Article

Efficient Sparse Bayesian Learning Model for Image Reconstruction Based on Laplacian Hierarchical Priors and GAMP

Wenzhe Jin [1], Wentao Lyu [1,*], Yingrou Chen [1], Qing Guo [2], Zhijiang Deng [3] and Weiqiang Xu [1]

[1] Key Laboratory of Intelligent Textile and Flexible Interconnection of Zhejiang Province, Zhejiang Sci-Tech University, Hangzhou 310018, China; 202230704066@mails.zstu.edu.cn (W.J.)
[2] Zhejiang Technical Innovation Service Center, Hangzhou 310007, China
[3] Fox-ess, Co., Ltd., Wenzhou 325024, China
* Correspondence: alvinlwt@zstu.edu.cn

Abstract: In this paper, we present a novel sparse Bayesian learning (SBL) method for image reconstruction. We integrate the generalized approximate message passing (GAMP) algorithm and Laplacian hierarchical priors (LHP) into a basic SBL model (called LHP-GAMP-SBL) to improve the reconstruction efficiency. In our SBL model, the GAMP structure is used to estimate the mean and variance without matrix inversion in the *E*-step, while LHP is used to update the hyperparameters in the *M*-step. The combination of these two structures further deepens the hierarchical structures of the model. The representation ability of the model is enhanced so that the reconstruction accuracy can be improved. Moreover, the introduction of LHP accelerates the convergence of GAMP, which shortens the reconstruction time of the model. Experimental results verify the effectiveness of our method.

Keywords: sparse Bayesian learning; generalized approximate message passing; Laplacian hierarchical priors

1. Introduction

The Sparse Bayesian learning (SBL) model has been successfully applied to sparse signal recovery (SSR) or image recovery in various fields [1–3]. The essence of SSR is to restore the sparse signal $x \in \mathbb{R}^{N \times 1}$ from the $M < N$ noisy measurement vector $y \in \mathbb{R}^{M \times 1}$. In theory, the model is expressed as follows:

$$y = \mathcal{D}x + e, \tag{1}$$

where $\mathcal{D} \in \mathbb{R}^{M \times N}$ is the known dictionary matrix and $e \in \mathbb{R}^{M \times 1}$ is the observation noise.

Accordingly, many methods [4–7] have been proposed to solve (1) that update parameters through the Expectation-Maximization (EM) [8] algorithm and achieve good reconstruction results. The EM algorithm has two steps. The first step is to calculate the expectation (*E*-step), which calculates the posterior distribution of hidden variables (the expectation of log-likelihood) based on current parameter estimates. The second step is maximization (*M*-step), which updates parameters by maximizing the expectation calculated in the *E*-step. To estimate mean and variance accurately (in the *E*-step) and hyperparameters (in the *M*-step), most SBL models involve operations of matrix inversion. Thus, the computational efficiencies of these methods may be inhibited.

To tackle this issue, multiple optimization strategies have been presented [9,10]. In [9], Shutin proposed a fast variational sparse Bayesian method (FV-SBL) to accelerate convergence by calculating the stationary points of variational iterative updates with noninformative hyperpriors to maximize the bound. Subsequently, Duan [10] presented a fast inverse-free SBL method (IF-SBL) based on maximizing the relaxed evidence lower bound (ELBO). The relaxed-ELBO is first obtained to avoid the operation of matrix inversion in the

E-step. Then, the variational EM algorithm is used to maximize the relaxed-ELBO, which improves computational efficiency by using inverse-free matrix operations.

In recent studies [11–14], GAMP was used in SBL models to speed up the reconstruction. Based on the transformation of the estimation problem, Gaussian approximation, and central limit theorem, GAMP can effectively estimate parameters without matrix inversion in the *E*-step. In [11], GAMP was used to perform the inference through information transformation between nodes of the iterative probability graph model. It applies quadratic approximation and Taylor expansion to the cyclic confidence propagation with low computational complexity. Based on this, Zou [12] proposed combining GAMP with the SBL model (GAMP-SBL) and embedding this into the EM framework in place of the *E*-step, which significantly reduces the computational complexity and shortens the running time. Because the GAMP framework can adapt to any prior distribution for the unknown vectors, Al-Shoukairi [13] applied Gaussian scale mixture (GSM) priors to SBL sparse coefficients (GGAMP-SBL) to enhance the robustness of the model and adopted multiple damping factors to achieve fast convergence of GAMP, thereby improving the reconstruction efficiency. In [14], Dong formulated GAMP and variational Bayesian (VB) EM structure into an SBL model (VGAMP-SBL). By replacing the traditional EM model with a VB-EM model and using the GAMP structure to estimate the mean in the *E*-step, the interaction between data fluctuations and parameter fluctuations was weakened, thereby improving reconstruction performance.

However, multiple random variables in these GAMP-based SBL models may result in a fat-tailed distribution, which cannot be well fitted by the prior distribution of Gaussian mixture functions. Therefore, local optimization may occur during the iteration process, which means that more time is required to approximate hyperparameters in the *M*-step. In contrast, Laplace distribution has good robustness against a fat-tailed distribution [15]. It can easily extract samples far from the mean (abnormal samples). Thus, a Laplacian prior is more suitable for the reconstruction of natural images. In [16], Babacan integrated Laplace hierarchical priori (LHP) into the basic SBL frame to speed up reconstruction. LHP can improve the sparsity of unknown signals and obtain a lower reconstruction error.

Inspired by this, we integrate GAMP and LHP into the basic SBL model (LHP-GAMP-SBL) to improve the efficiency of image reconstruction, reduce computational complexity, shorten running time, and accelerate convergence. The main contributions are summarized as follows:

(1) We propose an efficient sparse Bayesian learning model for image reconstruction based on Laplace hierarchical priors and GAMP. The GAMP structure is used to estimate the mean and variance in the *E*-step without matrix inversion. It outperforms several mainstream SBL image reconstruction models in terms of running time.

(2) Sufficient damping factors are used to constrain the output to prevent the divergence of the GAMP during iterations.

(3) In order to solve the problem that the prior distribution of the Gaussian mixture function in GAMP cannot fit a fat-tailed distribution well, we use the Laplace hierarchical prior (LHP) instead of the Gaussian prior to update the hyperparameters in the *M*-step, which improves the robustness of the model and speeds up the convergence of GAMP.

(4) The combination of GAMP and LHP further deepens the hierarchical structure of the model and improves the sparsity of the model. The higher the sparsity of the model, the fewer non-zero elements it has, and the stronger the representation ability of the model (using as few non-zero elements as possible to represent the main characteristic information of the original signal). The reconstruction accuracy is improved without affecting the computational efficiency. A large number of experiments have shown that our LHP-GAMP-SBL model outperforms other mainstream SBL image reconstruction models in overall reconstruction performance.

2. Methods

2.1. Laplace Hierarchical SBL Model

The Laplace distribution has a peak at zero, so it is easier to produce sparse solutions, so that signal coefficients close to zero are preferred [15,16]. In contrast, the Gaussian distribution is smooth and has no peak at zero, so it is less capable of modeling sparsity. According to sparse representation (SR) theory, the purpose of SR is to represent the main characteristic information of the original signal with as few non-zero elements as possible, which can make it easier to further process the signal. When the sparsity of the model is higher, there are fewer non-zero elements. Therefore, the representation ability of the model will be enhanced, thereby improving reconstruction accuracy. In addition, the Laplace distribution is more robust to abnormal values because its tail is heavier than that of the Gaussian distribution [15]. This means that in the presence of noise or abnormal values, the reconstruction performance of the SBL model using the Laplace prior is better. Therefore, this paper will adopt the Laplace hierarchical prior model to improve reconstruction accuracy.

In (1), all unknown parameters are treated as random quantities with a specified probability distribution. The unknown signal x is assigned a prior distribution $p(x|\lambda)$, and the observation y is a random process with conditional distribution $p(y|x,\theta)$, where $\theta = 1/\sigma^2$ is the inverse noise variance. These distributions depend on the model parameters, λ and θ, which can also be assigned additional prior distributions. The SBL model defines the joint distribution of all parameters and observations as follows:

$$p(x,\lambda,\theta,y) = p(y|x,\theta)p(x|\lambda)p(\lambda)p(\theta). \qquad (2)$$

Let noise e follow an independent Gaussian distribution with zero mean and θ^{-1} variance. Thus, one has:

$$p(y|x,\theta) = \mathcal{N}(y|\mathcal{D}x,\theta^{-1}). \qquad (3)$$

Let a Gamma prior be placed on θ, we have:

$$p(\theta|a^\theta,b^\theta) = \Gamma(\theta|a^\theta,b^\theta) = \frac{(b^\theta)^{a^\theta}}{\Gamma(a^\theta)}\theta^{a^\theta-1}exp(-b^\theta\theta). \qquad (4)$$

where $\theta > 0$ represents the hyperparameter, a is the scale parameter, and b is the shape parameter. The expectation of θ is given by the following equation.

$$\langle\theta\rangle = \frac{a^\theta}{b^\theta}. \qquad (5)$$

where $\langle\cdot\rangle$ denotes the expectation. Similarly, based on the sparsity of x in (1), x is assumed to follow the distribution of Laplace priori, that is:

$$p(x|\lambda) = \frac{\lambda}{2}exp(-\frac{\lambda}{2}\|x\|_1). \qquad (6)$$

However, the likelihood function in (3) and the prior probability in (6) are not conjugate. When they are conjugate, the posterior distribution and the prior probability are the same distribution. A conjugate prior can simplify the calculation of the posterior distribution. Therefore, we adopt the hierarchical prior model [16]. The first layer of the model is:

$$p(x|\omega) = \prod_{i=1}^{N}\mathcal{N}(x_i|0,\omega_i). \qquad (7)$$

where $\omega = (\omega_1,\omega_2,\cdots,\omega_N)$. The second layer of the hierarchical model (ω_i) is:

$$p(\omega_i|\lambda) = \Gamma(\omega_i|1,\lambda/2) = \frac{\lambda}{2}exp(-\frac{\lambda\omega_i}{2}), \omega_i,\lambda \geq 0. \qquad (8)$$

Based on (7) and (8), the distribution of Laplacian priors can be expressed by:

$$p(\boldsymbol{x}|\lambda) = \int p(\boldsymbol{x}|\omega)p(\omega|\lambda)\, d\omega = \prod_i \int p(\boldsymbol{x}_i|\omega_i)p(\omega_i|\lambda)\, d\omega_i$$

$$= \frac{\lambda^{N/2}}{2^N} exp(-\sqrt{\lambda}\sum_i |\boldsymbol{x}_i|). \tag{9}$$

Finally, the parameter λ is calculated by the Gamma hyperprior method.

$$p(\lambda|\gamma) = \Gamma(\lambda|\frac{\gamma}{2}, \frac{\gamma}{2}). \tag{10}$$

2.2. Laplace Hierarchical Bayesian Inference

According to Bayesian inference based on the maximum likelihood method, the posterior distribution can be written as:

$$p(\boldsymbol{x}, \omega, \lambda, \theta|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x}, \theta)p(\boldsymbol{x}|\omega)p(\omega|\lambda)p(\lambda)p(\theta)}{p(\boldsymbol{y})}. \tag{11}$$

where $p(\boldsymbol{y})$ represents the marginal likelihood function, which can be expressed as:

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{x}, \theta)p(\boldsymbol{x}|\omega)p(\omega|\lambda)p(\lambda)p(\theta)d\boldsymbol{x}$$

$$= (\frac{1}{2\pi})^{N/2}|\boldsymbol{C}|^{-1/2}exp(-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{C}^{-1}\boldsymbol{y})p(\omega|\lambda)p(\lambda)p(\theta). \tag{12}$$

where $\boldsymbol{C} = (\theta^{-1}\boldsymbol{I} + \mathcal{D}\boldsymbol{\Lambda}^{-1}\mathcal{D}^T)$ and $\boldsymbol{\Lambda} = diag(1/\omega_i)$, $\boldsymbol{I}$ represent the $M \times M$ identity matrix.

According to the properties of the Gaussian function, it can be obtained that the posterior distribution also approximately obeys the Gaussian distribution. Its posterior mean ($\boldsymbol{\mu_x}$), variance ($\boldsymbol{\Sigma_x}$) of $\boldsymbol{x}$, and hyperparameters (ω, λ, θ) can be iteratively updated by the EM algorithm, as summarized below:

$$\boldsymbol{\mu_x} = \theta\boldsymbol{\Sigma_x}\mathcal{D}^T\boldsymbol{y}, \quad \boldsymbol{\Sigma_x} = (\theta\mathcal{D}^T\mathcal{D} + \boldsymbol{\Lambda})^{-1}. \tag{13}$$

$$\omega_i = \frac{\sqrt{1 + 4\lambda\langle x_i^2 \rangle} - 1}{2\lambda}, \quad \lambda = \frac{2(N-1) + \gamma}{\sum_i \omega_i + \gamma},$$

$$\theta = \frac{N/2 + a^\theta}{\langle \|\boldsymbol{y} - \mathcal{D}\boldsymbol{x}\|^2 \rangle/2 + b^\theta}. \tag{14}$$

In addition, we can also estimate γ by maximizing the value of the logarithm of (12) with respect to γ. The solution formula is shown in (15):

$$lnp(\boldsymbol{y}) = -\frac{1}{2}ln|\boldsymbol{C}| - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{C}^{-1}\boldsymbol{y} + Nln\frac{\lambda}{2} - \frac{\lambda}{2}\sum_i \omega_i + \frac{\gamma}{2}ln\frac{\gamma}{2}$$

$$- ln\Gamma(\frac{\gamma}{2}) + (\frac{\gamma}{2} - 1)ln\lambda - \frac{\gamma}{2}\lambda + (a^\theta - 1)log\theta - b^\theta\theta, \tag{15}$$

$$\Rightarrow ln\frac{\gamma}{2} + 1 - \psi(\frac{\gamma}{2}) + ln\lambda - \lambda = 0.$$

where $\psi(\cdot)$ is the digamma function, which is the derivative of gamma function $\Gamma(\cdot)$ with respect to the logarithm of the independent variable. In (15), $\psi(\frac{\gamma}{2})$ is the derivative of $ln\Gamma(\frac{\gamma}{2})$ with respect to $\frac{\gamma}{2}$.

It can be seen that (15) is a nonlinear equation, which is complicated by the existence of the digamma function $\psi(\cdot)$. The digamma function has no simple analytical form, so (15) cannot be solved directly by simple algebraic manipulation. In order to solve this equation, numerical methods are usually required.

Unlike the method in [16], we use two layers of hierarchical priors. There are two reasons for this. First, the use of the LHP structure in the SBL model can accelerate the convergence of GAMP. Second, there is no direct solution in (15), and the last layer contributes little to reducing the recovery error, while more layers require more computing resources. It can be seen that the update of the posterior mean and covariance in (13) involves hyperparameters such as θ, ω, and λ, which need to be updated through (14). Therefore, through continuous iterative updates of (13) and (14), it will eventually converge to an estimate of x.

2.3. LHP-GAMP-SBL Model

The GAMP algorithm is based on the channel coding and message propagation principles of information theory. Through the transformation of the estimation problem, Gaussian approximation and the use of central limit theorem, it can effectively approximate the parameters μ_x and Σ_x solved in *E*-step of the SBL model shown in (13). In addition, it avoids the matrix inversion, greatly improving reconstruction efficiency.

In our LHP-GAMP-SBL model, GAMP is used to replace (13) in the *E*-step, while LHP is used to update (14) in the *M*-step. The structure of the LHP-GAMP-SBL model is shown in Figure 1.
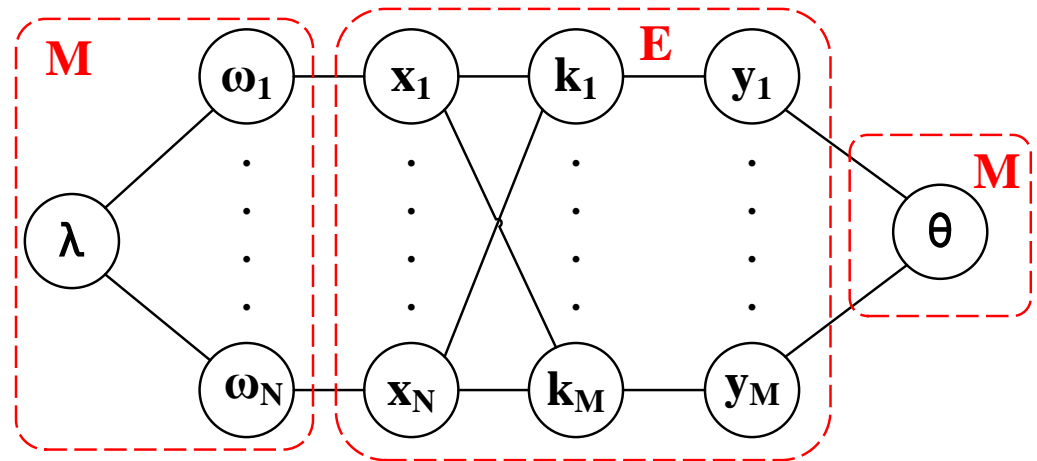

Figure 1. The structure of the LHP-GAMP-SBL model.

In Figure 1, $k \in \mathbb{R}^{M \times 1}$ represents the linear transformation of *n*-dimensional variable $x \in \mathbb{R}^{N \times 1}$, i.e.,

$$k = \mathcal{D}x. \tag{16}$$

For the given prior distribution $p(x|\omega)$ and marginal likelihood function $p(y|k)$, GAMP constructs iterative expressions by quadratic approximation and Taylor series expansions. The detailed derivation of GAMP can be found in [11]. Specifically, during the process of message propagation in Figure 1, the input function $G_k(\bar{x}, v_{\bar{k}})$ and the output function $G_x(\bar{x}, v_{\bar{x}})$ can be defined as:

$$[G_k(\bar{k}, v_{\bar{k}})]_m = \frac{\int k_m p(y_m|k_m)\mathcal{N}(k_m; \frac{\bar{k}_m}{v_{\bar{k}_m}}, \frac{1}{v_{\bar{k}_m}})dk_m}{\int p(y_m|k_m)\mathcal{N}(k_m; \frac{\bar{k}_m}{v_{\bar{k}_m}}, \frac{1}{v_{\bar{k}_m}})dk_m}, \tag{17}$$

$$[G_x(\bar{x}, v_{\bar{x}})]_n = \frac{\int x_n p(x_n|\omega_n)\mathcal{N}(x_n; \bar{x}_n, v_{\bar{x}_n})dx_n}{\int p(x_n|\omega_n)\mathcal{N}(x_n; \bar{x}_n, v_{\bar{x}_n})dx_n}. \tag{18}$$

where $m \in M$ represents the *m*-th atom of the input function, $n \in N$ represents the *n*-th atom of the output function, the intermediate variables $\bar{x}$ and $\bar{k}$ represent the approximation estimation of the sparse signal x and the noise-free model k, respectively, and $v_{\bar{x}}$ and $v_{\bar{x}}$

correspond to variances of x and k, respectively. Based on (3) and (7), the update rules of the input function and output function can be obtained by simplifying (17) and (18):

$$G_k(\bar{k}, v_{\bar{k}}) = \frac{\bar{k}/v_{\bar{k}} - y}{1/\theta + 1/v_{\bar{k}}}, \qquad\qquad G'_k(\bar{k}, v_{\bar{k}}) = \frac{\theta}{\theta + v_{\bar{k}}}, \qquad (19)$$

$$G_x(\bar{x}, v_{\bar{x}}) = \frac{\omega}{\omega + v_{\bar{x}}}\bar{x}, \qquad\qquad G'_x(\bar{x}, v_{\bar{x}}) = \frac{\omega}{\omega + v_{\bar{x}}}. \qquad (20)$$

Usually, the damped factors, $\eta_{\bar{k}}, \eta_{\bar{x}} \in (0, 1]$, are introduced to ensure the convergence in GAMP. The two damped factors correspond to two square roots in (21), respectively.

$$F(\eta_{\bar{k}}, \eta_{\bar{x}}) = \frac{\alpha \Omega_{max}}{\Sigma_\Omega}\eta^2 + \beta\eta + const, \qquad (21)$$

where Ω_{max} represents the maximum singular value of dictionary $\mathcal{D}$, and Σ_Ω represents the sum of singular values of $\mathcal{D}$. Both α and β are constant coefficients. The damped factor $\eta_{\bar{k}}$ is used to constrain $\bar{s}$ to enhance convergence and make it easier to obtain the estimate of x through continuous iterations. The damping factor $\eta_{\bar{x}}$ is used to control the sparse signal x.

Algorithm 1 shows the detailed process of the LHP-GAMP-SBL algorithm.

Algorithm 1: LHP-GAMP-SBL Algorithm

Input: $\mathcal{D} \in \mathbb{R}^{M \times N}, y$.

1 **Initialize:** $Iter_{em}, Iter_{gamp}, \varepsilon_{em}, \varepsilon_{gamp}, \theta, \lambda^0, \omega^0 \leftarrow 1, \boldsymbol{\mu_x}^0 \leftarrow 0, \hat{s}^0 \leftarrow 0, \boldsymbol{\Sigma_x}^0 \leftarrow 1;$

2 **for** $t = 1, 2, \cdots, Iter_{em}$ **do**

3 $\quad \boldsymbol{\mu_{\hat{x}}}^0 \leftarrow \boldsymbol{\mu_x}^{t-1}, \boldsymbol{\Sigma_{\hat{x}}}^0 \leftarrow \boldsymbol{\Sigma_x}^{t-1}, \bar{s}^0 \leftarrow \hat{s}^{t-1};$

4 $\quad$ // E-Step approximation;

5 $\quad$ **for** $i = 1, 2, \cdots, Iter_{gamp}$ **do**

6 $\quad\quad v_{\bar{k}}^i \leftarrow (|\mathcal{D}|^2 \boldsymbol{\Sigma_{\hat{x}}}^{i-1})^{-1};$

7 $\quad\quad \bar{k}^i \leftarrow \bar{s}^{i-1} + v_{\bar{k}}^i \mathcal{D} \boldsymbol{\mu_{\hat{x}}}^{i-1};$

8 $\quad\quad v_{\bar{s}}^i \leftarrow v_{\bar{k}}^i G'_k(\bar{k}^i, v_{\bar{k}}^i);$

9 $\quad\quad \bar{s}^i \leftarrow (1 - \eta_{\bar{k}})\bar{s}^{i-1} + \eta_{\bar{k}} G_k(\bar{k}^i, v_{\bar{k}}^i);$

10 $\quad\quad v_{\bar{x}}^i \leftarrow [(|\mathcal{D}|^2)^T v_{\bar{s}}^i]^{-1};$

11 $\quad\quad \bar{x}^i \leftarrow \boldsymbol{\mu_{\hat{x}}}^{i-1} - v_{\bar{x}}^i \mathcal{D}^T \bar{s}^i;$

12 $\quad\quad \boldsymbol{\Sigma_{\hat{x}}}^i \leftarrow v_{\bar{x}}^i G'_x(\bar{x}^i, v_{\bar{x}}^i);$

13 $\quad\quad \boldsymbol{\mu_{\hat{x}}}^i \leftarrow (1 - \eta_{\bar{x}})\boldsymbol{\mu_{\hat{x}}}^{i-1} + \eta_{\bar{x}} G_x(\bar{x}^i, v_{\bar{x}}^i);$

14 $\quad\quad$ **if** $\|\boldsymbol{\mu_{\hat{x}}}^i - \boldsymbol{\mu_{\hat{x}}}^{i-1}\|_2 / \|\boldsymbol{\mu_{\hat{x}}}^i\|_2 < \varepsilon_{gamp}$ **then**

15 $\quad\quad\quad$ break;

16 $\quad\quad$ **end**

17 $\quad$ **end**

18 $\quad \boldsymbol{\mu_x}^t \leftarrow \boldsymbol{\mu_{\hat{x}}}^i, \boldsymbol{\Sigma_x}^t \leftarrow \boldsymbol{\Sigma_{\hat{x}}}^i, \hat{s}^t \leftarrow \bar{s}^i;$

19 $\quad$ // M-Step approximation;

20 $\quad$ **for** $j = 1, 2, \cdots, N$ **do**

21 $\quad\quad \omega_{(j)}^t \leftarrow (\sqrt{1 + 4\lambda^{t-1}\boldsymbol{\mu_{x(j)}}^t} - 1)/(2\lambda^{t-1});$

22 $\quad$ **end**

23 $\quad \lambda^t \leftarrow 2(N-1)/(\sum_{j=1}^N \omega_{(j)}^{t-1});$

24 $\quad$ **if** $\|\boldsymbol{\mu_x}^t - \boldsymbol{\mu_x}^{t-1}\|_2 / \|\boldsymbol{\mu_x}^t\|_2 < \varepsilon_{em}$ **then**

25 $\quad\quad$ break;

26 $\quad$ **end**

27 **end**

Output: $x = \boldsymbol{\mu_x}^t$

In Algorithm 1, $|\mathcal{D}|$ represents the component wise magnitude squared of D; the variable $\bar{x}$ represents the approximate estimate of the sparse signal x with variance $v_{\bar{x}}$; and $\bar{k}$ represents the approximate estimate of the noise-free model $k = \mathcal{D}x$ with variance $v_{\bar{k}}$. The parameters $Iter_{em}$ and ε_{em} represent the maximum number of iterations and the normalized tolerance parameter of the EM algorithm, respectively. $Iter_{gamp}$ and ε_{gamp} represent the maximum number of iterations and the normalized tolerance parameter of the GAMP algorithm, respectively. These parameters are obtained through continuous optimization.

Overall, the computational complexities of IF-SBL, SBL-EM, and Laplace-SBL are, respectively, $\mathcal{O}(MN^2)$, $\mathcal{O}(N^3)$, and $\mathcal{O}(N^3)$. In contrast, GGAMP-SBL, VGAMP-SBL, and LHP-GAMP-SBL have the same computational complexity, $\mathcal{O}(MN)$. In most SBL models, mean and variance are estimated in the *E*-step, and hyperparameters are updated in the *M*-step. Table 1 lists respective computational complexities of the *E*-step and *M*-step for different SBL models. By using GAMP to estimate mean and variance in the *E*-step and using LHP to update hyperparameters in the *M*-step, our LHP-GAMP-SBL model decreases the computational complexity significantly.

Table 1. Computational complexity of *E*-step and *M*-step for different SBL models.

Methods	*E*-STEP	*M*-STEP
IF-SBL [10]	$\mathcal{O}(MN^2)$	$\mathcal{O}(MN)$
SBL-EM [3]	$\mathcal{O}(N^3)$	$\mathcal{O}(MN)$
Laplace-SBL [16]	$\mathcal{O}(N^3)$	$\mathcal{O}(N)$
GGAMP-SBL [13]	$\mathcal{O}(MN)$	$\mathcal{O}(MN)$
VGAMP-SBL [14]	$\mathcal{O}(MN)$	$\mathcal{O}(MN)$
LHP-GAMP-SBL	$\mathcal{O}(MN)$	$\mathcal{O}(N)$

3. Results and Discussion

The public DIOR [17] and DOTA [18] optical remote sensing image datasets and UC Merced Land Use(UCM) [19] remote sensing image datasets are used to verify the effectiveness of our method. In order to highlight different types of ground object information in remote sensing images and reduce the influence of factors such as atmosphere and illumination, the intensity information of each band was extracted in the experiment, and the amplitude map of the remote sensing image was generated to facilitate subsequent image reconstruction.

In our experiments, $\mathcal{D}$ is initialized as a Gaussian random matrix, $Iter_{em}$ is set as 5000, $Iter_{gamp} \in [5, 20]$. λ^0, ε_{gamp}, and ε_{em} are set as 1×10^{-6}, 1×10^{-4}, and 3×10^{-5}, respectively. The value of hyperparameter θ is empirically set as $0.01\|y\|_2^2$. Five state-of-the-art methods, namely, SBL-EM [3], Laplace-SBL [16], IF-SBL [10], GGAMP-SBL [13], and VGAMP-SBL [14], are utilized for comparison. The parameters in these methods are selected for optimal results. Normalized root-mean-square error (ERROR) [13], reconstruction time (TIME), peak signal-to-noise ratio (PSNR), and structural similarity (SSIM) are used to measure the performance of the different methods. Error is defined by $Error = \|\bar{S} - S\|_F^2 / \|S\|_F^2$, where $\bar{S}$ and S are estimated and real model parameters, respectively.

Forty images are selected randomly from each dataset (a total of 120 images) and used for experiments. These images are resized to 256×256 in pixels. They are then sparsely processed by using wavelet transform before optimization. The number of compressed samples and wavelet coefficients are 10,672 and 16,320, respectively. The programs are written in Matlab2017b and run in Windows 10 on a Lenovo Legion R9000X 2021 laptop with an NVIDIA GeForce RTX 3060 Laptop GPU, AMD Ryzen 7 5800H CPU, and 16GB RAM.

For the different SBL models, the reconstructed images of the three datasets are shown in Figure 2. Figure 2a–c represent two examples of 40 experimental images in each of the three image datasets DIOR, DOTA, and UCM. From left to right, they represent the original image and the reconstructed images of IF-SBL, SBL-EM, Laplace-SBL, GGAMP-

SBL, VGAMP-SBL, and the LHP-GAMP-SBL algorithm we proposed. The datas in brackets represent the reconstruction error (ERROR).
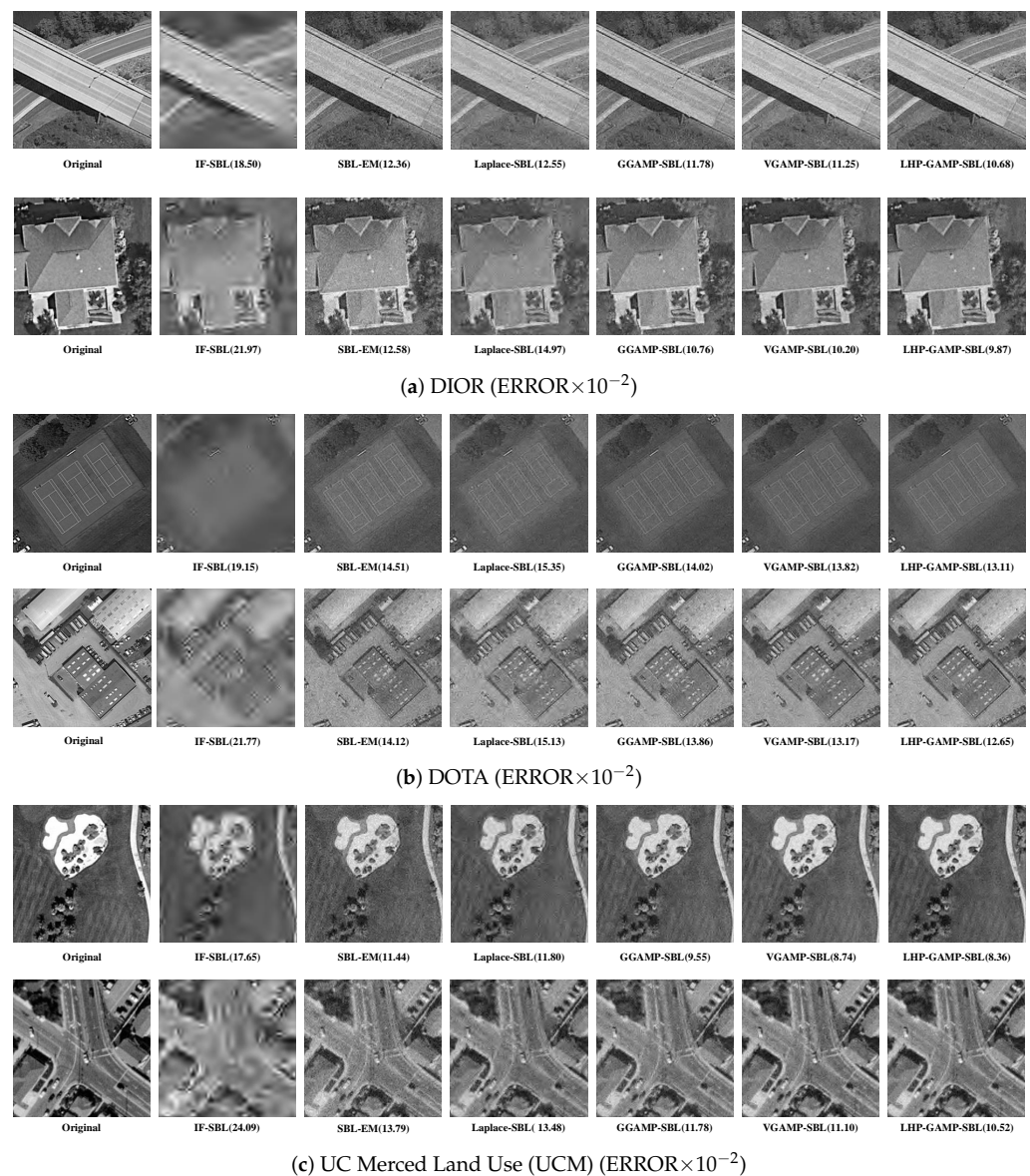

(**a**) DIOR (ERROR$\times 10^{-2}$)

(**b**) DOTA (ERROR$\times 10^{-2}$)

(**c**) UC Merced Land Use (UCM) (ERROR$\times 10^{-2}$)

Figure 2. The recovery results of different methods.

We can see that the three reconstruction methods based on GAMP have better results. The image reconstructed by the IF-SBL method is fuzzy, and it is difficult to extract valid information from the reconstructed image. To further compare the performance differences between different methods, we show the reconstructed ERROR and reconstructed TIME in detail in Table 2. Note that each value in Table 2 denotes the average value of the results for 120 images. As we see, our LHP-GAMP-SBL model achieves better reconstruction accuracies than other methods. The average ERROR values for the three datasets are, respectively, 0.0873, 0.1110, and 0.1036, which are lower than these of the GGAMP-SBL and VGAMP-SBL methods. In terms of TIME, our method has a larger advantage than the other GAMP-based models. By using Laplace prior to estimate hyperparameters in the *M*-step, our model can reconstruct the images with fewer iterations. This improves the reconstruction efficiency significantly.

Table 2. ERROR and TIME of different SBL models for the three datasets.

Methods	Error ($\times 10^{-2}$)			TIME (s)		
	A	**B**	**C**	**A**	**B**	**C**
IF-SBL [10]	16.54	17.51	20.55	458.2	449.2	431.2
SBL-EM [3]	10.66	12.28	12.89	978.2	962.4	1083.4
Laplace-SBL [16]	11.03	13.48	13.01	281.9	290.2	335.0
GGAMP-SBL [13]	9.49	11.87	11.53	1322.0	1414.6	1517.5
VGAMP-SBL [14]	9.04	11.56	10.78	578.1	764.5	600.1
LHP-GAMP-SBL	8.73	11.10	10.36	259.0	276.3	273.9

A (DIOR Dataset) B (DOTA Dataset) C (UCM Dataset).

In addition, Table 2 shows that the IF-SBL method obtains the lowest reconstruction accuracies. The main reason is that IF-SBL fails to obtain the optimal ELBO because of the use of a smoothing function in complex images.

Figure 3 shows the ERROR, TIME, PSNR, and SSIM results of 20 images (selected randomly) for different methods. From Figure 3a, the three GAMP-based SBL models have better reconstruction accuracies than the Laplace-SBL model. The best results are recorded by our LHP-GAMP-SBL model. Figure 3b plots the running time of different methods. By using GAMP to estimate mean and variance in the *E*-step and by employing LHP to update hyperparameters in the *M*-step, our method obtains better reconstruction efficiency than the GGAMP-SBL and VGAMP-SBL models. From Figure 3c,d, it can be seen that our proposed algorithm results in higher PSNR as well as SSIM. Combined with Figure 4, we can see that an algorithm with a higher PSNR value can obtain a lower ERROR and a higher SSIM.
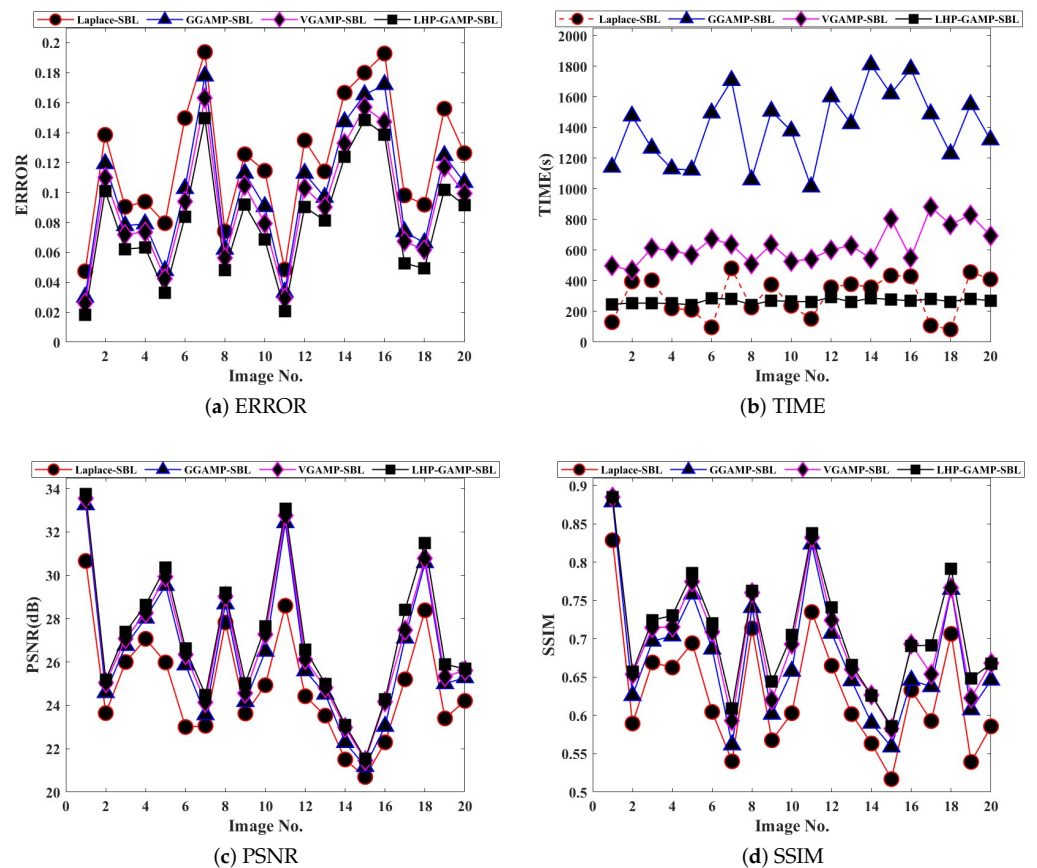

(**a**) ERROR

(**b**) TIME

(**c**) PSNR

(**d**) SSIM

Figure 3. ERROR, TIME, PSNR, and SSIM results of 20 images for different methods.

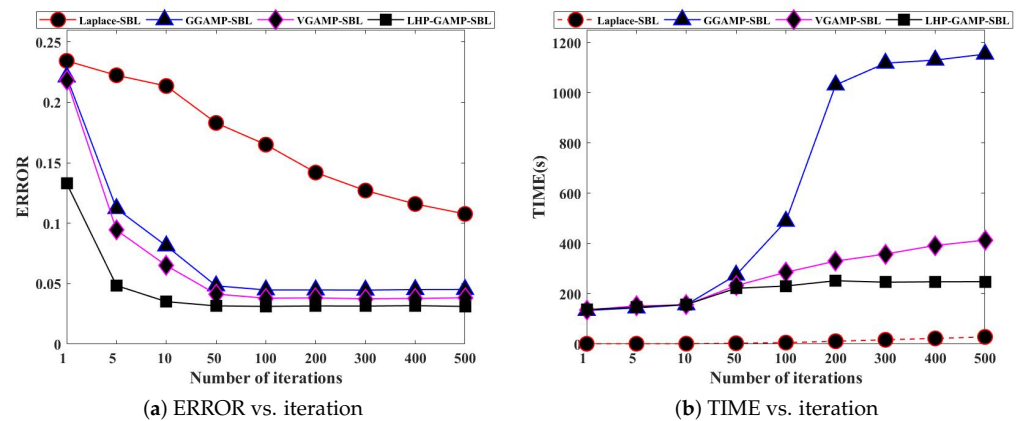(**a**) ERROR vs. iteration

(**b**) TIME vs. iteration

Figure 4. Convergence processes of different methods.

Figure 4 shows the average convergence processes of 20 images per iteration for different methods. Figure 4a shows that our LHP-GAMP-SBL method requires about 50 iterations to complete convergence and with low ERROR. In contrast, VGAMP-SBL and GGAMP-SBL require approximately 100 iterations to produce the desired results. Surprisingly, Laplace-SBL requires over 500 iterations to achieve image reconstruction. This is because the Laplace-SBL model employs a fast decision rule to determine whether hyperparameters in the *M*-step are updated or not. Accordingly, some small sparse solutions are discarded by a specific condition. This method achieves high reconstruction efficiency at the cost of restoration quality. This method still costs less time to terminate iterations, compared with the GGAMP-SBL and VGAMP-SBL methods. Figure 4b confirms this. From Table 1, we know that the three GAMP-based SBL models have the same computational complexities. Because VGAMP-SBL and LHP-GAMP-SBL have better sparsities in SBL models than GGAMP-SBL, thus, the former two methods require less computation resources. This means that the VGAMP-SBL and LHP-GAMP-SBL methods cost less running time per iteration, as shown in Figure 4b.

According to the experimental results, it can be seen that our LHP-GAMP-SBL model greatly reduces the computational complexity through the use of the GAMP structure, and gives the model better convergence through appropriate damping factors. Due to the introduction of the LHP structure, it has better sparsity compared to the existing methods and accelerates the convergence of GAMP. Therefore, the combination of the two structures results in a better reconstruction performance of the LHP-GAMP-SBL model.

It should be noted that our experiments are performed for optical images with wavelet transform before training, and the experimental results proved the effectiveness of the LHP-GAMP-SBL model. We also used SAR images for reconstruction, and the results show that the LHP-GAMP-SBL model is still effective for some sparse scenes.

4. Conclusions

This paper proposes an efficient SBL image reconstruction model based on Laplace hierarchical prior (LHP) and GAMP, called the LHP-GAMP-SBL model. We integrate the two structures of GAMP and LHP into the basic SBL model to improve the reconstruction efficiency. In our LHP-GAMP-SBL model, the GAMP structure is used to estimate mean and variance without matrix inversion in the *E*-step, while the LHP structure is used to update the hyperparameters in the *M*-step. In the LHP-GAMP-SBL model, the GAMP structure is used to estimate the posterior mean and variance in the *E*-step without matrix inversion, which greatly reduces the computational complexity, and introduces a damping factor to enhance convergence. The LHP structure is used to update the hyperparameters in the *M*-step, and the high sparsity of the Laplace prior is used to improve the reconstruction accuracy. In addition, the introduction of LHP accelerates the convergence of GAMP, thereby shortening the reconstruction time of the model. The combination of these two

structures further deepens the hierarchical structure of the model, so that the overall reconstruction performance of the model is enhanced. All experimental results verify the effectiveness of our LHP-GAMP-SBL model.

Future work will focus on further improving the reconstruction accuracy by using convex optimization strategies instead of the EM algorithm to update the hyperparameters in the *M*-step or by combining the LHP-GAMP-SBL model with a neural network.

Author Contributions: Conceptualization, W.J. and Y.C.; methodology, W.J., Y.C and W.L.; software (Matlab2017b) , W.J.; validation, W.L., Q.G. and Z.D.; data analysis W.J. and W.L.; writing—original draft preparation, W.J. and Y.C.; writing—review and editing, W.J., Y.C., W.L. and W.X. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: The DIOR and DOTA optical remote sensing datasets and the UC Merced Land Use (UCM) remote sensing datasets used in this study are available online at https://drive.google.com/drive/folders/1UdlgHk49iu6WpcJ5467iT-UqNPpx__CC (DIOR) (accessed on 8 December 2023) , https://captain-whu.github.io/DOTA/dataset.html (DOTA) (accessed on 21 December 2023) and http://weegee.vision.ucmerced.edu/datasets/landuse.html (UCM) (accessed on 6 January 2024) .

Conflicts of Interest: Author Qing Guo was employed by the company Zhejiang Technical Innovation Service Center. Author Zhijiang Deng was employed by the company Fox-ess, Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SBL	sparse Bayesian learning
GAMP	generalized approximate message passing
SSR	sparse signal recovery
EM	expectation maximization
VB	variational Bayesian
LHP	Laplace hierarchical priori
ELBO	evidence lower bound
PSNR	peak signal-to-noise ratio
SSIM	structural similarity

References

1. Zhou, W.; Zhang, H.T.; Wang, J. An efficient sparse Bayesian learning algorithm based on Gaussian-scale mixtures. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 3065–3078. [CrossRef] [PubMed]
2. Tipping, M.E. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.* **2001**, *1*, 211–244.
3. Wipf, D.; Rao, B. Sparse Bayesian learning for basis selection. *IEEE Trans. Signal Process.* **2004**, *52*, 2153–2164. [CrossRef]
4. Zhang, Y.; Qi, X.; Jiang, Y.C.; Li, H.B.; Liu, Z.T. Image reconstruction for low-oversampled staggered SAR based on sparsity Bayesian learning in the presence of a nonlinear PRI variation strategy. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–24. [CrossRef]
5. Chen, P.; Zhao, J.; Bai, X. Block Inverse-free Sparse Bayesian Learning for Block Sparse Signal Recovery. In Proceedings of the 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), Chongqing, China, 11–13 December 2019; pp. 1–4.
6. Sant, A.; Leinonen, M.; Rao, B.D. Block-sparse signal recovery via general total variation regularized sparse Bayesian learning. *IEEE Trans. Signal Process.* **2022**, *70*, 1056–1071. [CrossRef]
7. Yuan, S.Y.; Ji, Y.Z.; Shi, P.D.; Zeng, J.; Gao, J.H.; Wang, S.X. Sparse Bayesian learning-based seismic high-resolution time-frequency analysis. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 623–627. [CrossRef]
8. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–22. [CrossRef]
9. Shutin, D.; Buchgraber, T.; Kulkarni, S.R.; Poor, H.V. Fast variational sparse Bayesian learning with automatic relevance determination for superimposed signals. *IEEE Trans. Signal Process.* **2011**, *59*, 6257–6261. [CrossRef]

10. Duan, H.P.; Yang, L.X.; Fang, J.; Li, H.B. Fast inverse-free sparse Bayesian learning via relaxed evidence lower bound maximization. *IEEE Signal Process. Lett.* **2017**, *24*, 774–778. [CrossRef]

11. Rangan, S. Generalized approximate message passing for estimation with random linear mixing. In Proceedings of the 2011 IEEE International Symposium on Information Theory Proceedings, St. Petersburg, Russia, 31 July–5 August 2011; pp. 2168–2172.

12. Zou, X.B.; Li, F.W.; Fang, J.; Li, H.B. Computationally efficient sparse Bayesian learning via generalized approximate message passing. In Proceedings of the 2016 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB), Nanjing, China, 16–19 October 2016; pp. 2168–2172.

13. Al-Shoukairi, M.; Schniter, P.; Rao, B.D. A GAMP-based low complexity sparse Bayesian learning algorithm. *IEEE Trans. Signal Process.* **2017**, *66*, 294–308. [CrossRef]

14. Dong, J.Y.; Lyu, W.T.; Zhou, D.; Xu, W.Q. Variational Bayesian and Generalized Approximate Message Passing-Based Sparse Bayesian Learning Model for Image Reconstruction. *IEEE Signal Process. Lett.* **2022**, *29*, 2328–2332. [CrossRef]

15. Seeger, M.W.; Nickisch, H. Compressed sensing and Bayesian experimental design. In Proceedings of the 25th International Conference on Machine Learning(ICML), Helsinki, Finland, 5–9 July 2008; pp. 912–919.

16. Babacan, S.D.; Molina, R.; Katsaggelos, A.K. Bayesian compressive sensing using Laplace priors. *IEEE Trans. Image Process.* **2009**, *19*, 53–63. [CrossRef] [PubMed]

17. Cheng, G.; Wang, J.B.; Li, K.; Xie, X.X.; Lang, C.B.; Yao, Y.Q.; Han, J.W. Anchor-free oriented proposal generator for object detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [CrossRef]

18. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.

19. Peng, T.; Yi, J.J.; Fang, Y. A Local-global Interactive Vision Transformer for Aerial Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5. [CrossRef]