, i.e., the signal has been time-synchronized, as shown on the right in Figure D.3.

The top row of Figure D.4 shows the recovered constellation for six generated OFDM (BPSK) symbols and the bottom row shows the same for six generated OFDM (QPSK) signals. Note that since there are some null symbols, i.e., unused subcarriers in the frequency domain, there is an additional “constellation point” at the origin. In general the recovered constellations are clean, except for a few symbols that lie slightly off the constellation. When converting back to bits the symbols are mapped to the nearest constellation point.

D.3 CommSignal2

The CommSignal2 dataset was recorded over-the-air and hence, unlike the synthetic datasets, we cannot probe it without knowledge of the true system parameters and signal generation model, which is not provided in [5]. These complications serve as one of our motivations to develop data-driven source separation models. In the real-world, CommSignal2 could interfere with an SOI such as a QPSK signal for which the signal generation model is known. A learned prior for CommSignal2 could help mitigate such interference.

As shown in Figure D.5, we observe that the generated CommSignal2 waveforms generally display similar global temporal structure to the ground truth waveforms. To further demonstrate that our diffusion model has truly learned the statistical structure of the signal, we carry out source separation experiments and demonstrate that leveraging this diffusion model outperforms conventional and learning-based source separators.

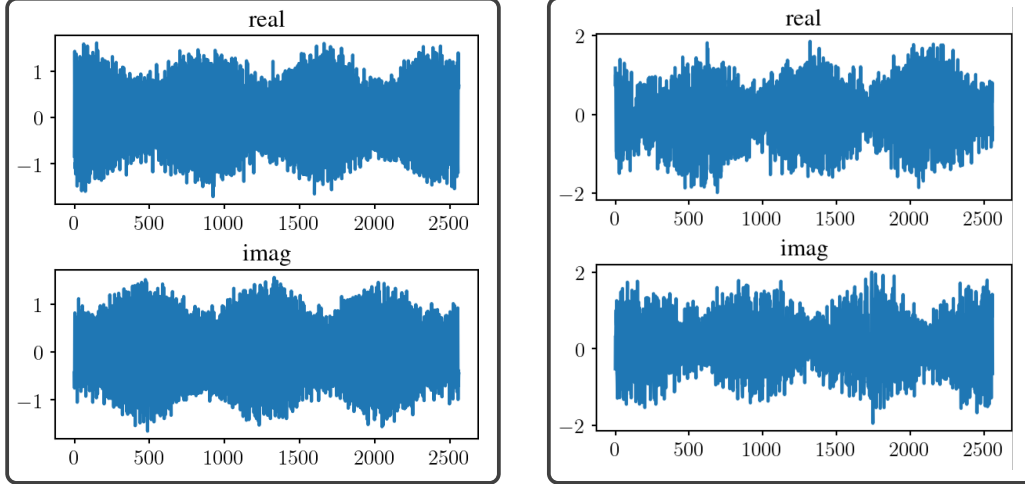


Figure D.5: **Left:** Ground truth CommSignal2 waveform from the dataset. **Right:** A generated CommSignal2 waveform from the learned diffusion model.

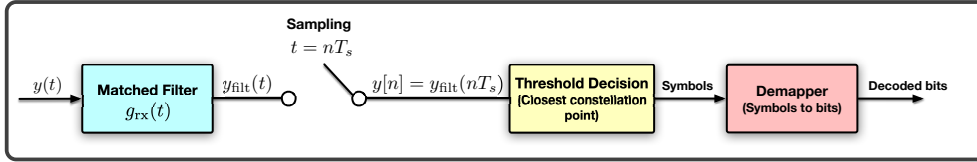


Figure D.6: Block diagram of the **matched filtering demodulation** pipeline used in our work. (First symbol at $\tau_s = 8$, symbol period $T_s = 16$, as detailed in Appendix C)

E Classical RF Interference Mitigation Techniques

E.1 Matched Filtering

Matched filtering (MF) is a signal processing technique commonly used in the demodulation of RF waveforms. It exploits knowledge about the signal waveform to enhance the detection and recovery of the transmitted symbols/bits, and is optimal in the maximum-SNR sense for signals contaminated with additive Gaussian noise.

The basic principle involves filtering (or similarly viewed as correlating) the received sampled RF waveform with a known reference waveform called the “matched filter”. The goal is to maximize the signal-to-interference-plus-noise ratio (SINR) at the filtered output, which consequently minimizes the error probability in the subsequent symbol detection when the noise is Gaussian. In the following segment, we develop the theory of the matched filter, with particular focus to our problem formulation.

Consider the baseband RRC-QPSK signal as described in Appendix C. Suppose, for the purposes of this exposition, that we adopt a simple additive white Gaussian noise (AWGN) channel model, thereby representing our received signal as

$$y(t) = \sum_p c_p g_{\text{tx}}(t - pT_s) + w(t) \quad (\text{E.1})$$

$$= g_{\text{tx}}(t) * \sum_p c_p \delta(t - pT_s) + w(t), \quad (\text{E.2})$$

where c_p are the symbols from a QPSK constellation (see Figure C.1b), $*$ denotes the convolution operator, $\delta(\cdot)$ is the dirac delta function, and $w(t) \sim \mathcal{N}(0, \sigma_{\text{AWGN}}^2)$ is the additive noise in the observed signal, statistically independent of all $\{c_p\}$. Of particular interest in this formulation is the transmit pulse shaping function $g_{\text{tx}}(t)$, where we chose to use the RRC function. (This is due its favorable properties in practice; however, the optimal transmit filter choices and waveform designs

are outside the scope of this work, and we defer readers to related resources for more information on this topic [4, 9].

At the receiver, we seek a receiver filter, $g_{\text{rx}}(t)$, such that the filtered and sampled output

$$y_{\text{filt}}(t) = \underbrace{g_{\text{rx}}(t) * g_{\text{tx}}(t)}_{:=g(t)} * \sum_p c_p \delta(t - pT_s) + g_{\text{rx}}(t) * w(t) \quad (\text{E.3})$$

$$y[n] = y_{\text{filt}}(nT_s) = \sum_p c_p g((n-p)T_s) + \underbrace{\int w(\tau) g_{\text{rx}}(nT_s - \tau) d\tau}_{:=v[n]} \quad (\text{E.4})$$

$$= \underbrace{c_n g(0)}_{:=y_s[n]} + \underbrace{\sum_{p \neq n} c_p g((n-p)T_s)}_{:=y_v[n]} + v[n] \quad (\text{E.5})$$

would maximize the output SINR. In other words, we are looking to maximize

$$\text{SINR} = \frac{\mathbb{E}[|y_s[n]|^2]}{\mathbb{E}[|y_v[n]|^2]} = \frac{\mathbb{E}[|c_n|^2] |g(0)|^2}{\mathbb{E}[|c_n|^2] \sum_{p \neq n} |g(pT_s)|^2 + \sigma_{\text{AWGN}}^2 \int |G_{\text{rx}}(f)|^2 df} \quad (\text{E.6})$$

(where $G_{\text{rx}}(f)$ is the Fourier transform of $g_{\text{rx}}(t)$) via an appropriate choice of $g(t)$ —and thereby, $g_{\text{rx}}(t)$. This can be done by finding an upper bound on the SINR that reaches equality for the appropriate filter choices. Ultimately, one such choice is $g_{\text{rx}}(t) = g_{\text{tx}}^*(-t)$ —termed as the *matched filter*—that leads to a maximized SINR. In the case of an RRC pulse shaping function (which is real and symmetric), the matched filter is also the same RRC function.

In this work, we refer to matched filtering more broadly to also include the detector/decoding step, as shown in Figure D.6. As part of the demodulation pipeline, the filtered output is sampled (as in (E.5)), and then mapped to the closest symbol in a predefined constellation (in the Euclidean distance sense). Finally, we can map these complex-valued symbols back to their corresponding bits to recover the underlying information. We use this as a standard demodulation/detection pipeline for our RRC-QPSK SOI following the corresponding signal separation/interference mitigation step (where applicable).

Demodulation with matched filtering is optimal for waveforms in the presence of additive Gaussian noise. However, in our signal separation problem, we consider the presence of an additive interference, which is not necessarily Gaussian. Thus, exploiting the non-Gaussian characteristics of the interference would likely lead to enhanced decoding performance.

E.2 LMMSE Estimation

Recall that our observation model is

$$\mathbf{y} = \mathbf{s} + \kappa \mathbf{b},$$

where we assume \mathbf{s} and \mathbf{b} are zero-mean and that they are statistically independent. The linear minimum mean square error (LMMSE) estimator is the estimator $\hat{\mathbf{s}} = \mathbf{W}_{\text{LMMSE}} \mathbf{y}$, such that

$$\mathbf{W}_{\text{LMMSE}} = \arg \min_{\mathbf{W} \in \mathbb{C}^{T \times T}} \mathbb{E} [\|\mathbf{s} - \mathbf{W} \mathbf{y}\|_2^2]. \quad (\text{E.7})$$

In this case, the optimal linear transformation (in the sense of (E.7)) can be written as

$$\mathbf{W}_{\text{LMMSE}} = \mathbf{C}_{sy} \mathbf{C}_{yy}^{-1} = \mathbf{C}_{ss} (\mathbf{C}_{ss} + \kappa^2 \mathbf{C}_{bb})^{-1}$$

where $\mathbf{C}_{sy} := \mathbb{E}[\mathbf{s} \mathbf{y}^H]$ corresponds to the cross-covariance between \mathbf{s} and \mathbf{y} , \mathbf{C}_{yy} , \mathbf{C}_{ss} , \mathbf{C}_{bb} are the auto-covariance of \mathbf{y} , \mathbf{s} and \mathbf{b} respectively. The second equality is obtained by statistical independence, thereby $\mathbf{C}_{sy} = \mathbf{C}_{ss}$, $\mathbf{C}_{yy} = \mathbf{C}_{ss} + \kappa^2 \mathbf{C}_{bb}$.

In relevant literature, this may also be referred to as linear MMSE receivers [10, Sec 8.3.3]. Note that we aim to mitigate the effects of an additive interference channel. For the purposes of this work, we use LMMSE as one of the baselines as a signal separation (interference mitigation) method. Thereafter, we assess performance based on the squared error between $\hat{\mathbf{s}}$ with the ground truth \mathbf{s} . To obtain the underlying bits, we perform a standard matched filtering operation on the estimator $\hat{\mathbf{s}}$.

To implement the LMMSE estimator described above, one simply requires the covariances \mathbf{C}_{yy} , \mathbf{C}_{ss} , \mathbf{C}_{bb} . Under the problem formulation, one could potentially obtain the sample covariance, whose precision scales with the number of samples used. In the case of the synthetic signals, we assume oracle knowledge of the covariance when computing this baseline, so as to eliminate confounding effects from poor estimation of these statistics. However, for CommSignal2, we assume that the signal is zero-mean (validated with the empirical mean), and compute the autocorrelation matrix for time windows of length 2560, across 10,000 sample realizations drawn randomly from the frames provided in the dataset. The empirical covariance estimate that we used will be provided in the accompanying code/data repository.

Note that in this problem, the interference also has random time offsets, which has to be accounted for when computing or estimating the covariance of our interference signals. The works in [11, 12] address the effects of such random time offsets in detail, and discuss the suboptimality of the above LMMSE in such a setting. We acknowledge that without an additional time synchronization step, the LMMSE estimator adopted herein may be modeling the interference as time series from a stationary process.

We remark that the LMMSE estimator is optimal if the components were Gaussian. However, as digital communication signals contain some underlying discreteness and undergo unknown time-shifts, these signals are typically non-Gaussian (and often, even far from Gaussian). Hence, better performance can generally be obtained through nonlinear methods.

F BASIS for RF Source Separation

We use the BASIS method [13] as a learning-based benchmark for SCSS in our experiments. Similar to our method, BASIS leverages pre-trained generative models as statistical priors and does not rely on end-to-end (supervised) training with paired data. The manner in which these priors are used for separation, however differs:

1. **Soft constraint vs. hard constraint:** In the context of our problem formulation, we are interested in decomposing a mixture $\mathbf{y} = \mathbf{s} + \kappa\mathbf{b}$ into its constituent components. As described in §3.1 our proposed method recovers the components by extending a MAP estimation rule with a hard constraint given by,

$$\hat{\mathbf{s}} = \arg \max_{\mathbf{s} \in \mathcal{S} \text{ s.t. } \mathbf{y} = \mathbf{s} + \kappa\mathbf{b}} p_{\mathbf{s}|\mathbf{y}}(\mathbf{s}|\mathbf{y}), \quad (\text{F.1})$$

where the resulting estimate of \mathbf{b} is $\hat{\mathbf{b}} = (\mathbf{y} - \hat{\mathbf{s}})/\kappa$. The result of imposing this constraint is a MAP objective that leverages two priors but only requires optimization with a single

Algorithm 2 BASIS Separation (original, as proposed in [13])

```

1: function SEPARATION( $\mathbf{y}, \kappa, N, \{\sigma_t^2\}_{t=1}^T, \psi_s^{(0)}, \psi_b^{(0)}$ )  $\triangleright$  Specially tuned noise schedule  $\{\sigma_t\}_{t=1}^T$ 
2:   for  $t \leftarrow 1, T$  do
3:      $\psi_s^{(t)} \leftarrow \psi_s^{(t-1)}, \psi_b^{(t)} \leftarrow \psi_b^{(t-1)}, \eta_t \leftarrow f(\sigma_t^2)$   $\triangleright$  Learning rate  $\eta_t$ 
4:     for  $i \leftarrow 0, N - 1$  do
5:        $\epsilon_s, \epsilon_b \sim \mathcal{N}(0, \mathbf{I})$ 
6:        $\hat{\mathbf{z}}_s, \hat{\mathbf{z}}_b = r_{\phi_s}(\psi_s^{(t)}, t), r_{\phi_b}(\psi_b^{(t)}, t)$   $\triangleright$  Compute scores at noise level  $\sigma_t$ 
7:        $\psi_s^{(t)} \leftarrow \psi_s^{(t)} - \eta_t \underbrace{(\hat{\mathbf{z}}_s / \sigma_t)}_{\text{approx. score}} - \eta_t (\mathbf{y} - \psi_s^{(t)} - \kappa\psi_b^{(t)}) / \sigma_t^2 + \sqrt{2\eta_t}\epsilon_s$ 
8:        $\psi_b^{(t)} \leftarrow \psi_b^{(t)} - \eta_t (\hat{\mathbf{z}}_b / \sigma_t) - \eta_t \kappa (\mathbf{y} - \psi_s^{(t)} - \kappa\psi_b^{(t)}) / \sigma_t^2 + \sqrt{2\eta_t}\epsilon_b$ 
9:     end for
10:  end for
11:  return  $\hat{\mathbf{s}} = \psi_s^{(T)}, \hat{\mathbf{b}} = \psi_b^{(T)}$ 
12: end function

```

variable,

$$\arg \max_{\mathbf{s} \in \mathcal{S}} p_{\mathbf{s}|\mathbf{y}}(\mathbf{s}|\mathbf{y}) = \arg \max_{\mathbf{s} \in \mathcal{S} \text{ s.t. } \mathbf{y}=\mathbf{s}+\kappa\mathbf{b}} p_{\mathbf{y}|\mathbf{s}}(\mathbf{y}|\mathbf{s})P_{\mathbf{s}}(\mathbf{s}) \quad (\text{F.2a})$$

$$= \arg \min_{\mathbf{s} \in \mathcal{S}} -\log P_{\mathbf{s}}(\mathbf{s}) - \log p_{\mathbf{b}}((\mathbf{y} - \mathbf{s})/\kappa). \quad (\text{F.2b})$$

In contrast, BASIS strives to estimate the underlying components by *sampling from the posterior*. In particular, they do not enforce a hard constraint during sampling and instead assume a likelihood of the form

$$p_{\mathbf{y}|\mathbf{s},\mathbf{b}}^{\text{BASIS}}(\mathbf{y}|\mathbf{s},\mathbf{b}) = \mathcal{N}(\mathbf{y}; \mathbf{s} + \kappa\mathbf{b}, \gamma^2\mathbf{I}), \quad (\text{F.3})$$

which actually corresponds to an alternate mixture model with auxilliary noise $\mathbf{w} \sim \mathcal{N}(0, \gamma^2\mathbf{I})$ such that,

$$\mathbf{y} = \mathbf{s} + \kappa\mathbf{b} + \mathbf{w}. \quad (\text{F.4})$$

Thus, the estimates $\hat{\mathbf{s}}^{\text{BASIS}}$ and $\hat{\mathbf{b}}^{\text{BASIS}}$ are obtained by sampling from the posterior,

$$p_{\mathbf{s},\mathbf{b}|\mathbf{y}}^{\text{BASIS}}(\mathbf{s},\mathbf{b}|\mathbf{y}) = p_{\mathbf{y}|\mathbf{s},\mathbf{b}}^{\text{BASIS}}(\mathbf{y}|\mathbf{s},\mathbf{b})P_{\mathbf{s}}(\mathbf{s})p_{\mathbf{b}}(\mathbf{b}). \quad (\text{F.5})$$

Notice that sampling from this posterior requires computing *two separate estimates* due to the soft constraint in (F.4).

2. **Annealed Langevin dynamics vs. randomized levels of Gaussian smoothing:** The aforementioned BASIS estimates satisfy $\mathbf{y} = \hat{\mathbf{s}}^{\text{BASIS}} + \kappa\hat{\mathbf{b}}^{\text{BASIS}}$ only when $\gamma \rightarrow 0$ in the soft constraint in (F.4) enforced via the likelihood term. As shown in Algorithm 2, the idea behind BASIS is to leverage annealed Langevin dynamics sampling to sample from the posterior by slowly decreasing the value of γ towards zero by *tuning a specially chosen noise schedule*. While generally a separate noise schedule is required per prior, the authors suggest sharing the same schedule across both priors. Apart from tuning the noise schedule, the learning rate schedule is another hyperparameter that must also be tuned.
3. **Optimization via sampling vs. optimization via gradient descent:** Starting with initializations $\psi_s^{(0)}$ and $\psi_b^{(0)}$ for the SOI and interference respectively, BASIS refines the estimate such that $\psi_s^{(t)}$ approximates a sample drawn from the distribution of the smoothed source $p_{\tilde{s}_t}$. Langevin dynamics leverages the approximate score of $p_{\tilde{s}_t}$, to update the estimate by *sampling from higher density regions*. A similar update is performed for the interference. The estimates are constrained to satisfy (F.4) through the Gaussian likelihood.

In contrast, as described in Algorithm 1 (see §3.2.2), we obtain an estimate $\boldsymbol{\theta}^{(t)}$ of the unsmoothed SOI for each randomly drawn noise level. The hard constraint allows us to readily get an estimate for the interference as $(\mathbf{y} - \boldsymbol{\theta}^{(t)})/\kappa$. As described in §3.2.2, we then update the SOI estimate using gradient descent, with a rule guided by (F.2b), using the *gradients of the log densities* with respect to $\boldsymbol{\theta}$ (which is functionally equivalent to the scaled scores). The ideal update gradient is,

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) := -\mathbb{E}_{\mathbf{t},z_t} \left[\sqrt{\alpha_t} S_{\tilde{s}_t}(\tilde{\mathbf{s}}_t(\boldsymbol{\theta})) \right] + \frac{\omega}{\kappa} \mathbb{E}_{\mathbf{u},z_u} \left[\sqrt{\alpha_u} S_{\tilde{\mathbf{b}}_u}(\tilde{\mathbf{b}}_u(\boldsymbol{\theta},\mathbf{y})) \right]. \quad (\text{F.6})$$

We first implement BASIS in its original form, with our best attempt at finding an appropriate learning rate. As shown in Figure F.1, on QPSK (SOI) + OFDM (QPSK) mixture, this leads to worse performance than matched filtering—likely due to poor matching of the observation model in (F.4). Note that we are limited in the annealing schedule by the set of possible noise levels, based on the training setup of our diffusion models. Thus, we use the diffusion model training noise schedule discretized into $T = 50$ timesteps (see §5.2) and additionally use $N = 100$. We seek to use the largest set of noise levels possible, consistent with approaches in most Langevin sampling based works. We tuned the learning rate schedule to the best of our abilities and found that $\eta_t = f(\sigma_t^2) = (2e-8)\sigma_t^2/\sigma_1^2$, gave the best results. All curves use an analytical score model for the SOI.

To make an even more conservative comparison (w.r.t. our proposed method), we replace the posterior using (F.5), which corresponds to the formulation in (F.4), with a hard constraint as in our setup. This method, as further described in Algorithm 3, leads to better performance, even outperforming LMMSE in some cases. We also experiment with $N = 400$ (for a total of 20,000 iterations), and

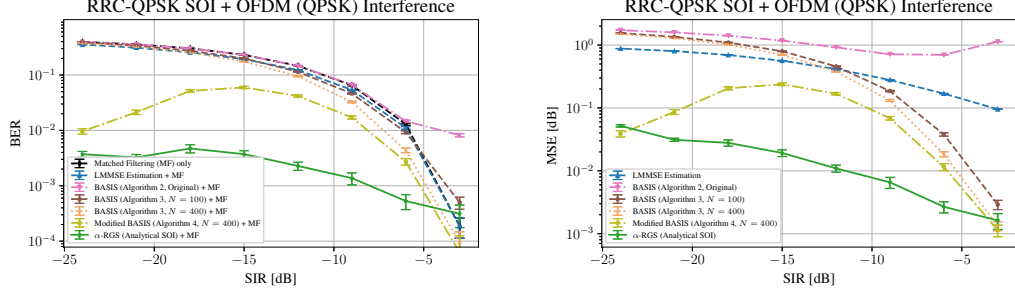


Figure F.1: Comparing BASIS and Modified BASIS against conventional RF source separation methods. The vanilla BASIS algorithm performs worse than MF, while the Modified BASIS algorithm, which is inspired by our usage of an α -posterior, is able to beat LMMSE. Still, our method outperforms all baselines.

Algorithm 3 BASIS Separation, using (F.2b)

```

1: function SEPARATION( $\mathbf{y}, \kappa, N, \{\sigma_t^2\}_{t=1}^T, \boldsymbol{\theta}^{(0)}$ )    ▷ Specially tuned noise schedule  $\{\sigma_t\}_{t=1}^T$ 
2:   for  $t \leftarrow 1, T$  do
3:      $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}, \eta_t \leftarrow f(\sigma_t^2)$     ▷ Learning rate  $\eta_t$ 
4:     for  $i \leftarrow 0, N - 1$  do
5:        $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ 
6:        $\hat{\mathbf{z}}_s, \hat{\mathbf{z}}_b = r_{\phi_s}(\boldsymbol{\theta}^{(t)}, t), r_{\phi_b}(\left(\frac{\mathbf{y} - \boldsymbol{\theta}^{(t)}}{\kappa}, t\right)$     ▷ Compute scores at noise level  $\sigma_t$ 
7:        $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t)} - \underbrace{\eta_t (\hat{\mathbf{z}}_s / \sigma_t) + \frac{1}{\kappa} \eta_t (\hat{\mathbf{z}}_b / \sigma_t)}_{\text{from MAP objective}} + \sqrt{2\eta_t} \boldsymbol{\epsilon}$ 
8:     end for
9:   end for
10:  return  $\hat{\mathbf{s}} = \boldsymbol{\theta}^{(T)}, \hat{\mathbf{b}} = (\mathbf{y} - \boldsymbol{\theta}^{(T)}) / \kappa$ 
11: end function

```

Algorithm 4 Modified BASIS Separation by introducing α -posterior

```

1: function SEPARATION( $\mathbf{y}, \kappa, N, \{\sigma_t^2\}_{t=1}^T, \boldsymbol{\theta}^{(0)}$ )    ▷ Specially tuned noise schedule  $\{\sigma_t\}_{t=1}^T$ 
2:   for  $t \leftarrow 1, T$  do
3:      $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}, \eta_t \leftarrow f(\sigma_t^2)$     ▷ Learning rate  $\eta_t$ 
4:     for  $i \leftarrow 0, N - 1$  do
5:        $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ 
6:        $\hat{\mathbf{z}}_s, \hat{\mathbf{z}}_b = r_{\phi_s}(\boldsymbol{\theta}^{(t)}, t), r_{\phi_b}(\left(\frac{\mathbf{y} - \boldsymbol{\theta}^{(t)}}{\kappa}, t\right)$     ▷ Compute scores at noise level  $\sigma_t$ 
7:        $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t)} - \underbrace{\eta_t (\hat{\mathbf{z}}_s / \sigma_t) + \frac{\omega}{\kappa} \eta_t (\hat{\mathbf{z}}_b / \sigma_t)}_{\text{from MAP objective with } \alpha\text{-posterior}} + \sqrt{2\eta_t} \boldsymbol{\epsilon}$ 
8:     end for
9:   end for
10:  return  $\hat{\mathbf{s}} = \boldsymbol{\theta}^{(T)}, \hat{\mathbf{b}} = (\mathbf{y} - \boldsymbol{\theta}^{(T)}) / \kappa$ 
11: end function

```

do not observe significant gains. Empirically we found this method to be typically stuck in local optimum which is in line with our characterization earlier in §4.

Finally, we augment the BASIS algorithm with an α -posterior with $\alpha = \omega$, as described in Algorithm 4. As shown in Figure F.1, the α -posterior augmented method obtains better performance than vanilla BASIS, but is generally unable to reach the performance of our method. We emphasize again that in contrast to our method, the aforementioned BASIS-based methods follow a prescribed annealing schedule, that can require cumbersome tuning for optimal performance. The implementations of Algorithm 3 and Algorithm 4 both use a learning rate schedule of $\eta_t = f(\sigma_t^2) = (2e-6)\sigma_t^2/\sigma_1^2$

G Complete Results

In this section we will provide the complete results of our study. We first present another baseline based on simulating the reverse diffusion process [7] as a denoiser, that complements the baselines already described in §5.2.

Reverse Diffusion. Given a mixture \mathbf{y} , we interpret the SOI as (scaled) additive noise on top of the interference \mathbf{b} . Since the reverse diffusion process can be interpreted as an iterative denoiser (see §2.2), we run a small chain of reverse diffusion for 10 timesteps starting at a noise variance of 0.005 and ending with variance of 0.0001, which was chosen based on different trials. Similar ideas have been used in prior works involving inverse problems for images [14, 15]. We note that it is generally cumbersome to find the optimal reverse diffusion noise range and it might in fact be dependent on the specific mixture’s SIR.

Extracting the underlying bits. Our primary metric for assessing the quality of the estimated SOI is through the BER between the estimated bits and the truly encoded bits. The estimate of the SOI typically equals the true SOI with a very small amount of noise due to the continuous nature of the optimization problem. Thus, we can expect the underlying constellation to contain some symbols concentrated around, but not exactly at the constellation points. This can be modeled as the true SOI corrupted with a small amount of Gaussian noise, for which MF is the optimal technique to decode the bits by mapping the symbols to the closest constellation point (see Section E.1).

The analytical SOI score. We use an RRC-QPSK signal as the SOI across all our source separation experiments. As detailed in Section B.2, the score for a smoothed QPSK source can be analytically computed via Proposition 2, where it is more amenable to model and compute it in the symbol space (i.e., as i.i.d. symbols). Nevertheless, for the problem of separating the SOI from an interference source, we have to consider the components jointly in the time domain. Thus, we relate the time-domain representation to the symbols via $\mathbf{s} = \mathbf{H}\mathbf{a}$, where \mathbf{H} represents the RRC filter matrix and \mathbf{a} is a vector of symbols. To compute this analytical score, we smooth the symbols via a Gaussian smoothing model at noise level corresponding to timestep t ,

$$\tilde{\mathbf{s}}_t = \mathbf{H}\tilde{\mathbf{a}}_t := \mathbf{H}(\gamma_t\mathbf{a} + \sigma_t\mathbf{z}_s). \quad (\text{G.1})$$

With this relation, we express the score of the smoothed source as,

$$\nabla_{\tilde{\mathbf{s}}_t} \log p_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{s}}_t) = \mathbf{H} \cdot \nabla_{\tilde{\mathbf{a}}_t} \log p_{\tilde{\mathbf{a}}_t}(\tilde{\mathbf{a}}_t) \quad (\text{G.2})$$

$$= \mathbf{H} \cdot \left(\frac{1}{\sigma_t^2} \left(-\tilde{\mathbf{a}}_t + \sum_{\mathbf{a} \in \mathcal{A}^K} \mathbf{a} \odot \phi_t(\mathbf{a}; \tilde{\mathbf{a}}_t) \right) \right) \quad (\text{G.3})$$

$$\approx \frac{1}{\sigma_t^2} \left(-\tilde{\mathbf{s}}_t + \mathbf{H} \sum_{\mathbf{a} \in \mathcal{A}^K} \mathbf{a} \odot \phi_t(\mathbf{a}; \mathbf{H}^\dagger \tilde{\mathbf{s}}_t) \right). \quad (\text{G.4})$$

where \odot represents element-wise product, ϕ_t is the softmax-like operator defined in (B.19) that is applied element-wise here, and \mathbf{H}^\dagger is the pseudo-inverse of \mathbf{H} . Note that (G.3) is obtained by applying (B.18) to a vector of i.i.d. smoothed QPSK symbols. In our implementation, the estimate of these smoothed symbols $\tilde{\mathbf{a}}_t$ is obtained by reversing the RRC filter using \mathbf{H}^\dagger , as in (G.4).

On the other hand, our RRC-QPSK diffusion model is trained directly on the time-domain waveform, and can be used directly to separate the SOI waveform from the mixture without conversion to the symbol domain. This once again sheds light on the practicality of using data-driven methods to circumvent otherwise computationally challenging statistical modeling technical problems.

Results. Figures G.1 and G.2 show the complete source separation results for mixtures with OFDM and CommSignal2 as interference, respectively. Our model that uses an analytical SOI score for the SOI and a diffusion-based score for the interference generally performs the best and outperforms all baselines. Furthermore, we show that using a learned SOI score still outperforms all baselines in terms of BER, despite the slight degradation. The trained SOI score models consistently outperform all methods at high SIR in terms of BER since the SOI diffusion model was trained with RRC-QPSK samples as opposed to the approximations made in (G.4) in implementing the analytical score.

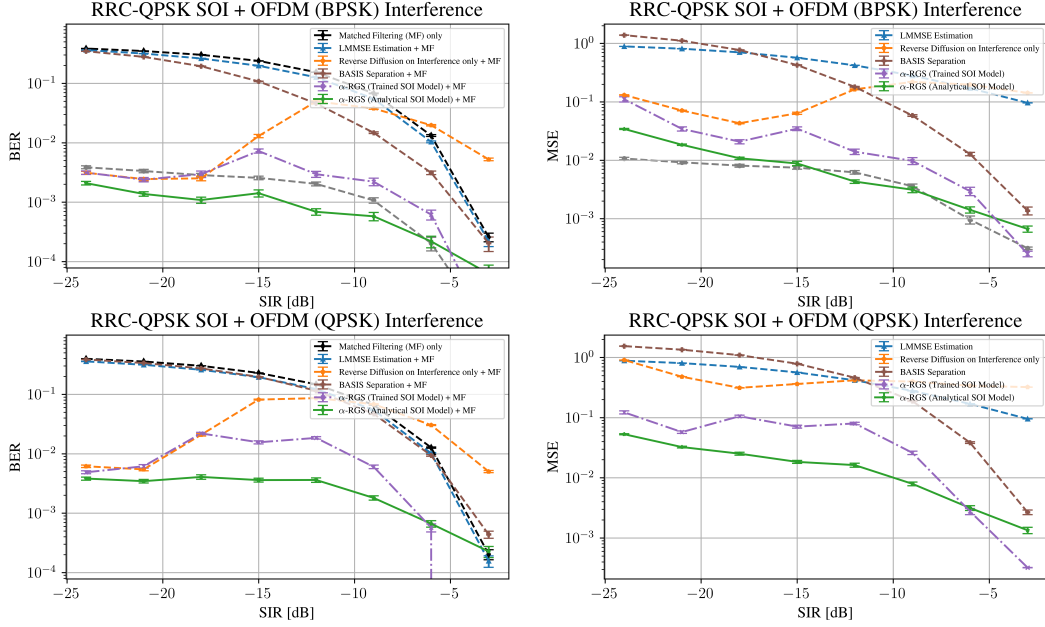


Figure G.1: Comparing our method against various baselines on separating (**top**) RRC-QPSK + OFDM (BPSK) mixtures and (**bottom**) RRC-QPSK + OFDM (QPSK) mixtures. Our model that uses an analytical SOI score outperforms all baselines in terms of BER. Reverse diffusion is competitive at low SIRs since the interference dominates the mixture in this regime and hence iterative denoising to separate the SOI is effective. The trained SOI diffusion models significantly outperform all baselines at high SIR. The reason the analytical SOI performs slightly worse is due to the approximations we make in (G.4).

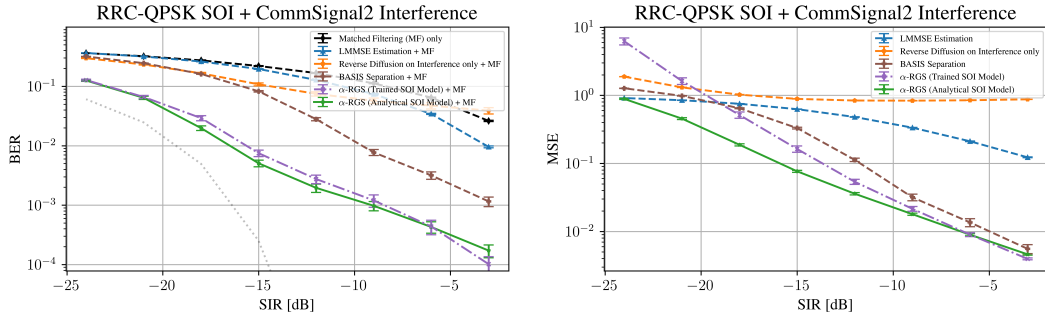


Figure G.2: Comparing our method against various baselines on separating RRC-QPSK + CommSignal2 mixtures. Our method significantly outperforms all baselines in terms of BER. The large MSE at low SIRs suggests that there is background noise present in the CommSignal2 source, which is amplified at lower SIR (large κ). We try to estimate the amount of noise and model it as additive Gaussian noise. Accounting for this noise could presumably lead to a lower bound on the BER, shown by dotted black line on the left.

Our method also outperforms baselines in terms of MSE for OFDM mixtures as shown in Figure G.1. The performance at low SIR in the context of CommSignal2 mixtures is not the same. As shown in Figure G.2, the large MSE at low SIRs suggest that there is background noise present in the CommSignal2 sources that is amplified for large values of κ , i.e., the interference is actually of the form $\mathbf{b} + \mathbf{w}$ for some background noise \mathbf{w} . We validated that this was indeed the case, by visualizing samples in both the time-domain and frequency domain using the RF challenge demo notebook⁶. As shown in Figure G.3, we noticed segments of lower magnitude at the start and end, which we believe

⁶https://github.com/RFChallenge/rfchallenge_singlechannel_starter/blob/main/notebook/Demo.ipynb

to be background noise that is not part of intended communication signal. During source separation, this presumably results in a noisier estimate of the SOI in comparison to mixtures with no additional background noise. We estimated the SNR to be 16.9 dB by averaging across multiple samples. The dotted black curve on the left of Figure G.2 is a presumable lower bound on the BER by accounting for the magnitude of the background noise and modeling it as additive white Gaussian noise.

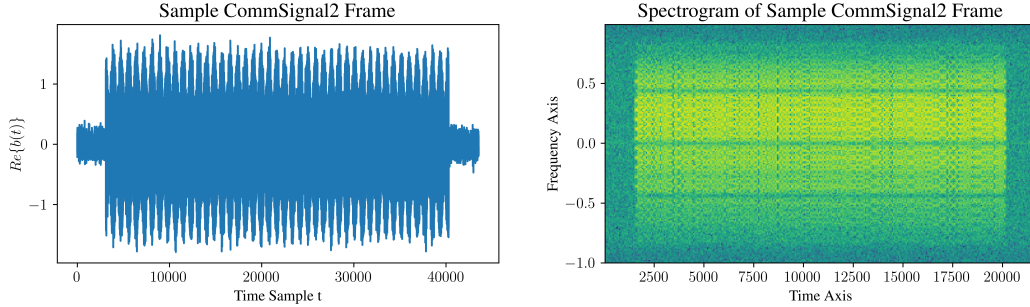


Figure G.3: Time-domain and frequency-domain plots of a single frame from the CommSignal2 dataset. **Left:** In the time-domain we observe segments of lower magnitude at the start and end, which we believe to be background noise. **Right:** In the frequency-domain we observe the bandlimited communication signal with spectrally flat features in the regions we identified as background noise.

Nevertheless, across all experiments, we demonstrate that our method sets a new state-of-the-art for applications such as interference mitigation where the interference can be learned from data recordings and the desired information can be decoded with knowledge of the SOI demodulation pipeline.

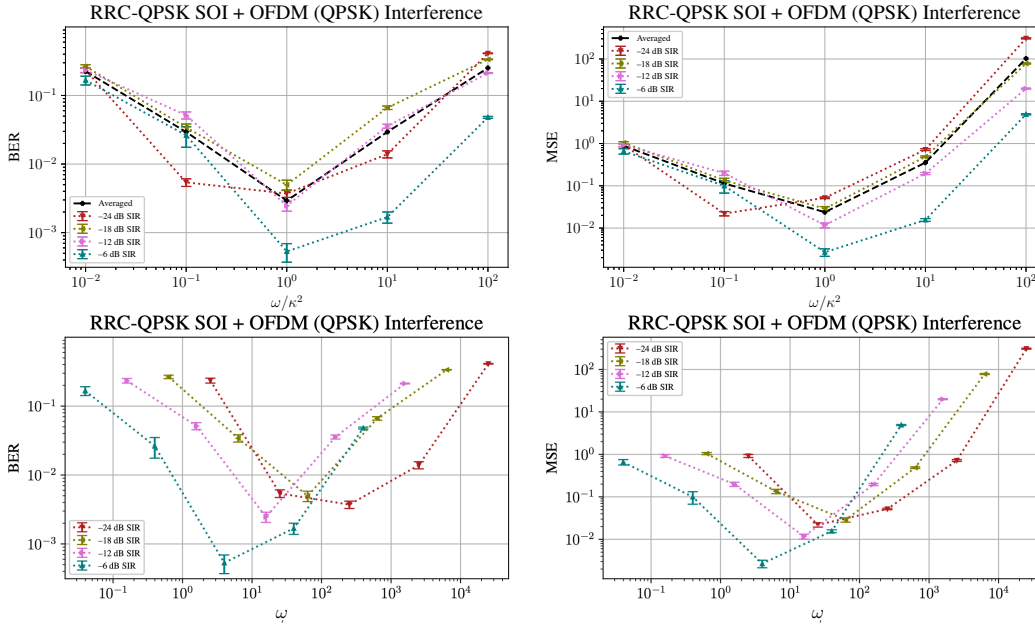


Figure G.4: **Top:** BER and MSE versus ω/κ^2 for different SIR levels. Evidently, a good choice of ω , on average, across different noise levels is $\omega = \kappa^2$. **Bottom:** By looking at individual SIRs we see that the minimum BER and MSE is achieved when ω increases with increasing κ^2 (decreasing SIR), visualized using a log scale on the x-axis.

Choice of ω . We numerically verify that $\omega = \kappa^2$ is a good choice of the α -posterior term. To this end, we first find a suitable order of magnitude for ω , by varying ω/κ^2 between 10^{-2} and 10^2 across different noise levels. As shown in the top row in Figure G.4, on average, the minimum BER and MSE is achieved when $\omega = \kappa^2$. We additionally validate that it is beneficial to adapt ω as the SIR

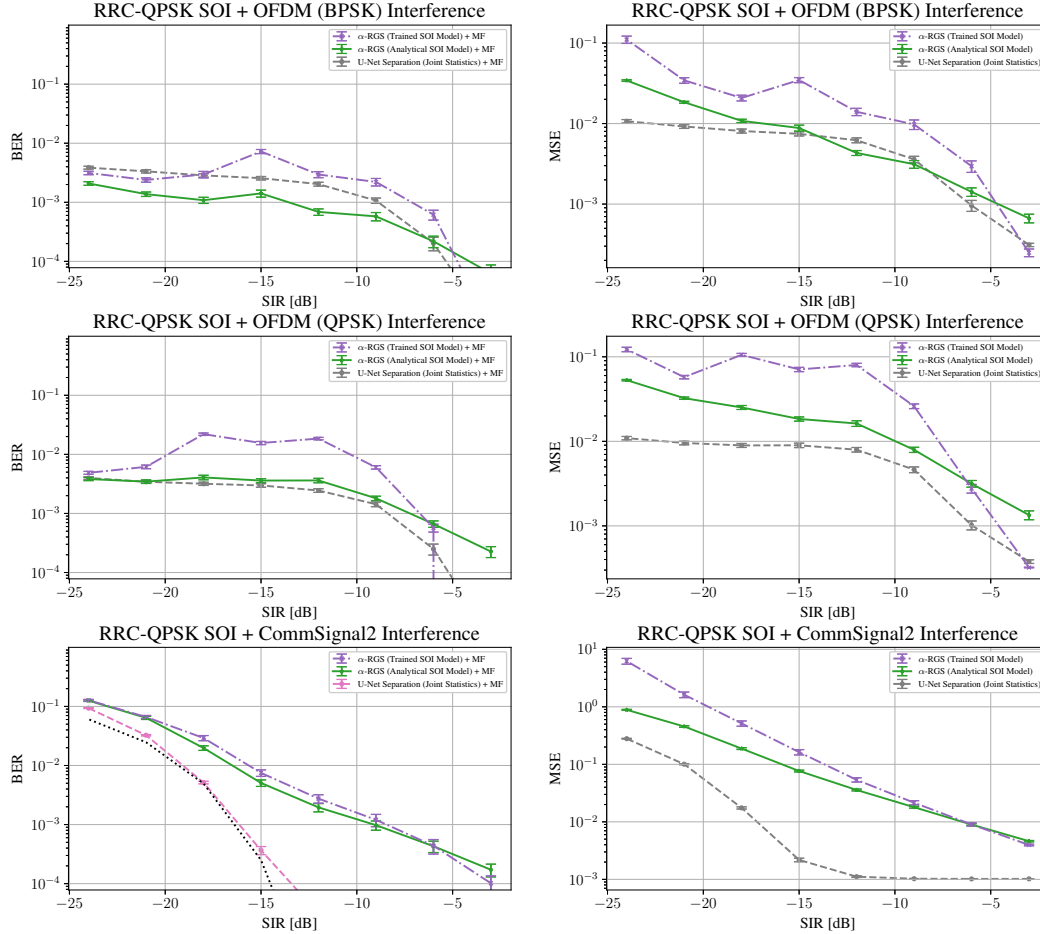


Figure G.5: Comparing our method against a supervised learning setup that trains a UNet on paired data samples with an ℓ_2 loss. We show that we are able to achieve competitive BERs, and even outperform the supervised method at certain SIRs. Our method is particularly competitive in the challenging strong interference regime (low SIR), demonstrating the fidelity of our trained interference diffusion models. The gap is larger for CommSignal2 mixtures as the supervised method is presumably able to leverage knowledge of the joint statistics between the SOI and interference to effectively deal with the background noise in the interference signal.

changes by varying ω and studying the BER and MSE curves at individual noise levels. As shown in the bottom row of Figure G.4, we observe that it ω should increase with κ^2 to achieve good results.

Comparing with supervised methods. As motivated in §1, we are interested in leveraging independently trained priors in our source separation setup. Nevertheless, we compare against a supervised setup that learns to separate mixtures end-to-end by training a UNet on *paired* data. We train three supervised models based on the recent work in [11], using their open-sourced training code⁷. These models are trained with an ℓ_2 loss. As such, we should expect the MSE performance to be better than our unsupervised approach, which is indeed the case across all the mixtures as shown in Figure G.5. However, we notice that our method, which uses an analytical SOI score, is able to perform similarly to the UNet and even better in terms of BER for some SIRs, especially in the OFDM interference setting. Furthermore, in the challenging low SIR (strong interference regime) our method performs well, showing that our interference diffusion models were able to learn the underlying statistical structures. Thus, if the mixture model changes in the future we can re-use our priors whereas the supervised approach could require an entire change to the architecture and additional training. We can potentially drive the BER in the CommSignal2 setting by modeling the background noise in these

⁷https://github.com/RFChallenge/SCSS_CSGaussian

signals either through a different mixture model or through additional priors might help in driving down the BER. This will be a focus of future work.

Computation Time. The inference time is as of now slightly longer than BASIS. To separate a single mixture, our method that uses $N = 20,000$ iterations takes 328 seconds on average, whereas BASIS takes 284 seconds for the same number of iterations. Meanwhile, the UNet only requires one forward pass through the model and can separate a mixture in less than one second. Our implementation was not optimized for computation time, but there may be strategies to speed up our algorithm in practice, which will be a focus of future work.

Code. The code for reproducing our results can be found at https://github.com/tkj516/score_based_source_separation and is also linked on our project webpage <https://alpha-rgs.github.io>.

References

- [1] Herbert E. Robbins. An empirical bayes approach to statistics. 1956.
- [2] Paul Bromiley. Products and convolutions of Gaussian probability density functions. *Tina-Vision Memo*, 3(4):1, 2003.
- [3] Jakob Hoydis, Sebastian Cammerer, Fayçal Ait Aoudia, Avinash Vem, Nikolaus Binder, Guillermo Marcus, and Alexander Keller. Sionna: An open-source library for next-generation physical layer research. *arXiv preprint*, Mar. 2022.
- [4] Amos Lapidoth. *A foundation in digital communication*. Cambridge University Press, 2017.
- [5] MIT RLE. Single-Channel RF Challenge. URL <https://rfchallenge.mit.edu/challenge-1/>. Accessed 2022-11-07.
- [6] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.
- [7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- [9] Robert W Heath Jr. *Introduction to wireless digital communication: a signal processing perspective*. Prentice Hall, 2017.
- [10] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge University Press, 2005.
- [11] Gary CF Lee, Amir Weiss, Alejandro Lancho, Jennifer Tang, Yuheng Bu, Yury Polyanskiy, and Gregory W Wornell. Exploiting temporal structures of cyclostationary signals for data-driven single-channel source separation. In *2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2022.
- [12] Alejandro Lancho, Amir Weiss, Gary CF Lee, Jennifer Tang, Yuheng Bu, Yury Polyanskiy, and Gregory W Wornell. Data-driven blind synchronization and interference rejection for digital communication signals. In *2022 IEEE Global Communications Conference (GLOBECOM)*, pages 2296–2302, 2022.
- [13] Vivek Jayaram and John Thickstun. Source separation with deep generative priors. In *International Conference on Machine Learning*, pages 4724–4735. PMLR, 2020.
- [14] Zahra Kadkhodaie and Eero Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Information Processing Systems*, 34: 13242–13254, 2021.
- [15] Vladimir Kulikov, Shahar Yadin, Matan Kleiner, and Tomer Michaeli. SinDDM: A single image denoising diffusion model. *arXiv preprint arXiv:2211.16582*, 2022.