



Article

Robust Multiple-Measurement Sparsity-Aware STAP with Bayesian Variational Autoencoder

Chenxi Zhang ¹, Huiliang Zhao ¹, Wenchao Chen ¹, Bo Chen ¹, Penghui Wang ^{1,*}, Changrui Jia ² and Hongwei Liu ¹

¹ National Lab of Radar Signal Processing, Xidian University, Xi'an 710071, China

² The 38th Research Institute of China Electronics Technology Corporation, Hefei 230088, China

* Correspondence: wangpenghui@mail.xidian.edu.cn; Tel.: +86-13609200460

Abstract: Due to the shortage of independent and identically distributed (i.i.d.) training samples, space–time adaptive processing (STAP) often suffers remarkable performance degradation in the heterogeneous clutter environment. Sparse recovery (SR) techniques have been introduced into STAP for the benefit of the drastically reduced training requirement, but they are incompletely robust for involving the tricky selection of hyper–parameters or the undesirable point estimation for parameters. Given this issue, we incorporate the Multiple–measurement Complex–valued Variational relevance vector machines (MCV) to model the space–time echoes and provide a Gibbs–sampling–based method to estimate posterior distributions of parameters accurately. However, the Gibbs sampler require quantities of iterations, as unattractive as traditional Bayesian type SR–STAP algorithms when the real–time processing is desired. To address this problem, we further develop the Bayesian Autoencoding MCV for STAP (BAMCV–STAP), which builds the generative model according to MCV and approximates posterior distributions of parameters with an inference network pre–trained off–line, to realize fast reconstruction of measurements. Experimental results on simulated and measured data demonstrate that BAMCV–STAP can achieve suboptimal clutter suppression in terms of the output signal to interference plus noise ratio (SINR) loss, as well as the attractive real–time processing property in terms of the convergence rate and computational loads.

Keywords: clutter suppression; variational autoencoder (VAE); space–time adaptive processing (STAP); clutter plus noise covariance matrix (CCM); sparse recovery (SR)



Citation: Zhang, C.; Zhao, H.; Chen, W.; Chen, B.; Wang, P.; Jia, C.; Liu, H. Robust Multiple-Measurement Sparsity-Aware STAP with Bayesian Variational Autoencoder. *Remote Sens.* **2022**, *14*, 3800. <https://doi.org/10.3390/rs14153800>

Academic Editor: Andreas Reigber

Received: 13 June 2022

Accepted: 20 July 2022

Published: 6 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Airborne phased–array radar is widely used to detect moving targets because of its mobility and freedom from the radius of the earth's curvature, but it is always plagued by the existence of the 'nuisance' clutter coupling in the space–time domain. To ensure the detection of weak targets, space-time adaptive processing (STAP) methods [1–4] are developed to mitigate the clutter and achieve better performance compared with traditional non–adaptive methods. For classical STAP methods, the performance is substantially determined by the accuracy of the clutter plus noise covariance matrix (CCM). Typically, the CCM of the cell under test (CUT) is estimated by training samples adjacent to the CUT, which are supposed to be independent and identically distributed (i.i.d.) [5]. According to the Reed–Mallett–Brennan (RMB) rule [6], the required training samples should be at least twice the system degree of freedom (DOF) to achieve steady performance. Nevertheless, since the clutter background is usually non–homogeneous and the DOF is usually high in practice, obtaining adequate qualified training samples is challenging, leading to dramatically degraded performance of CCM estimation and clutter suppression. A large number of methods to improve the performance of STAP with finite training samples have been proposed in the past few years, such as the reduced–dimension (RD) methods [7–10], the reduced–rank methods (RR) [11–13], the direct data domain (D3)

methods [14] and the knowledge-aided STAP (KA-STAP) methods [15–19]. However, the number of required samples is still large in the RD and RR methods, the DOF of D3 is significantly reduced resulting in severe performance loss, and the KA-STAP methods depend heavily on prior knowledge. By contrast, the sparse recovery STAP (SR-STAP) methods [20–23], which are not only more tolerant with the number of training samples but also more independent from the prior knowledge, have received significant attention. In SR-STAP, every atom of the dictionary is a space-time steering vector denoting a grid of the space-time plane, which is uniformly over-sampled in both the angle and Doppler. The space-time profile, recovered by solving the formulated optimization problem, will be leveraged to reconstruct the CCM for effective clutter suppression. Since the clutter suppression performance mainly relies on the accuracy of recovery results, many efforts have been put into the problem formulation of sparse recovery. The pioneering work, such as greedy algorithms [24–27], considers the problem of SR-STAP as an l_0 -norm minimization problem which is known as NP-hard [28,29]. To tackle the NP-hard problem, appealing approaches transform it into a convex problem [20,30–32], but the strong coherence among the atoms of sparse dictionary often leads to the severe performance degradation of sparse reconstruction. Besides, the recovery results obtained by convex optimization are particularly sensitive to the selection of regularization parameters.

Due to the superior ability to deal with the highly coherent dictionary situation and the tricky selection of regularization parameters, Bayesian type sparse recovery (SR) algorithms [22,33–35], represented by the sparse Bayesian learning (SBL) methods [36] and its extension for the multiple measurement vector (MMV) case, i.e., M-SBL [37], have received much attention in the field of STAP. However, the potential need for a large number of iterations per testing in these methods is a fatal drawback for STAP where real-time processing is desired. Some methods [33–35,38] are devoted to modifying the time-consuming process and accelerating the convergence, but they are at the price of either significant signal loss or expensive computational loads, which also hinder the real-time processing. In addition, all these Bayesian type SR methods are incompletely robust as a result of only providing point estimates for some parameters, such as the parameter controlling the variance of the space-time profile and the parameter denoting the noise power. In summary, the completely robust Bayesian type SR-STAP algorithms that ensure both the stable performance of sparse reconstruction and the feasibility of real-time processing are rarely studied.

To model the space-time echoes with a completely robust Bayesian framework, this paper introduces variational relevance vector machines (VRVM) [39] into STAP due to its attractive advantage of providing distributional estimates for all parameters without any user parameters (known as the parameter-free property) and sufficient ability to model the sparsity. However, the original VRVM is proposed in the real domain for the single measurement vector (SMV) case, which is not suitable for STAP where the complex-valued multiple measurement vector (MMV) case exists. As a result, we first develop the multiple-measurement complex-valued variational relevance vector machines (MCV), which can address the condition where complex-valued multiple measurements occur. To achieve accurate inference, we employ the Gibbs sampler as a general technique to estimate all parameters for MCV. Then, we present the novel STAP algorithm via MCV, named MCV-STAP, to design the adaptive filters. However, a large number of iterations, as well as expensive computational loads, are required in the Gibbs sampler, which limits the efficiency of the model for out-of-sample prediction. To render the real-time processing of the MCV-STAP, inspired by the structure of the variational autoencoder (VAE) [40–43], the powerful models for unsupervised learning in the field of deep learning, we introduce a VAE into our model to provide the variational inference and expand it to a VAE-structured model named Bayesian Autoencoding MCV-STAP (BAMCV-STAP), which is scalable in the training phase and fast in the testing phase. Then, our model can be optimized by minimizing the evidence lower bound (ELBO) via a gradient ascent scheme. We note that due to the gradient ascent scheme without involving inverse

operations when learning the parameters, BAMCV–STAP can approximate posterior densities of space–time profiles with low computational loads. Moreover, since the inference network can be pre–trained off–line, only a few iterations are needed before testing. Finally, the space–time profiles recovered via BAMCV–STAP are adopted to estimate the CCM for STAP. To the best of our knowledge, BAMCV–STAP is the first work that attempts a VRVM–with–VAE strategy in the context of STAP, and our method appears to outperform several other sparsity–enhanced methodologies in terms of clutter suppression performance and real–time processing, as will be demonstrated in the experimental results.

The main contributions of this paper are summarized as follows:

- Generalized from the original VRVM to the multiple measurements case existing in the complex domain, a parameter–free probabilistic model called MCV is derived to recover space–time profiles via the Gibbs sampling method for STAP.
- Since all parameters are estimated based on their posterior distributions in MCV, the robustness to the number of training samples and noise power estimation is significantly improved compared with other SR–STAP methods for the MMV case.
- Incorporating a suitable VAE into MCV, a novel method called BAMCV is developed to accelerate the convergence of iterative procedures for estimating parameters. As the inference network is pre–trained off–line, BAMCV–STAP can realize the sparse reconstruction with lower computational loads and much fewer iterations compared with conventional SR–STAP methods.
- As demonstrated on both simulated and measured data, the final proposed method BAMCV–STAP can process space–time echoes in real–time without degrading clutter suppression performance.

The rest of this paper is organized as follows. Section 2 introduces the signal model for airborne radar. Section 3 details the proposed algorithm. Numerical experiments with both simulated and measured data are shown in Section 4 before discussion in Section 5. Section 6 presents the conclusion and future work of findings.

2. System Model and Derivation of Optimal Filters

For tractability purpose, a side–looking airborne phased–array radar with a uniformly linear antenna (ULA) [34,44] consisting of M half wavelength spaced elements on the airborne radar platform is considered. As shown in Figure 1, the platform is at altitude h_p and moving with constant velocity v_p . φ and θ are the azimuth angle and the elevation angle, respectively. The radar transmits N pulses at a constant pulse repetition frequency (PRF) in a coherent processing interval (CPI). The model for the space–time clutter plus noise snapshot $\mathbf{x} \in \mathbb{C}^{MN \times 1}$ from the CUT is [45]

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_c + \mathbf{n} \\ &= \sum_{i=1}^{N_c} \alpha_i \mathbf{v}(f_{d,i}, f_{s,i}) + \mathbf{n}, \end{aligned} \quad (1)$$

where $\mathbf{x}_c \in \mathbb{C}^{MN \times 1}$ is the space–time clutter vector and $\mathbf{n} \in \mathbb{C}^{MN \times 1}$ is the Gaussian white thermal noise whose variance is σ^2 ; N_c denotes the number of independent clutter patches in the CUT; $\alpha_i, f_{d,i}, f_{s,i}, \mathbf{v}(f_{d,i}, f_{s,i}) \in \mathbb{C}^{MN \times 1}$ are the complex amplitude, the Doppler frequency, the spatial frequency and the space–time steering vector of the i th clutter patch, respectively. Considering the criterion of linearly constrained minimum variance (LCMV), the optimal STAP weight vector is [45]

$$\mathbf{S}_w = \kappa \mathbf{R}^{-1} \mathbf{v}_t, \quad (2)$$

where $\kappa = \left(\mathbf{v}_t^H \mathbf{R}^{-1} \mathbf{v}_t \right)^{-1}$ is the normalization constant, \mathbf{v}_t is the space–time steering vector of the target, and \mathbf{R} denotes the CCM, which is normally unknown and estimated

from i.i.d training samples. $(\cdot)^H$ denotes the conjugate operation. In conventional STAP algorithms, the CCM can be calculated by [45]

$$\mathbf{R} = \frac{1}{L} \sum_{l=1}^L \mathbf{x}_l \mathbf{x}_l^H, \tag{3}$$

where L is the number of training samples and $\mathbf{x}_l \in \mathbb{C}^{MN \times 1} (l = 1, 2, \dots, L)$ denotes the space–time clutter plus noise vector of l th training sample. In order to achieve stable performance, L is required to be at least twice of MN . However, it is difficult to obtain sufficient i.i.d. training samples in practice. As a result, SR–STAP algorithms, which exhibit better performance using far fewer training samples, stand out from the crowd. They start by dividing the whole space–time plane uniformly into $K = N_s N_t$ bins, where $N_s = \rho_s M (\rho_s > 1)$ and $N_t = \rho_t N (\rho_t > 1)$ are the number of spatial frequency grid points and Doppler frequency grid points, respectively; ρ_s and ρ_d determine the resolution of the spatial frequency and Doppler frequency axes. As each grid point can be represented as a space–time steering vector $\mathbf{v}_k (k = 1, 2, 3, \dots, K)$, STAP dictionary $\Phi \in \mathbb{C}^{MN \times K}$ can be denoted as a collection of all these space–time steering vectors, i.e.,

$$\Phi = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_K]. \tag{4}$$

For the SMV case, the signal model in Equation (1) can be re–expressed as

$$\mathbf{x} = \Phi \mathbf{w} + \mathbf{n}, \tag{5}$$

where $\mathbf{w} \in \mathbb{C}^{K \times 1}$ denotes the space–time profile with non–zero elements representing the presence of the clutter. The formulation of the CCM for the SMV case can be expressed as

$$\mathbf{R} = \Phi \mathbf{w} \mathbf{w}^H \Phi^H + \sigma^2 \mathbf{I}. \tag{6}$$

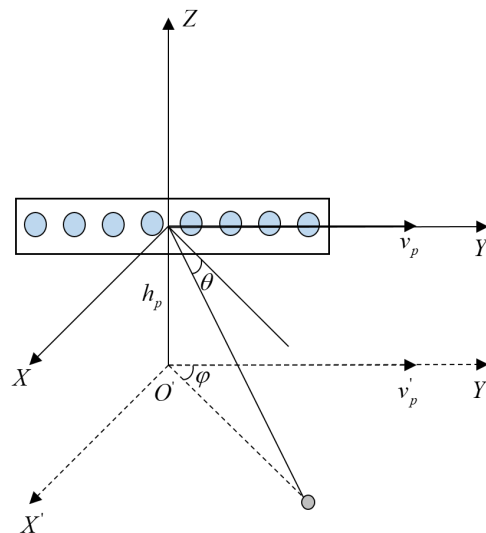


Figure 1. The side–looking airborne phased–array radar with a ULA.

For the MMV case, the signal model of L range cells $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L] \in \mathbb{C}^{MN \times L}$ can be expressed as

$$\mathbf{X} = \Phi \mathbf{W} + \mathbf{N}, \tag{7}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_L] \in \mathbb{C}^{K \times L}$ is an unknown solution matrix, $\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_L] \in \mathbb{C}^{MN \times L}$ is a Gaussian white thermal noise matrix. Inconsistent with the SMV case, row sparsity is encouraged simultaneously in the MMV case, i.e., each training sample in \mathbf{X} has

the same indices of direction of arrival (DOA). The calculation of the CCM for the MMV case can be expressed as

$$R = \frac{1}{L} \Phi W W^H \Phi^H + \sigma^2 I. \tag{8}$$

The optimal solution to W is discussed in the next section.

3. Proposed MCV–STAP Algorithm and BAMCV–STAP Algorithm

In this section, we first introduce the MCV approach by generalizing the complex–valued VRVM from the SMV case to the MMV case. Then we propose an MCV–STAP algorithm for designing the adaptive filters to suppress clutter and develop a Gibbs sampler to estimate parameters. To remedy the unattractive property requiring quantities of iterations in the Gibbs sampler of the MCV–STAP algorithm, we provide an efficient autoencoding variational inference network for the generative model built on the MCV approach and develop a VAE–structured method named BAMCV–STAP.

3.1. Derivation of MCV

We now derive the MCV approach by generalizing the complex–valued VRVM from the SMV case to the MMV case. Firstly, for the SMV case, assuming the noise is white complex Gaussian distribution with unknown power σ^2 , we can obtain the Gaussian likelihood function of the signal model in Equation (5) as

$$p(x | w, \sigma^2) \triangleq \mathcal{CN}(\Phi w, \sigma^2 I). \tag{9}$$

Drawing on the concept of automatic relevance determination (ARD) [46], we assign the Gaussian prior on the k th row of $w \in \mathbb{R}^K$ as

$$p(w_k | \gamma_k) \triangleq \mathcal{CN}(0, \gamma_k^{-1}) \tag{10}$$

where γ_k is an unknown variance parameter. Combining these row priors, a full prior of variable w can be expressed as

$$p(w | \gamma) \triangleq \prod_{k=1}^K p(w_k | \gamma_k) \tag{11}$$

where $\gamma = [\gamma_1, \gamma_2, \dots, \gamma_K]^T \in \mathbb{R}_+^K$ is a vector of hyper–parameters. In this way, every weight has an individual hyper–parameter to moderate the strength of the prior. To ensure the parameter–free property of our method and avoid point estimation of γ and σ^2 [36], the Gamma prior [47] over γ_k as well as over the noise variance σ^2 in the signal model are defined, i.e.,

$$p(\gamma) = \prod_{k=1}^K \text{Gamma}(\gamma_k | a_k, b_k), \tag{12}$$

$$p(\beta) = \text{Gamma}(\beta | c, d), \tag{13}$$

where $\beta = \sigma^{-2}$ and

$$\text{Gamma}(\gamma_k | a_k, b_k) = \Gamma(a_k)^{-1} b_k^{a_k} \gamma_k^{a_k-1} \exp(-b_k \gamma_k). \tag{14}$$

$\Gamma(a)$ is the gamma function expressed as

$$\Gamma(a) = \int_0^\infty t^{a-1} \exp(-t) dt. \tag{15}$$

By providing posterior distribution estimates for γ and β , our method increases the robustness of the noise power estimates while requiring fewer training samples.

Extending the above model for the SMV case to the MMV case to obtain the proposed model MCV, the Gaussian likelihood function of the signal model can be rewritten as

$$p(\mathbf{X} | \mathbf{W}, \beta) \triangleq \mathcal{CN}(\Phi \mathbf{W}, \beta^{-1} \mathbf{I}). \tag{16}$$

The prior of the k th row of \mathbf{W} in Equation (7) denoted by w_k is

$$p(w_k | \gamma_k) \triangleq \mathcal{CN}(\mathbf{0}, \gamma_k^{-1} \mathbf{I}). \tag{17}$$

So the prior of the l th column of \mathbf{W} denoted by $w_{.l}$ is

$$p(w_{.l} | \gamma) \triangleq \mathcal{CN}(\mathbf{0}, \mathbf{A}^{-1}) \tag{18}$$

where $\mathbf{A} = \text{diag}(\gamma)$ and $\text{diag}(\cdot)$ denotes the diagonal operation. Hence the full prior of \mathbf{W} is

$$p(\mathbf{W} | \gamma) \triangleq \prod_{l=1}^L p(w_{.l} | \gamma). \tag{19}$$

The priors of γ and β are the same as the case of SMV.

3.2. Proposed MCV–STAP Algorithm

The Gibbs sampler to estimate all parameters for MCV are derived from the SMV case in this subsection. Firstly, for the SMV case, the conditional posterior distribution over w can be written as

$$\begin{aligned} p(w | x, \gamma, \beta) &= \frac{p(x | w, \beta) p(w | \gamma)}{p(x | \gamma, \beta)} \\ &= \frac{|\Sigma|^{-1}}{\pi^K} \exp \left[-(w - \mu)^H \Sigma^{-1} (w - \mu) \right] \end{aligned} \tag{20}$$

where the posterior covariance and mean are respectively

$$\Sigma = \left(\beta \sum_{i=1}^{MN} \phi_i^H \phi_i + \mathbf{A} \right)^{-1}, \tag{21}$$

$$\mu = \beta \Sigma \Phi^H x, \tag{22}$$

with ϕ_i denoting the i -th row vector of Φ .

Similarly, the conditional posterior distribution over γ can be expressed as

$$\begin{aligned} p(\gamma | w, a_0, b_0) &= \frac{p(w | \gamma) p(\gamma | a_0, b_0)}{p(w | a_0, b_0)} \\ &= \prod_{k=1}^K \text{Gamma}(\gamma_k | a_k, b_k) \end{aligned} \tag{23}$$

where the shape parameter and the scale parameter can be expressed as

$$\begin{aligned} a_k &= a_{k,0} + 1, \\ b_k &= b_{k,0} + w_k^2. \end{aligned} \tag{24}$$

$\mathbf{a}_0 = [a_{1,0}, a_{2,0}, \dots, a_{K,0}]^T$ denotes the vector of initial values for shape parameters of γ , and $\mathbf{b}_0 = [b_{1,0}, b_{2,0}, \dots, b_{K,0}]^T$ denotes the vector of initial values for scale parameters of γ . The conditional posterior distribution over β is

$$\begin{aligned}
 p(\beta | \mathbf{x}, \mathbf{w}, c_0, d_0) &= \frac{p(\mathbf{x} | \mathbf{w}, \beta)p(\beta | c_0, d_0)}{p(\mathbf{x} | \mathbf{w}, c_0, d_0)} \\
 &= \text{Gamma}(\beta | c, d),
 \end{aligned}
 \tag{25}$$

where the shape parameter and the scale parameter respectively become

$$\begin{aligned}
 c &= c_0 + MN, \\
 d &= d_0 + \mathbf{x}^H \mathbf{x} - 2\mathbf{w}^H \Phi^H \mathbf{x} + \sum_{i=1}^{MN} \phi_i(\mathbf{w}\mathbf{w}^H)\phi_i^H.
 \end{aligned}
 \tag{26}$$

c_0 denotes the initial value for shape parameters of β and d_0 denotes the initial values for scale parameters of β .

Extending the above process to the MMV case, we develop the Gibbs sampler for MCV here. The conditional posterior distribution over the latent variable \mathbf{W} is thus given by

$$\begin{aligned}
 p(\mathbf{W} | \mathbf{X}, \gamma, \beta) &= \frac{p(\mathbf{X} | \mathbf{W}, \beta)p(\mathbf{W} | \gamma)}{p(\mathbf{X} | \gamma, \beta)} \\
 &= \frac{|\Sigma_{\mathbf{M}}|^{-1}}{\pi^K} \exp\{\text{tr}[-(\mathbf{W} - \mu_{\mathbf{M}})^H \Sigma_{\mathbf{M}}^{-1}(\mathbf{W} - \mu_{\mathbf{M}})]\}
 \end{aligned}
 \tag{27}$$

where the covariance and mean are respectively

$$\Sigma_{\mathbf{M}} = \left(\mathbf{f} \sum_{i=1}^{MN} \phi_i^H \phi_i + \mathbf{A} \right)^{-1},
 \tag{28}$$

$$\mu_{\mathbf{M}} = \beta \Sigma_{\mathbf{M}} \Phi^H \mathbf{X},
 \tag{29}$$

and $\text{tr}(\cdot)$ denotes the trace of a square matrix. The conditional posterior distribution over γ can be expressed as

$$\begin{aligned}
 p(\gamma | \mathbf{W}, \mathbf{a}_0, \mathbf{b}_0) &= \frac{p(\mathbf{W} | \gamma)p(\gamma | \mathbf{a}_0, \mathbf{b}_0)}{p(\mathbf{W} | \mathbf{a}_0, \mathbf{b}_0)} \\
 &= \prod_{k=1}^K \text{Gamma}(\gamma_k | a_k, b_k)
 \end{aligned}
 \tag{30}$$

where the shape parameter and the scale parameter can be expressed as

$$\begin{aligned}
 a_k &= a_{k,0} + L, \\
 b_k &= b_{k,0} + \|\mathbf{w}_k\|_2^2.
 \end{aligned}
 \tag{31}$$

The conditional posterior distribution over β is

$$\begin{aligned}
 p(\beta | \mathbf{X}, \mathbf{W}, c_0, d_0) &= \frac{p(\mathbf{X} | \mathbf{W}, \beta)p(\beta | c_0, d_0)}{p(\mathbf{X} | \mathbf{W}, c_0, d_0)} \\
 &= \text{Gamma}(\beta | c, d),
 \end{aligned}
 \tag{32}$$

where the shape parameter and the scale parameter respectively become

$$\begin{aligned}
 c &= c_0 + MNL, \\
 d &= d_0 + \sum_{l=1}^L \mathbf{x}_l^H \mathbf{x}_l - 2 \sum_{l=1}^L \mathbf{W}^H \Phi^H \mathbf{x}_l + \sum_{i=1}^{MN} \phi_i(\mathbf{W}\mathbf{W}^H)\phi_i^H.
 \end{aligned}
 \tag{33}$$

As conditional distributions over all parameters are specified clearly in the above, the space–time profiles estimated accurately by the Gibbs sampler of MCV will be adopted to design filters to suppress the clutter. For clarity, the pseudo–codes of the abovementioned MCV–STAP algorithm are provided in Algorithm 1.

Algorithm 1 MCV–STAP algorithm.

- Step1:** Give initial values $a_{k,0}=b_{k,0}=c_0=d_0=10^{-6}, k = 1, 2, 3, \dots, K$.
- Step2:** Sample $\gamma^{(0)}$ from Equation (12) and sample $\beta^{(0)}$ from Equation (13).
- Step3:** For $t = 1, 2, \dots$ **do**
 Calculate $\mu_M^{(t)}$ and $\Sigma_M^{(t)}$ via Equations (28) and (29);
 Sample $W^{(t)}$ from Equation (20);
 Calculate $a^{(t)}$ and $b^{(t)}$ via Equation (31);
 Sample $\gamma^{(t)}$ from Equation (30);
 Calculate $c^{(t)}$ and $d^{(t)}$ via Equation (33);
 Sample $\beta^{(t)}$ from Equation (32);
 Check for convergence.
End For
 Denote the last iter number as T .
- Step4:** Assume $W = \mu_M^{(T)}$, calculate the CCM via Equation (8) and the space–time adaptive optimal weight vector via Equation (1).
- Step5:** Denote the CUT as x_{vt} , the output of MCV–STAP algorithm solved by Gibbs sampling is $y = S_{vt}^H x_{vt}$.
-

While having closed–form updating equations and providing accurate estimation for parameters, the Gibbs sampler is still not attractive since it takes a large number of iterations to infer the sparse representation during the testing phase, which hinders real–time processing of the incoming CUT and motivates us to construct a VAE with fast testing, as described below.

3.3. BAMCV–STAP Algorithm

In this section, to ensure the real–time processing of MCV–STAP, we extend it into a VAE–structured method named BAMCV–STAP to map the observation x directly to the latent representation for out–of–sample prediction. Specifically, the generative model (decoder) of BAMCV–STAP is built based on MCV as shown in Figure 2a, and the inference network (encoder) is accomplished by the neural network with the structure shown in Figure 2b. Instead of needing expensive iterative inference schemes with Gibbs samplers introduced above, the variational inference network of BAMCV allows us to efficiently realize sparse reconstructions by fitting an approximate inference model to the unknown posterior using standard stochastic gradient methods. According to the theory of mean–field variational Bayes [48–50], the ELBO of BAMCV can be expressed as

$$L = -\mathbb{D}_{KL}[q(\gamma)||p(\gamma)] - \mathbb{D}_{KL}[q(w|x)||p(w|\gamma)] - \mathbb{D}_{KL}[q(\beta)||p(\beta)] + \mathbb{E}_{q(w,\gamma,\beta|x)}[p(x|w, \gamma, \beta)]. \tag{34}$$

where the approximate conditional posterior distributions $q(\gamma)$, $q(w|x)$ and $q(\beta)$ can be written as

$$\begin{aligned} q(\gamma) &= \text{Gamma}(a_h, b_h), \\ q(w|x) &= \mathcal{CN}(\mu_h, \Sigma_h), \\ q(\beta) &= \text{Gamma}(c_h, d_h). \end{aligned} \tag{35}$$

Noticeably, $\mathbb{D}_{KL}[q(\cdot)||p(\cdot)]$ denotes the Kullback–Leibler (KL) divergence between $q(\cdot)$ and $p(\cdot)$.

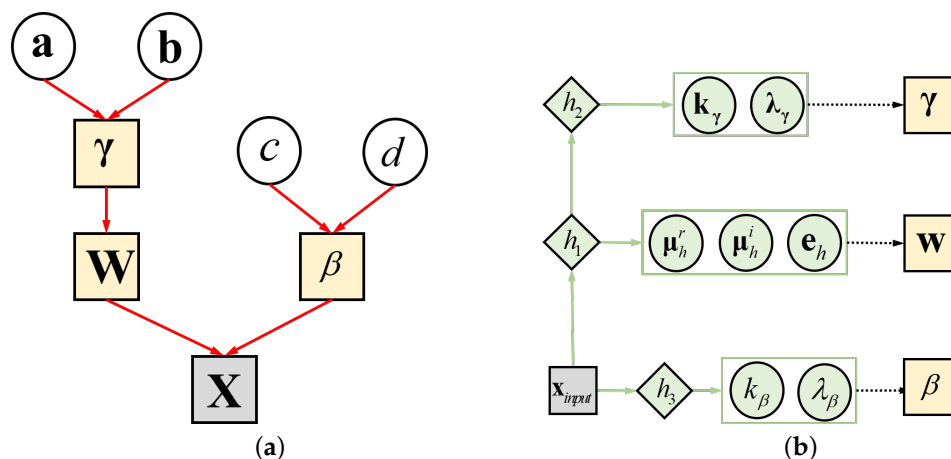


Figure 2. The structure of BAMCV. (a) Generative model of BAMCV. (b) Inference network of BAMCV. The red arrows denote the data generation. The green arrows denote the information propagation in the inference network. The black arrows with imaginary lines represent reparameterization tricks.

Different from traditional Gaussian distribution based VAEs, the gamma distribution, modeling the sparsity and satisfying non–negative constraint, is defined over γ and β here. However, it is hard to compute the gradient of the ELBO to γ and β due to the difficulty of reparameterizing gamma distributed random variables [51,52]. To reparameterize γ and β easily without deviating from gamma distributions, the Weibull distribution is considered an ideal choice to replace the gamma distribution, as demonstrated in [53]. Consequently, gamma distributed conditional posteriors, $q(\gamma)$ and $q(\beta)$, are approximated by Weibull distributions expressed as

$$q(\gamma) = \prod_{i=1}^K q(\gamma_i) = \prod_{i=1}^K \text{Weibull}(k_{\gamma_i}, \lambda_{\gamma_i}), \tag{36}$$

$$q(\beta) = \text{Weibull}(k_\beta, \lambda_\beta).$$

where k_{γ_i} and λ_{γ_i} are the parameters of γ_i , and λ_β and k_β are the parameters of γ_i . Latent variables γ_i and β can be easily reparameterized as [53,54]

$$\gamma_i = \lambda_{\gamma_i} (-\ln(1 - \varepsilon_{\gamma_i}))^{\frac{1}{k_{\gamma_i}}}, \varepsilon_{\gamma_i} \sim \text{Uniform}(0, 1), \tag{37}$$

$$\beta = \lambda_\beta (-\ln(1 - \varepsilon_\beta))^{\frac{1}{k_\beta}}, \varepsilon_\beta \sim \text{Uniform}(0, 1).$$

In addition, the Gaussian distribution based variable w can be simply reparameterized as [40]

$$w^r = \mu_h^r + \Sigma_h * z^r, z^r \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{38}$$

$$w^i = \mu_h^i + \Sigma_h * z^i, z^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

where the superscript r of all parameters represents their real part and the superscript i of all parameters represents their imaginary part. (*) denotes matrix multiplication. According to the illustration in [53], the KL divergence from the gamma distribution to the Weibull distribution, which is the first term of the ELBO in Equation (34), can be expressed analytically as

$$\mathbb{D}_{KL}[q(\gamma)||p(\gamma)] = \mathbb{D}_{KL}[\text{Weibull}(k_\gamma, \lambda_\gamma)||\text{Gamma}(a_0, b_0)]$$

$$= \sum_{i=1}^K \frac{\gamma e^{a_{i,0}}}{k_{\gamma_i}} - a_{i,0} \ln \lambda_{\gamma_i} + \ln k_{\gamma_i} + b_{i,0} \lambda_{\gamma_i} \Gamma(1 + \frac{1}{k_{\gamma_i}}) - \gamma e - 1 - a_{i,0} \ln(b_{i,0}) + \ln \Gamma(a_{i,0}). \tag{39}$$

where γ_e is the Euler–Mascheroni constant. The second term of the ELBO can be expressed analytically as

$$\begin{aligned} \mathbb{D}_{KL}[q(\mathbf{w}|\mathbf{x})||p(\mathbf{w}|\gamma)] &= \mathbb{D}_{KL}[\mathcal{CN}(\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h)||\mathcal{CN}(\mathbf{0}, \mathbf{A}^{-1})] \\ &= \sum_{i=1}^K (1 + \log \varepsilon_{kk} - |\mu_{h,i}|^2 - \varepsilon_{kk}). \end{aligned} \tag{40}$$

ε_{kk} means the k -th diagonal element of $\boldsymbol{\Sigma}_h$, and $\mu_{h,i}$ denotes the i -th element of $\boldsymbol{\mu}_h$. $|\cdot|$ means the modulus operation. The third term and the last term of the ELBO can be expressed analytically as

$$\begin{aligned} \mathbb{D}_{KL}[q(\beta)||p(\beta)] &= \mathbb{D}_{KL}[\text{Weibull}(k_\beta, \lambda_\beta)||\text{Gamma}(c_0, d_0)] \\ &= \frac{\gamma_e c_0}{k_\beta} - c_0 \ln \lambda_\beta + \ln k_\beta + d_0 \lambda_\beta \Gamma\left(1 + \frac{1}{k_\beta}\right) - \gamma_e - 1 - c_0 \ln(d_0) + \ln \Gamma(c_0). \end{aligned} \tag{41}$$

$$\mathbb{E}_{q(\mathbf{w}, \gamma, \beta|\mathbf{x})}[p(\mathbf{x}|\mathbf{w}, \gamma, \beta)] = \|\mathbf{x} - \Phi \mathbf{w}\|_2^2 \tag{42}$$

As illustrated in Figure 2a, all the parameters relative to conditional distributions are transformed from the observation \mathbf{x}_{input} with the neural network. Specifically,

$$\begin{aligned} \boldsymbol{\mu}_h^r &= \text{ReLU}(\mathbf{C}_\mu^r \mathbf{h}_1 + \mathbf{g}_\mu^r), \\ \boldsymbol{\mu}_h^i &= \text{ReLU}(\mathbf{C}_\mu^i \mathbf{h}_1 + \mathbf{g}_\mu^i), \\ \boldsymbol{\mu}_h &= \boldsymbol{\mu}_h^r + j \boldsymbol{\mu}_h^i, \end{aligned} \tag{43}$$

where $\boldsymbol{\mu}_h$ is the mean vector of \mathbf{w} ,

$$\mathbf{e}_h = \text{ReLU}(\mathbf{C}_e \mathbf{h}_1 + \mathbf{g}_e) \tag{44}$$

where $\mathbf{e}_h = \text{diag}(\boldsymbol{\Sigma}_h)$, and

$$\mathbf{h}_1 = \text{ReLU}(\mathbf{C}_{1h} \mathbf{x}_{input} + \mathbf{g}_{1h}) \tag{45}$$

where $\text{ReLU}(\cdot)$ denotes the non-linear activation function. Note that \mathbf{x}_{input} denotes the concatenation of the real part and the imaginary part of training data, as \mathbf{x} and latent representation \mathbf{w} are complex-valued. Similarly,

$$\begin{aligned} k_\gamma &= \text{Softplus}(\mathbf{C}_a \mathbf{h}_2 + \mathbf{g}_a), \lambda_\gamma = \text{Softplus}(\mathbf{C}_b \mathbf{h}_2 + \mathbf{g}_b) \\ k_\beta &= \text{Softplus}(\mathbf{C}_c \mathbf{h}_\beta + \mathbf{g}_c), \lambda_\beta = \text{Softplus}(\mathbf{C}_d \mathbf{h}_\beta + \mathbf{g}_d) \end{aligned} \tag{46}$$

$$\begin{aligned} \mathbf{h}_2 &= \text{ReLU}(\mathbf{C}_{2h} \mathbf{h}_1 + \mathbf{g}_{2h}) \\ \mathbf{h}_\beta &= \text{ReLU}(\mathbf{C}_\beta \mathbf{x}_{input} + \mathbf{g}_\beta) \end{aligned} \tag{47}$$

where $\mathbf{k}_\gamma = [k_{\gamma_1}, k_{\gamma_2}, \dots, k_{\gamma_K}] \in \mathbb{R}^{K \times 1}$, $\boldsymbol{\lambda}_\gamma = [\lambda_{\gamma_1}, \lambda_{\gamma_2}, \dots, \lambda_{\gamma_K}] \in \mathbb{R}^{K \times 1}$ and $\text{Softplus}(\cdot)$ applies $\log[1 + \exp(\cdot)]$ non-linearity to each element to ensure positive Weibull shape and scale parameters.

Obviously, $\mathbf{C}_\mu^r, \mathbf{C}_\mu^i, \mathbf{C}_e, \mathbf{C}_{1h}, \mathbf{g}_\mu^r, \mathbf{g}_\mu^i, \mathbf{g}_e, \mathbf{g}_{1h}, \mathbf{C}_a, \mathbf{C}_b, \mathbf{C}_c, \mathbf{C}_d, \mathbf{C}_{2h}, \mathbf{C}_\beta, \mathbf{g}_a, \mathbf{g}_b, \mathbf{g}_c, \mathbf{g}_d, \mathbf{g}_{2h}$ and \mathbf{g}_β are parameters in the neural network. The superscript r of all parameters represents the real part, and the superscript i of all parameters represents the imaginary part. For clarity, the dimensions of parameters in the inference network are summarized in Table 1.

Above all, due to the simple reparameterization tricks, our proposed VAE-structured MCV-STAP methods can obtain the latent representations directly with the aid of an autoencoding inference network.

Table 1. Parameters in the inference network.

Parameters	Dimensions	Parameters	Dimensions	Parameters	Dimensions
C_{μ}^r	$K \times K$	g_{1h}	$2MN \times 1$	g_a	$K \times 1$
C_{μ}^i	$K \times K$	C_a	$K \times K$	g_b	$K \times 1$
C_e	$K \times K$	C_b	$K \times K$	g_c	1×1
C_{1h}	$2MN \times K$	C_c	$1 \times 2MN$	g_d	1×1
g_{μ}^r	$K \times 1$	C_d	$1 \times 2MN$	g_{2h}	$K \times 1$
g_{μ}^i	$K \times 1$	C_{2h}	$K \times K$	g_{β}	$2MN \times 1$
g_e	$K \times 1$	C_{β}	$2MN \times 2MN$		

Moreover, to make the proposed method further attractive in real-time processing, we pre-train the inference network off-line with simulated data preparing for sequential fine-tuning facing with realistic clutter data to be processed [55]. The main steps of the proposed BAMCV for STAP (BAMCV-STAP) algorithm are summarized in Algorithm 2. For clarity, the flow chart of BAMCV is plotted in Figure 3.

Algorithm 2 BAMCV-STAP algorithm.

- Step1:** Simulate data for pre-training using radar system parameters of realistic clutter data under test. Denote the simulated dataset as D_{pre} and the realistic dataset as D_{test} .
- Step2:** Pre-train the inference network with dataset D_{pre} off-line until the ELBO converges.
- Step3:** Select CUT in D_{test} and choose L training samples around CUT.
- Step4:** Fine-tune the inference network with the selected L training samples until the ELBO converges again. Assume $W = \mu_h$.
- Step6:** Estimate the CCM via Equation (8) and the space-time adaptive optimal weight vector via Equation (2).
Denote the weight vector as S_{ot} .
- Step7:** Denote data in CUT as x_{cut} , the output of BAMCV-STAP solved by the inference network is $y = S_{ot}^H x_{cut}$.

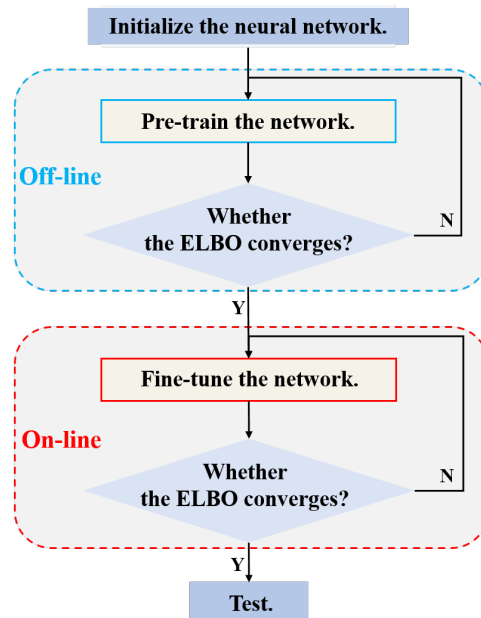


Figure 3. Flow chart of BAMCV-STAP.

Compared with MCV, BAMCV has two explicit advantages which can be summarized as follows:

- Decreasing the computational loads. Inspired by the gradient ascent scheme for parameters optimization, the parameters of BAMCV are updated by the backpropagation

of the gradient, which only involves some linear operations, exponential operations and logarithmic operations, instead of involving complex inverse operations and multiple samplings appeared in MCV.

- Improving the convergence rate for testing. As the inference network is pre-trained off-line, only a few iterations are taken in BAMCV to obtain the recovery results of observations rather than a large number of time-consuming iterations per testing in MCV.

Experimental results shown in the following section will verify these advantages.

4. Numerical Results

In this section, we evaluate the proposed algorithms, MCV-STAP and BAMCV-STAP, using both simulated data and measured data in terms of clutter suppression performance, time efficiency and computational complexity. Significantly, the metric to evaluate clutter suppression performance for the simulated data is the signal to interference plus noise ratio (SINR) loss [45], which is defined as follows

$$L_{\text{SINR}} = \frac{\sigma_n^2}{MN} \frac{|\mathbf{S}_w^H \mathbf{v}_t|^2}{\mathbf{S}_w^H \mathbf{R} \mathbf{S}_w} \quad (48)$$

where \mathbf{S}_w denotes the STAP weight vector, \mathbf{R} denotes the true CCM, \mathbf{v}_t denotes the target steering vector and σ_n^2 denotes the actual noise power. In addition, as the true CCM is unable to be obtained for the measured data, we evaluate the target detection performance for targets with normalized Doppler frequency of 0.1 by the probability of detection (PD) versus signal to noise ratio (SNR) curves, which are achieved by utilizing the adaptive matched filter (AMF) detector [56] and averaged over 10^5 Monte Carlo trials. The probability of the false alarm rate (PFA) is set as 10^{-6} , the PD is estimated based on 10^6 randomly sampled CUTs and the PD to SNR curves are averaged over 10^5 Monte Carlo trials. SNR is defined as the ratio of signal power (for a single pulse per antenna element) to the rough estimate of noise power (the average energy of echoes within the support region) here. In addition, to evaluate the time efficiency, we evaluate the time spent on the convergence rate per testing. The computational complexity is also evaluated by the comparison of computational loads per testing. For clarity, the performance metrics used in this paper are summarized in Table 2, in which the check mark means the metric is used in the data and the blank means not.

Table 2. Summary of Performance Metrics.

	Performance Metric	Simulated Data	Measured Data
Clutter Suppression	SINR loss	✓	
	PD		✓
Real-time Processing	Convergence rate	✓	✓
	Computational loads	✓	✓

Moreover, the proposed algorithm is compared with some other classical methods especially for the MMV case, such as loading sample matrix inversion (LSMI) for STAP [57,58], multiple orthogonal matching pursuit (M-OMP) for STAP [59], M-SBL for STAP and multiple fast converging SBL (M-FCSBL) for STAP [34]. The LSMI for STAP [57,58], which is the representative of classical STAP algorithm, is an adaptive algorithm imposing diagonal loading (DL) to improve the robustness of conventional Capon beamformers but fails to achieve satisfactory performance unless the number of training samples is more than twice the clutter rank. The M-OMP for STAP, orthogonal matching pursuit (OMP) [25] for the MMV case, obtains sparse representations through a greedy algorithm but is particularly unfriendly to the highly coherent dictionary. As the classical technique to solve sparse recovery issues utilizing the greedy strategy, M-OMP is

chosen as one of comparison methods in this paper. The M–SBL for STAP, suppressing the clutter imposing M–SBL algorithm based on Bayesian theory as well as ours, receives much attention in the field of STAP for the excellent performance with a highly coherent dictionary but potentially requires a large number of iterations per testing leading to unsuccessful real–time processing. The M–FCSBL for STAP, a fast converging M–SBL algorithm, improves the convergence by incorporating a simple approximation term and achieves a favorable recovery result with only a few iterations. However, the experimental results reveal that this approximation term leads to sub–optimal recovery results, as shown in its convergence curve.

Furthermore, the elements of all weight matrices in the neural network employed in this paper are initialized with Gaussian distributions with standard deviations set to 0.1, and all bias terms are set to zero. During the pre–training stage, we set the mini–batch size to 32. For optimization, the Adam optimizer [60] with a learning rate of 10^{-4} is used. On a Pentium PC with a 3.7 GHz CPU and 64 GB RAM, the inference network is pre–trained and fine–tuning in non–optimized Python software before being moved to MATLAB for testing. The amount of samples in the pre–training data set is ten times the number of parameters in the neural network to ensure the generalization of the pre–trained neural network. The settings of each layer in the neural network have been stated in Table 1.

4.1. Simulated Data

In this paper, a side–looking airborne phased array radar with a uniformly linear antenna (ULA), whose inter–element spacing is half wavelength, is considered. The parameters of radar system are shown in Table 3. In the simulations, we set $N_s = 4M$, $N_t = 4N$, $\sigma_n^2 = 1$ and 180 clutter patches uniformly distributed from $-\pi/2$ to $\pi/2$. According to Algorithm 2, the data set D_{pre} and D_{test} for BAMCV–STAP are simulated with the same radar system parameters, respectively. Obviously, MCV–STAP is executed on D_{test} directly without need of D_{pre} . To make the inference network easily generalized to the training samples of CUT, We simulated as many training samples as ten times the number of parameters with random clutter to signal ratio (CNR) sampled from 30 dB to 70 dB in D_{pre} . As for D_{test} , to investigate the performance of all methods under the heterogeneous clutter scenario conveniently, we assume the 150–th range cell as CUT which is located at the interested area with the average CNR = 45 dB and around by only 10 training samples satisfying the i.i.d constraint, as shown in Figure 4. Compared with the clairvoyant spectrum calculated by known CCM in Figure 5a, the clutter spectrum estimated by 128 training samples, as many as twice the DOF, exhibits too large value along the clutter ridge due to the heterogeneity of samples, as depicted in Figure 5b. Noticeably, the clutter suppression performance with the different number of i.i.d training samples in the interested area will also be investigated later.

Table 3. Radar System Parameters of Simulated data.

Parameters	Value	Parameters	Value
Carrier frequency (Hz)	1.25 G	Platform velocity (m/s)	125
Bandwidth (Hz)	2.5 M	Platform height (m)	6000
Mainbeam azimuth ($^\circ$)	0	Pulse number in one CPI	8
Mainbeam elevation ($^\circ$)	0	Antenna elements number	8
Pulse repetition frequency (Hz)	2000	Range cell number	400

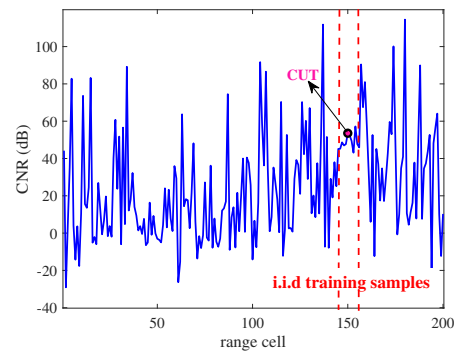


Figure 4. CNR versus range cell of D_{test} .

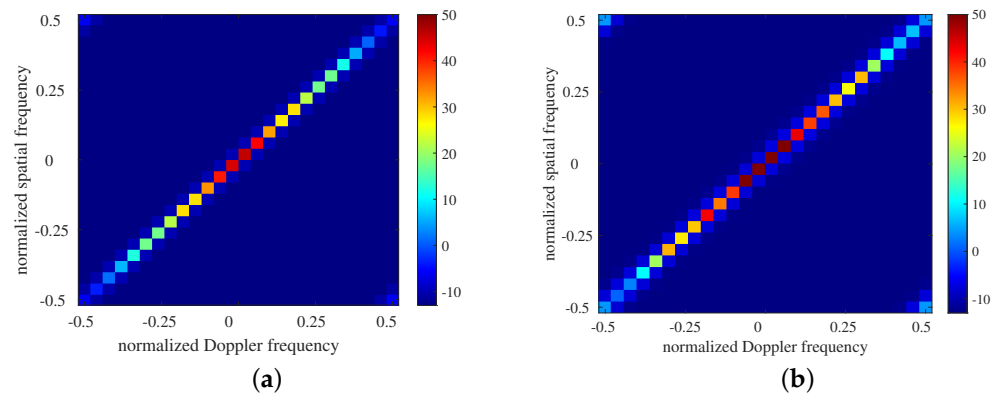


Figure 5. Estimated angle–Doppler clutter spectrum after pre–training. (a) Clairvoyant spectrum calculated by known CCM. (b) Estimated clutter spectrum using 128 training samples.

4.1.1. Analysis of Clutter Suppression Performance

The clutter suppression performance of the proposed method is evaluated here. The clutter spectrum of clutter data under test utilizing various approaches are depicted in Figure 6. The number of training samples is fixed to 10 since the number of i.i.d training samples is set as 10 according to Figure 4. As is customary in LSMI, the load factor is set at 1. The clutter spectrum computed by MCV in Figure 6f and BAMCV in Figure 6h is the closest to the ideal one in Figure 6a in terms of both position and value. As the posterior distributions are strictly derived via Bayesian theory, the CCM computed using Gibbs sampling is as close to the clairvoyant one as BAMCV–STAP. Because of limited i.i.d. training samples, the approximated spectrum by M–OMP in Figure 6c and LSMI in Figure 6b expose their powerlessness in clutter subspace estimation. Instead, the clutter spectrum obtained by M–SBL, as shown in Figure 6d, is extremely close to the ideal one, with a small spreading and a few noise peaks due to imprecise noise power assessment. In Figure 6e, M–FCSBL [34] achieves an accurate clutter subspace and performs no noise peaks. However, upon further investigation of these SR–STAP approaches, as shown in Figure 7, we discover that values along the clutter ridge in the clutter spectrum calculated by M–FCSBL are not as well as expected. The clutter ridge formed by our method, on the other hand, is closer to the ideal result.

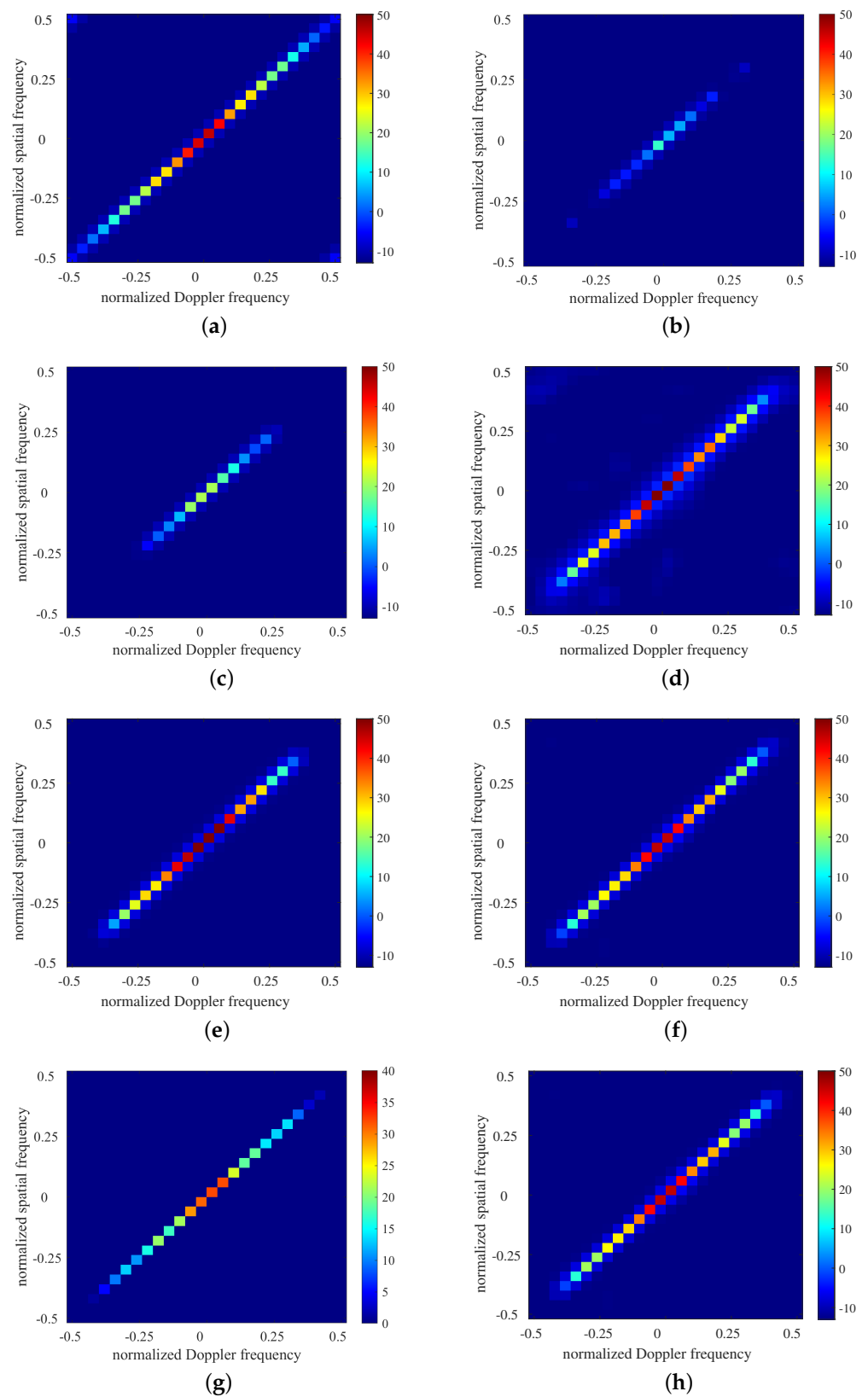


Figure 6. Estimated angle–Doppler clutter spectrum using different methods. (a) Clairvoyant angle–Doppler clutter spectrum. (b) LSML. (c) M–OMP. (d) M–SBL. (e) M–FCSBL. (f) Clutter spectrum via MCV. (g) BAMCV after pre–training. (h) BAMCV after fine–tuning.

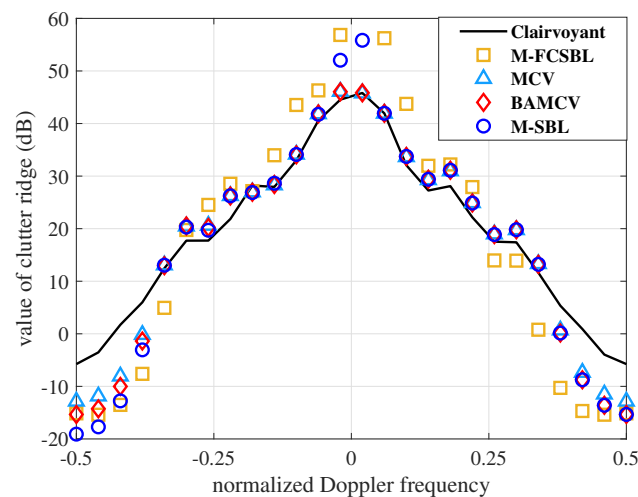


Figure 7. Analysis of clutter ridge estimated by different methods.

Moreover, to demonstrate the importance of the pre-training to the inference network, we utilize the inference network after pre-training to recover the space-time profile of clutter data under test, as shown in Figure 6g. Although the significant values along the clutter ridge are not accurate enough, the sparsity of the clutter spectrum and the shape of the clutter ridge are initialized successfully as the preparation for sequential fine-tuning. Apart from the sparse subspace, the noise power estimation is also an important issue for the CCM, according to Equation (8). Consequently, we enumerate the final results estimated by all comparison methods in Table 4. Clearly, our methods outperform M-SBL and M-FCSBL, proving the robustness to the estimation for noise power.

Table 4. Noise power estimation.

True Noise Power	M-SBL	M-FCSBL	MCV	BAMCV
0.01	2.49×10^{-10}	0.0111	0.00973	0.0101
0.1	4.15×10^{-9}	0.1197	0.1061	0.1103
1	7.92×10^{-8}	1.0117	0.9937	0.9925
5	2.98×10^{-6}	4.7692	4.9793	4.9001
10	7.57×10^{-6}	11.1382	9.9992	9.9856

When the number of training samples is 10, the clutter suppression performance measured by SINR loss is evaluated in Figure 8. The optimal SINR loss curve is completed using the clairvoyant CCM. In comparison to previous Bayesian SR-STAP approaches, LSMI and M-OMP display severe performance loss in the sidelobe region, as depicted in Figure 8a. Specifically, the output SINR loss of MCV-STAP is the closest to the optimal one among various Bayesian SR-STAP methods, as shown in Figure 8b (which is an enlargement of Figure 8a). BAMCV-STAP achieves the lower SINR loss than MCV-STAP as shown in Figure 8 but still surpasses other comparison methods.

Besides, we compare the effect of reducing the i.i.d training samples on the SINR loss of each method, and the results are shown in Figure 9. The average SINR loss is defined in this section as the mean of the SINR loss values across the entire normalized Doppler frequency range. According to Figure 9, on average sense, when L is a large value, all the methods have almost equivalent performance. But with an decrease in L , the Bayesian type SR-STAP methods, including M-SBL-STAP, MCV-STAP and BAMCV-STAP, have the larger SINR loss. Notably, our methods, both MCV-STAP and BAMCV-STAP, maintain stable performance as the number of training samples decreases and achieve a 3 dB higher SINR loss than M-SBL-STAP with only two training samples, demonstrating the robustness of the proposed methods to the number of training samples.

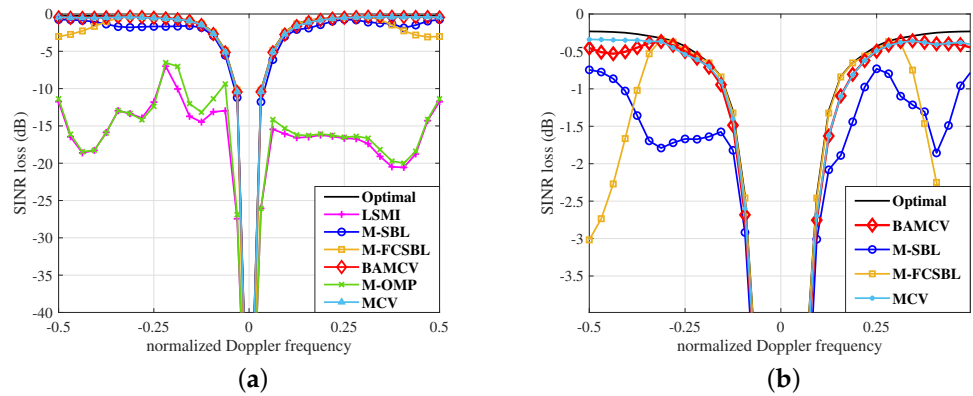


Figure 8. SINR loss curves. (a) SINR loss curves of all comparison methods. (b) SINR loss curves of Bayesian STAP methods.

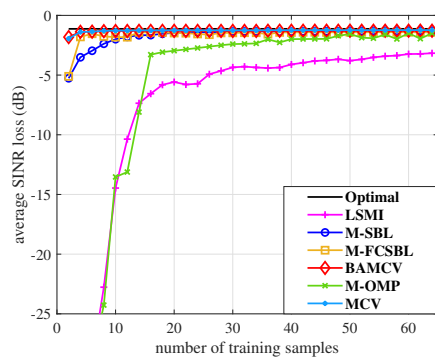


Figure 9. SINR loss versus the number of training samples.

4.1.2. Analysis of Computational Loads and Convergence Rate

To study the real-time processing of several SR-STAP algorithms with similarly superior performance i.e., M-SBL, M-FCSBL and our methods, we detail the analysis of the convergence rate and computational loads in Figure 10.

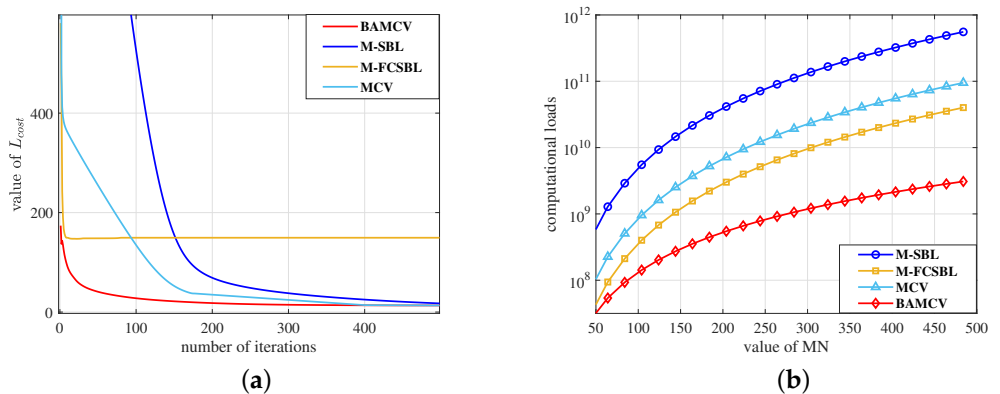


Figure 10. Analysis of convergence and computational loads. (a) Convergence curves of different methods using simulated data. (b) Comparison of computational loads.

For the on-line phase, convergence curves, utilizing the negative log-likelihood as the unified loss function [34] for fair comparison, are shown in Figure 10a, i.e.,

$$L_{cost} = \ln|M| + \text{tr}(M^{-1}R_{cost}) \tag{49}$$

where

$$\mathbf{M} = \Phi^H \mathbf{A}^{-1} \Phi + \beta^{-1} \mathbf{I}$$

$$\mathbf{R}_{\text{cost}} = \frac{1}{L} \mathbf{X} \mathbf{X}^H.$$

Different from MCV converging as slowly as M-SBL, BAMCV, which is pre-trained with lots of simulated data off-line, exhibits the dominant superiority at the beginning of the iteration. Then, after fine-tuning with several snapshots around CUT over a few iterations, BAMCV converges to a stable and optimal state. The time spent on each iteration of each method is also listed in Table 5. Indeed, since it uses a gradient ascent scheme without any inverse operation or sampling, BAMCV takes much less time per iteration than the other three approaches. However, M-SBL and MCV require not only a large number of iterations but also a long time for each iteration. Furthermore, even though M-FCSBL quickly converges to its steady-state value, the final solution is not optimal, which is exactly coincident with the estimated clutter spectrum in Figures 6e and 7.

Table 5. Comparison of running time per iteration using simulated data.

Approach	Running Time Per Iteration (s)
M-SBL	0.125
M-FCSBL	0.055
MCV	0.103
BAMCV	0.011

In addition, the computational loads of M-SBL, M-FCSBL, MCV and BAMCV are detailed in Table 6. Specifically, N_s is the iteration number of the M-SBL algorithm, N_f is the iteration number of the M-FCSBL algorithm and N_g is the iteration number of MCV for our method. Due to the intractable inverse operations and the large number of iterations, the computational loads of M-SBL and M-FCSBL, as well as MCV are about three times of the DOF. By contrast, since the time-consuming training process of neural networks is finished before testing, BAMCV achieves fast super-resolution by directly mapping out sparse representation vectors of training samples with only twice of the DOF. To provide an explicit illustration, we also display the curve of computational loads versus DOF in Figure 10b with setting $K = 16MN$, $L = 10$ and $N_s = N_f = N_g = 1$. Obviously, the computational complexity of BAMCV is much lower than that of other methods, and the gap between BAMCV and other methods diverges as DOF increasing.

Table 6. Comparison of computational cost.

Approach	Computational Loads
M-SBL	$((K)^3 + (MN)^3 + 3K^2MN + [2(MN)^2 + 2MNL + L + 1]K + MNL + 1)N_s$
M-FCSBL	$((MN)^3 + K^2MN + 6K(MN)^2 + (2KL + 2K)MN + 1)N_f$
MCV	$((MN)^3 + 3K^2MN + 4K(MN)^2 + (L^2 + L + 1)MNK + MNL^2 + MNL + 1)N_g$
BAMCV	$2KMNL + 5K^2L + 4(MN)^2L + 4MNL$

In summary, the modified method BAMCV-STAP not only requires less online running time and lower computational loads than other comparison methods while achieving comparable clutter suppression performance, but also remains robust to the number of training samples and noise power. Consequently, BAMCV-STAP is suitable for airborne early warning (AEW) radar applications.

4.2. Measured Data

In this subsection, MCV-STAP and BAMCV-STAP are applied to the public available data set, the Multi-Channel Airborne Radar Measurements (MCARM) data set [61], to verify their attractiveness. Some indispensable parameters of the radar system are listed

in Table 7. The clutter spectrum estimated using 10 training samples around the selected target range cell by different methods is exhibited in Figure 11. Consistent with simulated results, limited by the number of training samples, the clutter spectrum estimated by the LSMI method in Figure 11a is as rough as the one by the M–OMP method in Figure 11b. Due to the approximation term, the clutter spectrum estimated by M–FCSBL in Figure 11d method is too sparse to reconstruct the information in the sidelobe region. M–SBL exhibits a better capability of recovering the clutter spectrum using only 10 training samples but with a slight broadening of the clutter ridge as shown in Figure 11c.

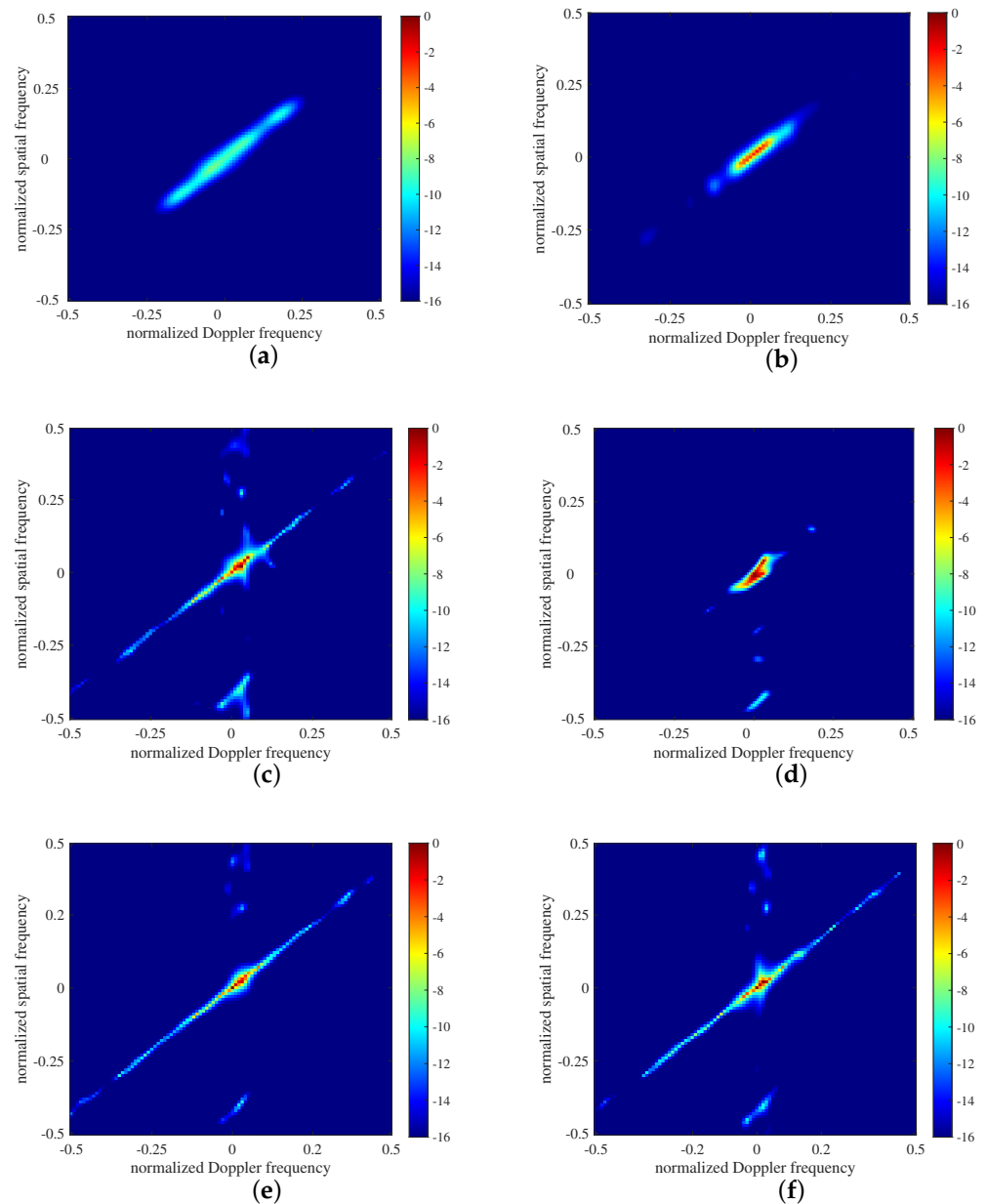


Figure 11. Estimated clutter spectrum of MCARM data. (a) LSMI. (b) M–OMP. (c) M–SBL. (d) M–FCSBL. (e) MCV. (f) BAMCV.

Table 7. Radar System Parameters of MCARM data.

Parameters	Value	Parameters	Value
Pulse repetition frequency (Hz)	1984	Antenna array spacing of azimuth (m)	0.1029
Wavelength (m)	0.24	Antenna array spacing of elevation (m)	0.5629
Pulse number in one CPI	128	Platform height (m)	10,188
Antenna elements number of azimuth	11	Range cell number	400
Antenna elements number of elevation	2		

As it is difficult to judge whether our methods or M–SBL achieve the best performance according to the clutter spectrum estimated shown in Figure 11c,e,f, we explore the detection performance averaged over all range cells for the targets with the normalized Doppler frequency of 0.1 (corresponding velocity is 10 m/s) by the PD versus SNR curves as depicted in Figure 12a. The probability of the false alarm rate (PFA) is set as 10^{-6} , the PD to SNR for each CUT is averaged over 10^5 Monte Carlo trials. To further explore the detection performance of targets with different velocity values, we exhibit the PD versus velocity curves (SNR = 20 dB), which are averaged over 10^5 Monte Carlo trials, in Figure 12b.

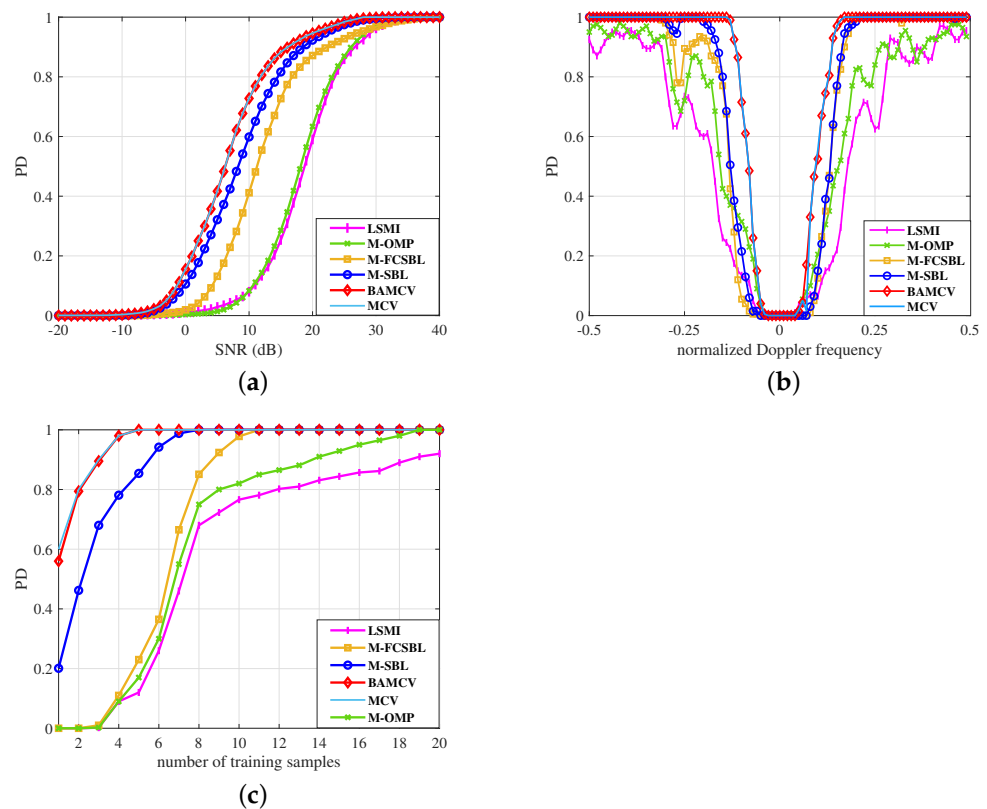


Figure 12. Average detection results of measured data. (a) PD versus SNR ($f_d = 0.1$, $L = 10$). (b) PD versus normalized Doppler frequency (SNR = 20 dB, $L = 10$). (c) PD versus the number of training samples ($f_d = 0.1$, SNR = 20 dB). f_d denotes the normalized Doppler frequency of targets. L is the number of training samples.

Moreover, the detection performance to explore the robustness to the number of training samples is provided in Figure 12c. As shown in Figure 12, similar to the LSMI method, M–FCSBL suffers a severe performance degradation as the information in the sidelobe region is ultimately given up due to the approximation term, leading to the too sparse reconstructions to obtain the accurate CCM. M–SBL method achieves sub–optimal performance among comparison methods but with very slow convergence, as shown in Figure 13 and Table 8. On the contrary, BAMCV–STAP achieves the sub–optimal target

detection performance. Besides that, according to the computational loads analyzed in Figure 10b and Table 6, the computational loads of BAMCV is about $\mathcal{O}(MN)$ but the other methods are about $\mathcal{O}((MN)^3)$. Consequently, BAMCV–STAP can achieve favorable detection of targets and keep robust to the number of training samples, with the most efficient convergence and the lowest computational loads among comparison methods.

Table 8. Comparison of running time per iteration using MCARM data.

Approach	Running Time Per Iteration (s)
M–SBL	1.102
M–FCSBL	0.333
MCV	0.951
BAMCV	0.025

To sum up, among these comparison approaches, our method exhibits its superiority in recovering the clutter spectrum in terms of both sparsity and accuracy on measured data, which is consistent with the results on simulated data.

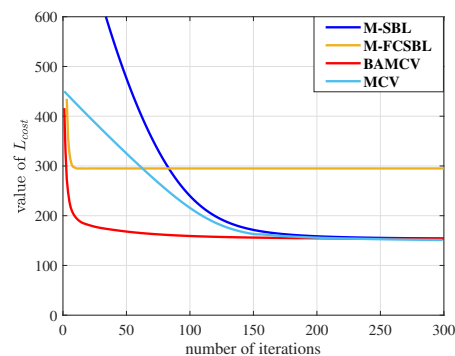


Figure 13. Convergence curves of different methods using measured data.

5. Discussion

Along with the impressive performance achieved by our methods, the above–mentioned experimental results indicate that it is substantially more challenging for traditional SR–STAP algorithms, such as the proposed MCV–STAP, to ensure both the stable performance of sparse reconstruction and real–time processing. However, deep learning techniques, such as the proposed BAMCV–STAP, can efficiently realize parameter estimates for real–time on–line processing at the expense of nonsignificant off–line pre–training.

6. Conclusions

Traditional SR–STAP algorithms are unattractive as they only provide point estimates for several parameters and rely on a large number of iterations per testing at the testing stage. To avoid the iterative procedure, we first propose MCV–STAP, a parameter–free SR–STAP method that incorporates a modified VRVM into the field of STAP, to model the space–time echoes with a completely robust Bayesian framework. Then, to realize the real–time processing of the incoming space–time echoes, we introduce BAMCV–STAP, a VAE–structured method that builds its encoder from MCV but with an efficient inference network instead of the time–consuming Gibbs sampler in MCV–STAP. Compared with the conventional Bayesian type SR–STAP algorithms, the inference network of BAMCV–STAP can be pre–trained off–line using a gradient ascent scheme, enabling more efficient sparse reconstruction. The experimental results show that the BAMCV–STAP algorithm can achieve comparable clutter suppression with lower computational cost.

This paper focuses on the suppression of non–homogeneous clutter environments with side–looking airborne phased–array radar by means of deep learning methods. Inspired by their attractive characteristics, approaches that can solve the other tricky

problems of radar signal processing, such as the suppression of the non-stationary clutter and the detection of slow-moving targets, are the promising directions that we plan to explore in the future.

Author Contributions: Conceptualization, P.W. and B.C.; methodology, C.Z.; software, H.Z. and C.Z.; validation, H.Z. and C.Z.; formal analysis, H.Z. and C.Z.; investigation, C.Z.; resources, P.W.; data curation, C.J.; writing—original draft preparation, C.Z.; writing—review and editing, C.Z. and W.C.; visualization, C.Z. and W.C.; supervision, C.Z. and W.C.; project administration, B.C. and H.L.; funding acquisition, B.C. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation Of China, Grant No. 61771361.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, X.; Cheng, Y.; Wu, H.; Wang, H. Heterogeneous clutter suppression for airborne radar STAP based on matrix manifolds. *Remote Sens.* **2021**, *13*, 3195. [[CrossRef](#)]
2. Li, H.; Liao, G.; Xu, J.; Zeng, C. Sub-CPI STAP based clutter suppression and target refocusing with airborne radar system. *Digit. Signal Process.* **2022**, *123*, 103418. [[CrossRef](#)]
3. Liu, K.; Wang, T.; Wu, J.; Chen, J. A Two-Stage STAP Method Based on Fine Doppler Localization and Sparse Bayesian Learning in the Presence of Arbitrary Array Errors. *Sensors* **2022**, *22*, 77. [[CrossRef](#)] [[PubMed](#)]
4. Xiao, H.; Wang, T.; Zhang, S.; Wen, C. A robust refined training sample reweighting space-time adaptive processing method for airborne radar in heterogeneous environment. *IET Radar Sonar Nav.* **2021**, *15*, 310–322. [[CrossRef](#)]
5. Yifeng, W.; Tong, W.; Jianxin, W.; Jia, D. Robust training samples selection algorithm based on spectral similarity for space-time adaptive processing in heterogeneous interference environments. *IET Radar Sonar Nav.* **2015**, *9*, 778–782. [[CrossRef](#)]
6. Liu, J.; Liu, W.; Liu, H. A simpler proof of rapid convergence rate in adaptive arrays. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 135–136. [[CrossRef](#)]
7. Zhang, W.; He, Z.; Li, J.; Liu, H.; Sun, Y. A method for finding best channels in beam-space post-Doppler reduced-dimension STAP. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 254–264. [[CrossRef](#)]
8. Zhang, W.; He, Z.; Li, J.; Li, C. Beamspace reduced-dimension space-time adaptive processing for multiple-input multiple-output radar based on maximum cross-correlation energy. *IET Radar Sonar Nav.* **2015**, *9*, 772–777. [[CrossRef](#)]
9. Wang, X.; Yang, Z.; Huang, J.; de Lamare, R.C. Robust two-stage reduced-dimension sparsity-aware STAP for airborne radar With coprime arrays. *IEEE Trans. Signal Process.* **2020**, *68*, 81–96. [[CrossRef](#)]
10. Zhang, W.; An, R.; He, N.; He, Z.; Li, H. Reduced dimension STAP based on sparse recovery in heterogeneous clutter environments. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 785–795. [[CrossRef](#)]
11. Guerci, J.R.; Goldstein, J.S.; Reed, I.S. Optimal and adaptive reduced-rank STAP. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 647–663. [[CrossRef](#)]
12. Fa, R.; de Lamare, R.C. Reduced-Rank STAP Algorithms using Joint Iterative Optimization of Filters. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 1668–1684. [[CrossRef](#)]
13. Yang, Z.; Wang, X. Reduced-rank space-time adaptive processing algorithm based on multistage selections of angle-Doppler filters. *IET Radar Sonar Nav.* **2022**, *16*, 327–345. [[CrossRef](#)]
14. Sarkar, T.K.; Wang, H.; Park, S.; Adve, R.; Koh, J.; Kim, K.; Zhang, Y.; Wicks, M.C.; Brown, R.D. A deterministic least-squares approach to space-time adaptive processing (STAP). *IEEE Trans. Aerosp. Electron. Syst.* **2001**, *49*, 91–103. [[CrossRef](#)]
15. Guerci, J.R.; Baranoski, E.J. Knowledge-aided adaptive radar at DARPA: An overview. *IEEE Signal Process. Mag.* **2006**, *23*, 41–50. [[CrossRef](#)]
16. Capraro, C.T.; Capraro, G.T.; Bradaric, I.; Weiner, D.D.; Wicks, M.C.; Baldygo, W.J. Implementing digital terrain data in knowledge-aided space-time adaptive processing. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 1080–1099. [[CrossRef](#)]
17. Cui, N.; Xing, K.; Duan, K.; Yu, Z. Knowledge-aided block sparse Bayesian learning STAP for phased-array MIMO airborne radar. *IET Radar Sonar Navig.* **2021**, *15*, 1628–1642. [[CrossRef](#)]
18. Liu, M.; Zou, L.; Yu, X.; Zhou, Y.; Wang, X.; Tang, B. Knowledge Aided Covariance Matrix Estimation via Gaussian Kernel Function for Airborne SR-STAP. *IEEE Access* **2020**, *8*, 5970–5978. [[CrossRef](#)]
19. Zhu, X.; Li, J.; Stoica, P. Knowledge-aided space-time adaptive processing. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 1325–1336. [[CrossRef](#)]
20. Yang, Z.; de Lamare, R.C.; Li, X. L1 regularized STAP algorithms With a generalized sidelobe canceler architecture for airborne radar. *IEEE Trans. Signal Process.* **2012**, *60*, 674–686. [[CrossRef](#)]

21. Sun, K.; Zhang, H.; Li, G.; Meng, H.; Wang, X. A Novel STAP Algorithm using Sparse Recovery Technique. In Proceedings of the IEEE International Geoscience & Remote Sensing Symposium (IGARSS 2009), Cape Town, South Africa, 12–17 July 2009; pp. 336–339.
22. Wu, Q.; Zhang, Y.D.; Amin, M.G.; Himed, B. Space-Time Adaptive Processing and Motion Parameter Estimation in Multistatic Passive Radar Using Sparse Bayesian Learning. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 944–957. [[CrossRef](#)]
23. Yang, X.; Sun, Y.; Zeng, T.; Long, T.; Sarkar, T.K. Fast STAP Method Based on PAST with Sparse Constraint for Airborne Phased Array Radar. *IEEE Trans. Signal Process.* **2016**, *64*, 4550–4561. [[CrossRef](#)]
24. Mallat, S.; Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **1993**, *41*, 3397–3415. [[CrossRef](#)]
25. Pati, Y.; Rezaifar, R.; Krishnaprasad, P. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1–3 November 1993; pp. 40–44.
26. Cotter, S.F.; Rao, B.D.; Engan, K.; Kreutz-Delgado, K. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Trans. Signal Process.* **2005**, *53*, 2477–2488. [[CrossRef](#)]
27. Blumensath, T.; Davies, M.E. Gradient Pursuits. *IEEE Trans. Signal Process.* **2008**, *56*, 2370–2382. [[CrossRef](#)]
28. Bruckstein, A.M.; Donoho, D.L.; Elad, M. From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images. *SIAM Rev.* **2009**, *51*, 34–81. [[CrossRef](#)]
29. Tibshirani, R. Regression shrinkage and selection via the lasso: A retrospective. *J. R. Stat. Soc. Series B Stat. Methodol.* **2011**, *73*, 273–282. [[CrossRef](#)]
30. Zibulevsky, M.; Elad, M. L1-L2 Optimization in Signal and Image Processing. *IEEE Signal Process. Mag.* **2010**, *27*, 76–88. [[CrossRef](#)]
31. Chen, S.S.; Donoho, D.L.; Saunders, M.A. Atomic Decomposition by Basis Pursuit. *SIAM Rev.* **2001**, *43*, 129–159. [[CrossRef](#)]
32. Wei, S.; Zhang, L.; Ma, H.; Liu, H. Sparse Frequency Waveform Optimization for High-Resolution ISAR Imaging. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 546–566. [[CrossRef](#)]
33. Wu, Q.; Zhang, Y.D.; Amin, M.G.; Himed, B. Complex multitask Bayesian compressive sensing. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014), Florence, Italy, 4–9 May 2014; pp. 3375–3379.
34. Wang, Z.; Xie, W.; Duan, K.; Wang, Y. Clutter suppression algorithm based on fast converging sparse Bayesian learning for airborne radar. *Signal Process.* **2017**, *130*, 159–168. [[CrossRef](#)]
35. Poli, L.; Oliveri, G.; Viani, F.; Massa, A. MT-BCS-based microwave imaging approach through minimum-norm current expansion. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *61*, 4722–4732. [[CrossRef](#)]
36. Tipping, M.E. Sparse Bayesian Learning and the Relevance Vector Machine. *J. Mach. Learn. Res.* **2001**, *1*, 211–244.
37. Wipf, D.P.; Rao, B.D. An Empirical Bayesian Strategy for Solving the Simultaneous Sparse Approximation Problem. *IEEE Trans. Signal Process.* **2007**, *55*, 3704–3716. [[CrossRef](#)]
38. Duan, K.; Chen, H.; Xie, W.; Wang, Y. Deep learning for high-resolution estimation of clutter angle-Doppler spectrum in STAP. *IET Radar Sonar Nav.* **2022**, *16*, 193–207. [[CrossRef](#)]
39. Bishop, C.M.; Tipping, M.E. Variational Relevance Vector Machines. In Proceedings of the the 16th Conference in Uncertainty in Artificial Intelligence, Stanford University, Stanford, CA, USA, 30 June–3 July 2000; pp. 46–53.
40. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the International Conference on Learning Representations (ICLR 2014), Banff, AB, Canada, 14–16 April 2014.
41. Notin, P.; Hernández-Lobato, J.M.; Gal, Y. Improving black-box optimization in VAE latent space using decoder uncertainty. In Proceedings of the Neural Information Processing Systems 2021 (NeurIPS 2021), Virtual, 6–14 December 2021; pp. 802–814.
42. Zhou, Y.; Liang, X.; Zhang, W.; Zhang, L.; Song, X. VAE-based Deep SVDD for anomaly detection. *Neurocomputing* **2021**, *453*, 131–140. [[CrossRef](#)]
43. Ding, M. The road from MLE to EM to VAE: A brief tutorial. *AI Open* **2022**, *3*, 29–34. [[CrossRef](#)]
44. Hussain, A.; Anjum, U.; Channa, B.A.; Afzal, W.; Hussain, I.; Mir, I. Displaced Phase Center Antenna Processing For Airborne Phased Array Radar. In Proceedings of the 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST), Islamabad, Pakistan, 12–16 January 2021; pp. 988–992.
45. Ward, J. Space-time adaptive processing for airborne radar. In Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing, ICASSP, Detroit, MI, USA, 8–12 May 1995; pp. 2809–2812.
46. Wipf, D.; Nagarajan, S. A New View of Automatic Relevance Determination. In Proceedings of the Twentieth Annual Conference On Neural Information Processing Systems (NIPS 2007), Vancouver, BC, Canada, 4–7 December 2006; Platt, J., Koller, D., Singer, Y., Roweis, S., Eds.; Curran Associates, Inc.: West Chester, PA, USA, 2007; Volume 20.
47. Berger, J.O. *Statistical Decision Theory and Bayesian Analysis*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 1985.
48. Jordan, M.I.; Ghahramani, Z.; Jaakkola, T.S.; Saul, L.K. An Introduction to Variational Methods for Graphical Models. *Mach. Learn.* **1999**, *37*, 183–233. [[CrossRef](#)]
49. Worley, B. Scalable Mean-Field Sparse Bayesian Learning. *IEEE Trans. Signal Process.* **2019**, *67*, 6314–6326. [[CrossRef](#)]
50. Tzikas, D.G.; Likas, A.C.; Galatsanos, N.P. The variational approximation for Bayesian inference. *IEEE Signal Process. Mag.* **2008**, *25*, 131–146. [[CrossRef](#)]
51. Kingma, D.P.; Welling, M. Stochastic gradient VB and the variational auto-encoder. In Proceedings of the the 2nd International Conference on Learning Representations (ICLR 2014), London, UK, 8–9 July 2014; Volume 19, p. 121.

52. Ruiz, F.J.R.; Titsias, M.K.; Blei, D.M. The Generalized Reparameterization Gradient. In Proceedings of the the 29th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; pp. 460–468.
53. Zhang, H.; Chen, B.; Guo, D.; Zhou, M. WHAI: Weibull Hybrid Autoencoding Inference for Deep Topic Modeling. In Proceedings of the the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 April–3 May 2018.
54. Duan, Z.; Wang, D.; Chen, B.; Wang, C.; Chen, W.; Li, Y.; Ren, J.; Zhou, M. Sawtooth Factorial Topic Embeddings Guided Gamma Belief Network. In Proceedings of the the 38th International Conference on Machine Learning (ICML 2021), Virtual Event, 18–24 July 2021; pp. 2903–2913.
55. Hendrycks, D.; Lee, K.; Mazeika, M. Using Pre-Training Can Improve Model Robustness and Uncertainty. In Proceedings of the the 36th International Conference on Machine Learning (ICML 2019), Long Beach, CA, USA, 9–15 June 2019; pp. 2712–2721.
56. Robey, F.; Fuhrmann, D.; Kelly, E.; Nitzberg, R. A CFAR adaptive matched filter detector. *IEEE Trans. Aerosp. Electron. Syst.* **1992**, *28*, 208–216. [[CrossRef](#)]
57. Carlson, B. Covariance matrix estimation errors and diagonal loading in adaptive arrays. *IEEE Trans. Aerosp. Electron. Syst.* **1988**, *24*, 397–401. [[CrossRef](#)]
58. Cox, H.; Zeskind, R.M.; Owen, M.M. Robust adaptive beamforming. *IEEE Trans. Acoust. Speech Signal Process.* **1987**, *35*, 1365–1376. [[CrossRef](#)]
59. Chen, J.; Huo, X. Theoretical Results on Sparse Representations of Multiple-Measurement Vectors. *IEEE Trans. Signal Process.* **2006**, *54*, 4634–4643. [[CrossRef](#)]
60. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
61. Little, M.; Perry, W. Real-time multichannel airborne radar measurements. In Proceedings of the the 1997 IEEE National Radar Conference, Syracuse, NY, USA, 13–15 May 1997; pp. 138–142.