Modeling Mask Uncertainty in Hyperspectral Image Reconstruction

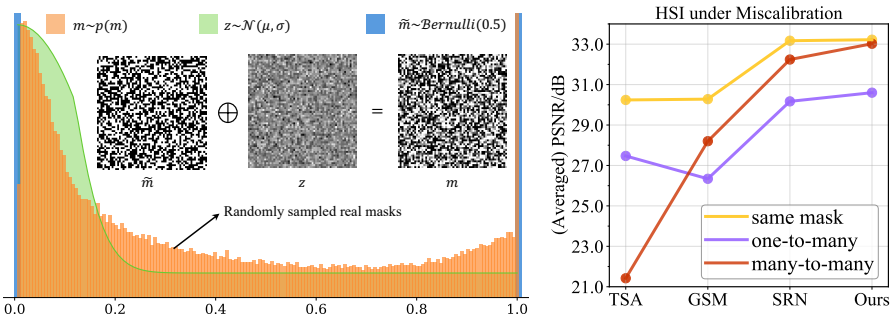
Jiamian Wang[1], Yulun Zhang[2], Xin Yuan[3], Ziyi Meng[4], and Zhiqiang Tao[1]

[1]Department of Computer Science and Engineering, Santa Clara University, USA
[2]ETH Zürich, Switzerland [3]Westlake University, China
[4]Kuaishou Technology, China
{jwang16,ztao}@scu.edu, yulun100@gmail.com, xyuan@westlake.edu.cn, mengziyi64@gmail.com

Abstract. Recently, hyperspectral imaging (HSI) has attracted increasing research attention, especially for the ones based on a coded aperture snapshot spectral imaging (CASSI) system. Existing deep HSI reconstruction models are generally trained on paired data to retrieve original signals upon 2D compressed measurements given by a particular optical hardware mask in CASSI, during which the mask largely impacts the reconstruction performance and could work as a "model hyperparameter" governing on data augmentations. This mask-specific training style will lead to a hardware miscalibration issue, which sets up barriers to deploying deep HSI models among different hardware and noisy environments. To address this challenge, we introduce mask uncertainty for HSI with a complete variational Bayesian learning treatment and explicitly model it through a mask decomposition inspired by real hardware. Specifically, we propose a novel Graph-based Self-Tuning (GST) network to reason uncertainties adapting to varying spatial structures of masks among different hardware. Moreover, we develop a bilevel optimization framework to balance HSI reconstruction and uncertainty estimation, accounting for the hyperparameter property of masks. Extensive experimental results validate the effectiveness (over 33/30 dB) of the proposed method under two miscalibration scenarios and demonstrate a highly competitive performance compared with the state-of-the-art well-calibrated methods. Our source code and pre-trained models are available at https://github.com/Jiamian-Wang/mask_uncertainty_spectral_SCI

1 Introduction

Hyperspectral imaging (HSI) provides richer signals than the traditional RGB vision and has broad applications across agriculture [28,30], remote sensing [54,59], medical imaging [17,29], etc. Various HSI systems have been built and studied in recent years, among which, the coded aperture snapshot spectral imaging (CASSI) system [11,52] stands out due to its passive modulation property and has attracted increasing research attentions [16,27,34,36,47,50,55] in the computer vision community. The CASSI system adopts a hardware encoding & software decoding schema. It first utilizes an optical hardware mask to compress



(a) Real mask decomposition (b) Miscalibration performance

Fig. 1. (a) A real mask $m \sim p(m)$ can be decomposed into an unknown clean mask $\tilde{m}$ plus random noise z. The mask distribution is plotted by realistic hardware mask values. Note that the distributions are demonstrated in a `symlog` scale. (b) Performance comparison under three different settings, including 1) the same mask for training/testing, 2) training on one mask and testing on multiple masks (one-to-many), and 3) training with random masks and testing on a held-out mask set (many-to-many).

hyperspectral signals into a 2D measurement and then develops software algorithms to retrieve original signals upon the coded measurement conditioning on one particular mask used in the system. Therefore, the hardware mask generally plays a key role in reconstructing hyperspectral images and may exhibit a strongly-coupled (*one-to-one*) relationship with its reconstruction model.

While deep HSI networks [16,34,44,46,56] have shown a promising performance on high-fidelity reconstruction and real-time inference, they mainly treat the hardware mask as a fixed "model hyperparameter" (governing data augmentations on the compressed measurements) and train the reconstruction network on the paired hyperspectral images and measurements given the same mask. Empirically, this will cause a **hardware miscalibration** issue – the mask used in the pre-trained model is mismatched with the real captured measurement, when 1) deploying a single deep network among arbitrary uncalibrated hardware systems of different masks, or 2) having distinct responses of the same mask due to the fabrication errors. As shown in Fig. 1, the performance of deep reconstruction networks pre-trained with one specific mask will badly degrade when applied to multiple unseen masks (*one-to-many*). Rather than re-training models on each new mask, which is inflexible for practical usage, we are more interested in training a single model that could adapt to different hardware by exploring and exploiting uncertainties among masks.

One possible solution is to train a deep network over multiple CASSI systems, *i.e.*, using multiple masks and their corresponding encoded measurements following the deep ensemble [23] strategy. However, due to the distinct spatial patterns of each mask and its hyperparameter property, directly training the network with randomly sampled masks still cannot achieve a well-calibrated performance and sometimes performs even worse, *e.g.*, *many-to-many* of TSA-

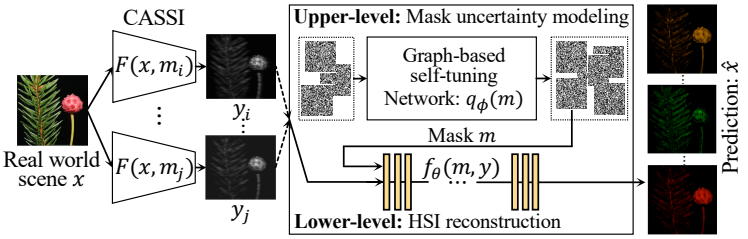


Fig. 2. Illustration of the proposed bilevel optimization framework. The upper-level models the mask uncertainty by approximating a mask posterior distribution, and the lower-level adopts a reconstruction network $f_\theta(m, y)$ which takes masks as hyperparameters. Our model could be applied in multiple CASSIs using different masks.

Net [34] in Fig. 1. Hence, we delve into one possible mask decomposition observed from the real hardware, which treats a mask as the unknown clean one plus random noise like Gaussian (see Fig. 1). We consider the noise stemming from two practical sources: 1) the hardware fabrication in real CASSI systems and 2) the functional mask values caused by different lighting environments. Notably, rather than modeling the entire mask distribution, which is challenging due to the high-dimensionality of a 2D map, we explicitly model the mask uncertainty as Gaussian noise centering around a given mask through its decomposition and resort to learn *self-tuning variances* adapting to different mask spatial patterns.

In this study, we propose a novel Graph-based Self-Tuning (GST) network to model mask uncertainty upon variational Bayesian learning and hyperparameter optimization techniques. On the one hand, we approximate the mask posterior distribution with variational inference under the given prior from real mask values, leading to a smoother mask distribution with smaller variance supported by empirical evidence. On the other hand, we leverage graph convolution neural networks to instantiate a stochastic encoder to reason uncertainties varying to different spatial structures of masks. Moreover, we develop a bilevel optimization framework (Fig. 2) to balance the HSI reconstruction performance and the mask uncertainty estimation, accounting for the high sensitive network responses to the mask changes. We summarize the contributions of this work as follows.

- We introduce *mask uncertainty* for CASSI to calibrate a single deep reconstruction network applying in multiple hardware, which brings a promising research direction to improve the robustness and flexibility of deploying CASSI systems to retrieve hyperspectral signals in real-world applications. To our best knowledge, this is the first work to explicitly explore and model mask uncertainty in the HSI reconstruction problem.
- A complete variational Bayesian learning framework has been provided to approximate the mask posterior distribution based on a mask decomposition inspired by real hardware mask observations. Moreover, we design and develop a bilevel optimization framework (see Fig. 2) to jointly achieve high-fidelity HSI reconstruction and mask uncertainty estimation.

- We propose a novel Graph-based Self-Tuning (GST) network to automatically capture uncertainties varying to different spatial structures of 2D masks, leading to a smoother mask distribution over real samples and working as an effective data augmentation method.
- Extensive experimental results on both simulation and real data demonstrate the effectiveness (over 33/30 dB) of our approach under two miscalibration cases. Our method also shows a highly competitive performance compared with state-of-the-art methods under the traditional well-calibrated setting.

2 Related Work

Recently, many advanced algorithms have been designed from diverse perspectives to reconstruct the HSI data from measurements encoded by CASSI system. Among them, the optimization-based methods solve the problem by introducing different priors, *e.g.*, GPSR [7], TwIST [2] GAP-TV [51], and DeSCI [27]. Another mainstream direction is to empower optimization-based method by deep learning. For example, deep unfolding methods [13,31,45] and Plug-and-Play (PnP) structures [36,38,39,53] have been raised. Despite their interpretability and robustness to masks to a certain degree, they may suffer from low efficiency and unstable convergence. Besides, a number of deep reconstruction networks [4,15,16,25,34,35,37,44] have been proposed for HSI, yielding the state-of-the-art performance with high inference efficiency. For instances, TSA-Net [34] retrieves hyperspectral images through modeling spatial and spectral attentions. SRN [44] provides a lightweight reconstruction backbone based on nested residual learning. More recently, a Gaussian Scale Mixture (GSM) based method [16] shows robustness on masks by enabling an approximation on multiple sensing matrices. However, all the above pre-trained networks perform unsatisfactorily on distinct unseen masks, raising the question of how to deploy a single reconstruction network among different hardware systems.

Previous works mainly consider mask calibration from a hardware perspective. For example, a high-order model [1] is proposed to calibrate masks with a single fixed wavelength for various wavelengths adaptation, enabling a band-limited signal approximation. One recent work [41] proposes to calibrate the point-spread-function upon existing CASSI setups for better quality. Yet, the impact of software (reconstruction model) has been barely considered in the mask calibration process. In this work, we calibrate a single deep reconstruction network to adapt to different real masks (hardware systems) by estimating mask uncertainties with a Bayesian variational approach. Popular uncertainty estimation methods include 1) Bayesian neural networks (BNN) [3,10,32] and 2) deep ensemble [8,23,26]. The former usually approximates the weight posterior distribution by using variational inference [3] or MC-dropout [10], while the latter generally trains a group of networks from random weight initializations. However, it is challenging to directly quantify mask uncertainty via BNNs or deep ensemble, since treating masks as model weights contraries to their hyperparameter properties. The proposed method solves this challenge by marrying uncertainty estimation to hyperparameter optimization in a bilevel framework [33,42,58].

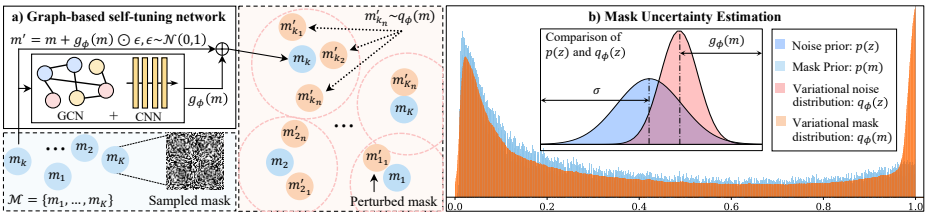


Fig. 3. Illustration of modeling mask uncertainty with the proposed Graph-based Self-Tuning (GST) network. a) GST takes as input a real mask m_k randomly sampled from different hardware masks $\mathcal{M}$ and obtains perturbed masks m'_{k_n} by learning self-tuning variance centering on m_k. b) GST estimates mask uncertainty by approximating the mask posterior with a variational distribution $q_\phi(m)$, leading to a smoother mask distribution over the mask prior $p(m)$. More discussions are given in Section 4.2.

3 Methodology

3.1 Preliminaries

HSI reconstruction. The reconstruction based on the CASSI system [34,52] generally includes a hardware-encoding forward process and a software-decoding inverse process. Let x be a 3D hyperspectral image with the size of $H \times W \times \Lambda$, where H, W, and Λ represent the height, width, and the number of spectral channels. The optical hardware encoder will compress the datacube x into a 2D measurement y upon a fixed physical mask m. The forward model of CASSI is

$$y = F(x; m) = \sum\nolimits_{\lambda}^{\Lambda} \texttt{shift}(x)_\lambda \odot \texttt{shift}(m)_\lambda + \zeta, \tag{1}$$

where λ refers to a spectral channel, $\odot$ represents the element-wise product, and ζ denotes the measurement noise. The shift operation is implemented by a single disperser as $\texttt{shift}(x)(u, v, i) = x(h, w + d(i - \lambda), i)$. In essence, the measurement y is captured by spectral modulation[1] conditioning on the hardware mask m.

In the inverse process, we adopt a deep reconstruction network as the decoder: $\widehat{x} = f_\theta(m, y)$ where $\widehat{x}$ is the retrieved hyperspectral image, and θ represents all the learnable parameters. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be the dataset. The reconstruction network f_θ is generally trained to minimize an ℓ_1 or ℓ_2 loss as the following:

$$\min_\theta \sum_{x,y \in \mathcal{D}} \ell(f_\theta(m, y) - x) \quad \text{where} \quad y = F(x; m). \tag{2}$$

We instantiate f_θ as a recent HSI backbone model provided in [44], which benefits from nested residual learning and spatial/spectral-invariant learning. We employ this backbone for its lightweight structure to simplify the training.

[1] We used a two-pixel shift for neighbored spectral channels following [34,44]. More details about spectral modulation could be found in [52].

Hardware miscalibration. As shown in Eq. (2), there is a *paired relationship* between the parameter θ and mask m in deep HSI models. Thus, for different CASSI systems (i.e., distinct masks), multiple pairs $\{m_1;\theta_1\},...,\{m_K;\theta_K\}$ are expected for previous works. Empirically, the miscalibration between m and θ will lead to obvious performance degradation. This miscalibration issue inevitably impairs the flexibility and robustness of deploying deep HSI models across real systems, considering the expensive training time and various noises existing in hardware. To alleviate such a problem, one straight-forward solution is to train the model f_θ with multiple masks, i.e., $\mathcal{M} = \{m_1,...,m_K\}$, falling in a similar strategy to deep ensemble [23]. However, directly training a single network with random masks cannot provide satisfactory performance to unseen masks (see Section 4), since the lack of explicitly exploring the relationship between uncertainties and different mask structures.

3.2 Mask Uncertainty

Modeling mask uncertainty is challenging due to the high dimensionality of a 2D mask, limited mask set size (*i.e.*, K for $\mathcal{M}$), and the varying spatial structures among masks. In this section, we first estimate uncertainties around each mask through one possible mask decomposition, and then we adapt the mask uncertainty to the change of mask structures with a self-tuning network in Section 3.3.

Inspired by the distribution of real mask values (Fig. 1 and Fig. 3), which renders two peaks at 0 and 1 and appears a Gaussian shape spreading over the middle, we decompose a mask as two components:

$$m = \tilde{m} + z, \tag{3}$$

where we assume each pixel in z follows a Gaussian distribution. For simplicity, we slightly abuse the notations by denoting the noise prior as $p(z) = \mathcal{N}(\mu, \sigma)$. The $\tilde{m}$ denotes the underlying clean binary mask with a specific spatial structure.

We estimate the mask uncertainty by approximating the mask posterior $p(m|X,Y)$ following [3,9,49], where $X = \{x_1,\dots,x_N\}$ and $Y = \{y_1,\dots,y_N\}$ indicate hyperspectral images and their corresponding measurements. To this end, we aim to learn a variational distribution $q_\phi(m)$ parameterized by ϕ to minimize the KL-divergence between $q_\phi(m)$ and $p(m|X,Y)$, $\min_\phi KL[q_\phi(m)||p(m|X,Y)]$, equivalent to maximizing the evidence lower bound (ELBO) [14,21] as

$$\max_{\phi} \underbrace{\mathbb{E}_{q_\phi(m)}[\log p(X|Y,m)]}_{\text{reconstruction}} - \underbrace{KL[q_\phi(m)||p(m)]}_{\text{regularization}}, \tag{4}$$

where the first term measures the reconstruction (i.e., reconstructing the observations X based on the measurements Y and mask m via $f_\theta(m,y)$), and the second term regularizes $q_\phi(m)$ given the mask prior $p(m)$. Following the mask decomposition in (3), we treat the clean mask $\tilde{m}$ as a 2D constant and focus on mask uncertainties arising from the noise z. Thus, the variational distribution $q_\phi(m)$ is defined as a Gaussian distribution centering on a given $m \in \mathcal{M}$ by

$$q_\phi(m) = \mathcal{N}(m, g_\phi(m)), \tag{5}$$

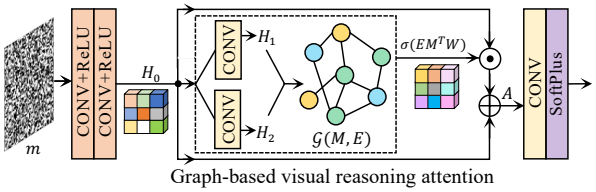


Fig. 4. Structure of Graph-based Self-Tuning (GST) network. The model takes mask m as input and outputs a 2D variance map, globally handling mask in a graph domain.

where $g_\phi(m)$ learns *self-tuning variance* to model the uncertainty adapting to real masks sampled from $\mathcal{M}$. Correspondingly, the underlying variational noise distribution $q_\phi(z)$ follows Gaussian distribution with variance $g_\phi(m)$. We adopt the reparameterization trick [21] for computing stochastic gradients for the expectation w.r.t $q_\phi(m)$. Specifically, let $m' \sim q_\phi(m)$ be a random variable sampled from the variational distribution, we have

$$m' = t(\phi, \epsilon) = m + g_\phi(m) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, 1). \tag{6}$$

Notably, we clamp all the pixel values of m' in range $[0, 1]$.

The first term in Eq. (4) reconstructs x with $p(x|y, m) \propto p(x|\widehat{x} = f_\theta(m, y))$, yielding a squared error when $x|\widehat{x}$ follows a Gaussian distribution [43]. Similar to AutoEncoders, we implement the negative log-likelihood $\mathbb{E}_{q_\phi(m)}[-\log p(X|Y, m)]$ as a ℓ_2 loss and compute its Monte Carlo estimates with Eq. (6) as

$$\ell(\phi, \theta; \mathcal{D}) = \frac{N}{B} \sum_{i=1}^{B} \|f_\theta(y_i, t(\phi, \epsilon_i)) - x_i\|^2, \tag{7}$$

where $(x_i, y_i) \in \mathcal{D}$, B denotes the mini-batch size, and $t(\phi, \epsilon_i)$ represents the i-th sample from $q_\phi(m)$. We leverage $t(\phi, \epsilon_i)$ to sample B perturbed masks from $q_\phi(m)$ centering on one randomly sampled mask $m \in \mathcal{M}$ per batch.

Since $p(m)$ is unknown due to various spatial structures of masks, we resort to approximating the KL term in Eq. (4) with the entropy of $q_\phi(m)$. Eventually, we implement the ELBO($q(m)$) with the following loss:

$$\mathcal{L}(\phi, \theta; \mathcal{D}) = \ell(\phi, \theta; \mathcal{D}) + \beta \mathbb{H}[\log q_\phi(m)], \tag{8}$$

where $\mathbb{H}[\log q_\phi(m)]$ is computed by $\ln(g_\phi(m)\sqrt{2\pi e})$ and $\beta{>}0$ interprets the objective function between variational inference and variational optimization [19,33].

3.3 Graph-based Self-Tuning Network

We propose a graph-based self-tuning (GST) network to instantiate the variance model $g_\phi(m)$ in Eq. (5), which captures uncertainties around each mask and leads to a smoother mask distribution over real masks (see Fig. 3). The key of handling unseen masks (new hardware) is to learn how the distribution will change along

with the varying spatial structures of masks. To this end, we implement the GST as a visual reasoning attention network [5,24,57]. It firstly computes pixel-wise correlations (visual reasoning) based on neural embeddings and then generates attention scores based on graph convolutional networks (GCN) [22,40]. Unlike previous works [5,24,57], the proposed GST model is tailored for building a stochastic probabilistic encoder to capture the mask distribution.

We show the network structure of GST in Fig. 4. Given a real mask m, GST produces neural embedding H_0 by using two concatenated `CONV-ReLU` blocks. Then, we employ two `CONV` layers to convert H_0 into two different embeddings H_1 and H_2, and generate a graph representation by matrix multiplication $H_1^T H_2$, resulting in $\mathcal{G}(M, E)$, where the node matrix M represents mask pixels and the edge matrix E denotes the pixel-wise correlations. Let W be the weight matrix of GCN. We obtain an enhanced attention cube by pixel-wise multiplication

$$A = H_0 \odot (\sigma(EM^T W) + \mathbf{1}), \tag{9}$$

where σ is the sigmoid function. Finally, the self-tuning variance is obtained by

$$g_\phi(m) = \delta(\texttt{CONV}(A)), \tag{10}$$

where δ denotes the softplus function and ϕ denotes all the learnable parameters. Consequently, GST enables adaptive variance modeling to multiple real masks.

3.4 Bilevel Optimization

Algorithm 1: GST Training Algorithm

Input: $\mathcal{D}^{trn}, \mathcal{D}^{val}, \mathcal{M}$; initialized θ, ϕ;
Output: θ^*, ϕ^*
Pre-train $f_\theta(\cdot)$ on $\mathcal{D}^{trn}$ with α_0 for T^{init} epochs;
while *not converge* **do**
  for $t = 1, ..., T^{trn}$ **do**
    $\{(x_i, y_i)\}_{i=1}^B \sim \mathcal{D}^{trn}$;
    $\theta \leftarrow \theta - \alpha_1 \frac{\partial}{\partial \theta} \ell(\phi, \theta; \mathcal{D}^{trn})$;
  end
  for $t = 1, ..., T^{val}$ **do**
    $\{(x_i, y_i)\}_{i=1}^B \sim \mathcal{D}^{val}, m \sim \mathcal{M}, \epsilon \sim \mathcal{N}(0, 1)$;
    $\phi \leftarrow \phi - \alpha_2 \frac{\partial}{\partial \phi} \mathcal{L}(\phi, \theta; \mathcal{D}^{val})$;
  end
end

While it is possible to jointly train the HSI reconstruction network f_θ and the self-tuning network g_ϕ using the Eq. (8), it is more proper to formulate the training of these two networks as a bilevel optimization problem accounting for the hyperparameter properties of masks. Deep HSI methods [34,44] usually employ a single mask and shifting operations to lift the 2D measurements as multi-channel inputs, where the mask works as a hyperparameter similar to the data augmentation purpose. Thus, the reconstruction network is highly-sensitive to the change/perturbation of masks (model weight θ is largely subject to a mask m).

To be specific, we define the lower-level problem as HSI reconstruction and the upper-level problem as mask uncertainty estimation, and propose the final objective function of our GST model as the following:

$$\min_{\phi} \mathcal{L}(\phi, \theta^*; \mathcal{D}^{val}) \quad \text{s.t.} \quad \theta^* = \operatorname*{argmin}_{\theta} \ell(\phi, \theta; \mathcal{D}^{trn}), \tag{11}$$

where $\ell(\phi, \theta; \mathcal{D}^{trn})$ is provided in Eq. (7) with a training set and $\mathcal{L}(\phi, \theta^*; \mathcal{D}^{val})$ is given by Eq. (8) in a validation set. Upon Eq. (11), f_θ and g_ϕ are alternatively updated by computing gradients $\frac{\partial l}{\partial \theta}$ and $\frac{\partial \mathcal{L}}{\partial \phi}$. To better initialize the parameter θ, we pre-train the reconstruction network $f_\theta(m, y)$ for several epochs. The entire training procedure of the proposed method is summarized in Algorithm 1. Notably, introducing Eq. (11) brings two benefits. 1) It could balance the solutions of HSI reconstruction and mask uncertainty estimation. 2) It enables the proposed GST as a hyperparameter optimization method, which could provide high-fidelity reconstruction even working on a single mask (see Table 3).

4 Experiments

Simulation data. We adopt the training set provided in [34]. Simulated measurements are obtained by mimicking the compressing process of SD-CASSI system [34]. For metric and perceptual comparisons, we employ a benchmark test set that contains ten 256×256×28 hyperspectral images following [16,36,44]. We build a validation set by splitting 40 hyperspectral images from the training set.

Real data. We adopt five real 660×714 measurements provided in [34] for the qualitative evaluation. We train the model on the expanded simulation training set by augmenting 37 HSIs originating from the KAIST dataset [6]. Also, the Gaussian noise ($\mathcal{N}(0, \varphi), \varphi \sim U[0, 0.05]$) is added on the simulated measurements during training, for the sake of mimicking practical measurement noise ζ. All the other settings are kept the same as the compared deep reconstruction methods.

Mask set. We adopt two 660×660 hardware masks in our experiment. Both are produced by the same fabrication process. For the training, the mask set $\mathcal{M}$ is created by randomly cropping (256×256) from the mask provided in [34]. For the testing, both masks are applied. In simulation, testing masks are differentiated from the training ones. For real HSI reconstruction, the second mask [35] is applied, indicating a hardware miscalibration scenario.

Implementation details. The training procedures (Algorithm 1) for simulation and real case follow the same schedule: We apply the `xavier uniform` [12] initializer with `gain=1`. Before alternating, the reconstruction network is trained for T^{init}=20 epochs (learning rate α_0=4× 10^{-4}). Then, the reconstruction network $f_\theta(\cdot)$ is updated on training phase for T^{trn}=5 epochs (α_1=4× 10^{-4}) and the GST network is updated on validation phase for T^{val}=3 epochs (α_2=1× 10^{-5}). The learning rates are halved per 50 epochs and we adopt Adam optimizer [20] with the default setting. In this work, we adopt SRN (`v1`) [44] as the reconstructive backbone, i.e., the full network without rescaling pairs. All the experiments were conducted on four NVIDIA GeForce GTX 3090 GPUs.

Compared methods. For hardware miscalibration, masks for data pair setup (i.e., CASSI compressing procedure) and network training should be different from those for testing. We specifically consider two scenarios: 1) many-to-many, i.e., training the model on mask set $\mathcal{M}$ and testing it by unseen masks; 2) One-to-many, i.e., training the model on single mask and testing it by diverse unseen masks, which brings more challenges. For quantitative performance

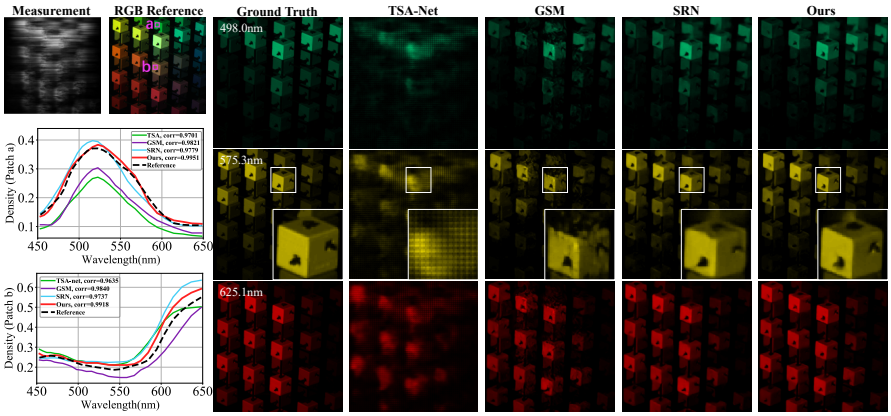


Fig. 5. Reconstruction results on one simulation scene under hardware miscalibration (many-to-many). All methods are trained on the mask set $\mathcal{M}$ and tested by one unseen mask. Density curves computed on chosen patches are compared to analysis the spectra.

comparison, in this work all the testing results are computed upon 100 testing trials (100 random unseen masks). We compare with four state-of-the-art methods: TSA-Net [34], GSM-based method [16], SRN [44], and PnP-DIP [36], among which the first three are deep networks and the last one is an iterative optimization-based method. Note that 1) PnP-DIP is a self-supervised method. We test it by feeding the data encoded by different masks in the testing mask set and compute the performance over all obtained results. 2) For real-world HSI reconstruction, all models are trained on the same mask while tested on the other. Specifically, the network inputs are initialized by testing mask for TSA-Net and SRN. For GSM, as demonstrated by the authors, we directly compute the sensing matrix of testing mask and replace the corresponding approximation in the network. We use PSNR and SSIM [48] as metrics for quantitative comparison.

4.1 HSI Reconstruction Performance

We evaluate our method under different settings on both simulation and real data. More visualizations and analyses are provided in the supplementary.

Miscalibration (many-to-many). Training the deep reconstruction networks with a mask ensemble strategy could improve the generalization ability, such as training TSA-Net, GSM, and SRN on a mask set. However, as shown in Table 1 and Table 3, these methods generally suffer from a clear performance degradation under miscalibration compared with their well-calibrated performance. Benefiting from modeling mask uncertainty, our approach achieves high-fidelity results (over 33dB) on both cases, with only a 0.2dB drop. As shown in Fig. 5, our method retrieves more details at different spectral channels.

Miscalibration (one-to-many). In Table 2, all the methods are trained on a single mask and tested on multiple unseen masks. We pose this setting to

Table 1. PSNR(dB)/SSIM by different methods on 10 simulation scenes under the **many-to-many** hardware miscalibration. All the methods are trained with a mask set $\mathcal{M}$ and tested by random unseen masks. TSA-Net [34], GSM [16], and SRN [44] are obtained with a mask ensemble strategy. We report $\texttt{mean}_{\pm\texttt{std}}$ among 100 testing trials.

Scene	TSA-Net [34]		GSM [16]		PnP-DIP† [36]		SRN [44]		GST (Ours)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	$23.45_{\pm0.29}$	$0.6569_{\pm0.0051}$	$31.38_{\pm0.20}$	$0.8826_{\pm0.0032}$	$29.24_{\pm0.98}$	$0.7964_{\pm0.0532}$	$33.26_{\pm0.16}$	$0.9104_{\pm0.0018}$	$\mathbf{33.99}_{\pm0.14}$	$\mathbf{0.9258}_{\pm0.0013}$
2	$18.52_{\pm0.12}$	$0.5511_{\pm0.0049}$	$25.94_{\pm0.22}$	$0.8570_{\pm0.0041}$	$25.73_{\pm0.54}$	$0.7558_{\pm0.0117}$	$29.86_{\pm0.23}$	$0.8809_{\pm0.0029}$	$\mathbf{30.49}_{\pm0.17}$	$\mathbf{0.9002}_{\pm0.0022}$
3	$18.42_{\pm0.30}$	$0.5929_{\pm0.0127}$	$26.11_{\pm0.20}$	$0.8874_{\pm0.0034}$	$29.61_{\pm0.45}$	$0.8541_{\pm0.0125}$	$31.69_{\pm0.20}$	$0.9093_{\pm0.0020}$	$\mathbf{32.63}_{\pm0.16}$	$\mathbf{0.9212}_{\pm0.0013}$
4	$30.44_{\pm0.15}$	$0.8940_{\pm0.0043}$	$34.72_{\pm0.35}$	$0.9473_{\pm0.0023}$	$38.21_{\pm0.66}$	$0.9280_{\pm0.0078}$	$39.90_{\pm0.22}$	$0.9469_{\pm0.0012}$	$\mathbf{41.04}_{\pm0.23}$	$\mathbf{0.9667}_{\pm0.0014}$
5	$20.89_{\pm0.23}$	$0.5648_{\pm0.0077}$	$26.15_{\pm0.24}$	$0.8256_{\pm0.0061}$	$28.59_{\pm0.79}$	$0.8481_{\pm0.0183}$	$30.86_{\pm0.16}$	$0.9232_{\pm0.0019}$	$\mathbf{31.49}_{\pm0.17}$	$\mathbf{0.9379}_{\pm0.0017}$
6	$23.04_{\pm0.19}$	$0.6099_{\pm0.0060}$	$30.97_{\pm0.29}$	$0.9224_{\pm0.0025}$	$29.70_{\pm0.51}$	$0.8484_{\pm0.0186}$	$34.20_{\pm0.23}$	$0.9405_{\pm0.0014}$	$\mathbf{34.89}_{\pm0.29}$	$\mathbf{0.9545}_{\pm0.0009}$
7	$15.97_{\pm0.14}$	$0.6260_{\pm0.0042}$	$22.58_{\pm0.24}$	$0.8459_{\pm0.0054}$	$27.13_{\pm0.31}$	$0.8666_{\pm0.0079}$	$27.27_{\pm0.16}$	$0.8515_{\pm0.0026}$	$\mathbf{27.63}_{\pm0.16}$	$\mathbf{0.8658}_{\pm0.0024}$
8	$22.64_{\pm0.18}$	$0.6366_{\pm0.0066}$	$29.76_{\pm0.22}$	$0.9059_{\pm0.0021}$	$28.38_{\pm0.35}$	$0.8325_{\pm0.0203}$	$32.35_{\pm0.22}$	$0.9320_{\pm0.0015}$	$\mathbf{33.02}_{\pm0.26}$	$\mathbf{0.9471}_{\pm0.0013}$
9	$18.91_{\pm0.11}$	$0.5946_{\pm0.0083}$	$27.23_{\pm0.11}$	$0.8899_{\pm0.0021}$	$33.63_{\pm0.26}$	$0.8779_{\pm0.0073}$	$32.83_{\pm0.13}$	$0.9205_{\pm0.0016}$	$\mathbf{33.45}_{\pm0.13}$	$\mathbf{0.9317}_{\pm0.0013}$
10	$21.90_{\pm0.18}$	$0.5249_{\pm0.0110}$	$28.05_{\pm0.21}$	$0.8877_{\pm0.0055}$	$27.24_{\pm0.43}$	$0.7957_{\pm0.0226}$	$30.25_{\pm0.14}$	$0.9053_{\pm0.0019}$	$\mathbf{31.49}_{\pm0.15}$	$\mathbf{0.9345}_{\pm0.0015}$
Avg.	$21.42_{\pm0.07}$	$0.6162_{\pm0.0030}$	$28.20_{\pm0.01}$	$0.8852_{\pm0.0001}$	$29.66_{\pm0.38}$	$0.8375_{\pm0.0093}$	$32.24_{\pm0.10}$	$0.9121_{\pm0.0010}$	$\mathbf{33.02}_{\pm0.01}$	$\mathbf{0.9285}_{\pm0.0001}$

†PnP-DIP is a mask-free method which reconstructs from measurements encoded by random masks.

Table 2. PSNR(dB)/SSIM by different methods on 10 simulation scenes under the **one-to-many** hardware miscalibration. All the methods are trained by a single mask and tested by random unseen masks. We report $\texttt{mean}_{\pm\texttt{std}}$ among 100 testing trials.

Scene	TSA-Net [34]		GSM [16]		PnP-DIP† [36]		SRN [44]		GST (Ours)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	$28.49_{\pm0.58}$	$0.8520_{\pm0.0081}$	$28.20_{\pm0.95}$	$0.8553_{\pm0.0185}$	$29.24_{\pm0.98}$	$0.7964_{\pm0.0532}$	$31.24_{\pm0.77}$	$0.8878_{\pm0.0117}$	$\mathbf{31.72}_{\pm0.76}$	$\mathbf{0.8939}_{\pm0.0119}$
2	$24.96_{\pm0.51}$	$0.8332_{\pm0.0064}$	$24.46_{\pm0.96}$	$0.8330_{\pm0.0189}$	$25.73_{\pm0.54}$	$0.7558_{\pm0.0117}$	$27.87_{\pm0.82}$	$0.8535_{\pm0.0131}$	$\mathbf{28.22}_{\pm0.85}$	$\mathbf{0.8552}_{\pm0.0144}$
3	$26.14_{\pm0.76}$	$0.8829_{\pm0.0108}$	$23.71_{\pm1.18}$	$0.8077_{\pm0.0221}$	$29.61_{\pm0.45}$	$0.8541_{\pm0.0125}$	$28.31_{\pm0.88}$	$\mathbf{0.8415}_{\pm0.0213}$	$28.77_{\pm1.13}$	$0.8405_{\pm0.0257}$
4	$35.67_{\pm0.47}$	$0.9427_{\pm0.0028}$	$31.55_{\pm0.75}$	$0.9385_{\pm0.0074}$	$38.21_{\pm0.66}$	$0.9280_{\pm0.0078}$	$\mathbf{37.93}_{\pm0.72}$	$\mathbf{0.9476}_{\pm0.0057}$	$37.60_{\pm0.81}$	$0.9447_{\pm0.0071}$
5	$25.40_{\pm0.59}$	$0.8280_{\pm0.0108}$	$24.44_{\pm0.96}$	$0.7744_{\pm0.0291}$	$28.59_{\pm0.79}$	$0.8481_{\pm0.0183}$	$27.99_{\pm0.79}$	$0.8680_{\pm0.0194}$	$\mathbf{28.58}_{\pm0.79}$	$\mathbf{0.8746}_{\pm0.0208}$
6	$29.32_{\pm0.60}$	$0.8796_{\pm0.0047}$	$28.28_{\pm0.92}$	$0.9026_{\pm0.0094}$	$29.70_{\pm0.51}$	$0.8484_{\pm0.0186}$	$32.13_{\pm0.87}$	$\mathbf{0.9344}_{\pm0.0061}$	$\mathbf{32.72}_{\pm0.79}$	$0.9339_{\pm0.0061}$
7	$22.80_{\pm0.65}$	$0.8461_{\pm0.0101}$	$21.45_{\pm0.79}$	$0.8147_{\pm0.0162}$	$27.13_{\pm0.31}$	$0.8666_{\pm0.0079}$	$24.84_{\pm0.73}$	$0.7973_{\pm0.0150}$	$\mathbf{25.15}_{\pm0.76}$	$\mathbf{0.7935}_{\pm0.0173}$
8	$28.09_{\pm0.43}$	$0.8738_{\pm0.0043}$	$28.08_{\pm0.76}$	$0.9024_{\pm0.0089}$	$28.38_{\pm0.35}$	$0.8325_{\pm0.0203}$	$31.32_{\pm0.59}$	$0.9324_{\pm0.0043}$	$\mathbf{31.84}_{\pm0.56}$	$\mathbf{0.9323}_{\pm0.0042}$
9	$27.75_{\pm0.55}$	$0.8865_{\pm0.0054}$	$26.80_{\pm0.78}$	$0.8773_{\pm0.0144}$	$33.63_{\pm0.26}$	$0.8779_{\pm0.0073}$	$31.06_{\pm0.66}$	$0.8997_{\pm0.0091}$	$\mathbf{31.11}_{\pm0.72}$	$\mathbf{0.8988}_{\pm0.0104}$
10	$26.05_{\pm0.48}$	$0.8114_{\pm0.0072}$	$26.40_{\pm0.77}$	$0.8771_{\pm0.0124}$	$27.24_{\pm0.43}$	$0.7957_{\pm0.0226}$	$29.01_{\pm0.61}$	$0.9028_{\pm0.0092}$	$\mathbf{29.50}_{\pm0.68}$	$\mathbf{0.9030}_{\pm0.0098}$
Avg.	$27.47_{\pm0.46}$	$0.8636_{\pm0.0060}$	$26.34_{\pm0.06}$	$0.8582_{\pm0.0012}$	$29.66_{\pm0.38}$	$0.8375_{\pm0.0093}$	$30.17_{\pm0.63}$	$0.8865_{\pm0.0108}$	$\mathbf{30.60}_{\pm0.08}$	$\mathbf{0.8881}_{\pm0.0013}$

†PnP-DIP is a mask-free method which reconstructs from measurements encoded by random masks.

further demonstrate the hardware miscalibration challenge. Except for the mask-free method PnP-DIP, the others usually experience large performance descent compared with those in Table 1. This observation supports the motivation of modeling mask uncertainty – 1) simply using mask ensemble may aggravate the miscalibration (TSA-Net using ensemble performs even worse) and 2) the model trained with a single mask cannot be effectively deployed in different hardware.

Same mask (one-to-one). Table 3 reports the well-calibrated performance for all the methods, *i.e.*, training/testing models on the same real mask. While our approach is specially designed for training with multiple masks, it still consistently outperforms all the competitors by leveraging a bilevel optimization.

Results on real data. Fig. 6 visualizes reconstruction results on the real dataset, where the *left* corresponds to the same mask and the *right* is under the one-to-many setting. For the same mask, the proposed method is supposed to perform comparably. For the one-to-many, we train all the models on a single real mask provided in [34] and test them on the other one [35]. The proposed method produces plausible results and improves over other methods visually.

Table 3. PSNR (dB) and SSIM values by different algorithms on the simulation dataset under the well-calibrated setting (training/test on the ***same mask***). We adopt the same 256×256 real mask provided in previous works [16,34] for a fair comparison.

Scene	λ-net [37]		HSSP [45]		TSA-Net [34]		GSM [16]		PnP-DIP [36]		SRN [44]		GST (Ours)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	30.82	0.8492	31.07	0.8577	31.26	0.8920	32.38	0.9152	31.99	0.8633	34.13	0.9260	**34.19**	**0.9292**
2	26.30	0.8054	26.30	0.8422	26.88	0.8583	27.56	0.8977	26.56	0.7603	30.60	0.8985	**31.04**	**0.9014**
3	29.42	0.8696	29.00	0.8231	30.03	0.9145	29.02	0.9251	30.06	0.8596	32.87	0.9221	**32.93**	**0.9224**
4	37.37	0.9338	38.24	0.9018	39.90	0.9528	36.37	0.9636	38.99	0.9303	**41.27**	**0.9687**	40.71	0.9672
5	27.84	0.8166	27.98	0.8084	28.89	0.8835	28.56	0.8820	29.09	0.8490	31.66	0.9376	**31.83**	**0.9415**
6	30.69	0.8527	29.16	0.8766	31.30	0.9076	32.49	0.9372	29.68	0.8481	35.14	**0.9561**	**35.14**	0.9543
7	24.20	0.8062	24.11	0.8236	25.16	0.8782	25.19	0.8860	27.68	0.8639	27.93	**0.8638**	**28.08**	0.8628
8	28.86	0.8307	27.94	0.8811	29.69	0.8884	31.06	0.9234	29.01	0.8412	33.14	**0.9488**	**33.18**	0.9486
9	29.32	0.8258	29.14	0.8676	30.03	0.8901	29.40	0.9110	33.35	0.8802	33.49	0.9326	**33.50**	**0.9332**
10	27.66	0.8163	26.44	0.8416	28.32	0.8740	30.74	0.9247	27.98	0.8327	31.43	**0.9338**	**31.59**	0.9311
Avg.	29.25	0.8406	28.93	0.8524	30.24	0.8939	30.28	0.9166	30.44	0.8529	33.17	0.9288	**33.22**	**0.9292**

Table 4. Ablation study and complexity analysis. All the methods are tested on simulation test set under the many-to-many setting with one NVIDIA RTX 3090 GPU. We report the PSNR (dB)/SSIM among 100 testing trials, the total training time, and the test time per sample. PnP-DIP is self-supervised, thus no training is required.

Settings	PSNR	SSIM	#params (M)	FLOPs (G)	Training (day)	Test (sec.)
TSA-Net [34]	$21.42_{\pm 0.07}$	$0.6162_{\pm 0.0030}$	44.25	110.06	1.23	0.068
GSM [16]	$28.20_{\pm 0.01}$	$0.8852_{\pm 0.0001}$	3.76	646.35	6.05	0.084
PnP-DIP [36]	$29.66_{\pm 0.38}$	$0.8375_{\pm 0.0093}$	33.85	64.26	–	482.78
`w/o GST`	$32.24_{\pm 0.10}$	$0.9121_{\pm 0.0010}$	1.25	81.84	1.14	0.061
`w/o Bi-Opt`	$32.43_{\pm 0.02}$	$0.9206_{\pm 0.0001}$	1.27	82.87	1.83	0.061
`w/o GCN`	$32.82_{\pm 0.01}$	$0.9262_{\pm 0.0001}$	1.27	82.78	1.63	0.062
Ours (full model)	$33.02_{\pm 0.01}$	$0.9285_{\pm 0.0001}$	1.27	82.87	2.56	0.062

4.2 Model Discussion

Ablation study. Table 4 compares the performance and complexity of the proposed `full model` with three ablated models as follows. 1) The model `w/o GST` is equivalent to training the reconstruction backbone SRN [44] with a mask ensemble strategy. 2) The model `w/o Bi-Opt` is implemented by training the proposed method without using the Bilevel optimization framework. 3) In the model `w/o GCN`, we replace the GCN module in GST with convolutional layers carrying a similar size of parameters. The bilevel optimization achieves 0.59dB improvement without overburdening the complexity. The GCN contributes 0.2dB with 0.09G FLOPs increase. Overall, the proposed GST yields 0.8dB improvement with negligible costs (i.e., +0.02M #params, +1.03G FLOPs, and +1.14 days training), and could be used in multiple unseen masks without re-training.

Complexity comparison. In Table 4, we further compare the complexity of the proposed method with several recent HSI methods. The proposed method possess one of the smallest model size. Besides, our method shows a comparable

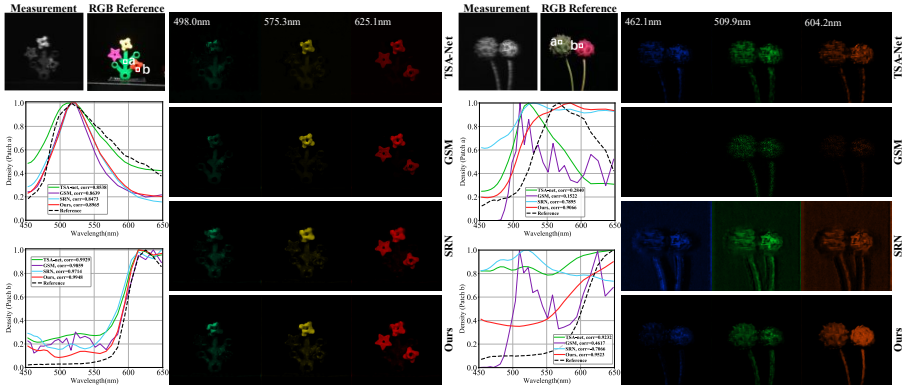


Fig. 6. Real HSI reconstruction. ***Left***: same mask (one-to-one) reconstruction, i.e., all methods are trained and tested on the same 660×660 real mask. ***Right***: miscalibration (one-to-many) setting, i.e., all methods are trained on a single mask and tested by unseen masks (Here we adopt another 660×660 real mask).

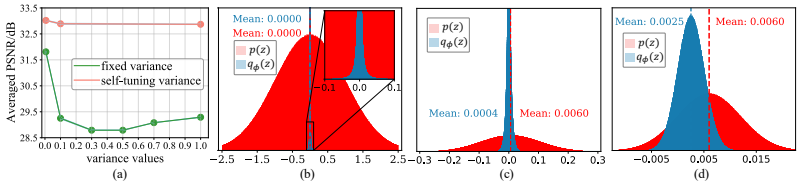


Fig. 7. Discussion on self-tuning variance. (a) Performance comparison between self-tuning variance and fixed ones. (b) The standard normal prior $\mathcal{N}(0, 1)$. (c) Set the prior as $\mathcal{N}(0.006, 0.1)$ by observing real masks. (d) Set the prior as $\mathcal{N}(0.006, 0.005)$ by observing real masks and the performance curve in (a).

FLOPs and training time as others. Notably, given M distinct masks, TSA-Net, GSM, and SRN require $M\times$ training time as reported to achieve well-calibrated performance. Instead, the proposed method only needs to be trained one time to provide calibrated reconstructions over multiple unseen masks.

Self-tuning variance under different priors. We first validate the effectiveness of the self-tuning variance by comparing it with the fix-valued variance, i.e., scalars from 0 to 1. As shown by the green curve in Fig. 7 (a), fixed variance only achieves less than 32dB performance. The best performance by 0.005 indicates a strong approximation nature to the mask noise. The self-tuning variance upon different noise priors achieves no less than 32.5dB performance (red curve in Fig. 7 (a)). Specifically, we implement the noise prior $p(z)$ by exchanging the standard normal distribution of auxiliary variable ϵ in Eq. (6). We start from $\mathcal{N}(0, 1)$, which is so broad that the GST network tries to centralize variational noise and restrict the randomness as Fig. 7 (b) shown. Then, we constraint the variance and approximate the mean value by the minimum of the real mask histogram to emphasize the near-zero noise, proposing $\mathcal{N}(0.006, 0.1)$.

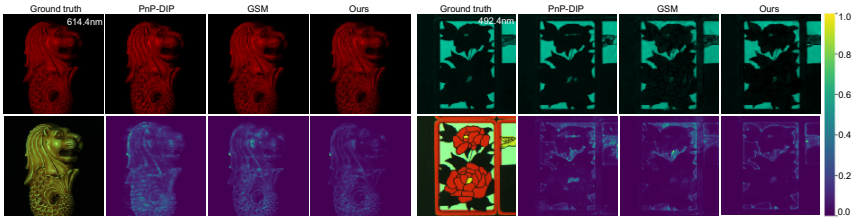


Fig. 8. Illustration of epistemic uncertainty induced by multiple masks. For each block, the first row shows the averaged reconstruction results of selected channels given by different methods and the second demonstrates the corresponding epistemic uncertainty.

Fig. 7 (c) indicates the underlying impact of GST network. We further combine the previous fixed-variance observation and propose $\mathcal{N}(0.006, 0.005)$. The best performance is obtained by observing the red curve in Fig. 7 (a). In summary, the proposed method restricts the posited noise prior, leading to the variational noise distribution with a reduced range.

From mask uncertainty to epistemic uncertainty. The hardware mask plays a similar role to model hyperparameter and largely impacts the weights of reconstruction networks. Thus, marginalizing over the mask posterior distribution will induce the epistemic uncertainty (also known as model uncertainty [9,18]) and reflect as pixel-wise variances (the second row in Fig. 8) of the reconstruction results over multiple unseen masks. As can be seen, the mask-free method PnP-DIP [36] still produces high uncertainties given measurements of the same scene coded by different hardware masks. While employing a deep ensemble strategy could alleviate this issue, such as training GSM [16] with mask ensemble, it lacks an explicit way to quantify mask uncertainty and may lead to unsatisfactory performance (see Table 1). Differently, the proposed GST method models mask uncertainty by approximating the mask posterior through a variational Bayesian treatment, exhibiting high-fidelity reconstruction result with low epistemic uncertainties across different masks as shown in Fig. 8.

5 Conclusions

In this work, we have explored a practical hardware miscalibration issue when deploying deep HSI models in real CASSI systems. Our solution is to calibrate a single reconstruction network via modeling mask uncertainty. We proposed a complete variational Bayesian learning treatment upon one possible mask decomposition inspired by observations on real masks. Bearing the objectives of variational mask distribution modeling and HSI retrieval, we introduced and implemented a novel Graph-based Self-Tuning (GST) network that proceeds HSI reconstruction and uncertainty reasoning under a bilevel optimization framework. The proposed method enabled a smoothed distribution and achieved promising performance under two different miscalibration scenarios. We hope the proposed insight will benefit future work in this novel research direction.

References

1. Arguello, H., Rueda, H., Wu, Y., Prather, D.W., Arce, G.R.: Higher-order computational model for coded aperture spectral imaging. Applied optics **52**(10), D12–D21 (2013) 4
2. Bioucas-Dias, J.M., Figueiredo, M.A.: A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. IEEE Transactions on Image processing **16**(12), 2992–3004 (2007) 4
3. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: ICML (2015) 4, 6
4. Cai, Y., Lin, J., Hu, X., Wang, H., Yuan, X., Zhang, Y., Timofte, R., Van Gool, L.: Mask-guided spectral-wise transformer for efficient hyperspectral image reconstruction. In: CVPR (2022) 4
5. Chen, X., Li, L.J., Fei-Fei, L., Gupta, A.: Iterative visual reasoning beyond convolutions. In: CVPR (2018) 8
6. Choi, I., Kim, M., Gutierrez, D., Jeon, D., Nam, G.: High-quality hyperspectral reconstruction using a spectral prior. Tech. rep. (2017) 9
7. Figueiredo, M.A., Nowak, R.D., Wright, S.J.: Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. IEEE Journal of selected topics in signal processing (2007) 4
8. Fort, S., Hu, H., Lakshminarayanan, B.: Deep ensembles: A loss landscape perspective. arXiv preprint arXiv:1912.02757 (2019) 4
9. Gal, Y.: Uncertainty in Deep Learning. Ph.D. thesis, University of Cambridge (2016) 6, 14
10. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: ICML (2016) 4
11. Gehm, M.E., John, R., Brady, D.J., Willett, R.M., Schulz, T.J.: Single-shot compressive spectral imaging with a dual-disperser architecture. Optics express **15**(21), 14013–14027 (2007) 1
12. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256. JMLR Workshop and Conference Proceedings (2010) 9
13. Hershey, J.R., Roux, J.L., Weninger, F.: Deep unfolding: Model-based inspiration of novel deep architectures. arXiv preprint arXiv:1409.2574 (2014) 4
14. Hoffman, M.D., Johnson, M.J.: Elbo surgery: yet another way to carve up the variational evidence lower bound. In: NeurIPS Workshop (2016) 6
15. Hu, X., Cai, Y., Lin, J., Wang, H., Yuan, X., Zhang, Y., Timofte, R., Van Gool, L.: Hdnet: High-resolution dual-domain learning for spectral compressive imaging. In: CVPR (2022) 4
16. Huang, T., Dong, W., Yuan, X., Wu, J., Shi, G.: Deep gaussian scale mixture prior for spectral compressive imaging. In: CVPR (2021) 1, 2, 4, 9, 10, 11, 12, 14
17. Johnson, W.R., Wilson, D.W., Fink, W., Humayun, M.S., Bearman, G.H.: Snapshot hyperspectral imaging in ophthalmology. Journal of biomedical optics **12**(1), 014036 (2007) 1
18. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: NeurIPS (2017) 14
19. Khan, M., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., Srivastava, A.: Fast and scalable bayesian deep learning by weight-perturbation in adam. In: ICML (2018) 7

20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 9
21. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013) 6, 7
22. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017) 8
23. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: NeurIPS (2017) 2, 4, 6
24. Li, K., Zhang, Y., Li, K., Li, Y., Fu, Y.: Visual semantic reasoning for image-text matching. In: ICCV (2019) 8
25. Lin, J., Cai, Y., Hu, X., Wang, H., Yuan, X., Zhang, Y., Timofte, R., Van Gool, L.: Coarse-to-fine sparse transformer for hyperspectral image reconstruction. arXiv preprint arXiv:2203.04845 (2022) 4
26. Liu, J.Z., Paisley, J., Kioumourtzoglou, M.A., Coull, B.: Accurate uncertainty estimation and decomposition in ensemble learning. arXiv preprint arXiv:1911.04061 (2019) 4
27. Liu, Y., Yuan, X., Suo, J., Brady, D.J., Dai, Q.: Rank minimization for snapshot compressive imaging. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(12), 2990–3006 (2018) 1, 4
28. Lorente, D., Aleixos, N., Gómez-Sanchis, J., Cubero, S., García-Navarrete, O.L., Blasco, J.: Recent advances and applications of hyperspectral imaging for fruit and vegetable quality assessment. Food and Bioprocess Technology **5**(4), 1121–1142 (2012) 1
29. Lu, G., Fei, B.: Medical hyperspectral imaging: a review. Journal of biomedical optics **19**(1), 010901 (2014) 1
30. Lu, R., Chen, Y.R.: Hyperspectral imaging for safety inspection of food and agricultural products. In: Pathogen Detection and Remediation for Safe Eating. vol. 3544, pp. 121–133. International Society for Optics and Photonics (1999) 1
31. Ma, J., Liu, X.Y., Shou, Z., Yuan, X.: Deep tensor admm-net for snapshot compressive imaging. In: ICCV (2019) 4
32. MacKay, D.J.C.: Bayesian Methods for Adaptive Models. Ph.D. thesis, California Institute of Technology (1992) 4
33. MacKay, M., Vicol, P., Lorraine, J., Duvenaud, D., Grosse, R.: Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions. arXiv preprint arXiv:1903.03088 (2019) 4, 7
34. Meng, Z., Ma, J., Yuan, X.: End-to-end low cost compressive spectral imaging with spatial-spectral self-attention. In: ECCV (2020) 1, 2, 3, 4, 5, 8, 9, 10, 11, 12
35. Meng, Z., Qiao, M., Ma, J., Yu, Z., Xu, K., Yuan, X.: Snapshot multispectral endomicroscopy. Optics Letters **45**(14), 3897–3900 (2020) 4, 9, 11
36. Meng, Z., Yu, Z., Xu, K., Yuan, X.: Self-supervised neural networks for spectral snapshot compressive imaging. In: ICCV (2021) 1, 4, 9, 10, 11, 12, 14
37. Miao, X., Yuan, X., Pu, Y., Athitsos, V.: l-net: Reconstruct hyperspectral images from a snapshot measurement. In: ICCV (2019) 4, 12
38. Qiao, M., Liu, X., Yuan, X.: Snapshot spatial–temporal compressive imaging. Optics letters **45**(7), 1659–1662 (2020) 4
39. Qiao, M., Meng, Z., Ma, J., Yuan, X.: Deep learning for video compressive sensing. Apl Photonics **5**(3), 030801 (2020) 4
40. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE transactions on neural networks **20**(1), 61–80 (2008) 8

41. Song, L., Wang, L., Kim, M.H., Huang, H.: High-accuracy image formation model for coded aperture snapshot spectral imaging. IEEE Transactions on Computational Imaging **8**, 188–200 (2022) 4
42. Tao, Z., Li, Y., Ding, B., Zhang, C., Zhou, J., Fu, Y.: Learning to mutate with hypergradient guided population. In: NeurIPS (2020) 4
43. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A., Bottou, L.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research **11**(12) (2010) 7
44. Wang, J., Zhang, Y., Yuan, X., Fu, Y., Tao, Z.: A new backbone for hyperspectral image reconstruction. arXiv preprint arXiv:2108.07739 (2021) 2, 4, 5, 8, 9, 10, 11, 12
45. Wang, L., Sun, C., Fu, Y., Kim, M.H., Huang, H.: Hyperspectral image reconstruction using a deep spatial-spectral prior. In: CVPR (2019) 4, 12
46. Wang, L., Sun, C., Zhang, M., Fu, Y., Huang, H.: Dnu: Deep non-local unrolling for computational spectral imaging. In: CVPR (2020) 2
47. Wang, L., Xiong, Z., Shi, G., Wu, F., Zeng, W.: Adaptive nonlocal sparse representation for dual-camera compressive hyperspectral imaging. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(10), 2104–2111 (2016) 1
48. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004) 10
49. Wilson, A.G., Izmailov, P.: Bayesian deep learning and a probabilistic perspective of generalization. arXiv preprint arXiv:2002.08791 (2020) 6
50. Yuan, X., Liu, Y., Suo, J., Durand, F., Dai, Q.: Plug-and-play algorithms for video snapshot compressive imaging. IEEE Transactions on Pattern Analysis and Machine Intelligence (01), 1–1 (2021) 1
51. Yuan, X.: Generalized alternating projection based total variation minimization for compressive sensing. In: ICIP (2016) 4
52. Yuan, X., Brady, D.J., Katsaggelos, A.K.: Snapshot compressive imaging: Theory, algorithms, and applications. IEEE Signal Processing Magazine **38**(2), 65–88 (2021) 1, 5
53. Yuan, X., Liu, Y., Suo, J., Dai, Q.: Plug-and-play algorithms for large-scale snapshot compressive imaging. In: CVPR (2020) 4
54. Yuan, Y., Zheng, X., Lu, X.: Hyperspectral image superresolution by transfer learning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **10**(5), 1963–1974 (2017) 1
55. Zhang, S., Wang, L., Fu, Y., Zhong, X., Huang, H.: Computational hyperspectral imaging based on dimension-discriminative low-rank tensor recovery. In: ICCV (2019) 1
56. Zhang, T., Fu, Y., Wang, L., Huang, H.: Hyperspectral image reconstruction using deep external and internal learning. In: ICCV (2019) 2
57. Zhang, Y., Li, K., Li, K., Fu, Y.: Mr image super-resolution with squeeze and excitation reasoning attention network. In: CVPR (2021) 8
58. Zhu, R., Tao, Z., Li, Y., Li, S.: Automated graph learning via population based self-tuning GCN. In: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2096–2100. ACM (2021) 4
59. Zou, Y., Fu, Y., Zheng, Y., Li, W.: Csr-net: Camera spectral response network for dimensionality reduction and classification in hyperspectral imagery. Remote Sensing **12**(20), 3294–3314 (2020) 1