# Diffusion Model with Disentangled Modulations for Sharpening Multispectral and Hyperspectral Images

Zihan Cao[a], Shiqi Cao[a], Liang-Jian Deng[a, *], Xiao Wu[a], Junming Hou[b], Gemine Vivone[c, d]

[a] *University of Electronic Science and Technology of China, Chengdu, 611731, China*
[b] *Southeast University, Nanjing, 210000, China*
[c] *Institute of Methodologies for Environmental Analysis (CNR-IMAA), Tito, 85050, Italy*
[d] *National Biodiversity Future Center (NBFC), Palermo, 90133, Italy*

**Abstract**

The denoising diffusion model has received increasing attention in the field of image generation in recent years, thanks to its powerful generation capability. However, diffusion models should be deeply investigated in the field of multi-source image fusion, such as remote sensing pansharpening and multispectral and hyperspectral image fusion (MHIF). In this paper, we introduce a novel supervised diffusion model with two conditional modulation modules, specifically designed for the task of multi-source image fusion. These modules mainly consist of a coarse-grained style modulation (CSM) and a fine-grained wavelet modulation (FWM), which aim to disentangle coarse-grained style information and fine-grained frequency information, respectively, thereby generating competitive fused images. Moreover, some essential strategies for the training of the given diffusion model are well discussed, e.g., the selection of training objectives. The superiority of the proposed method is verified compared with recent state-of-the-art (SOTA) techniques by extensive experiments on two multi-source image fusion benchmarks, i.e., pansharpening and MHIF. In addition, sufficient discussions and ablation studies in the experiments are involved to demonstrate the effectiveness of our approach. Code will be available after possible acceptance.

*Keywords:* Denoising diffusion model, wavelet transformation, pansharpening, multi-source image fusion, multispectral and hyperspectral image fusion, end-to-end network, remote sensing

## 1. Introduction

Recently, multi-source image fusion (MSIF) has attracted much attention in the area of image processing and computer vision. The MSIF leverages on different images with domain-specific information to generate a fused image that cannot be obtained under practical conditions, such as a high-resolution multi-spectral image (HRMS), or a high-resolution hyperspectral image (HRHS). In this work, we mainly focus on two practical MSIF tasks, i.e., *pansharpening* [1] and *multispectral and hyperspectral image fusion (MHIF)* [2], to propose our method.

Pansharpening, as a relevant problem in remote sensing image processing, is attracting more and more interest from the research community and commercial companies. Specifically, pansharpening requires the fusion of a high spatial resolution panchromatic (PAN) image and a low spatial resolution multispectral (LRMS) image to obtain a high spatial resolution multispectral image (HRMS), which preserves the advantages of the two images belonging to two different domains. Most satellites can simultaneously capture PAN and MS images, such as WorldView-3 and GaoFen-2. Pansharpening methods can be divided into four categories [3], i.e., component substitution (CS) methods [4, 5, 6, 7], multi-resolution analysis (MRA) methods [8, 9, 10, 11], variational optimized

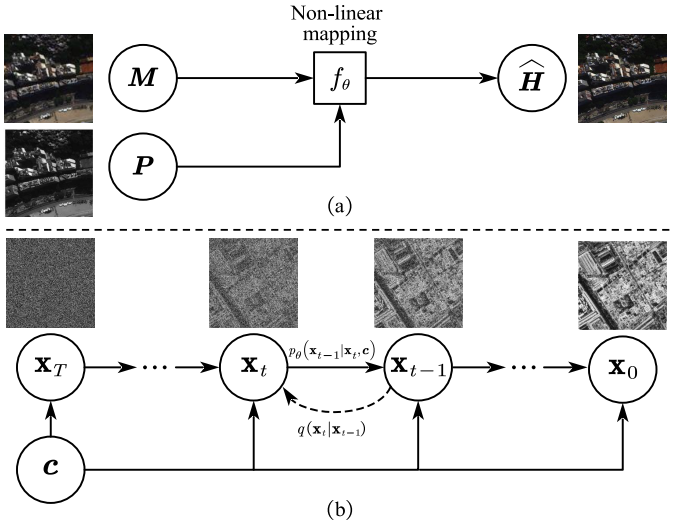



Figure 1: (a) Previous DL-based methods directly learn a non-linear mapping to fuse under the one-step preconditioned framework. The $\boldsymbol{M}$, $\boldsymbol{P}$, and $\widehat{\boldsymbol{H}}$ indicate the LRMS, the PAN, and the fused HRMS in pansharpening. (b) The denoising diffusion model is enabled with a multistep denoising process while breaking the preconditioned learning process. The $q(\mathbf{x}_t|\mathbf{x}_{t-1})$, $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{c})$, and $\boldsymbol{c}$ represent the noise adding forward process, the denoising backward process, and the condition, respectively.

* Corresponding Author.

(VO) techniques [12, 13, 14, 15], and deep learning (DL) approaches [16, 17, 18, 19, 20, 21]. The first three conventional approaches have their own advantages and disadvantages, but they frequently result in spectral or spatial distortions because of the limitations of the various techniques. CS methods often consider a linear projection into a transformed domain of the MS image by substituting the separated MS spatial information with the PAN image. MRA methods relied upon a multiscale decomposition (often exploiting linear filtering) to the PAN image to extract its spatial components and inject them into the MS image. Thus, both are based on linear operations and, hence, they cannot accurately extract spectral/spatial information suffering from spectral and/or spatial distortions. VO methods introduce regularization terms to take into account of prior information. However, regularization terms that are not well-designed can cause degraded pansharpening performance. About the DL-based approaches, which are mostly related to this work, we will discuss their advantages and disadvantages in conjunction with the DL-based methods for MHIF.

MHIF techniques aim to fuse multispectral and hyperspectral images captured from the same scene to get information from phenomena that cannot be detected by the sole HRMS. More specifically, MHIF can also be seen as a preprocessing method for further high-level applications, such as change detection [22, 23], mineral exploration [24, 25], and segmentation [26]. The recent literature of MHIF [2] mainly involves model-based methods [27, 28, 29, 30] and DL-based methods [31, 32, 33]. Although the model-based methods utilize various image priors, it is still difficult to obtain high-fidelity, less distorted high-resolution hyperspectral fusion images due to the lack of training with large-scale datasets.

Over recent years, supervised DL-based methods have achieved significant advances in various vision tasks, such as convolution neural networks (CNNs) [34, 35] and Vision Transformers (ViTs) [36, 37]. In the task of MSIF, supervised DL-based methods can fuse an image by learning a non-linear functional mapping from spatial and spectral degradation related to pansharpening or MHIF task. Previous supervised DL-based methods can be regarded as the *preconditioned fusion framework*, which injects the image features and priors into the learnable mapping process. To promote the non-linear capability, these DL-based works mainly focus on better architecture designing (e.g., [17, 38, 39, 19, 40] for pansharpening and [41, 42, 32, 43] for MHIF), more effective modules [44, 45, 46], and more novel model-driven networks [33, 47, 48, 49]. However, the learnable mapping process tries to fuse the image features and priors in just one step (i.e., model evaluation) based on the preconditioned fusion framework (see Fig. 1-a). Moreover, supervised DL-based methods cannot leverage the learning ability of the network based on the one-step preconditioned framework, exacerbating the domain gap of the multi-source image. Additionally, the conditions for the network are entangled and may not be suitable for the fusion task. Recently, many unsupervised methods have been proposed by exploiting adversarial learning [50, 51], novel spatial/spectral loss [52], cycle consistency [53], and learning degraded processes [54]. In this work, we mainly focus on supervised methods, and some discussions about unsupervised methods can be found in Sect. 5.5.

Diffusion denoising probability model (DPM), as proposed for unconditional image generation [23, 55], conditional image generation [23, 56], text-to-image generation [57, 58], image-to-image translation [59], and discrete nature language generation [60] tasks, has shown its power providing extra feature details and good generation ability. Another advantage is that DPM owns a more stable training process and no model collapse compared to the GAN-based model. More specifically, the training and testing phases of DPM are dubbed as the forward process and backward process [61, 62]. In the forward process, the input image is corrupted by a pure Gaussian noise, then the model tries to denoise and recover the original input during the training. In the backward process, the input is a pure Gaussian noise and the trained model is responsible for removing, step-by-step, a little noise. Finally, after large enough time steps, the noise is recovered to the generated images. Furthermore, to control the generated images, Ho *et al.* [63] proposed the conditional DPM where the condition is fed into the diffusion model to control the trajectory of the stochastic differential equation (SDE) [23, 61, 62] ordinary differential equation (ODE) [64, 65]. DPM, which has a multistep forward/backward process (see Fig. 1-b), produces intermediate time-dependent variables and breaks the one-step preconditioned framework. However, a few works noted that DPM can be applied to the task of MSIF. The methods that are most closely related to our work are DDPM-CD [66], Dif-fuse [67], and PanDiff [68]. The common design of the first two works is that they both utilize a trained diffusion model and an additional task-specific pathway, i.e., detection head and fusion head that involve a tedious two-stage training. PanDiff directly inputs the PAN and LRMS images as conditions for diffusion model and designs a task-agnostic architecture to complete the pansharpening task, but neglecting the entangled conditions.

To address the aforementioned issues, we propose a novel diffusion model with two conditional modulation modules, named DDIF, specifically designed for the task of MSIF. The two modules mainly include a coarse-grained style modulation (CSM) and a fine-grained wavelet modulation (FWM), aiming to disentangle coarse-grained style information and fine-grained frequency information, thus overcoming the limitation that the entangled conditions are not suitable for the fusion task. The issue of the degraded learning ability is addressed by the multistep denoising process inherited by the diffusion model. Experiments show that our DDIF can fuse images with better visual quality and performance metrics. Moreover, DPM with SDE sampling can introduce new noise that contains many high-frequency components when sampling for better fusion. Compared with existing diffusion models, our DDIF is specifically designed for the multi-source fusion task, as well as it is able to yield better fusion outcomes.

The contribution of this work is three-fold:

1. We propose a novel supervised diffusion model with disentangled modulations for sharpening multispectral and hyperspectral images (dubbed as DDIF), which can address the degraded learning ability brought by the one-step pre-

conditioned framework in previous DL-based models. A preliminary manuscript can be found on the preprint website[1].

2. The given DDIF contains two novel modulation modules, i.e., coarse-grained style modulation (CSM) and fine-grained wavelet modulation (FWM). These two modules disentangle style information and frequency components from different domain conditions, thus suiting the fusion task and bringing improved fusion results.
3. Our DDIF gets state-of-the-art (SOTA) performance on the pansharpening task considering two widely used pansharpening datasets, as well as yields competitive performance on one benchmark MHIF dataset. Discussions and ablation studies assess the effectiveness of the proposed method.

The remaining of the paper is organized as follows. Sect. 2 describes the related works. The proposed methodology is presented in Sect. 3. A broad experimental analysis is shown in Sect. 4. Discussions are provided to the readers in Sect. 5. Finally, conclusions are drawn in Sect. 7.

## 2. Related Works

This section mainly introduces supervised DL-based and diffusion methods that are mostly related to our work, as well as mainstream generative tasks and methods related to diffusion models.

### 2.1. Supervised Preconditioned Models for Image Fusion

Previous DL-based methods are mainly based on a preconditioned fusion framework. Ma *et al* [69] first proposed transformer-based framework for image fusion, in which explains the importance of transformer's long distance dependence on image fusion tasks. Tang *et al* [70] integrated image matching, fusion and semantic awareness into a unified framework, and gets promising outcomes. While unsupervised methods have led to significant advancements in recent years, it is important to note that our work is related to supervised learning. Thus, in this paper, we mainly focus on supervised methods. For pansharpening, PNN [17], inspired by a related single image super-resolution technique, proposed first a convolutional neural network for pansharpening. To enable the network to handle the high-frequency information, PanNet [18] was proposed to explicitly inject high-passed information from the PAN image into the upsampled LRMS image, resulting in a better performance. Furthermore, DiCNN [38] has been presented as a details-inject-based CNN, which is powerful in preserving frequency details. Another approach, the so-called FusionNet [19], implemented an end-to-end residual network, allowing it to explicitly learn high-frequency details.

To deal with the static kernel of conventional convolutional operators, LAGNet [45] introduced a dynamic kernel based on the input. Building on this idea, AKD [44] further explored the concept of LAGNet and proposed two dynamically generated branches responsible for spectral and spatial details extraction, respectively. SpanConv [71] presented an interpretable span strategy, which effectively constructed a kernel space and reduced the redundancy of the convolution while maintaining good performance. Besides, PMACNet [72], a parallel convolutional neural network structure, has been employed with a pixel-wise attention constraint module and a novel multireceptive-field attention block. This architecture successfully addressed the small receptive field issue caused by CNNs.

Moving on to MHIF, CNN-FUSE [31] learned the subspace from the low-resolution hyperspectral (LRHS) image via singular value decomposition and then approximated the high-resolution hyperspectral (HRHS) image with subspace coefficients. Furthermore, SSRNet [42] proposed three models for MHIF, including a cross-mode message inserting model for producing preliminary fused HRSR, a spatial reconstruction model, and a spectral reconstruction model. Recently, DHIF [33] analyzed the importance of physical imaging models for MHIF and introduced a spatio-spectral regularized deep hyperspectral fusion model.

The aforementioned works conducted deep research on features of DL-based methods helping models in generating better fusion results. However, these DL-based models require fixed priors and then fuse the image features, which means they are under the preconditioned framework. In contrast, denoising diffusion models produce a predicted distribution as close as possible to the posterior distribution in each step of optimization, breaking the preconditioned learning process.

### 2.2. Diffusion Models

Diffusion models have recently been proposed for the generation task, including conditional or unconditional generation [23, 55, 61], text-to-image translation [58, 73], image super-resolution [74], image restoration [75, 76], and other high-level image manipulation tasks [77, 78, 59]. Wherein, Song *et al.* [79] introduced first a score-based model that produces samples via Langevin dynamics using gradients of the data distribution estimated with score matching. Ho *et al.* [23] proposed DDPM from the direction of weighted variational bound, and their equivalence is proven in [62]. To accelerate the sampling of DPM, DDIM [64] designed a non-Markov chain sampling process. In addition, DPM-solver [65] simplified the solution to an exponentially weighted integral of the neural network by computing the linear part of the ODE and applying change-of-variable, further accelerating the sampling process. To free the design of DPM from cumbersome mathematical requirements, EDM [56] decoupled various design components of the diffusion model and designed a second-order ODE sampler, which further improved the performance of the diffusion model to reach state-of-the-art performance.

Besides, generative approaches mainly consist of GAN-based and flow-based models (beyond diffusion models). GAN-based models [80, 81, 82, 83] generate a sample following a data distribution using a discriminate model, $D$, having the role

[1] https://arxiv.org/abs/2304.04774

of estimating the probability that a sample comes from the training data rather than the generative model, $G$. Instead, flow-based models [84, 85] learn the underlying distribution of data by transforming a simple input distribution (e.g., Gaussian) into the target distribution through a series of invertible transformations. Compared to the two above-mentioned generative models, diffusion models can generate images with more details and higher fidelity. The advantages of diffusion models include stable training, minimal mode collapse, and the ability to train with only a single mean squared error (MSE) loss. In comparison to GAN-based models, which suffer from instability issues in their adversarial training, and flow-based models, which are limited in their network performance due to the reversibility requirement, diffusion models are easier to train and design.

Despite being so powerful in the field of image generation, diffusion models have not received much attention in the field of MSIF. The most related works are as follows. DDPM-CD [66] utilized diffusion models for landform change detection. Firstly, an unconditional diffusion model is trained on a large dataset, and the features of a specific layer of the diffusion model during the sampling process are used as additional information input to the segmentation head, producing segmentation results. As a result, DDPM-CD has achieved satisfactory performance in the field of landform change detection. Similar to DDPM-CD, Dif-fuse [67] fused red-green-blue (RGB) and near-infrared images getting clear advances. These works focus on using deep semantic feature maps of DPM feeding them into another segmentation or fusion head, which is not straightforward, also (unnecessary) requiring a two-stage training. These previous works raise a question that is how and if we can fuse images from two different domains by exploiting just a unique diffusion model, training it in an end-to-end manner.

### *2.3. Motivations*

Diffusion models in the MSIF task are mostly based on a straightforward approach. They exploit an ideal image corrupted by noise as input, i.e., the noised version of the ground-truth (GT) image. Afterwards, the combination of multi-source images is fed into the diffusion model as the condition. Obviously, the noised GT can be concatenated with them to fuse the two domains, which has been proven to be effective in [86]. However, we believe that this condition, just using concatenation, is entangled and will lead to insufficient convergence and poor results (see Sect. 4 for details). To achieve better results, we consider conditional style and frequency modulation to disentangle conditions in the denoising diffusion process, thus well adapting to the fusion task.

About the style modulation, the related style information is encoded as a condition to help the model control the general features of objects, such as, shapes and colors, which has been inspired by the extracting style code proposed in [82]. In the context of sharpening multispectral and hyperspectral images, we describe style information as the coarse-grained spectral changes that are intertwined with low-frequency spatial information as shown in Fig. 4-a. The frequency modulation can be introduced by the added Gaussian noise in the backward process of the diffusion model and the domain-related image (e.g., the PAN image). Based on this, a condition can be disentangled into the style codes and frequency information. It is feasible to employ the diffusion model with two conditioning modules that handle coarse-grained style information and fine-grained frequency information, respectively. Benefiting from this disentanglement, we can make the diffusion U-Net encoder and decoder responsible for style and frequency information separately, thus easing the learning process. Finally, the combination of the multi-source images is injected into the final (generated) fused image.

Table 1: Some notations used in this work.

| Notation | Explanation |
|---|---|
| $t$ | The timestep |
| $\mathbf{x}_t$ | An image at diffusion timestep $t$ |
| $\epsilon$ | Gaussian noise |
| $\mathbf{v}$ | Training objective defined in Eq. (11) |
| $\boldsymbol{M}$ | The upsampled LRMS |
| $\boldsymbol{P}$ | The PAN image |
| $\boldsymbol{H}$ | The original HRMS |
| $\widehat{\boldsymbol{H}}$ | The fused HRMS |
| $\boldsymbol{F}^l$ | The $l$-th layer's feature map |
| $\boldsymbol{c}$ | The diffusion condition |
| $\boldsymbol{LL}, \boldsymbol{LH}, \boldsymbol{HL}, \boldsymbol{HH}$ | The wavelet coefficients |
| $\boldsymbol{Z}$ | The scale used for style modulation in CSM |
| $\boldsymbol{S}$ | The shift used for style modulation in CSM |
| $(\cdot)_{\downarrow l}$ | Bilinear downsample with factor $l$ |

## 3. Methodology

This section introduces first some notations that will be used in the description of the proposed method. Then, we will briefly review the mechanism of the diffusion denoising model. Subsequently, a detailed introduction to the architecture of the diffusion model and the two conditional disentangled modulation modules will be shown. Finally, the effects of the proposed two modules and some discussions about training and sampling techniques will be provided.

### *3.1. Notations*

Tab. 1 reports all the related notations used in the paper, focusing on the pansharpening task. From the table, we denote the PAN image, the upsampled LRMS image, the original HRMS image, and the fused HRMS image as $\boldsymbol{P} \in \mathbb{R}^{H\times W\times c}$, $\boldsymbol{M} \in \mathbb{R}^{H\times W\times C}$, $\boldsymbol{H} \in \mathbb{R}^{H\times W\times C}$, and $\widehat{\boldsymbol{H}} \in \mathbb{R}^{H\times W\times C}$, respectively, where, $H$, $W$, and $C$ (or $c$) indicate the image spatial size and the channel (or spectral) number, respectively.

Considering the MHIF task, $\boldsymbol{P}$ correspondingly represents the HRMS image, $\boldsymbol{M}$ is the upsampled LRHS image, $\boldsymbol{H}$ is the original HRHS image, and $\widehat{\boldsymbol{H}}$ is the fused HRHS image.

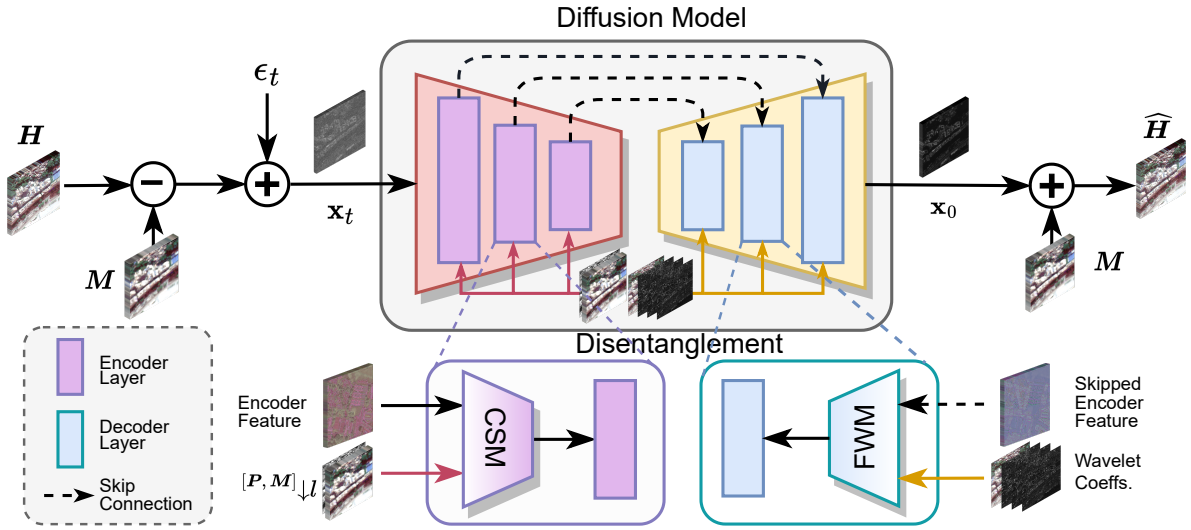



Figure 2: The flowchart of our DDIF. The input is the residual related to the difference between HRMS and LRMS images. A Gaussian noise $\epsilon_t = \sqrt{1-\bar{\alpha}_t}\epsilon$ is sampled to be added to the residual. The diffusion U-Net exploiting our effective disentangled modulations (i.e., CSM and FWM) produces the undegraded $\mathbf{x}_0$. Finally, $\mathbf{x}_0$ is added to LRMS to get the HRMS image. "wavelet coeffs." are $\boldsymbol{LL_M}, \boldsymbol{LH_P}, \boldsymbol{HL_P}, \boldsymbol{HH_P}$ as presented in Eq. (10), and $(\cdot)_l$ is the bilinear downsample operator by a factor of $l$.

### 3.2. A Brief Review of Diffusion Model

Diffusion models can generate a realistic image from a Gaussian distribution by reversing a noise process. The diffusion model accomplishes its task in two steps, i.e., forward and backward processes, which are illustrated in Fig. 1.

The forward process aims to make the origin image, $\mathbf{x_0} \sim p_{data}(\mathbf{x_0})$, noisy due to a $T$ step Markov chain that gradually converts it into a Gaussian distribution. The forward step is defined as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t;\ \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where $t \in [0, T]$, $\beta_t \in (0, 1)$ is a pre-defined variance, that is function of the steps, $\mathcal{N}$ is a Gaussian distribution with mean, $\sqrt{1-\beta_t}\mathbf{x}_{t-1}$, and standard deviation, $\beta_t\mathbf{I}$, and $\mathbf{I}$ is the identity matrix. Through the reparameterization trick, it can get $\mathbf{x}_t$ as follows:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{2}$$

where $\epsilon \sim \mathcal{N}(0, 1)$ is the standard Gaussian noise, and $\bar{\alpha}_t = \prod_{i=0}^{t}\alpha_i$ with $\alpha_t = 1 - \beta_t$.

Then, the backward process is related to the denoising of $\mathbf{x}_t$ by the following procedure:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)), \tag{3}$$

where $\theta$ denotes the learnable model parameters, and $\mu_\theta(\mathbf{x}_t, t)$ and $\Sigma_\theta(\mathbf{x}_t, t)$ are the mean and the standard deviation, respectively, of the Gaussian distribution.

In summary, the forward process degrades the data distribution into a standard Gaussian distribution. Instead, the backward process, modeled by a neural network, aims to learn how to remove the degradation generated in the forward process, i.e., the denoising task.

The training of a diffusion model, by maximizing its variational lower bound (VLB), is done exploiting a simple supervised loss [23], written as follows:

$$\min_\theta\ L_{simple} = \mathbb{E}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2]. \tag{4}$$

Please, note that in Eq. (4), the model prediction of the added Gaussian noise, $\epsilon_\theta(\mathbf{x}_t, t)$, is used as the training objective, but it can be substituted by the original input, $\mathbf{x}_0$, or the "velocity", $\mathbf{v}$ (see [57] and the ablation study in Sect. 4 for more information).

After training the model, we can sample data starting from a standard Gaussian noise, $\mathbf{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$, and, according to Eq. (3), the mean and variance can be computed as follows:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right), \tag{5}$$

$$\Sigma_\theta(\mathbf{x}_t, t) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{6}$$

To this aim, the image $\mathbf{x}_t$ can be sampled from the previous step. Through $T$ iterative sampling steps, the initial image, $\mathbf{x}_0$, can be obtained.

In the MSIF task, there are usually two (or even more) input images from different domains. Focusing on pansharpening, the PAN and LRMS images are from different domains (specifically from different sensors) and the HRMS image is from the target domain. To guide the diffusion model, we empirically set the PAN, the LRMS, and the constructed wavelet coefficients (presented in Sect. 3.5) to conditions denoted as $\boldsymbol{c}$. Afterwards, Eqs. (3)-(6) have to be conditioned to $\boldsymbol{c}$.

### 3.3. Overall Model Architecture

As shown in Fig. 2, the overall network architecture of our DDIF consists of an encoder, decoder, and two middle self-

attention modules, which resemble a U-Net architecture of convolutional layers. There are two downsampling operations between the two layers of the encoder to reduce the spatial resolution and increase the number of channels. Similarly, there is an upsampling operation by a factor of 2 between the two layers of the decoder to increase the spatial resolution and reduce the number of channels.

The encoder and decoder have the same number of layers. The encoder takes in input, not only the encoded features from the previous layer, but also the modulated coarse-grained style conditions (see Sect. 3.4). The features of the encoder at the corresponding layer will be concatenated with the input features of the decoder at the corresponding layer and the additional wavelet features (see Sect. 3.5) along the channel dimension, feeding them into the decoder. The related output can be modeled as the added Gaussian noise, $\epsilon$, the original image, $\mathbf{x}_0$, or the "velocity", $\mathbf{v}$, which will be discussed in Sect. 3.7.

### 3.4. Coarse-grained Style Modulation (CSM)

As mentioned above, we consider disentangling the style and the detailed information in the MS and PAN images proposing a CSM as shown in Fig. 3-a. More specifically, the MS and PAN images are concatenated along the channel dimension and are considered conditional guidance. We add this conditional guidance into CSM, then feeding the output feature into each layer of the encoder to modulate the encoder feature, $\boldsymbol{F}^l$. It is worth noting that we avoid encoding a too much detailed spatial information, but reconstructing it in the decoder. To this end, the encoder only needs to consider the overall style information without the detailed spatial information. It is also meaningful for the encoder to progressively decrease the spatial size of the feature map while increasing the number of channels. By incorporating style information into the encoder, the model can focus on style-related aspects without being influenced by irrelevant spatial information.

For CSM, it is feasible to generate the corresponding style feature by outputting the scale, $\boldsymbol{Z}$, and the shift, $\boldsymbol{S}$, based on the concatenation of the PAN and the MS images. Thus, we have:

$$\begin{aligned} \boldsymbol{Z}, \boldsymbol{S} &= \text{Split}(\text{MLP}([\boldsymbol{P}, \boldsymbol{M}])), \\ \boldsymbol{F}^l &= \boldsymbol{F}^l \cdot (\boldsymbol{I} + \boldsymbol{Z}) + \boldsymbol{S}, \end{aligned} \tag{7}$$

where "Split" splits the feature into two parts with equal size along the channel dimension. The MLP is implemented with several convolutional layers, where a SiLU [87] activation and a GroupNorm [88] are added in-between. More specifically, the scale, $\boldsymbol{Z} \in \mathbb{R}^{h \times w \times d}$, and the shift, $\boldsymbol{S} \in \mathbb{R}^{h \times w \times d}$, are produced by an MLP. Besides, $\boldsymbol{F^l} \in \mathbb{R}^{h \times w \times d}$ is the CSM output, linearly modulated by $\boldsymbol{Z}$ and $\boldsymbol{S}$, where $h, w, d$ are determined by the size of the feature outputted from the previous encoder layer.

### 3.5. Fine-grained Wavelet Modulation (FWM)

The encoder is responsible for encoding style information into a high dimensional, low spatial resolution feature by treating the LRMS and PAN images as conditions. The decoder (whose objective is to decode high-dimensional, low spatial resolution features from the encoder as faithfully as possible) uses

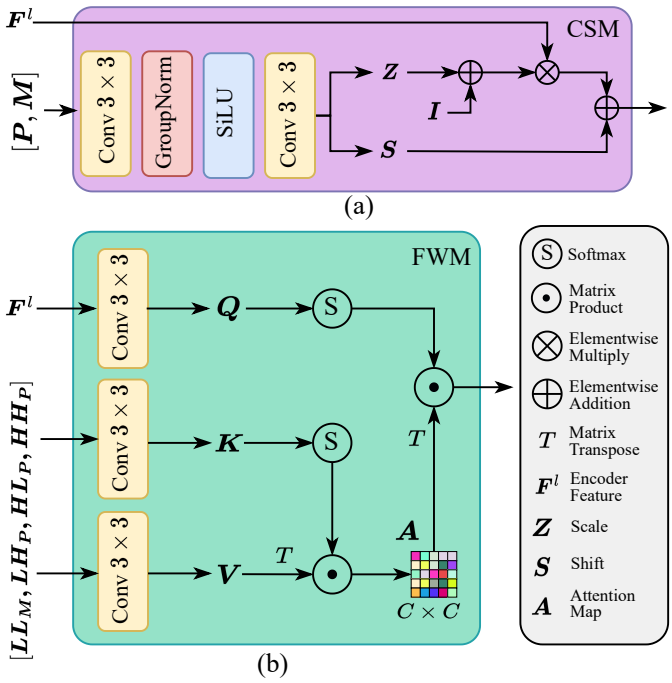



Figure 3: Two conditioning modulations that have been proposed: (a) coarse-grained style modulation (CSM), (b) fine-grained wavelet modulation (FWM). $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are query, key, and value defined in Eq. 10. For the other notations, please refer to Tab. 1.

the features encoded from the encoder to support style information and frequency components disentanglement. However, relying solely on the features outputted by the encoder is not enough for the decoder. We found that introducing appropriate high-frequency spatial components during the decoding stages is instead advantageous to produce images with richer details. An intuitive approach is to concatenate the high-frequency detailed components extracted from the PAN image with the features outputted by the encoder along the channel dimension, then feeding the concatenated features into the decoder for decoding. Considering that previous traditional methods often exploit wavelet information [89, 90, 91] to complement additional frequency details, we can decompose images into wavelets and incorporate them into the diffusion model.

Based on this cue, we apply wavelet decomposition on the PAN image to extract the horizontal, vertical, and diagonal high-frequency components and to use a cross-attention mechanism in the decoder to introduce these high-frequency details into the decoding process. However, considering only high-frequency details could be inadequate, thus, we also introduced the low-frequency main component of the LRMS, which is also extracted using wavelet decomposition. The high-frequency and low-frequency components are concatenated along the channel dimension, then passing them through the fine-grained attention module on the U-Net skip connection. We refer to this process as FWM, see Fig. 3-b. The advantage of this modulation is that it separates the low-frequency and high-frequency components, with PAN providing the high-frequency

details and LRMS providing the low-frequency spatial component, thus making the learning process easier for the decoder.

To decompose both the LRMS and PAN images into four components, we used the DB1 wavelet decomposition [92], which includes one main (low-frequency) component and three details components in the horizontal, vertical, and diagonal directions. Thus, we have:

$$\begin{aligned} \boldsymbol{LL_M}, \boldsymbol{LH_M}, \boldsymbol{HL_M}, \boldsymbol{HH_M} &= \text{DB1}(\boldsymbol{M}), \\ \boldsymbol{LL_P}, \boldsymbol{LH_P}, \boldsymbol{HL_P}, \boldsymbol{HH_P} &= \text{DB1}(\boldsymbol{P}), \end{aligned} \tag{8}$$

where $\boldsymbol{LL} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times d}$ denotes the main (low-frequency) component, $\boldsymbol{LH} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times d}$, $\boldsymbol{HL} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times d}$, and $\boldsymbol{HH} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times d}$ indicate the details components in the horizontal, vertical, and diagonal directions, respectively.

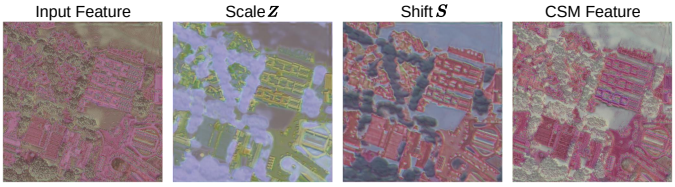
Afterward, the attention operation can be used in FWM for modulation. However, the memory complexity of the attention is $O(n^2)$, where $n = HW$. When approaching the head of the decoder, the output image size increases and it can become unacceptable when dealing with large images due to the quadratic complexity with respect to the image size. Hence, we adopt a linear-memory cross-attention mechanism to incorporate the frequency components into the decoder. The linear-memory attention mechanism is as follows:

$$\begin{aligned} \boldsymbol{Q} &= \text{Reshape}(\text{Softmax}(\boldsymbol{Q}, 1)), \\ \boldsymbol{K} &= \text{Reshape}(\text{Softmax}(\boldsymbol{K}, 2)), \\ \boldsymbol{V} &= \text{Reshape}(\boldsymbol{V}), \\ \boldsymbol{A} &= \boldsymbol{K} \odot \boldsymbol{V}^\top, \\ \boldsymbol{O} &= \boldsymbol{A}^\top \odot \boldsymbol{Q}, \end{aligned} \tag{9}$$

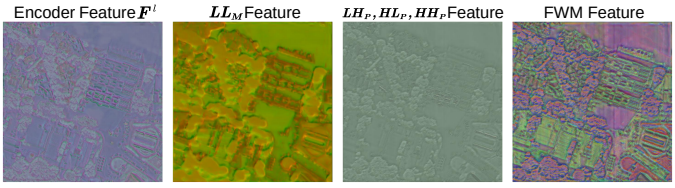
where the "Reshape" operation is employed to flatten the spatial dimensions of $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$, into a single dimension, i.e., $\mathbb{R}^{H\times W\times d} \rightarrow \mathbb{R}^{HW\times d}$, and Softmax$(\cdot, i)$ represents the Softmax operation along the $i$-th dimension. As a result, we reduce the memory complexity to $O\left(d^2\right)$, where $d$ is much smaller than $n$. This significantly reduces memory usage while ensuring the effective introduction of the conditions. In the above equations, $\boldsymbol{Q}$ is the query tensor, and $\boldsymbol{K}$ and $\boldsymbol{V}$ are the tensors obtained by concatenating the $\boldsymbol{LL_M}, \boldsymbol{LH_P}, \boldsymbol{HL_P}$ and the $\boldsymbol{HH_P}$ components along the channel dimension. These latter terms are projected by several convolutional layers as follows:

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{W}_Q \otimes \boldsymbol{F} + \boldsymbol{b}_Q, \\ [\boldsymbol{K}, \boldsymbol{V}] &= \boldsymbol{W}_{K,V} \otimes [\boldsymbol{LL_M}, \boldsymbol{LH_P}, \boldsymbol{HL_P}, \boldsymbol{HH_P}] + \boldsymbol{b}_{K,V}, \end{aligned} \tag{10}$$

where $\otimes$ is the convolution operation, $\boldsymbol{W}_*$ is the weight, $\boldsymbol{b}_*$ is the bias, and $\boldsymbol{F}$ is the feature from the corresponding encoder layer. This linear-memory cross-attention in the proposed FWM can be used to learn a global response related to the frequency information. However, it exhibits some limitations in adequately capturing spatial information. Recognizing that the vanilla self-attention mechanism has enough representational capability to recover spatial details and offers the advantage of a quadratic computational overload, we chose to harness the linear-memory cross-attention mechanism along the spectral dimension. In doing so, we introduced frequency components extracted by wavelet coefficients to modulate features from the corresponding encoder layer, employing a cross-attention approach.




(a) Features by CSM modulation




(b) Features by FWM modulation

Figure 4: Feature maps of the two proposed modulations.

### 3.6. The Effects of CSM and FWM

CSM generates the scale, $\boldsymbol{Z}$, and the shift, $\boldsymbol{S}$, for modulation. The scale term controls the amplitude of the encoder feature, $\boldsymbol{F}^l$, and the shift directly injects the condition into the feature. As depicted in Fig. 4-a, the input feature has low contrast (buildings, forest, and lands have similar feature intensity). Moreover, the CSM feature has higher contrast than the input feature under the control of the scale, as well as the color and shape (e.g., buildings) share some similarities with the shift. Exploiting this strategy, the style information is decomposed into the scale and shifted to coarse-grained modulating the encoder feature.

FWM uses wavelet coefficients extracted from the MS and PAN images. The MS image contributes to the low-frequency coefficient, $\boldsymbol{LL_M}$, and the PAN image contributes to the high-frequency coefficients, $\boldsymbol{LH_P}, \boldsymbol{HL_P}, \boldsymbol{HH_P}$. As shown in Fig. 4-b, the input encoder feature gets meaningful contrastive regions due to CSM, but still lacking fine-grain frequency information. Thanks to the help of the wavelet coefficients, the modulated FWM feature has higher spatial fidelity.

In summary, the style and frequency information are well disentangled with the proposed two modulation modules and separately handled in the encoder and decoder. This can solve the preconditioned fusion problem of the previously developed DL-based methods.

### 3.7. Boost the Performance Further

**Residual Learning**. Because of CNNs tend to learn low-frequency information, they lack in representing high-frequency information. Previous works attempted to address this issue by filtering the input with high-pass filters and directly incorporating the related details as input [18] by designing specialized high-frequency injection modules or supervising the process in the frequency domain [93]. Here, we do not aim to design complex high-frequency modules that significantly increase the number of parameters in the network, nor do we want to perform complex operations such as computing loss in the frequency domain.

**Algorithm 1:** Training stage of our method.

**Data:** LRMS image, $\boldsymbol{M}$, PAN image, $\boldsymbol{P}$, GT image, $\mathbf{x}_0$, diffusion model, $\mathbf{x}_\theta$, with its parameters, $\theta$, timestep, $t$, and denoised objective, $\hat{\mathbf{x}}_0$.
**Result:** Optimized diffusion model $\mathbf{x}^*_\theta$.

$\mathbf{c} \leftarrow \boldsymbol{P}, \boldsymbol{M}, \mathrm{DB1}(\boldsymbol{P}, \boldsymbol{M})$; // Modulation
**while** until convergence **do**
  $t \leftarrow \mathrm{Uniform}(0, T)$;
  $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$;
  $\mathbf{x}_t \leftarrow \sqrt{\bar{\alpha}_t}(\mathbf{x}_0 - \boldsymbol{M}) + \sqrt{1-\bar{\alpha}_t}\epsilon$;
  $\hat{\mathbf{x}}_0 \leftarrow \mathbf{x}_\theta(\mathbf{x}_t, \mathbf{c}) + \boldsymbol{M}$; // Residual learning
  $\theta \leftarrow \nabla_\theta L_{simple}(\hat{\mathbf{x}}_0, \mathbf{x}_0)$. // Eq. (12)
**end**

Inspired by FusionNet [19], we change the input of the diffusion model during the training process from HRMS to the difference between HMRS and LRMS, i.e., HRMS-LRMS, and then adding Gaussian noise as in Eq. (2). We find that using noisy residuals as input, it converges faster and produces better results than when HRMS is used as input. Regarding to the use of residual learning, the related ablation experiments are shown in Sect. 4.8.2.

**Training Objective** There are three choices for the training objective of the diffusion model[2], i.e., $\epsilon$, $\mathbf{x}_0$, and $\mathbf{v}$ [57]. Note that $\mathbf{v}$ is a combination of $\epsilon$ and $\mathbf{x_0}$ defined as follows:

$$\mathbf{v} = \sqrt{\bar{\alpha}_t}\epsilon - \sqrt{1-\bar{\alpha}_t}\mathbf{x}_0. \tag{11}$$

In the previous diffusion works [23, 55], the training objective has often been selected as the added Gaussian noise, $\epsilon$, which is suitable for large-scale datasets. However, datasets for MSIF are much smaller than the ones related to natural images. We will show, in Sect. 3.7, that for small datasets, predicting $\mathbf{x}_0$ is a better choice than $\epsilon$ and $\mathbf{v}$.

Thus, considering $\mathbf{x}_0$ as goal, we have that the loss is:

$$L_{simple} = \mathbb{E}\left[\|\mathbf{x}_0 - \mathbf{x}_\theta(\mathbf{x}_t, t, \mathbf{c})\|_2^2\right]. \tag{12}$$

**Fast Iterative Sampling** Regarding to the backward process (sampling process), the diffusion model needs to iterate hundreds or thousands of times to generate an image, leading to a slow generation speed. To solve this problem, we convert the original DDPM SDE sampler [23] of the original backward sampling process into the DDIM ODE sampler [64], which allows for fast sampling in a non-Markov chain form as:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_\theta(\mathbf{x}_t, t, \boldsymbol{c}) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\epsilon_\theta(\mathbf{x}_t, t, \boldsymbol{c}) + \sigma_t^2\epsilon, \tag{13}$$

where $\sigma_t$ is an established function of $t$, $\mathbf{x}_\theta(\mathbf{x}_t, t, \mathbf{c})$ is the model prediction of $\mathbf{x}_0$, and

$$\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_\theta(\mathbf{x}_t, t, \mathbf{c})}{\sqrt{1-\bar{\alpha}_t}}. \tag{14}$$

[2] We do not discuss the score function [62] $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) = -\frac{\epsilon_\theta(\mathbf{x}_t)}{\sqrt{1-\bar{\alpha}_t}}$, since it has a similar form as $\epsilon$.

**Algorithm 2:** Inference stage of our method.

**Data:** LRMS image, $\boldsymbol{M}$, PAN image, $\boldsymbol{P}$, trained diffusion model, $\mathbf{x}_{\theta^*}$, with its parameters, $\theta^*$, sampled image, $\mathbf{x}_t$, at timestep $t$.
**Result:** Sampled image $\mathbf{x}_0$.

$t \leftarrow T$;
$\mathbf{c} \leftarrow \boldsymbol{P}, \boldsymbol{M}, \mathrm{DB1}(\boldsymbol{P}, \boldsymbol{M})$; // Modulation
**while** $t > 0$ **do**
  $\mathbf{x}_{t-1} \leftarrow \mathrm{DDIM\_Sample}(\mathbf{x}_t, \mathbf{c})$; // Eq. (13)
  $t \leftarrow t - 1$.
**end**
$\mathbf{x}_0 \leftarrow \mathbf{x}_0 + \boldsymbol{M}$.

From Eq. (13), the denoised $\mathbf{x}_{t-1}$ can be a combination of the predicted $\mathbf{x}_\theta$, the inferred noise, $\epsilon_\theta$, based on Eq. (14), and the new noise, $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$. Therefore, we can accelerate sampling by $\tau = [\tau_1, \tau_2, \cdots, \tau_{dim(\tau)}]$. More specifically, $\tau$ is a subset of $[1, 2, \cdots, T]$, and we can make a fast sampling after using $\tau$:

$$\mathbf{x}_\theta(\mathbf{x}_{\tau_i}, \tau_i, \boldsymbol{c}) := \frac{\mathbf{x}_{\tau_i} - \sqrt{1-\bar{\alpha}_{\tau_i}}\epsilon_\theta(\mathbf{x}_{\tau_i}, \tau_i, \boldsymbol{c})}{\sqrt{\bar{\alpha}_{\tau_i}}}. \tag{15}$$

According to the above sampling equations, the sampling speed can be improved. The overall training and sampling stages are presented in Algs. 1 and 2. The existing efficient ODE solvers can still be employed to accelerate our DDIF, such as the DPM solver [65], PNDM solver [94], and Heun solver [56]. However, we opted for the straightforward DDIM approach to accelerate the sampling process, ensuring both simplicity and excellent sharpening performance.

## 4. Experiments

In this section, we will provide a comprehensive overview of the implementation details, including the methodologies and the technical details. Furthermore, we will delve into the datasets, aiming to highlight their characteristics and relevance to this work. Additionally, we will present the benchmarks employed to evaluate the performance of our approach on two image fusion tasks, i.e., pansharpening, and multispectral and hyperspectral image fusion (MHIF). Finally, the main results and ablation studies will be provided to quantitatively illustrate the effectiveness of the proposed method.

### *4.1. Implementation Details*

The proposed DDIF method is implemented in PyTorch 1.13.1 and Python 3.10.9 using AdamW [95] optimizer with a learning rate of $1 \times 10^{-4}$ to minimize $L_{simple}$ on a Linux operating system with two NVIDIA GeForce RTX4090 GPUs. The initialization of convolution modules is based on a Kaiming initialization [96]. To obtain wavelet coefficients, we use the DB1 wavelets decomposition method decomposing the image into four wavelet coefficients. The chosen diffusion denoising model is a cosine schedule [55] for $\alpha_t$:

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos\left(\frac{t/T + \alpha}{1+\alpha} \cdot \frac{\pi}{2}\right), \tag{16}$$

where we set $\alpha$ to $8 \times 10^{-3}$. The model does not learn the variance term $\Sigma_\theta$ introduced in [55]. The total training diffusion timestep is set to 500 for both pansharpening and MHIF experiments. Additionally, the exponential moving average (EMA) ratio is set to 0.995. Moreover, the total training iterations for the WorldView-3 (WV3), GaoFen-2 (GF2), and CAVE datasets are set to 100k, 100k, and 300k iterations, respectively. According to Eqs. (14) and (15), we set the number of sampling steps to 25 for the pansharpening task, and 100 for the MHIF task, rather than 1000 or 2000 DDPM sampling steps as in [97] and [68]. Regarding our diffusion model configuration, we employed 4 layers each for both the encoder and decoder. The initial number of channels in the encoder layer is set to 32. Following each encoder/decoder layer, the number of channels is adjusted by multiplication or division using the factors 1, 2, 2, and 4, which correspond to the sequence of 4 encoder/decoder layers.

### 4.2. Datasets

To show the effectiveness of our DDIF, we conduct experiments over a standard pansharpening data-collection (i.e., PanCollection dataset[3]), which includes WV3 (8 bands) and GF2 (4 bands) data. To evaluate the performance, we perform the reduced-resolution and full-resolution experiments to compute the reference and non-reference metrics, respectively. Note that the same training and data augmentation strategy are applied to Pancollection for a fair comparison.

For the MHIF, we choose the CAVE indoor dataset[4] to further evaluate our DDIF. The CAVE dataset contains 31 hyperspectral images (HSIs) captured under controlled illumination with a spatial size of $512 \times 512$ and 31 spectral bands ranging from 400nm to 700nm at 10 nm steps. The multispectral images (consisting of RGB bands) can be generated by using HSIs and the spectral response functions of the Nikon D700 camera, which has a spatial size of $512 \times 512 \times 3$. 20 images are randomly selected for training and validation, and the remaining 11 images are used for testing. The test set is shown in Fig. 5. Afterward, we cropped 20 selected HSIs and MSIs into 3920 overlapping patches for training and validation with the size of $64 \times 64$. A Gaussian blurred kernel (size $3 \times 3$) with a standard deviation of 0.5 is used to get blurred HSIs that are decimated by a factor of 4 to produce the LRHS images. The cropped MSIs are used as HRMS and the original cropped HSIs are used as GT. Finally, the pairs are divided into training (80%) and validation (20%). To verify the performance of each method on hyperspectral real remote sensing data, we utilize the GF5-GF1 public dataset [98]. The GF5-GF1 dataset contains HSIs and MSIs, where the spatial size of MSIs is twice that of HSIs (i.e., $1161 \times 1120 \times 150$ for HSIs and $2332 \times 2258 \times 4$ for MSIs). We randomly cropped HSIs and MSIs into patches of size $40 \times 40$ and $80 \times 80$ with an overlap of 10 and 20, respectively, to generate real data. Based on the same patching scheme, furthermore, we can get the HRHSI (80×80) and HRMSI (160×160) patches for simulated data. We applied the provided modulated transfer functions (MTFs) to the patched HRHSI and the patched HRMSI following Wald's protocol. To get the final HRMSI, we adjusted the simulated HRMSI by using the modified $\boldsymbol{M} = (\boldsymbol{M} - \boldsymbol{B} \cdot \boldsymbol{R})/\boldsymbol{A}$ (as proposed in [98]), where $A$ and $B$ are the correlated weight tensors, and $R$ is the spectral response function. Finally, we obtained 150 simulated LRHSI and HRMSI patches with sizes $40 \times 40 \times 150$ and $80 \times 80 \times 4$, respectively, with the original LRHSI serving as ground-truth. We divided the 150 patches into train/validation/test sets using the following percentages 80%/10%/10%.

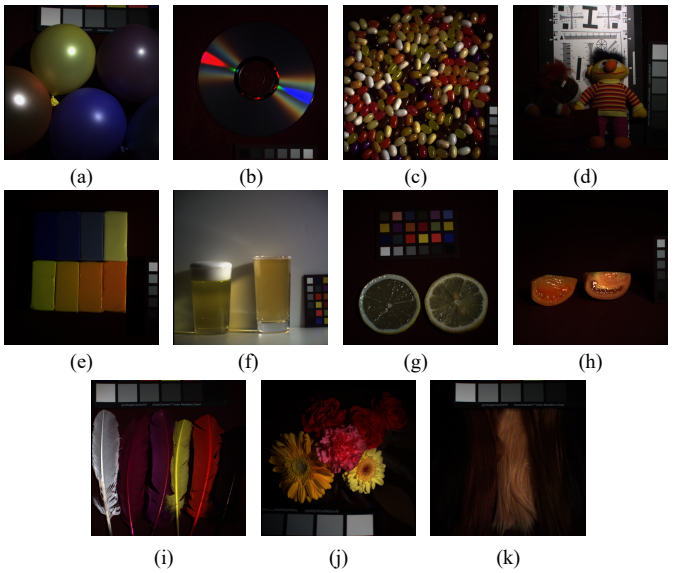



Figure 5: Testing images from the CAVE dataset: (a) *Balloons*, (b) *Compact disc (CD)*, (c) *Jelly beans*, (d) *Chart and stuffed toy*, (e) *Clay*, (f) *Fake and real beers*, (g) *Fake and real lemon slices*, (g) *Fake and real tomatoes*, (i) *Feathers*. (j) *Flowers*, (k) *Hairs*. The true color representation is used to depict the images.

### 4.3. Benchmark

To assess the performance of our DDIF, we compare it with previous state-of-the-art pansharpening methods (on the WV3 and GF2 datasets) and MHIF approaches (on the CAVE dataset).

For the pansharpening task, we choose some representative model-based methods as optimized Brovey transform with haze correction method (BT-H) [99], band-dependent spatial-detail with physical constraints approach (BDSD-PC) [100], generalized Laplacian pyramid (GLP) with modulation transfer function-matched filters and a full scale (FS) regression-based injection model (MTF-GLP-FS) [101] and a set of some competitive DL-based methods including PNN [17], DiCNN [38], MSDCNN [39], MMNet [102], FusionNet [19], CTINN [103], LAGConv [45], and DCFNet [40]. A recently proposed diffusion-based pansharpening method, PanDiff [68], is also compared. We compare them with our DDIF under the benchmark in [16]. Note that we do not compare our approach with

[3] https://liangjiandeng.github.io/PanCollection.html
[4] https://www.cs.columbia.edu/CAVE/databases/multispectral/

other GAN-based models as we found that almost all the GAN-based models are designed for unsupervised pansharpening. A comparison with them leads to unfairness, but we still discuss on pros and cons in Sect. 5.5.

For the MHIF task, we also chose model-based and DL-based methods for comparisons, where model-based methods include the coupled sparse tensor factorization (CSTF) method [30], the fast fusion of multi-band images based on solving a Sylvester equation approach (FUSE) [104], the modulation transfer function-matched generalized Laplacian pyramid hyper-sharpening (MTF-GLP-HS) [101], the iterative regularization method based on tensor subspace representation method (IR-TenSR) [29], the low tensor-train rank-based (LTTR) method [27] and the subspace-based low tensor multi-rank regularization method (LTMR) [105]. For real MHIF data, we chose some other competitive methods including the coupled nonnegative matrix factorization unmixing method (CNMF) [106], the subspace-based regularization method (Hysure) [107], and a component substitution method, i.e., the Gram-Schmidt adaptive (GSA) [108]. Regarding to DL-based methods, we perform a comparison with CNN-FUSE [31], SSRNet [42], ResTFNet [41], FusFormer [32], HSRNet [109], and the recent state-of-the-art DHIF [33] technique. To ensure a fair comparison, all the DL-based methods are retrained using identical input pairs and trained until convergence.

### 4.4. Quality Metrics and Runtime

For reduced-resolution datasets in pansharpening, we use the spectral angle mapper (SAM) [110], the relative dimensionless global error in synthesis (ERGAS) [111], the universal image quality index ($Q2^n$) [112], and the spatial correlation coefficient (SCC) [113] as metrics. The ideal values for $Q2^n$ and SCC are 1, and for SAM and ERGAS are 0. Since these metrics require the GT, they are considered reference metrics and are used in the reduced-resolution test case. For the full-resolution datasets, since there is no GT available, we use non-reference metrics to validate the accuracy of our DDIF. These metrics include $D_\lambda$, $D_s$, and hybrid quality with no reference (HQNR) [1]. The HQNR index has an ideal value of 1, while $D_\lambda$ and $D_s$ have ideal values of 0.

For the MHIF dataset, we use another two commonly-used metrics for the evaluation, i.e., the peak signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM) [114]. The optimal values for PSNR and SSIM are $+\infty$ and 1, respectively.

The runtime of traditional pansharpening methods is evaluated using an Intel 12900k CPU on a reduced-resolution image from the WorldView-3 dataset, while DL-based methods are tested on an NVIDIA 3090 GPU. Likewise, the runtimes of the MHIF algorithms are assessed on the CAVE dataset using the same hardware devices.

### 4.5. Results on WorldView-3 Dataset

In this section, we conduct experiments to evaluate our DDIF on the WV3 dataset assessing performance on 20 testing images from the PanCollection dataset. We compared our method with three traditional methods and some recent state-of-the-art DL-based methods. The reduced-resolution and full-resolution results are reported in Tab. 2. To clearly demonstrate the advantages of our method, we present the visual comparisons in Fig. 6, proposing some close-ups to better show some details. Additionally, the error maps are shown accordingly.

On average, it can be seen that DL-based methods have significantly better performance than traditional methods. Among DL-based methods, our approach achieves the best performance on the reduced-resolution dataset and competitive performance on the full-resolution dataset. Our DDIF can reach the values of 2.73 (SAM) and 2.02 (ERGAS) on the reduced dataset, which outperforms all the compared DL-based techniques. The error maps also indicate that the image fused by DDIF is closer to the GT (as it has darker blue colors). The full-resolution images obtained by DDIF on the WV3 dataset can achieve state-of-the-art performance. Fig. 9 shows the full-resolution outcomes, as well as their HQNR maps, where a HQNR score closer to 1 indicates better fusion quality for the full-resolution image. The obtained results indicate that our DDIF can fuse HRMS images reducing spatial and spectral distortions, thus demonstrating that it has a good generalization at full-resolution.

### 4.6. Results on GaoFen-2 Dataset

In this section, we test the performance of the compared approaches on 20 GaoFen-2 test samples from the PanCollection dataset. As shown in Tab. 3, our DDIF outperforms the benchmark for almost all the reference and the non-reference metrics for reduced-resolution and full-resolution datasets. More specifically, we observe an improvement of $\approx 18\%/17\%/0.5\%$ in the SAM/ERGAS/Q4 metrics when comparing with the second best method, i.e., LAGConv [45]. Compared with the third best method, i.e., DCFNet [40], our DDIF improves $\approx 28\%/29\%/1.3\%$ on the SAM/ERGAS/Q4 metrics. The high performance on the full-resolution dataset reflects the one at reduced resolution. Besides, Fig. 7 depicts some close-ups in the rectangular boxes and the error maps among the compared DL-based methods. The colors and edge details of the objects in the figure are closer to the GT. Moreover, the error maps indicate that the residual between the outcome of DDIF and the GT is minimal since the color of the map is the darkest blue one. In terms of runtime, our method achieves state-of-the-art performance while maintaining favorable execution times compared to other DL-based methods. Since DDIF is an iterative approach, an increase in runtime is expected. However, our method significantly reduces the sampling time by employing Alg. 2 for accelerated sampling, enhancing its practical utility. In contrast, PanDiff [68] as a diffusion model, utilizing 2000 steps of DDPM sampling, incurs substantial time consumption for sampling.

### 4.7. Results on QuickBird Dataset

We also conducted experiments on the QB dataset assessing the performance both at reduced-resolution and full-resolution. Similarly, the reference and non-reference metrics are obtained

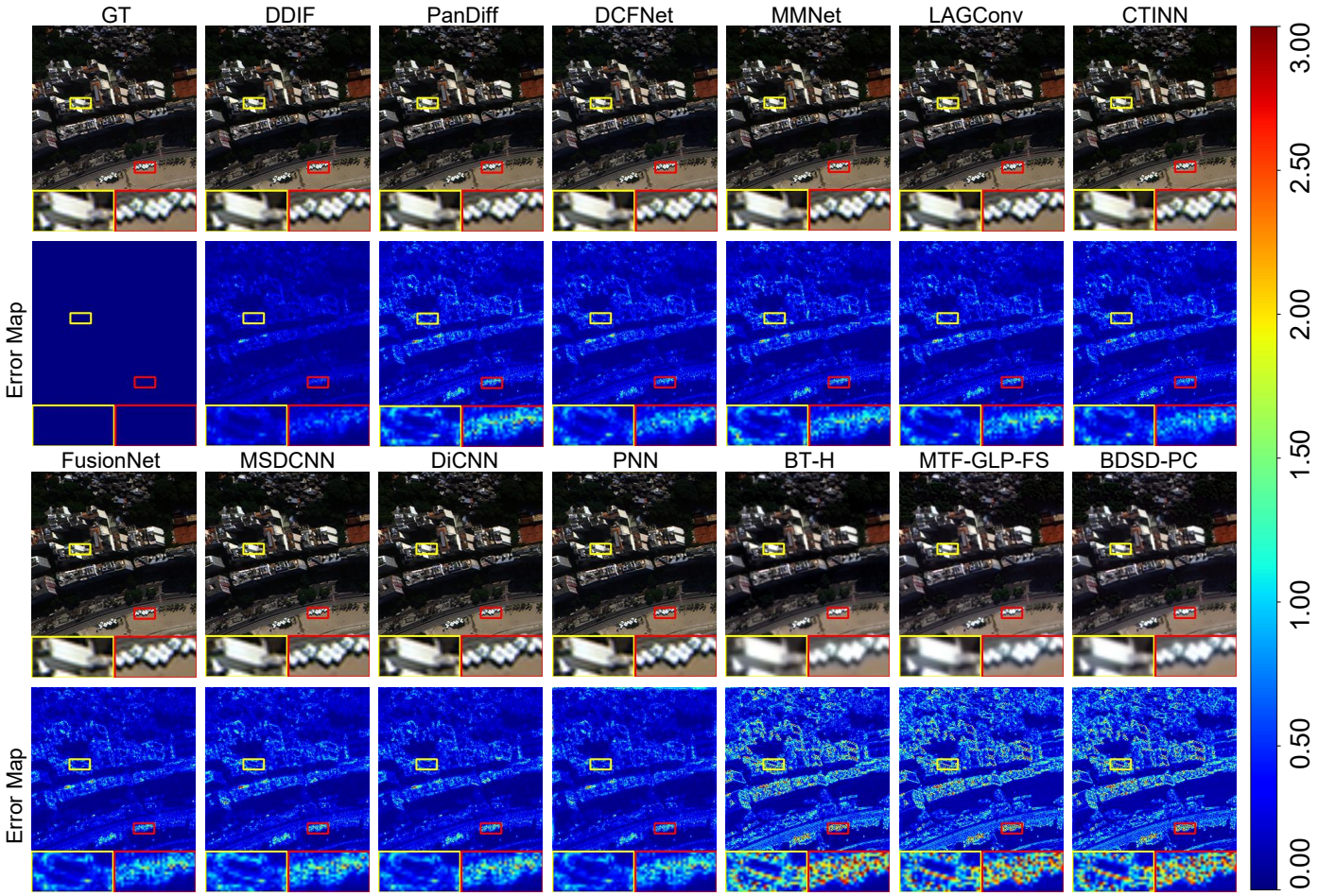



Figure 6: Visual comparisons with pansharpening methods on the WV3 dataset. The first and third rows show the RGB bands of the fused images. The second and fourth rows show the error maps between the GT and fused images. Some close-ups are depicted in red and yellow rectangles. The numerical errors are displayed on the color bar, with darker colors indicating closer proximity to the ground-truth (GT) (i.e., better fusion performance).

Table 2: Results on the WV3 reduced-resolution and full-resolution datasets. Some conventional methods (the first four rows) and DL-based approaches are compared. The best DL-based results are in red and the second best results are in blue.

| method | Reduced | | | | Full | | | Runtime(s) |
|---|---|---|---|---|---|---|---|---|
| | SAM(± std) | ERGAS(± std) | Q4(± std) | SCC(± std) | $D_\lambda$(± std) | $D_s$(± std) | HQNR(± std) | |
| BDSD-PC [100] | 5.4675±1.7185 | 4.6549±1.4667 | 0.8117±0.1063 | 0.9049±0.0419 | 0.0625±0.0235 | 0.0730±0.0356 | 0.8698±0.0531 | 0.059 |
| MTF-GLP-FS [101] | 5.3233±1.6548 | 4.6452±1.4441 | 0.8177±0.1014 | 0.8984±0.0466 | **0.0206±0.0082** | 0.0630±0.0284 | 0.9180±0.0346 | 0.023 |
| BT-H [99] | 4.8985±1.3028 | 4.5150±1.3315 | 0.8182±0.1019 | 0.9240±0.0243 | 0.0574±0.0232 | 0.0810±0.0374 | 0.8670±0.0540 | 0.321 |
| PNN [17] | 3.6798±0.7625 | 2.6819±0.6475 | 0.8929±0.0923 | 0.9761±0.0075 | **0.0213±0.0080** | 0.0428±0.0147 | 0.9369±0.0212 | 0.042 |
| DiCNN [38] | 3.5929±0.7623 | 2.6733±0.6627 | 0.9004±0.0871 | 0.9763±0.0072 | 0.0362±0.0111 | 0.0462±0.0175 | 0.9195±0.0258 | 0.083 |
| MSDCNN [39] | 3.7773±0.8032 | 2.7608±0.6884 | 0.8900±0.0900 | 0.9741±0.0076 | 0.0230±0.0091 | 0.0467±0.0199 | 0.9316±0.0271 | 0.112 |
| FusionNet [19] | 3.3252±0.6978 | 2.4666±0.6446 | 0.9044±0.0904 | 0.9807±0.0069 | 0.0239±0.0090 | 0.0364±0.0137 | **0.9406±0.0197** | 0.065 |
| CTINN [103] | 3.2523±0.6436 | 2.3936±0.5194 | 0.9056±0.0840 | 0.9826±0.0046 | 0.0550±0.0288 | 0.0679±0.0312 | 0.8815±0.0488 | 1.329 |
| LAGConv [45] | 3.1042±0.5585 | 2.2999±0.6128 | 0.9098±0.0907 | 0.9838±0.0068 | 0.0368±0.0148 | **0.0418±0.0152** | 0.9230±0.0247 | 1.381 |
| MMNet [102] | 3.0844±0.6398 | 2.3428±0.6260 | 0.9155±0.0855 | 0.9829±0.0056 | 0.0540±0.0232 | 0.0336±0.0115 | 0.9143±0.0281 | 0.348 |
| DCFNet [40] | **3.0264±0.7397** | **2.1588±0.4563** | **0.9051±0.0881** | **0.9861±0.0038** | 0.0781±0.0812 | 0.0508±0.0342 | 0.8771±0.1005 | 0.548 |
| PanDiff [68] | 3.2968±0.6010 | 2.4667±0.5837 | 0.8980±0.0880 | 0.9800±0.0063 | 0.0273±0.0123 | 0.0542±0.0264 | 0.9203±0.0360 | 261.410 |
| DDIF(ours) | **2.7386±0.5080** | **2.0165±0.4508** | **0.9202±0.0824** | **0.9882±0.0031** | 0.0258±0.0187 | **0.0231±0.0075** | **0.9517±0.0173** | 2.602 |
| Ideal value | **0** | **0** | **1** | **1** | **0** | **0** | **1** | **0** |

on 20 testing samples, randomly selected from the QB dataset. Tab. 4 reports the quality indexes. It is clear that our DDIF method significantly outperforms all the compared approaches at reduced resolution, leading to a substantial improvement in SAM scores, ranging from 4.538 to 4.349. Also, our DDIF can get competitive performance for full-resolution test cases. To

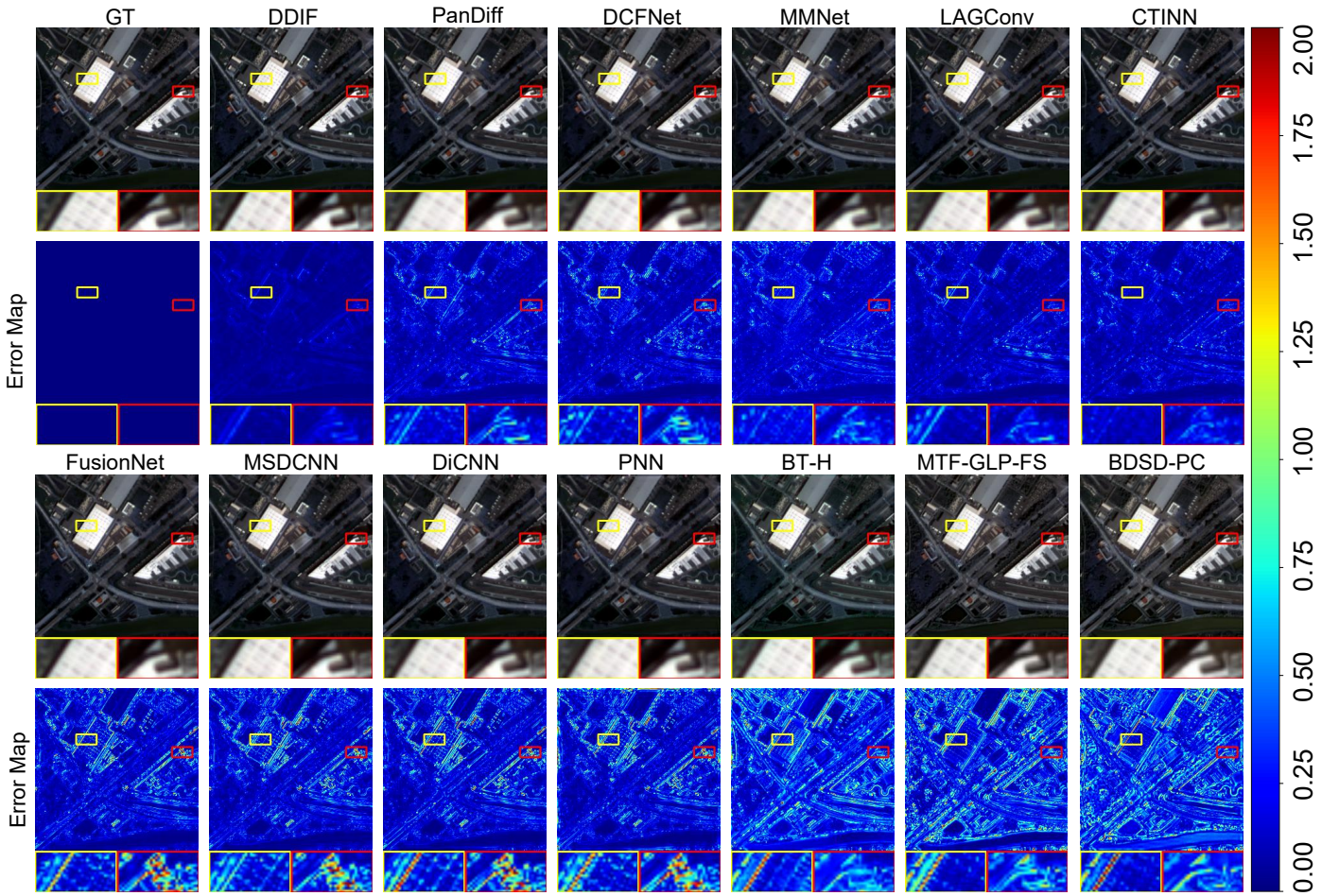



Figure 7: Visual comparisons on the GF2 dataset. The first and third rows show the RGB bands of the fused images. The second and fourth rows show the error maps between the GT and fused images. Some close-ups are depicted in red and yellow rectangles. The numerical errors are displayed on the color bar, with darker colors indicating closer proximity to the ground-truth (GT) (i.e., better fusion performance).

Table 3: Result on the GF2 reduced-resolution and full-resolution datasets. Some conventional methods (the first four rows) and the DL-based approaches are compared. The best results are in red and the second best results are in blue.

| method | Reduced | | | | Full | | |
|---|---|---|---|---|---|---|---|
| | SAM(± std) | ERGAS(± std) | Q4(± std) | SCC(± std) | $D_\lambda$(± std) | $D_s$(± std) | HQNR(± std) |
| BDSD-PC [100] | 1.7110±0.3210 | 1.7025±0.4056 | 0.9932±0.0308 | 0.9448±0.0166 | 0.0759±0.0301 | 0.1548±0.0280 | 0.7812±0.0409 |
| MTF-GLP-FS [101] | 1.6757±0.3457 | 1.6023±0.3545 | 0.8914±0.0256 | 0.9390±0.0197 | 0.0336±0.0129 | 0.1404±0.0277 | 0.8309±0.0334 |
| BT-H [99] | 1.6810±0.3168 | 1.5524±0.3642 | 0.9089±0.0292 | 0.9508±0.0150 | 0.0602±0.0252 | 0.1313±0.0193 | 0.8165±0.0305 |
| PNN [17] | 1.0477±0.2264 | 1.0572±0.2355 | 0.9604±0.0100 | 0.9772±0.0054 | 0.0367±0.0291 | 0.0943±0.0224 | 0.8726±0.0373 |
| DiCNN [38] | 1.0525±0.2310 | 1.0812±0.2510 | 0.9594±0.0101 | 0.9771±0.0058 | 0.0413±0.0128 | 0.0992±0.0131 | 0.8636±0.0165 |
| MSDCNN [39] | 1.0472±0.2210 | 1.0413±0.2309 | 0.9612±0.0108 | 0.9782±0.0050 | 0.0269±0.0131 | 0.0730±0.0093 | 0.9020±0.0128 |
| FusionNet [19] | 0.9735±0.2117 | 0.9878±0.2222 | 0.9641±0.0093 | 0.9806±0.0049 | 0.0400±0.0126 | 0.1013±0.0134 | 0.8628±0.0184 |
| CTINN [103] | 0.8251±0.1386 | 0.6995±0.1068 | 0.9772±0.0117 | 0.9803±0.0015 | 0.0586±0.0260 | 0.1096±0.0149 | 0.8381±0.0237 |
| LAGConv [45] | **0.7859±0.1478** | **0.6869±0.1125** | **0.9804±0.0085** | **0.9906±0.0019** | 0.0324±0.0130 | 0.0792±0.0136 | 0.8910±0.0204 |
| MMNet [102] | 0.9929±0.1411 | 0.8117±0.1185 | 0.9690±0.0204 | 0.9859±0.0024 | 0.0428±0.0300 | 0.1033±0.0129 | 0.8583±0.0269 |
| DCFNet [40] | 0.8896±0.1577 | 0.8061±0.1369 | 0.9727±0.0100 | 0.9853±0.0024 | **0.0234±0.0116** | **0.0659±0.0096** | **0.9122±0.0119** |
| PanDiff [68] | 0.8881±0.1197 | 0.7461±0.1032 | 0.9792±0.0097 | 0.9887±0.0020 | 0.0265±0.0195 | 0.0729±0.0103 | 0.9025±0.0209 |
| DDIF(ours) | **0.6408±0.1203** | **0.5668±0.1010** | **0.9855±0.0078** | **0.9859±0.0035** | **0.0201±0.0109** | **0.0408±0.0103** | **0.9398±0.0137** |
| Ideal value | **0** | **0** | **1** | **1** | **0** | **0** | **1** |

better visualize the performance gap, Fig. 8 depicts the fused images and the error maps.

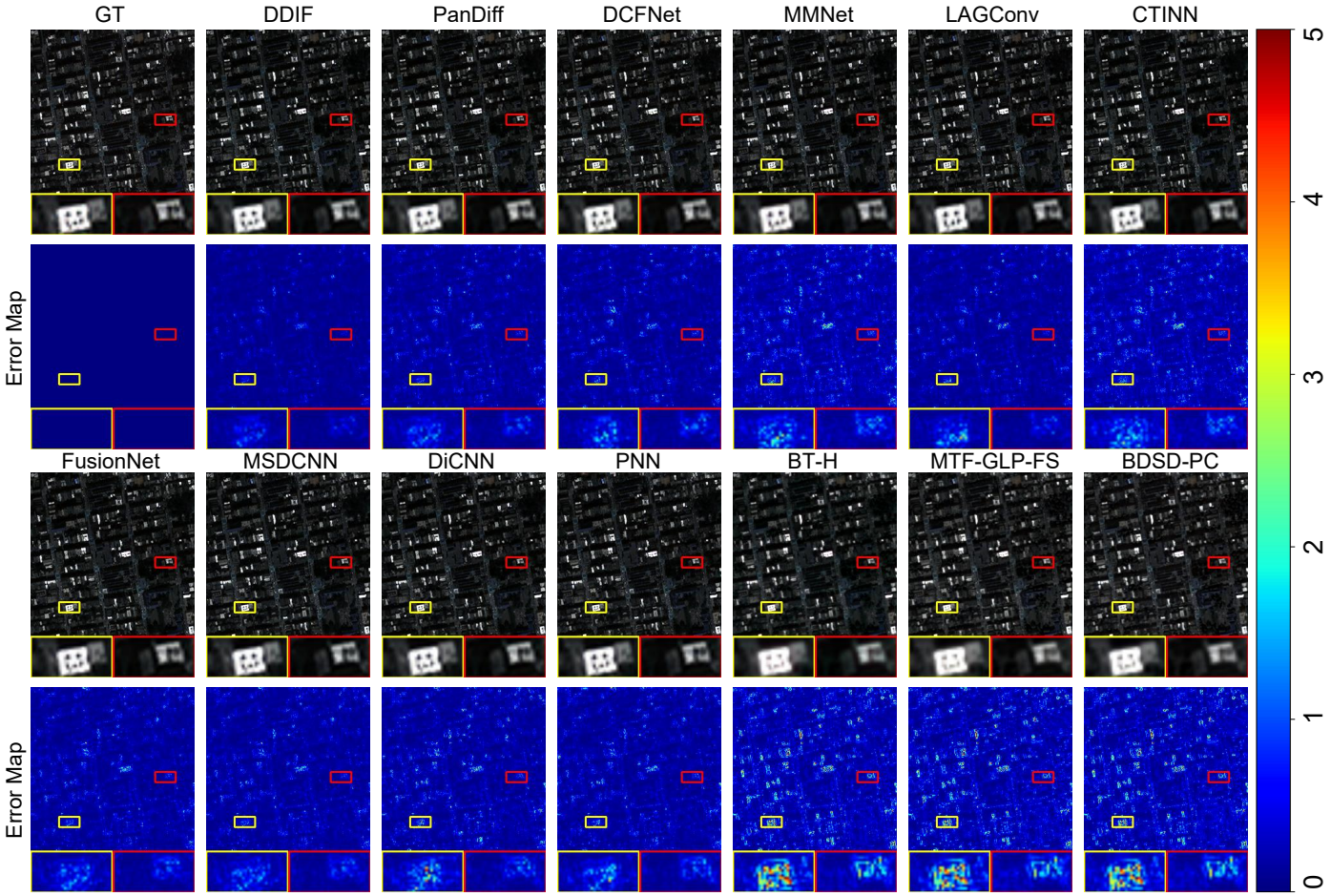



Figure 8: Visual comparisons on the QB dataset. The first and third rows show the RGB bands of the fused images. The second and fourth rows show the error maps between the GT and fused images. Some close-ups are depicted in red and yellow rectangles. The numerical errors are displayed on the color bar, with darker colors indicating closer proximity to the ground-truth (GT) (i.e., better fusion performance).

Table 4: Quantitative results on the QuickBird reduced-resolution and full-resolution datasets. Some conventional methods (the first three rows) and DL-based methods are compared. The best results are in red and the second best results are in blue.

| method | Reduced | | | | Full | | |
|---|---|---|---|---|---|---|---|
| | SAM(± std) | ERGAS(± std) | Q4(± std) | SCC(± std) | $D_\lambda$(± std) | $D_s$(± std) | HQNR(± std) |
| BDSD-PC [100] | 8.2620±2.0497 | 7.5420±0.8138 | 0.8323±0.1013 | 0.9030±0.0181 | 0.1975±0.0334 | 0.1636±0.0483 | 0.6722±0.0577 |
| MTF-GLP-FS [101] | 8.1131±1.9553 | 7.5102±0.7926 | 0.8296±0.0905 | 0.8998±0.0196 | **0.0489±0.0149** | 0.1383±0.0238 | 0.8199±0.0340 |
| BT-H [99] | 7.1943±1.5523 | 7.4008±0.8378 | 0.8326±0.0880 | 0.9156±0.0152 | 0.2300±0.0718 | 0.1648±0.0167 | 0.6434±0.0645 |
| PNN [17] | 5.2054±0.9625 | 4.4722±0.3734 | 0.9180±0.0938 | 0.9711±0.0123 | 0.0569±0.0112 | 0.0624±0.0239 | 0.8844±0.0304 |
| DiCNN [38] | 5.3795±1.0266 | 5.1354±0.4876 | 0.9042±0.0942 | 0.9621±0.0133 | 0.0920±0.0143 | 0.1067±0.0210 | 0.8114±0.0310 |
| MSDCNN [39] | 5.1471±0.9342 | 4.3828±0.3400 | 0.9188±0.0966 | 0.9689±0.0121 | 0.0602±0.0150 | 0.0667±0.0289 | 0.8774±0.0388 |
| FusionNet [19] | 4.9226±0.9077 | 4.1594±0.3212 | 0.9252±0.0902 | 0.9755±0.0104 | 0.0586±0.0189 | **0.0522±0.0088** | **0.8922±0.0219** |
| CTINN [103] | 4.6583±0.7755 | 3.6969±0.2888 | 0.9320±0.0072 | 0.9829±0.0072 | 0.1738±0.0332 | 0.0731±0.0237 | 0.7663±0.0432 |
| LAGConv [45] | 4.5473±0.8296 | **3.8259±0.4196** | **0.9335±0.0878** | **0.9807±0.0091** | 0.0844±0.0238 | 0.0676±0.0136 | 0.8536±0.0178 |
| MMNet [102] | 4.5568±0.7285 | 3.6669±0.3036 | 0.9337±0.0941 | 0.9829±0.0070 | 0.0890±0.0512 | 0.0972±0.0382 | 0.8225±0.0319 |
| DCFNet [40] | **4.5383±0.7397** | 3.8315±0.2915 | 0.9325±0.0903 | 0.9741±0.0101 | **0.0454±0.0147** | 0.1239±0.0269 | 0.8360±0.0158 |
| PanDiff [68] | 4.5754±0.7359 | 3.7422±0.3099 | 0.9345±0.0902 | 0.9818±0.0902 | 0.0587±0.0223 | 0.0642±0.0252 | 0.8813±0.0417 |
| DDIF(ours) | **4.3496±0.7313** | **3.5223±0.2703** | **0.9375±0.0904** | **0.9845±0.0069** | 0.0583±0.0126 | **0.0492±0.0103** | **0.8954±0.0206** |
| Ideal value | **0** | **0** | **1** | **1** | **0** | **0** | **1** |

### *4.8. Ablation Study*

In this section, ablation studies are conducted on CSM, FWM, and the other performance-boosting techniques to verify their effectiveness.

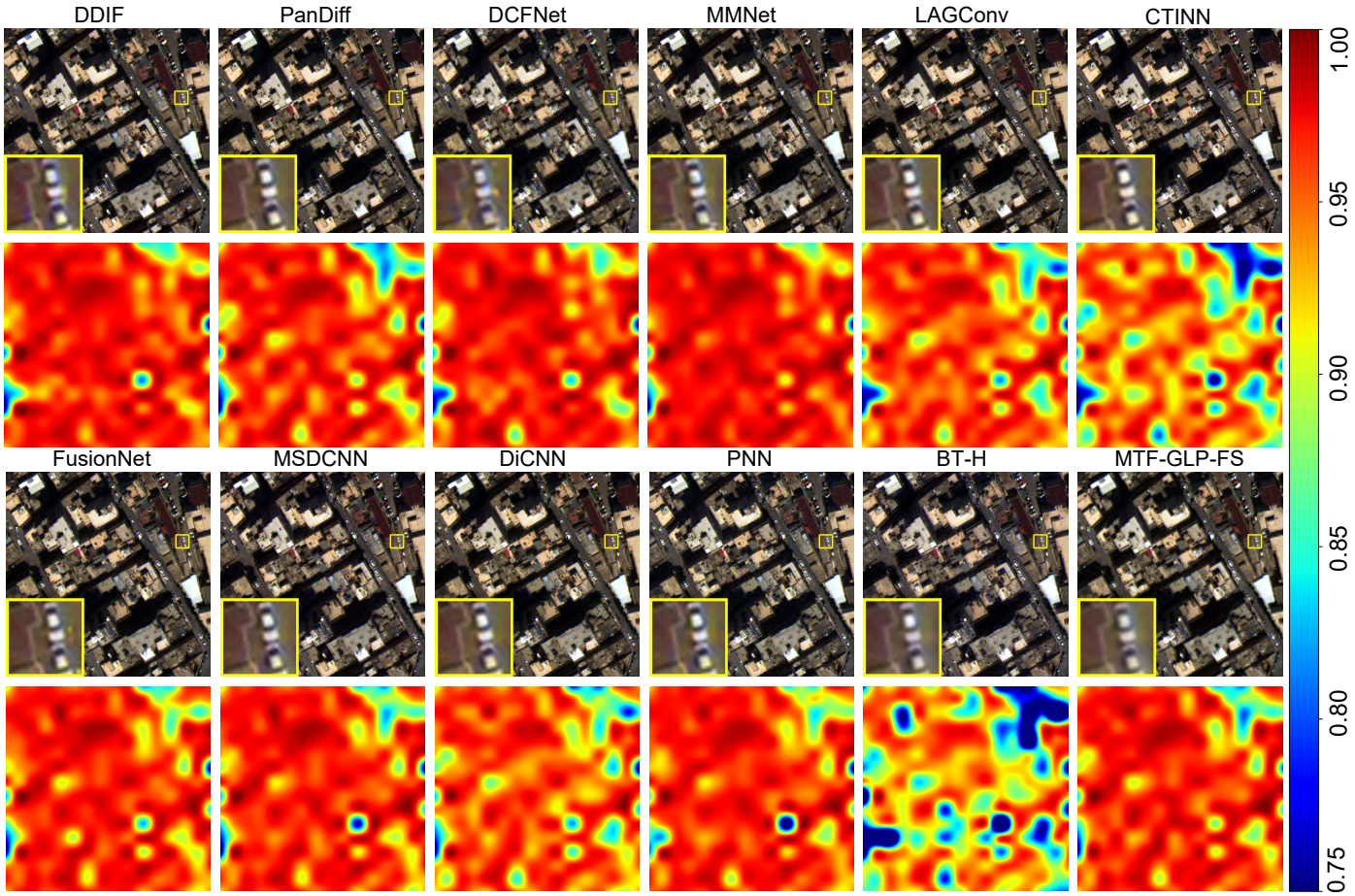



Figure 9: Fused WV3 full-resolution data and their corresponding HQNR map. The high value in the HQNR map means better full-resolution fusion performance.

#### 4.8.1. *Disentangled Modulation*

We propose two disentangled conditional modulation modules, dubbed CSM and FWM, which are respectively responsible for coarse-grain style condition modulation and fine-grain frequency modulation. Without using them, one intuitive way is to concatenate conditions (i.e., the LRMS and PAN images) with the noisy input along the channel dimension. We ablate the proposed two modulations by training the diffusion U-Net in the following forms:

1. Only concatenating conditions and input together as proposed in [86, 74].
2. Only using CSM.
3. Conducting two modulations (i.e., CSM and FWM) in the diffusion U-Net.

Each of the above-mentioned models has been trained until convergence for a fair comparison. The performance on the WV3 reduced-resolution test set is reported in Tab. 5. With the addition of the style modulation and the wavelet modulation, the fusion performance exhibits a monotonically increase. It should be noted that without these two modulations, our DDIF degrades to DDPM [23] just adding the residual learning, thus indicating that directly applying DDPM to the fusion task has poor performance.

Table 5: Ablation study on the two modulation modules. Best results are in red.

| Style Transfer Modulation | Wavelet Modulation | SAM(±std) | ERGAS(±std) | Q8(±std) | SCC(±std) |
|---|---|---|---|---|---|
| ✗ | ✗ | 3.2851±0.6828 | 2.5501±0.6141 | 0.8981±0.0904 | 0.9796±0.0067 |
| ✓ | ✗ | 3.1418±0.5789 | 2.3527±0.5116 | 0.8971±0.0940 | 0.9837±0.0047 |
| ✓ | ✓ | **2.7386±0.5080** | **2.0165±0.4508** | **0.9202±0.0824** | **0.9882±0.0031** |

Table 6: Ablation study on using residual learning. Best results are in red.

| Residual Learning | SAM(±std) | ERGAS(±std) | Q8(±std) | SCC(±std) |
|---|---|---|---|---|
| ✗ | 3.2397±0.4546 | 3.2096±1.003 | 0.9061±0.0837 | 0.9729±0.0113 |
| ✓ | **2.7386±0.5080** | **2.0165±0.4508** | **0.9202±0.0824** | **0.9882±0.0031** |

Table 7: Ablation study on training objectives of the diffusion model. Best results are in red.

| Objective | SAM(±std) | ERGAS(±std) | Q8(±std) | SCC(±std) |
|---|---|---|---|---|
| $\epsilon$ | 3.7702±0.6397 | 2.7954±0.6516 | 0.8388±0.1181 | 0.9794±0.0064 |
| $\mathbf{v}$ | 3.4853±0.6080 | 2.6624±0.5912 | 0.8715±0.1025 | 0.9808±0.0052 |
| $\mathbf{x}_0$ | **2.7386±0.5080** | **2.0165±0.4508** | **0.9202±0.0824** | **0.9882±0.0031** |

#### 4.8.2. *Residual Learning*

To verify the effectiveness of the residual learning technique, we return the input to the noisy GT as $\mathbf{x}_t$. Then, we retrain the

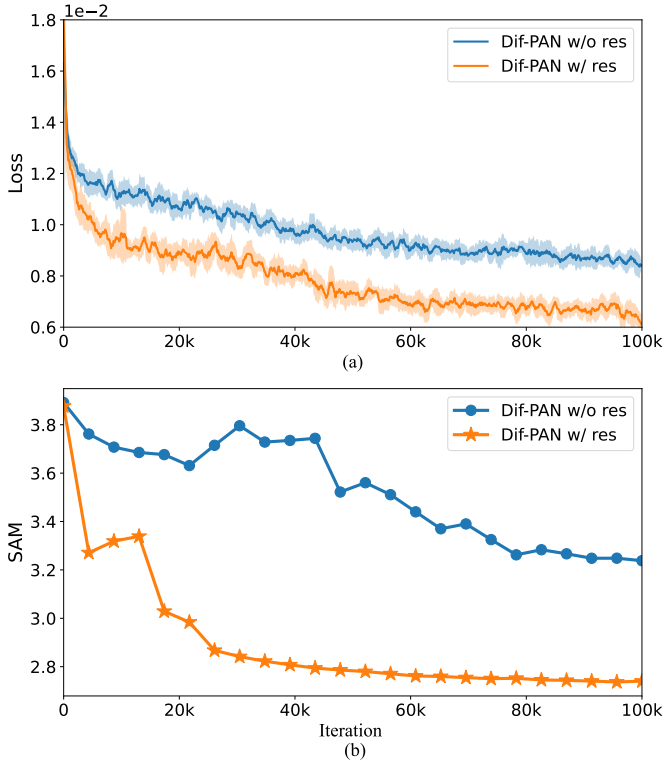



Figure 10: Changes in loss (a) and SAM metric (b) on the WV3 dataset of our DDIF over iterations with and without the residual learning technique (w/ res, w/o res).

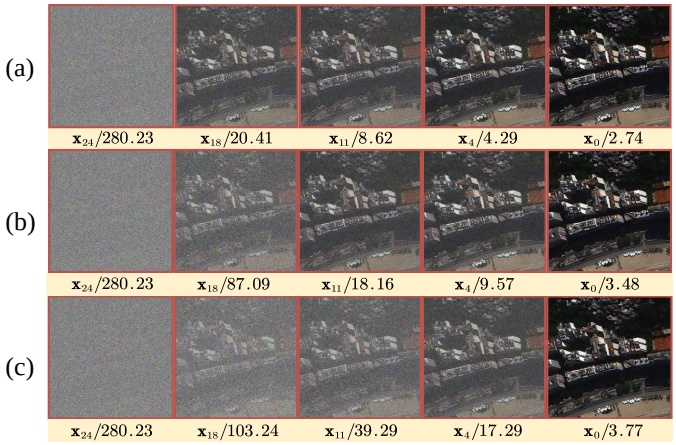



Figure 11: The denoised $\mathbf{x}_t$ and the related SAM metrics over different timesteps for different training objectives focusing on pansharpening. (a), (b), and (c) represent the training objectives $\mathbf{x}_0$, $\mathbf{v}$, and $\epsilon$, respectively.

diffusion model until the model converges calculating its performance on the WV3 dataset as shown in Tab. 6. With residual learning, the diffusion model can get fused images closer to the GT. Besides, the loss convergence is faster, as shown in Fig. 10. One hypothesis is that the residual distribution is simpler than the GT distribution, thus learning in a more efficient manner. A direct way to express the complexity of a distribution is the entropy. We computed the pixel entropy of the GT and the residual image (i.e., HRMS − LRMS) and we found that the entropy of the GT is 6.498 bits-per-pixel (bpp) and the one of the residual image is only 4.461 bpp, which indicates that the residual image distribution is simpler. The entropy, $\mathcal{H}_I$, is defined as follows:

$$\mathcal{H}_I = \frac{1}{C}\sum_{c=0}^{C}\sum_{p\in I_c} p\log_2(p), \tag{17}$$

where $C$ is the number of channels, $I_c$ denotes the channel image, and $p$ is the value of a pixel.

#### 4.8.3. Training Objectives

The training objectives of the model can be $\epsilon$, $\mathbf{x}_0$, or $\mathbf{v}$, where $\epsilon$ indicates that the model needs to predict the added Gaussian noise, $\mathbf{x_0}$ denotes that the model predicts the original (noiseless) image from the noisy image, and $\mathbf{v}$ reflects the fact that the model predicts the weighted sum of $\epsilon$ and $\mathbf{x_0}$. We separately treat these three training objectives, ensuring the convergence of the diffusion model in all three cases on the WV3 dataset. Their performance is reported in Tab. 7. As we can see, predicting $\mathbf{x_0}$ gets the best performance. The backward process of the three objectives is illustrated in Fig. 11. Starting from the same $\mathbf{x}_T$, the $\mathbf{x}_0$ objective denoises much faster than the other two options. We guess that for small-scale datasets, $\mathbf{x_0}$ is more straightforward than $\epsilon$ and $\mathbf{v}$ since the sample density is more concentrated, but, for large-scale datasets, predicting $\epsilon$ and $\mathbf{v}$ may force the network to learn detailed denoising trajectory, which is useful to generate images with high-level conditions, such as, classification labels [116], texts [58], and bounding boxes [117].

## 5. Discussion

In this section, we will analyze the generalization ability, and then extend our DDIF to the MHIF task. Finally, we will compare the proposed approach with traditional deep regressive and GAN-based models.

### 5.1. Generalization Ability on WorldView-2 Dataset

Thanks to the powerful fitting ability of neural networks, they often perform well on data sharing the same domain. However, once test data are shifted to a different domain (never seen by the network during the training), the network usually performs poorly. To assess the generalization ability, we evaluate our DDIF model, trained on the WV3 dataset, in a zero-shot manner[5] on WorldView-2 data. We also compare our method with other DL-based methods as reported in Tab. 8. The proposed method achieves competitive results, thus demonstrating its good generalization. These results are in agreement with some related works that apply the diffusion model in a zero-shot manner to tasks as image inverse problem [76], image classification [118], and semantic segmentation [119].

[5]"Zero-shot" means that we directly use the trained model to test on another dataset without any fine-tuning.

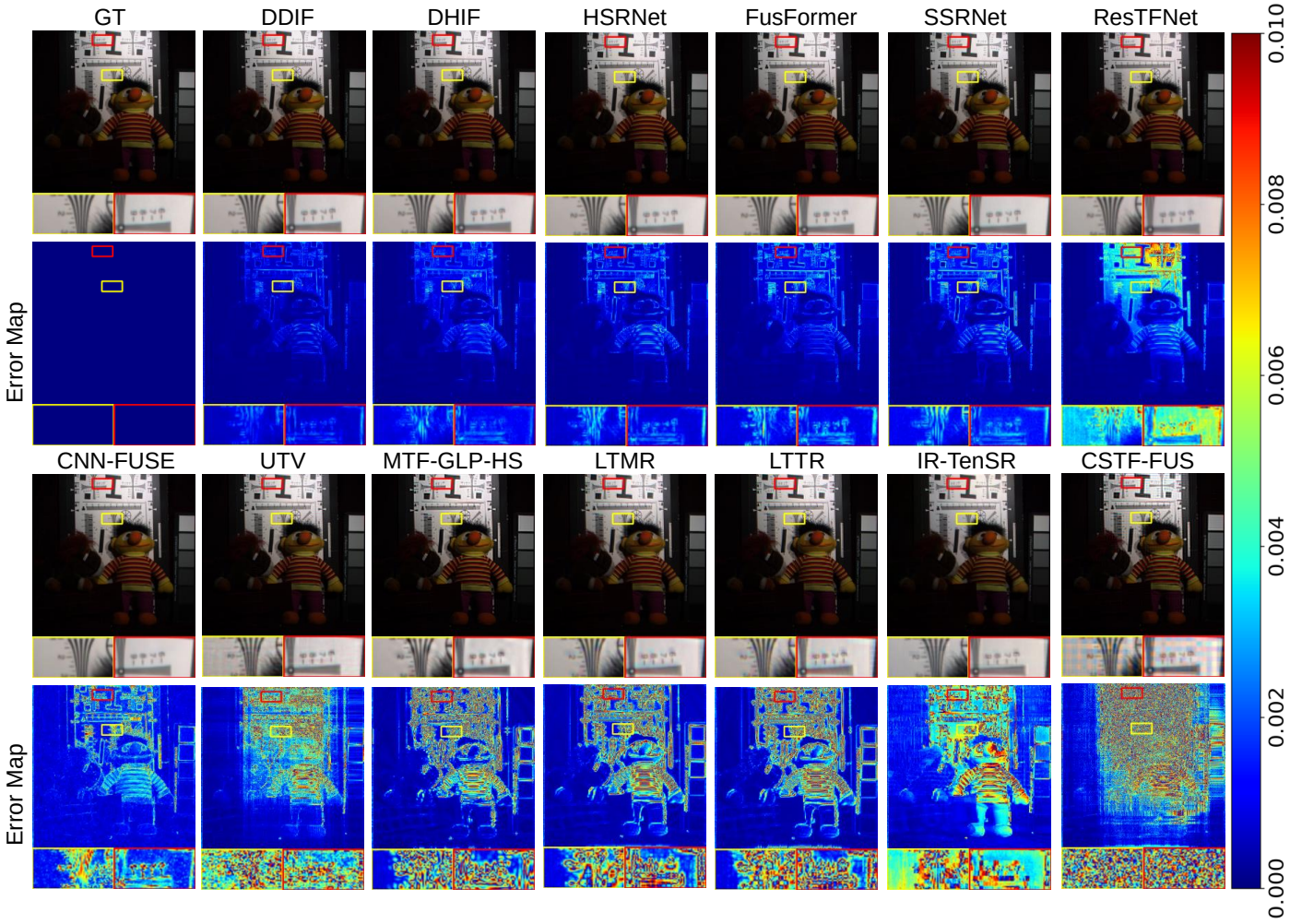



Figure 12: The first and third rows show the natural color results on "*chart and stuffed toy*". Some close-ups are depicted in the red and yellow rectangles. The second and fourth rows show the error maps between the GT and the fused images. The darker blue error maps indicate better performance.

Table 8: Generalization ability of DL-based methods. The best results are in red and the second best results are in blue.

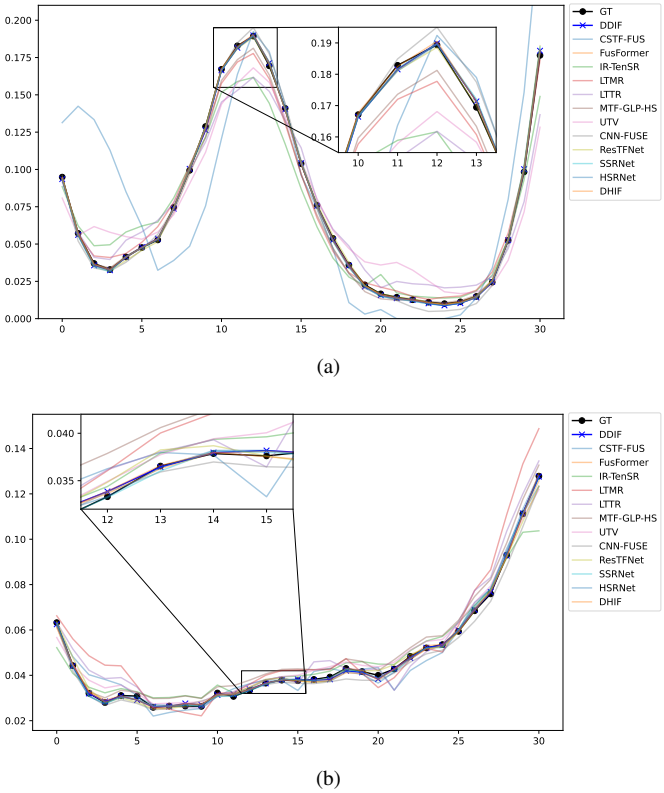
| method | Reduced | | | | Full | | |
|---|---|---|---|---|---|---|---|
| | SAM(± std) | ERGAS(± std) | Q4(± std) | SCC(± std) | $D_\lambda$(± std) | $D_s$(± std) | HQNR(± std) |
| PNN [17] | 7.1158±1.6812 | 5.6152±0.9431 | 0.7619±0.0928 | 0.8782±0.0175 | 0.1484±0.0957 | 0.0771±0.0169 | 0.7869±0.0959 |
| DiCNN [38] | 6.9216±0.7898 | 6.2507±0.5745 | 0.7205±0.0746 | 0.8552±0.0289 | 0.1412±0.0661 | 0.1023±0.0195 | 0.7700±0.0505 |
| MSDCNN [39] | 6.0064±0.6377 | 4.7438±0.4939 | **0.8241±0.0799** | 0.8972±0.0109 | 0.0589±0.0421 | **0.0290±0.0138** | **0.9143±0.0516** |
| FusionNet [19] | 6.4257±0.8602 | 5.1363±0.5151 | 0.7961±0.0737 | 0.8746±0.0134 | **0.0519±0.0292** | 0.0559±0.0146 | 0.8948±0.0187 |
| CTINN [103] | 6.4103±0.5953 | 4.6435±0.3792 | 0.8172±0.0873 | **0.9147±0.0102** | 0.1722±0.0373 | 0.0375±0.0065 | 0.7967±0.0360 |
| LAGConv [45] | 6.9545±0.4739 | 5.3262±0.3185 | 0.8054±0.0837 | 0.9125±0.0101 | 0.1302±0.0856 | 0.0547±0.0159 | 0.8229±0.0884 |
| MMNet [102] | 6.6109±0.3209 | 5.2213±0.2133 | 0.8143±0.0790 | 0.9136±0.0201 | 0.0897±0.0340 | 0.0688±0.0209 | 0.8476±0.0569 |
| DCFNet [40] | 5.6194±0.6039 | **4.4887±0.3764** | **0.8292±0.0815** | **0.9154±0.0083** | 0.0649±0.0357 | 0.0700±0.0219 | 0.8690±0.0233 |
| DDIF(ours) | **5.3827±0.5737** | **4.6712±0.4155** | 0.8217±0.0777 | 0.8993±0.0129 | **0.0313±0.0376** | **0.0312±0.0111** | **0.9388±0.0453** |
| Ideal value | **0** | **0** | **1** | **1** | **0** | **0** | **1** |

### *5.2. Extension to the MHIF Task*

We additionally conduct experiments on the CAVE dataset. The results and comparisons with other methods are provided in Tab. 9. It is worth noting that, compared to the state-of-the-art MHIF method, i.e., DHIF [33], our method achieves a 0.1dB higher PSNR and a higher Q2n, as well as competitive performance on the other metrics. Fusion images and error maps are depicted in Fig. 12. To analyze the spectral accuracy, the spectral signatures, obtained by extracting two different pixels in two testing images, are compared in Fig. 13 showing that our

Table 9: Quantitative results on the CAVE (×4) dataset. Some conventional methods and DL-based approaches are compared. The best results are in red and the second best results are in blue.

| method | PSNR(± std) | SSIM(± std) | Q2n(± std) | SAM(± std) | ERGAS(± std) | Runtime(s) |
|---|---|---|---|---|---|---|
| CSTF-FUS [30] | 34.4632±4.2806 | 0.8662±0.0747 | 0.6659±0.1586 | 14.3683±5.3020 | 8.2885±5.2848 | 12.766 |
| IR-TenSR [29] | 35.6081±3.4461 | 0.9451±0.0267 | 0.7774±0.1228 | 12.2950±4.6825 | 5.8969±3.0455 | 14.073 |
| LTTR [27] | 35.8505±3.4883 | 0.9562±0.0288 | 0.8404±0.0979 | 6.9895±2.5542 | 5.9904±2.9211 | 241.207 |
| LTMR [105] | 36.5434±3.2995 | 0.9631±0.0208 | 0.8416±0.1031 | 6.7105±2.1934 | 5.3868±2.5286 | 174.807 |
| MTF-GLP-HS [115] | 37.6920±3.8528 | 0.9725±0.0158 | 0.8716±0.0847 | 5.3281±1.9119 | 4.5749±2.6605 | 12.766 |
| UTV [28] | 38.6153±4.0640 | 0.9410±0.0434 | 0.7752±0.1416 | 8.6488±3.3764 | 4.5189±2.8173 | 613.535 |
| CNN-FUSE [31] | 42.4281±3.1994 | 0.9782±0.0079 | 0.9419±0.0240 | 5.7599±2.1509 | 2.8420±1.7590 | 0.481 |
| ResTFNet [41] | 45.5842±5.4647 | 0.9939±0.0055 | 0.9581±0.0315 | 2.7643±0.6988 | 2.3134±2.4377 | 0.485 |
| SSRNet [42] | 48.6196±3.9182 | 0.9954±0.0024 | 0.9598±0.0309 | 2.5415±0.8369 | 1.6358±1.2191 | 0.594 |
| Fusformer [32] | 49.9831±8.0965 | 0.9943±0.0114 | 0.9624±0.0362 | 2.2033±0.8510 | 2.5337±5.3052 | 17.143 |
| HSRNet [109] | 50.3805±3.3802 | 0.9970±0.0015 | 0.9666±0.0290 | 2.2272±0.6575 | **1.2002±0.7506** | 0.690 |
| DHIF [33] | **51.0721±4.1648** | **0.9973±0.0017** | **0.9695±0.0267** | **2.0080±0.6304** | **1.2216±0.9653** | 0.837 |
| DDIF(ours) | **51.1758±4.6148** | **0.9971±0.0026** | **0.9737±0.0106** | **2.0952±0.6471** | 1.2996±1.2822 | 10.706 |
| Ideal value | ∞ | **1** | **1** | **0** | **0** | **0** |




(a)

(b)

Figure 13: Spectral vectors of the compared approaches: (a) spectral vectors in "*feather*" located at position (400, 200), (b) spectral vectors in "*chart and stuffed toy*" located at position (400, 200). The horizontal and vertical axes represent the band number and the pixel values, respectively.

DDIF has a more accurate spectral consistency.

To better illustrate the superiority of our DDIF on real hyperspectral data, we conducted fusion experiments on the GF5-GF1 dataset. As shown in Tab. 10, our approach achieves state-of-the-art performance on reduced-resolution data, also demonstrating excellent results at full-resolution.

We test the runtime of various methods on the CAVE dataset. Although our method exhibits a slightly longer runtime compared to other DL-based methods, considering that DDIF is an iterative approach, the proposed method's runtime remains within an acceptable range. It is worth noting that the runtime of FusFormer [32] is significantly higher than that of other methods. This is attributed to FusFormer's inability to directly operate on a $512 \times 512$ image. It requires the image to be partitioned into smaller patches, and the network performs fusion on each patch before assembling them back together, resulting in an extended runtime. Our designed network does not encounter this issue.

### 5.3. Comparisons with Other Unsupervised Methods

Unsupervised methods are directly trained at full-resolution, thus getting good performance when full-resolution metrics are considered. In contrast, our DDIF only accesses reduced-resolution data during its training. This train-test resolution mismatch poses challenges for supervised methods. We selected a recent state-of-the-art unsupervised regressive method, LDPNet [54], and a representative GAN-based method, ZerGAN [51], for comparison. Tab. 11 shows the comparisons with LDPNet and ZerGAN at full-resolution. It can be seen that, even without accessing full-resolution data during the training, our DDIF can still generalize well and obtain high performance on full-resolution data in comparison with state-of-the-art unsupervised methods.

### 5.4. Differences with Regressive Models

Regressive models can be trained in both the supervised and unsupervised manner. The mainstream works, such as [19, 40], focus on designing powerful architectures or neural operators [45] and then training the model in a supervised manner. Other works [120] chose to bound the regressive mapping by adding a regularized loss on fused outcomes training in an unsupervised way. The proposed DDIF is still a supervised training model, which benefits from the noiseless supervised signal from the GT rather than more complex regularized terms. However, our method can also be transformed into an approach that relies upon unsupervised training by adding constraints on the training loss, $L_{simple}$ (as done by most of the unsupervised works).

Table 10: Result on the GF5-GF1 reduced-resolution and full-resolution datasets. Some conventional methods (the first six rows) and the DL-based approaches are compared. The best results are in red and the second best results are in blue.

| method | Simulated | | | | | Real | | |
|---|---|---|---|---|---|---|---|---|
| | PSNR(± std) | SSIM(± std) | Q2n(± std) | SAM(± std) | ERGAS(± std) | $D_\lambda$(± std) | $D_s$(± std) | HQNR(± std) |
| CNMF [106] | 44.2525±3.8884 | 0.9823±0.0122 | 0.7420±0.1767 | 0.8509±0.2133 | 2.7609±0.7674 | 0.0447±0.0830 | 0.0592±0.0500 | 0.8979±0.0842 |
| Hysure [107] | 42.5184±4.5175 | 0.9728±0.0137 | 0.7323±0.1464 | 1.3046±0.4059 | 3.6770±1.3169 | 0.0414±0.0760 | 0.0741±0.1046 | 0.8865±0.1169 |
| GSA [108] | 44.9948±5.3369 | 0.9795±0.0119 | 0.7544±0.1313 | 1.2003±0.3322 | 2.8978±0.9557 | 0.0526±0.1029 | 0.0674±0.0615 | 0.8818±0.1006 |
| LTTR [27] | 47.1451±2.9068 | 0.9897±0.0028 | 0.8442±0.0983 | 2.1593±0.2858 | 5.8080±2.7321 | 0.0989±0.1231 | 0.0465±0.0233 | 0.8596±0.1063 |
| LTMR [105] | 45.5163±2.4245 | 0.9898±0.0030 | 0.8494±0.1079 | 1.5950±0.3435 | 2.7199±1.2766 | 0.0567±0.1029 | 0.0361±0.0177 | 0.9098±0.1048 |
| MTF-GLP-HS [115] | 45.5954±5.8205 | 0.9837±0.0120 | 0.7772±0.1462 | 0.8561±0.2646 | 2.8475±1.1541 | 0.0303±0.0482 | 0.0747±0.1054 | 0.8966±0.1063 |
| ResTFNet [41] | 46.9763±2.1124 | 0.9934±0.0022 | 0.8504±0.1000 | 0.9060±0.1301 | 3.3234±3.1232 | 0.0423±0.0782 | 0.0880±0.0587 | 0.8742±0.0994 |
| SSRNet [42] | 45.4872±2.6866 | 0.9880±0.0047 | 0.8500±0.0942 | 1.0389±0.2101 | 4.8631±4.1605 | 0.1169±0.1401 | 0.0538±0.0190 | 0.8357±0.1341 |
| Fusformer [32] | 49.7373±4.6446 | 0.9914±0.0031 | 0.8908±0.0762 | 0.6382±0.1552 | 4.7612±0.5921 | **0.0302±0.0558** | 0.0401±0.0250 | 0.9312±0.0636 |
| HSRNet [109] | 49.8109±3.0477 | 0.9964±0.0016 | 0.8883±0.0806 | 0.6925±0.1386 | 0.9006±0.4513 | 0.0377±0.0734 | 0.0473±0.0201 | 0.9170±0.0748 |
| DHIF [33] | **55.3538±4.2007** | **0.9982±0.0009** | **0.9285±0.0758** | **0.3088±0.0622** | **0.8852±0.3882** | **0.0312±0.0574** | **0.0335±0.0215** | **0.9365±0.0624** |
| DDIF(ours) | **56.4022±3.8194** | **0.9984±0.0007** | **0.9382±0.0638** | **0.2726±0.0489** | **0.8449±0.5053** | 0.0327±0.0571 | **0.0350±0.0213** | **0.9333±0.0573** |
| Ideal value | $+\infty$ | **1** | **1** | **0** | **0** | **0** | **0** | **1** |

Table 11: Quantitative comparisons with recent state-of-the-art unsupervised methods on the WV3 full-resolution dataset.

| method | $D_\lambda$ | $D_s$ | HQNR |
|---|---|---|---|
| LDPNet [54] | 0.0235±0.0085 | 0.0364±0.0192 | 0.9411±0.0192 |
| ZerGAN [51] | 0.0221±0.0092 | 0.0210±0.0082 | 0.9574±0.0170 |
| DDIF(ours) | 0.0258±0.0187 | 0.0231±0.0075 | 0.9517±0.0173 |

### *5.5. Differences with Unsupervised GAN-based Models*

GAN-based pansharpening models can be seen as an extension of an unsupervised regressive model by adding one or multiple discriminators and a GAN loss[6] [50, 51]. Benefiting from additional supervised discriminator signals, those models can get more realistic images. However, GAN-based models suffer from training instability and need careful tuning because of the adversarial training. Our DDIF does not use adversarial training exploiting a corrupt-reconstruct supervised training, which avoids unstable training but still produces high-quality fused images.

## 6. Acknowledge

This research is supported by National Key Research and National Natural Science Foundation of China (12271083), Development Program of China (Grant No. 2020YFA0714001), and Natural Science Foundation of Sichuan Province (2022NSFSC0501, 2023NSFSC1341, 2022NSFSC1821).

## 7. Conclusions

In this paper, we proposed a denoising diffusion model, the so-called DDIF, for two MSIF tasks. Our motivation is that supervised DL-based methods suffer from degraded learning ability under the preconditioned framework and the conditions are entangled, which is not suitable for the fusion task. Hence, we designed two novel feature modulation modules, i.e., CSM and FWM, to leverage on the learning ability by using DPM. Experiments conducted on widely used pansharpening datasets and an additional MHIF set of data demonstrated that the proposed approach can outperform (both qualitatively and quantitatively) recent state-of-the-art image fusion approaches. Furthermore, we provided to the readers some discussions on the proposed method, even showing ablation studies to verify the effectiveness of the proposed technique.

[6]Note that GAN-based models can also be trained in a supervised manner, but, unfortunately, we did not find these kinds of works in the related literature.

## References

[1] A. Arienzo, G. Vivone, A. Garzelli, L. Alparone, J. Chanussot, Full-resolution quality assessment of pansharpening: Theoretical and hands-on approaches, IEEE Geoscience and Remote Sensing Magazine 10 (2022) 168–201.

[2] G. Vivone, Multispectral and hyperspectral image fusion in remote sensing: A survey, Information Fusion 89 (2023) 405–417.

[3] G. Vivone, M. Dalla Mura, A. Garzelli, R. Restaino, G. Scarpa, M. O. Ulfarsson, L. Alparone, J. Chanussot, A new benchmark based on recent advances in multispectral pansharpening: Revisiting pansharpening with classical and emerging pansharpening methods, IEEE Geoscience and Remote Sensing Magazine 9 (2020) 53–81.

[4] P. S. Chavez, A. Y. Kwarteng, Extracting spectral contrast in landsat thematic mapper image data using selective principal component analysis, Photogrammetric Engineering and Remote Sensing 55 (1989) 339–348.

[5] C. A. Laben, B. V. Brower, Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening, 2000. US Patent 6,011,875.

[6] B. Aiazzi, S. Baronti, M. Selva, Improving component substitution pansharpening through multivariate regression of ms + pan data, IEEE Transactions on Geoscience and Remote Sensing 45 (2007) 3230–3239.

[7] J. Qu, Y. Li, W. Dong, Hyperspectral pansharpening with guided filter, IEEE Geoscience and Remote Sensing Letters 14 (2017) 2152–2156.

[8] X. Otazu, M. González-Audícana, O. Fors, J. Núñez, Introduction of sensor spectral response into image fusion methods. application to wavelet-based methods, IEEE Transactions on Geoscience and Remote Sensing 43 (2005) 2376–2385.

[9] J. Liu, Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details, International Journal of Remote Sensing 21 (2000) 3461–3472.

[10] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, M. Selva, MTF-tailored multiscale fusion of high-resolution ms and pan imagery, Photogrammetric Engineering and Remote Sensing 72 (2006) 591–596.

[11] G. Vivone, R. Restaino, M. Dalla Mura, G. Licciardi, J. Chanussot, Contrast and error-based fusion schemes for multispectral image pansharpening, IEEE Geoscience and Remote Sensing Letters 11 (2013) 930–934.

[12] X. He, L. Condat, J. M. Bioucas-Dias, J. Chanussot, J. Xia, A new pansharpening method based on spatial and spectral sparsity priors, IEEE Transactions on Image Processing 23 (2014) 4160–4174.

[13] M. Moeller, T. Wittman, A. L. Bertozzi, A variational approach to hyperspectral image fusion, in: Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV, volume 7334, SPIE, 2009, pp. 502–511.

[14] T. Wang, F. Fang, F. Li, G. Zhang, High-quality bayesian pansharpening, IEEE Transactions on Image Processing 28 (2018) 227–239.

[15] J. Duran, A. Buades, B. Coll, C. Sbert, A nonlocal variational model for pansharpening image fusion, SIAM Journal on Imaging Sciences 7 (2014) 761–796.

[16] L.-J. Deng, G. Vivone, M. E. Paoletti, G. Scarpa, J. He, Y. Zhang, J. Chanussot, A. Plaza, Machine learning in pansharpening: A benchmark, from shallow to deep networks, IEEE Geoscience and Remote Sensing Magazine 10 (2022) 279–315.

[17] G. Masi, D. Cozzolino, L. Verdoliva, G. Scarpa, Pansharpening by convolutional neural networks, Remote Sensing 8 (2016) 594.

[18] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding, J. W. Paisley, PanNet: A deep network architecture for pan-sharpening, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2017, pp. 1753–1761.

[19] L.-J. Deng, G. Vivone, C. Jin, J. Chanussot, Detail injection-based deep convolutional neural networks for pansharpening, IEEE Transactions on Geoscience and Remote Sensing 59 (2020) 6995–7010.

[20] T.-J. Zhang, L.-J. Deng, T.-Z. Huang, J. Chanussot, G. Vivone, A triple-double convolutional neural network for panchromatic sharpening, IEEE Transactions on Neural Networks and Learning Systems (2022).

[21] Z.-C. Wu, T.-Z. Huang, D. Liang-Jian, J. Hu, G. Vivone, VO+Net: An adaptive approach using variational optimization and deep learning for panchromatic sharpening, IEEE Transactions on Geoscience and Remote Sensing (2021) 1–16.

[22] C. Wu, B. Du, X. Cui, L. Zhang, A post-classification change detection method based on iterative slow feature analysis and bayesian soft fusion, Remote Sensing of Environment 199 (2017) 241–255.

[23] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, in: Advances in Neural Information Processing Systems (NeurIPS), volume 33, 2020, pp. 6840–6851.

[24] C. A. Bishop, J. G. Liu, P. J. Mason, Hyperspectral remote sensing for mineral exploration in pulang, yunnan province, china, International Journal of Remote Sensing 32 (2011) 2409–2426.

[25] T. A. Carrino, A. P. Crósta, C. L. B. Toledo, A. M. Silva, Hyperspectral remote sensing applied to mineral exploration in southern peru: A multiple data integration approach in the chapi chiara gold prospect, International Journal of Applied Earth Observation and Geoinformation 64 (2018) 287–300.

[26] M. D. Hossain, D. Chen, Segmentation for object-based image analysis (obia): A review of algorithms and challenges from remote sensing perspective, ISPRS Journal of Photogrammetry and Remote Sensing 150 (2019) 115–134.

[27] R. Dian, S. Li, L. Fang, Learning a low tensor-train rank representation for hyperspectral image super-resolution, IEEE Transactions on Neural Networks and Learning Systems 30 (2019) 2672–2683.

[28] T. Xu, T.-Z. Huang, L.-J. Deng, X.-L. Zhao, J. Huang, Hyperspectral image superresolution using unidirectional total variation with tucker decomposition, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 13 (2020) 4381–4398.

[29] T. Xu, T.-Z. Huang, L.-J. Deng, N. Yokoya, An iterative regularization method based on tensor subspace representation for hyperspectral image super-resolution, IEEE Transactions on Geoscience and Remote Sensing 60 (2022) 1–16.

[30] S. Li, R. Dian, L. Fang, J. M. Bioucas-Dias, Fusing hyperspectral and multispectral images via coupled sparse tensor factorization, IEEE Transactions on Image Processing 27 (2018) 4118–4130.

[31] R. Dian, S. Li, X. Kang, Regularizing hyperspectral and multispectral image fusion by cnn denoiser, IEEE Transactions on Neural Networks and Learning Systems 32 (2021) 1124–1135.

[32] J.-F. Hu, T.-Z. Huang, L.-J. Deng, H.-X. Dou, D. Hong, G. Vivone, Fusformer: A transformer-based fusion network for hyperspectral image super-resolution, IEEE Geoscience and Remote Sensing Letters 19 (2022) 1–5.

[33] T. Huang, W. Dong, J. Wu, L. Li, X. Li, G. Shi, Deep hyperspectral image fusion network with iterative spatio-spectral regularization, IEEE Transactions on Computational Imaging 8 (2022) 201–214.

[34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[35] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, 2015, pp. 234–241.

[36] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 10012–10022.

[37] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 568–578.

[38] L. He, Y. Rao, J. Li, J. Chanussot, A. Plaza, J. Zhu, B. Li, Pansharpening via detail injection based convolutional neural networks, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 12 (2019) 1188–1204.

[39] Q. Yuan, Y. Wei, X. Meng, H. Shen, L. Zhang, A multiscale and multidepth convolutional neural network for remote sensing imagery pansharpening, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 11 (2018) 978–989.

[40] X. Wu, T.-Z. Huang, L.-J. Deng, T.-J. Zhang, Dynamic cross feature fusion for remote sensing pansharpening, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 14687–14696.

[41] X. Liu, Q. Liu, Y. Wang, Remote sensing image fusion based on two-stream fusion network, Information Fusion 55 (2020) 1–15.

[42] X. Zhang, W. Huang, Q. Wang, X. Li, SSR-NET: Spatial–spectral reconstruction network for hyperspectral and multispectral image fusion, IEEE Transactions on Geoscience and Remote Sensing 59 (2020) 5953–5965.

[43] X. Wang, Q. Hu, J. Jiang, J. Ma, A group-based embedding learning and integration network for hyperspectral image super-resolution, IEEE Transactions on Geoscience and Remote Sensing 60 (2022) 1–16.

[44] S. Peng, L.-J. Deng, J.-F. Hu, Y.-W. Zhuo, Source-adaptive discriminative kernels based network for remote sensing pansharpening, in: International Joint Conference on Artificial Intelligence (IJCAI), 2022.

[45] Z.-R. Jin, T.-J. Zhang, T.-X. Jiang, G. Vivone, L.-J. Deng, LAGConv: Local-context adaptive convolution kernels with global harmonic bias for pansharpening, in: AAAI Conference on Artificial Intelligence (AAAI), volume 36, 2022, pp. 1113–1121.

[46] X. Wang, Y. Cheng, X. Mei, J. Jiang, J. Ma, Group shuffle and spectral-spatial fusion for hyperspectral image super-resolution, IEEE Transactions on Computational Imaging 8 (2022) 1223–1236.

[47] X. Tian, K. Li, W. Zhang, Z. Wang, J. Ma, Interpretable model-driven deep network for hyperspectral, multispectral, and panchromatic image fusion, IEEE Transactions on Neural Networks and Learning Systems (2023) 1–14.

[48] X. Tian, K. Li, Z. Wang, J. Ma, VP-Net: An interpretable deep network for variational pansharpening, IEEE Transactions on Geoscience and Remote Sensing 60 (2022) 1–16.

[49] D. Liu, J. Li, Q. Yuan, L. Zheng, J. He, S. Zhao, Y. Xiao, An efficient unfolding network with disentangled spatial-spectral representation for hyperspectral image super-resolution, Information Fusion 94 (2023) 92–111.

[50] J. Ma, W. Yu, C. Chen, P. Liang, X. Guo, J. Jiang, Pan-gan: An unsupervised pan-sharpening method for remote sensing image fusion, Information Fusion 62 (2020) 110–120.

[51] W. Diao, F. Zhang, J. Sun, Y. Xing, K. Zhang, L. Bruzzone, Zergan: Zero-reference gan for fusion of multispectral and panchromatic images, IEEE Transactions on Neural Networks and Learning Systems (2022) 1–15.

[52] Q. Xu, Y. Li, J. Nie, Q. Liu, M. Guo, Upangan: Unsupervised pansharpening based on the spectral and spatial loss constrained generative adversarial network, Information Fusion 91 (2023) 31–46.

[53] S. Shi, L. Zhang, Y. Altmann, J. Chen, Unsupervised hyperspectral and multispectral images fusion based on the cycle consistency, arXiv preprint arXiv:2307.03413 (2023).
[54] J. Ni, Z. Shao, Z. Zhang, M. Hou, J. Zhou, L. Fang, Y. Zhang, Ldp-net: An unsupervised pansharpening network based on learnable degradation processes, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 15 (2022) 5468–5479.
[55] A. Q. Nichol, P. Dhariwal, Improved denoising diffusion probabilistic models, in: International Conference on Machine Learning (ICML), PMLR, 2021, pp. 8162–8171.
[56] T. Karras, M. Aittala, T. Aila, S. Laine, Elucidating the design space of diffusion-based generative models, in: Advances in Neural Information Processing Systems (NeurIPS), volume 35, 2022, pp. 26565–26577.
[57] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al., Photorealistic text-to-image diffusion models with deep language understanding, in: Advances in Neural Information Processing Systems (NeurIPS), volume 35, 2022, pp. 36479–36494.
[58] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10684–10695.
[59] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, M. Norouzi, Palette: Image-to-image diffusion models, in: ACM SIGGRAPH 2022 Conference Proceedings (ACM SIGGRAPH), 2022, pp. 1–10.
[60] X. Li, J. Thickstun, I. Gulrajani, P. S. Liang, T. B. Hashimoto, Diffusion-lm improves controllable text generation, in: Advances in Neural Information Processing Systems (NeurIPS), volume 35, 2022, pp. 4328–4343.
[61] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, B. Poole, Score-based generative modeling through stochastic differential equations, in: International Conference on Learning Representations (ICLR), 2021.
[62] Y. Song, S. Ermon, Generative modeling by estimating gradients of the data distribution, in: Advances in Neural Information Processing Systems (NeurIPS), volume 32, 2019.
[63] J. Ho, T. Salimans, Classifier-free diffusion guidance, in: Advances in Neural Information Processing Systems (NeurIPS), Workshop on Deep Generative Models and Downstream Applications, 2022.
[64] J. Song, C. Meng, S. Ermon, Denoising diffusion implicit models, in: International Conference on Learning Representations (ICLR), 2021.
[65] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, J. Zhu, Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps, in: Advances in Neural Information Processing Systems (NeurIPS), volume 35, 2022, pp. 5775–5787.
[66] W. G. C. Bandara, N. G. Nair, V. M. Patel, DDPM-CD: Remote sensing change detection using denoising diffusion probabilistic models, arXiv preprint arXiv:2206.11892 (2022).
[67] J. Yue, L. Fang, S. Xia, Y. Deng, J. Ma, Dif-Fusion: Towards high color fidelity in infrared and visible image fusion with diffusion models, arXiv preprint arXiv:2301.08072 (2023).
[68] Q. Meng, W. Shi, S. Li, L. Zhang, Pandiff: A novel pansharpening method based on denoising diffusion probabilistic model, IEEE Transactions on Geoscience and Remote Sensing 61 (2023) 1–17. doi:10.1109/TGRS.2023.3279864.
[69] J. Ma, L. Tang, F. Fan, J. Huang, X. Mei, Y. Ma, Swinfusion: Cross-domain long-range learning for general image fusion via swin transformer, IEEE/CAA Journal of Automatica Sinica 9 (2022) 1200–1217.
[70] L. Tang, Y. Deng, Y. Ma, J. Huang, J. Ma, Superfusion: A versatile image registration and fusion network with semantic awareness, IEEE/CAA Journal of Automatica Sinica 9 (2022) 2121–2137.
[71] Z.-X. Chen, C. Jin, T.-J. Zhang, X. Wu, L.-J. Deng, Spanconv: A new convolution via spanning kernel space for lightweight pansharpening, in: International Joint Conference on Artificial Intelligence (IJCAI), 2022.
[72] Y. Liang, P. Zhang, Y. Mei, T. Wang, Pmacnet: Parallel multiscale attention constraint network for pan-sharpening, IEEE Geoscience and Remote Sensing Letters 19 (2022) 1–5.
[73] K. Preechakul, N. Chatthee, S. Wizadwongsa, S. Suwajanakorn, Diffusion autoencoders: Toward a meaningful and decodable representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10619–10629.
[74] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, M. Norouzi, Image super-resolution via iterative refinement, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2022) 4713–4726.
[75] B. Kawar, M. Elad, S. Ermon, J. Song, Denoising diffusion restoration models, in: Advances in Neural Information Processing Systems (NeurIPS), volume 35, 2022, pp. 23593–23606.
[76] Y. Wang, J. Yu, J. Zhang, Zero-shot image restoration using denoising diffusion null-space model, in: International Conference on Learning Representations (ICLR), 2023.
[77] W. Wang, J. Bao, W. Zhou, D. Chen, D. Chen, L. Yuan, H. Li, Sindiffusion: Learning a diffusion model from a single natural image, arXiv preprint arXiv:2211.12445 (2022).
[78] J. Zhang, J. Guo, S. Sun, J.-G. Lou, D. Zhang, Layoutdiffusion: Improving graphic layout generation by discrete diffusion probabilistic models, arXiv preprint arXiv:2303.11589 (2023).
[79] Y. Song, S. Ermon, Generative modeling by estimating gradients of the data distribution, in: Advances in neural information processing systems (NeurIPS), volume 32, 2019.
[80] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems (NeurIPS), volume 27, 2014.
[81] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: International Conference on Machine Learning (ICML), PMLR, 2017, pp. 214–223.
[82] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4401–4410.
[83] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of StyleGAN, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[84] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real NVP, in: International Conference on Learning Representations (ICLR), 2017.
[85] D. P. Kingma, P. Dhariwal, Glow: Generative flow with invertible 1x1 convolutions, in: Advances in Neural Information Processing Systems (NeurIPS), volume 31, 2018.
[86] H. Li, Y. Yang, M. Chang, S. Chen, H. Feng, Z. Xu, Q. Li, Y. Chen, Srdiff: Single image super-resolution with diffusion probabilistic models, Neurocomputing 479 (2022) 47–59.
[87] S. Elfwing, E. Uchibe, K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, Neural Networks 107 (2018) 3–11.
[88] Y. Wu, K. He, Group normalization, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.
[89] Y. Zhang, G. Hong, An ihs and wavelet integrated approach to improve pan-sharpening visual quality of natural colour ikonos and quickbird images, Information fusion 6 (2005) 225–234.
[90] R. B. Gomez, A. Jazaeri, M. Kafatos, Wavelet-based hyperspectral and multispectral image fusion, in: Geo-Spatial Image and Data Exploitation II, volume 4383, SPIE, 2001, pp. 36–42.
[91] H. Phung, Q. Dao, A. Tran, Wavelet diffusion models are fast and scalable image generators, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 10199–10208.
[92] I. Daubechies, Ten lectures on wavelets, SIAM, 1992.
[93] M. Zhou, J. Huang, K. Yan, H. Yu, X. Fu, A. Liu, X. Wei, F. Zhao, Spatial-frequency domain information integration for pan-sharpening, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2022, pp. 274–291.
[94] L. Liu, Y. Ren, Z. Lin, Z. Zhao, Pseudo numerical methods for diffusion models on manifolds, in: International Conference on Learning Representations (ICLR), 2022.
[95] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations (ICLR), 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.
[96] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of

the IEEE international conference on computer vision, 2015, pp. 1026–1034.
[97] X. Rui, X. Cao, Z. Zhu, Z. Yue, D. Meng, Unsupervised pansharpening via low-rank diffusion model, ArXiv abs/2305.10925 (2023).
[98] A. Guo, R. Dian, S. Li, A deep framework for hyperspectral image fusion between different satellites, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2023) 7939–7954. doi:10.1109/TPAMI.2022.3229433.
[99] S. Lolli, L. Alparone, A. Garzelli, G. Vivone, Haze correction for contrast-based multispectral pansharpening, IEEE Geoscience and Remote Sensing Letters 14 (2017) 2255–2259.
[100] G. Vivone, Robust band-dependent spatial-detail approaches for panchromatic sharpening, IEEE Transactions on Geoscience and Remote Sensing 57 (2019) 6421–6433.
[101] G. Vivone, R. Restaino, J. Chanussot, Full scale regression-based injection coefficients for panchromatic sharpening, IEEE Transactions on Image Processing 27 (2018) 3418–3431.
[102] K. Yan, M. Zhou, L. Zhang, C. Xie, Memory-augmented model-driven network for pansharpening, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2022, pp. 306–322.
[103] M. Zhou, J. Huang, Y. Fang, X. Fu, A. Liu, Pan-sharpening with customized transformer and invertible neural network, in: AAAI Conference on Artificial Intelligence (AAAI), 2022.
[104] Q. Wei, N. Dobigeon, J.-Y. Tourneret, Fast fusion of multi-band images based on solving a sylvester equation, IEEE Transactions on Image Processing 24 (2015) 4109–4121.
[105] R. Dian, S. Li, Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization, IEEE Transactions on Image Processing 28 (2019) 5135–5146.
[106] N. Yokoya, T. Yairi, A. Iwasaki, Coupled non-negative matrix factorization (cnmf) for hyperspectral and multispectral data fusion: Application to pasture classification, in: 2011 IEEE International Geoscience and Remote Sensing Symposium, 2011, pp. 1779–1782.
[107] M. Simoes, J. Bioucas-Dias, L. B. Almeida, J. Chanussot, A convex formulation for hyperspectral image superresolution via subspace-based regularization, IEEE Transactions on Geoscience and Remote Sensing 53 (2014) 3373–3388.
[108] B. Aiazzi, S. Baronti, M. Selva, Improving component substitution pansharpening through multivariate regression of ms + pan data, IEEE Transactions on Geoscience and Remote Sensing 45 (2007) 3230–3239.
[109] J.-F. Hu, T.-Z. Huang, L.-J. Deng, T.-X. Jiang, G. Vivone, J. Chanussot, Hyperspectral image super-resolution via deep spatiospectral attention convolutional neural networks, IEEE Transactions on Neural Networks and Learning Systems 33 (2021) 7251–7265.
[110] R. H. Yuhas, A. F. Goetz, J. W. Boardman, Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm, in: JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop, 1992.
[111] L. Wald, Data fusion: definitions and architectures: fusion of images of different spatial resolutions, Presses des MINES, 2002.
[112] A. Garzelli, F. Nencini, Hypercomplex quality assessment of multi/hyperspectral images, IEEE Geoscience and Remote Sensing Letters 6 (2009) 662–665.
[113] J. Zhou, D. L. Civco, J. A. Silander, A wavelet transform method to merge landsat tm and spot panchromatic data, International Journal of Remote Sensing 19 (1998) 743–757.
[114] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (2004) 600–612.
[115] M. Selva, B. Aiazzi, F. Butera, L. Chiarantini, S. Baronti, Hyper-sharpening: A first approach on sim-ga data, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 8 (2015) 3008–3024.
[116] P. Dhariwal, A. Nichol, Diffusion models beat gans on image synthesis, in: Advances in Neural Information Processing Systems (NeurIPS), volume 34, 2021, pp. 8780–8794.
[117] L. Zhang, M. Agrawala, Adding conditional control to text-to-image diffusion models, arXiv preprint arXiv:2302.05543 (2023).
[118] A. C. Li, M. Prabhudesai, S. Duggal, E. Brown, D. Pathak, Your diffusion model is secretly a zero-shot classifier, arXiv preprint arXiv:2303.16203 (2023).
[119] W. Wu, Y. Zhao, M. Z. Shou, H. Zhou, C. Shen, DiffuMask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models, arXiv preprint arXiv:2303.11681 (2023).
[120] H. Zhang, H. Wang, X. Tian, J. Ma, P2sharpen: A progressive pansharpening network with deep spectral transformation, Information Fusion 91 (2023) 103–122.