Manuscript submitted to AIMS Journal

doi:10.3934/xx.xxxxxxx

NEURAL SHRÖDINGER BRIDGE MATCHING FOR PANSHARPENING

Zihan Cao[✉1], Xiao Wu[✉1], Liang-Jian Deng[✉*1]

[1]School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, 611731, China

Abstract. Recent diffusion probabilistic models (DPM) in the field of pansharpening have been gradually gaining attention and have achieved state-of-the-art (SOTA) performance. In this paper, we identify shortcomings in directly applying DPMs to the task of pansharpening as an inverse problem: 1) initiating sampling directly from Gaussian noise neglects the low-resolution multispectral image (LRMS) as a prior; 2) low sampling efficiency often necessitates a higher number of sampling steps. We first reformulate pansharpening into the stochastic differential equation (SDE) form of an inverse problem. Building upon this, we propose a Schrödinger bridge matching method that addresses both issues. We design an efficient deep neural network architecture tailored for the proposed SB matching. In comparison to the well-established DL-regressive-based framework and the recent DPM framework, our method demonstrates SOTA performance with fewer sampling steps. Moreover, we discuss the relationship between SB matching and other methods based on SDEs and ordinary differential equations (ODEs), as well as its connection with optimal transport. Code will be available.

1. **Introduction.** Pansharpening is a special case of super-resolution and falls into the category of classical inverse image restoration problems in remote sensing. Due to the inherent limitations of the hardware system, existing imaging devices such as Gaofen-2 (GF2), WorldView-4 (WV4), and WorldView-3 (WV3) can only measure low-resolution multispectral (LRMS) $\in \mathbb{R}^{h\times w\times C}$ images while capturing finer spatial information into grayscale panchromatic (PAN) $\in \mathbb{R}^{H\times W\times c}$ images (where $c < C$ and $h < H, w < W$). In general, such a trade-off imposes restrictions on many downstream applications that rely on high-resolution multispectral (HRMS) $\in \mathbb{R}^{H\times W\times C}$ images.

Various approaches have been proposed in the past to address the pansharpening problem. These include model-based methods such as component substitution (CS) [2, 44, 70], multi-resolution analysis (MRA) [1, 51, 66, 94], and variational optimization (VO) methods [28, 61, 98, 99], as well as deep regression models [15, 18, 38, 59, 88, 101, 108] and the recent diffusion models [6, 60, 100]. Previous model-based methods typically involve transforming pansharpening into an optimization problem using physical models, often yielding suboptimal results. In contrast, deep regression models leverage deep learning by feeding LRMS and PAN inputs into a carefully designed deep neural network to fuse and generate HRMS.

Key words and phrases. Pansharpening, Schrödinger bridge matching, Diffusion Model, Score-based Model, SDE, ODE..

*Corresponding author: Liang-Jian Deng.

Much of the prior work has focused on contributing to the design of more efficient deep neural networks. There are also some works that integrate physical models into the design of their models. They incorporate physical formulas into the forward process of the network or design the solution of an optimization problem as a module within the network. While these methods often achieve good performance, they may introduce additional computational overhead.

The recent diffusion-based model (DPM) has garnered significant attention in various fields such as image and video generation [30,54,74,76], molecular structure generation [103], and reinforcement learning [114]. Many works have applied DPM to pansharpening, achieving state-of-the-art (SOTA) results. The approach of DPM relies on stochastic differential equations (SDEs) or ordinary differential equations (ODEs). Its generation process starts from an easily samplable distribution (*e.g.*, Gaussian distribution) and progressively denoises the Gaussian distribution into the target distribution. Due to the nature of being an SDE/ODE model, DPM requires multiple steps of network forward evaluation (NFE) during sampling, resulting in a lengthy sampling process. This limitation hinders the current development of DPM. Despite recent efforts to develop efficient SDE or ODE DPM samplers, the quality of sampled outputs remains unsatisfactory when NFE is low. Furthermore, some works utilize distillation to transfer knowledge from a large NFE network to a smaller NFE one, but this still requires additional data and computational resources for distillation. For solving inverse problems, DPM also finds many applications. Some methods based on singular value decomposition (SVD) [42] and optimization [11] modify the sampling process of DPM, integrating the solution process of the inverse problem into it, yielding promising results. However, these methods require DPM trained on corresponding data; otherwise, due to the significant domain gap (*e.g.,* using Imagenet trained DPM model to restore the remote sensing images), the quality of sampling may degrade.

Recently, DDIF [6] and PanDiff [60] inheriting the concept of DPM, utilizes LRMS and PAN as conditional inputs into a diffusion network based on the VP SDE [31], performing denoising to transform the Gaussian samples into HRMS. Although they have achieved satisfactory pansharpening performance, they all overlook a crucial fact: *pansharpening, as a type of inverse problem, benefits from the known degradation of LRMS to HRMS, which can serve as a prior, they still initiate the sampling process from a Gaussian distribution*, which is unreasonable and counterintuitive.

The Schrödinger Bridge (SB) problem was first introduced in quantum mechanics [77] and expanded to broader fields such as OT problem. It seeks two optimal policies that transform back-and-forth between two arbitrary distributions in a finite horizon. In recent times, there has been a surge in efforts to develop more efficient neural SB solvers for solving SB problems. Examples include solvers based on IPF [14,80] and likelihood-based approaches [9,62,97]. However, these methods involve *trajectory simulation (simulating the entire SDE trajectory), training multiple forward-backward networks simultaneously, complex learning objectives, and intricate training procedures*. Consequently, applying these methods to large-scale image tasks is not practical.

To address the above issues of 1) long sampling time, 2) starting sampling from a Gaussian distribution in pansharpening of current DPM-based methods, and 3) the impracticality of neural SB solvers for large-scale image problems, we first characterize the pansharpening task based on the characteristics of traditional CS and

MRA methods and deduce the form of the SDE in Sect. 3.1. We then extend this to degradation SDE and its ODE form in Sect. 3.2. Building on this, we simplify the degradation SDE/ODE to linear SDE forms using the SB formulation in Sect. 3.3, thereby reducing the sampling difficulty. Furthermore, we provide the parameterization of the proposed SDE and ODE in Sect. 3.5. The main conception of this paper is illustrated in Fig. 2. Extensive experiments are conducted to verify the proposed SB SDE and ODE in Sect. 5. Moreover, further discussion with other works is provided in Sect. 6.

To summarize, the contributions of this paper are:

1. We summarize the shortcomings of existing DPM-based pansharpening frameworks. Based on the inverse problem formulation of pansharpening, we express it as a unified degradation-recovery SDE/ODE.
2. Furthermore, by leveraging the SB formula, we improve the degradation-recovery SDE/ODE to linear forward-backward SDE/ODE. This formulation features an analytic forward process, eliminating the need for simulation and exhibiting excellent properties of optimal transport (OT). This makes both training and sampling highly convenient.
3. We design a more efficient deep neural network architecture for the proposed SB matching, which enhances the effectiveness of SB SDE.
4. The proposed SB matching method achieves new SOTA performances on three commonly used pansharpening datasets.

2. Priliminary.

In this section, we will provide a brief review of the Diffusion Probabilistic Model (DPM) [31, 76, 85, 87], Optimal Transport [43, 47, 91] that can be linked with Schrödinger Bridge (SB) [9, 14, 62, 80, 97].

2.1. Diffusion Probabilistic Model.

Diffusion Probabilistic Model (DPM) uses a little isotropy Gaussian noise to degrade the input image step by step until pure Gaussian noise in the forward process. In the backward process, the trained diffusion model tries to remove the noise from the noisy latent images. DPM forward/backward processes are operated by ODEs or SDEs [31, 40, 85]. From the viewpoint of score matching [85, 87] (e.g., a kind of diffusion model), the forward SDE can be formed as,

$$dX_t = f_t(X_t)dt + g_t dW_t,\ X_0 \sim p_X,\ X_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \tag{1}$$

where f_t is the drift term and the g_t is the diffusion term, and W_t is the Winner process. We can derive the backward process by using the Fokker Plank (FP) equation [23],

$$\frac{\partial}{\partial t}p_t(X_t) = -\nabla \cdot (f_t(X_t)p_t(X_t)) + \frac{1}{2}g_t^2 \Delta p_t(X_t), \tag{2}$$

where $\nabla \cdot (\cdot)$ is the divergence operator and $\Delta(\cdot)$ is the Laplacian operator. Thus, the backward process can be accomplished,

$$dX_t = [f_t(X_t) - g_t^2 \nabla_{X_t} \log p_t(X_t, t)]dt + g_t d\bar{W}_t. \tag{3}$$

$\bar{w}_t$ is the reversed Winner process and $\nabla_{X_t} \log p_t$ is known as the score function.

In practice, it is feasible to analytically sample X_t given t and X_0 and to train a neural network s_θ (often U-Net [73]) to predict the score function $\mathbb{E}_{X_t,t}\|s_\theta(X_t, t; \theta) -$

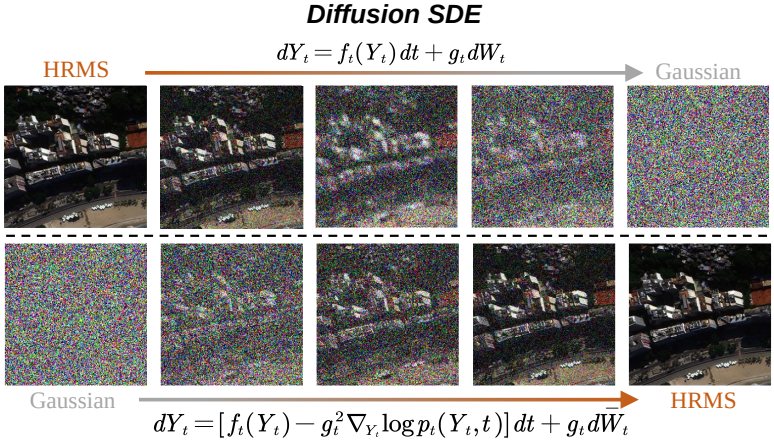


FIGURE 1. Illustration of diffusion framework. It connects the Gaussian distribution with the HRMS distribution, which is inefficient in handling the pansharpening task. Previous works [6, 60] adopt this scheme.

$\nabla_{X_t} \log p(X_t, t|X_0)\|_2^2$ as same as minimizing the variational lower bound (ELBO),

$$\begin{aligned} L_{vlb} = &- \log p_\theta(X_0|X_1) + D_{KL}(p_T(X_T|X_0)||p_T(X_T)) \\ &+ \sum_{t>1} D_{KL}(p_{t-1}(x_{t-1}|x_t, x_0)||p_\theta(X_{t-1}|X_t)). \end{aligned} \tag{4}$$

Meanwhile, FPE provides an ODE,

$$dX_t = \left[f_t(X_t) - \frac{1}{2} g_t^2 \nabla_{X_t} \log p_t(X_t, t) \right] dt, \tag{5}$$

that shares the same marginal distribution which can be effectively solved by some well-studied ODE solvers. Additive parameterization of $f_t(X_t)$ and g_t can be found in the supplementary.

2.2. **Optimal Transport.** The Optimal Transport problem [91] tries to find the minimal displacement cost from one distribution p_0 to another distribution p_1 when given a ground cost. Takes a usual two-Wasserstein distance as an example,

$$\mathcal{W}_2(p_0, p_1)^2 = \inf_{\pi \in \Pi} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y)^2 \pi(dx, dy), \tag{6}$$

where Π is the set of all joint probability measures on space $\mathcal{X} \times \mathcal{Y}$ marginalized on p_X and p_Y and $c(\cdot, \cdot)$ denotes the non-negative ground cost. π is the OT plan where discrete and continuous OT methods are tried to search. Based on it, we can define the kinetic form given by

$$\mathcal{W}_2(q_0, q_1)^2 = \inf_{p_t, f_t} \int_{\mathbb{R}^d} \int_0^1 p_t(X_t) \|f_t\|^2 dt dX, \tag{7}$$

where p_t is the marginal distribution that obeys the transport equation $\partial_t p_t + \nabla \cdot (p_t f_t) = 0$ and has the boundary p_X, p_Y. The proposed Schrödinger Bridge Matching in Sect. 3.3 can be linked with Optimal Transport and is discussed in Sect. 6.2.

2.3. **Schrödinger Bridge.** SB problem is often defined as

$$\min_{\mathbb{Q}\in\mathcal{F}(p_X,p_Y)} D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}), \tag{8}$$

where $\mathcal{F}(p_X, p_Y) \subset \mathscr{P}(\Omega)$ is the path measure with the probability density p_X and p_Y at the bridge endpoints and usually p_X is defined as data distribution and p_Y is prior distribution. $\mathbb{Q}$ and $\mathbb{P}$ are path measures. The goal of an SB problem is to find a $\mathbb{P}$ that has the same marginal distribution p_X and p_Y at times $t = 0$ and $t = 1$. It is also linked with the stochastic optimal control [13] which is formulated as follows,

$$\begin{aligned} &\min_{u(X_t,t)} \mathbb{E}_{p_t}\left[\int_0^1 \frac{1}{2}\|u(X_t,t)\|dt\right], \\ &s.t.,\ dX_t = [f_t(X_t) + u_t(X_t,t)]\,dt + g_t dW_t. \end{aligned} \tag{9}$$

This control problem can be converted into a PDE by applying the Hopf-Cole transform [33],

$$\begin{cases} \frac{\partial\Psi(X_t,t)}{\partial t} = -\nabla\Psi^\top f_t - \frac{1}{2}\mathrm{Tr}(g_t^2\nabla^2_{X_t}\Psi) \\ \frac{\partial\widehat{\Psi}(X_t,t)}{\partial t} = -\nabla\cdot(\widehat{\Psi}^\top f_t) + \frac{1}{2}\mathrm{Tr}(g_t^2\nabla^2_{X_t}\widehat{\Psi}) \end{cases} \tag{10a}$$

$$s.t.,\ \Psi(X_t,0)\widehat{\Psi}(X_t,0) = p_{\mathcal{A}}(X), \tag{10b}$$

$$\Psi(X_t,1)\widehat{\Psi}(X_t,1) = p_{\mathcal{B}}(X), \tag{10c}$$

where $\Psi, \widehat{\Psi} \in (\mathbb{R}^d, [0,1])$ are the energy potentials and $\mathrm{Tr}(\cdot)$ is the matrix trace. From the above PDEs, we can use FPE and non-linear Feynman-Kac formula [39] to get an SB-inspired SDE:

$$\begin{cases} dX_t = (f_t + g_t Z_t)dt + g_t dW_t \\ dY_t = \frac{1}{2}Z_t^\top Z_t dt + Z_t^\top dW_t \\ d\hat{Y}_t = \left(\frac{1}{2}\hat{Z}_t^\top\hat{Z}_t + \nabla_{X_t}\cdot(g_t\hat{Z}_t - f_t) + \hat{Z}_t^\top Z_t\right)dt + \hat{Z}_t dW_t \end{cases}$$

where $Z_t = g_t\nabla_{X_t}Y_t(X_t) = g_t\nabla_{X_t}\log\Psi(X_t,t)$ and $\hat{Z}_t = g_t\nabla_{X_t}\hat{Y}_t(X_t) = g_t\nabla_{X_t}\log\hat{\Psi}(X_t,t)$ are the SB optimal forward and backward drifts of the SDE. Akin to the DPM SDE, we can still perform simulated SDE trajectories [9] by defining any f_t and g_t which can ensure the convergence to the boundaries. However, simulating trajectories needs to run forward and backward in time and takes huge computational resources, meanwhile, it also requires two networks (*i.e.,* forward and backward networks) simultaneously learning the forwarding and backwarding. Proofs of the derivations from SB problems to PDEs and SB-inspired SDEs can be found in the Supplementary. Hence, developing a *simulation-free* SB method is both practical and advantageous.

3. Inverse Problem of Pansharpening and Learning Schrödinger Bridge Matching.

3.1. **Inverse Problems of Pansharpening.** First, we introduce the general formula of the inverse problem,

$$Y = A(X) + Z, Z \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d), \tag{11}$$

where $A(\cdot)$ is a degraded operator that can degrade the clean ground truth, and Y is the measurement added by Gaussian noise with variance σ^2. For various inverse

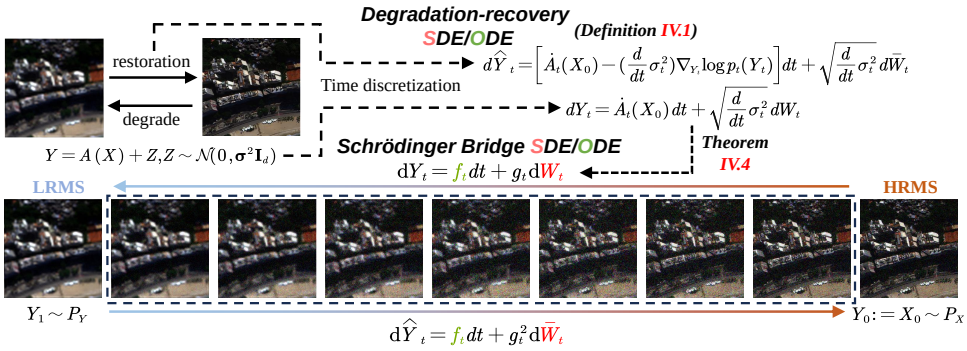


FIGURE 2. Main concept of the proposed SB SDE and ODE. HRMS is degraded by the forward SB SDE/ODE process (denoted as q process) and recovered by the backward process (denoted as p process). PAN, as the condition in the SB SDE or ODE, is omitted for a clear illustration.

problems, such as superresolution [48, 74–76], colorization [71, 74], MRI reconstruction [11, 86] et al, there are different degraded operators. Eq. (11) also suits the pansharpening task. Recall the two widely-used traditional pansharpening methods (i.e., CS [2, 44, 70], and MRA methods [1, 51, 66, 94]):

$$\text{CS}: \quad \mathbf{M} = \mathbf{H} - \mathbf{g} \odot (\mathbf{P^D} - \mathbf{I_L^D}), \tag{12a}$$

$$\text{MRA}: \quad \mathbf{M} = \mathbf{H} - \mathbf{g} \odot (\mathbf{P^D} - \mathbf{P_L^D}), \tag{12b}$$

where $\mathbf{P^D}, \mathbf{P_L^D}, \mathbf{I_L^D}, \mathbf{M}$, and $\mathbf{H}$ are the channel-duplicated PAN, channel-duplicated low-passed PAN, channel-duplicated intensity of LRMS, LRMS, HRMS images and $\mathbf{g}$ is a weighting vector. From the above equations, a general degraded operator A can be defined by considering $\mathbf{M}$ and $\mathbf{H}$ are measurement Y and clean ground truth X_0,

$$A(X|\mathbf{P}, \mathbf{M}) = X - \mathcal{J}(X, \mathbf{P}, \mathbf{M}), \tag{13}$$

where $\mathcal{J}(\cdot)$ is a function and it can be noticed that $\mathbf{H}$ is the clean ground truth and $\mathbf{M}$ denotes the observation but $Z = 0$ in the context of pansharpening. Without considering the Gaussian term Z, Eq. (13) tells us the pansharpening task be formulated as a typical inverse problem and different As contribute different inverse problems.

3.2. **Degradation SDE and ODE of Pansharpening.** In the previous section, we represent pansharpening as the classical inverse problem formulation (11). Next, we represent this process in the form of SDE in the following definition:

Definition 3.1 (Stochastic degradation equation)**.** The operator A is time-dependent (A_t) and first-order differentiable. The same requirement applies to $\mathcal{J}$. The following SDE defines the degraded process by the operator A_t:

$$dY_t = \dot{A}_t(X_0)dt + \sqrt{\frac{d}{dt}\sigma_t^2}dW_t, \tag{14}$$

where Y_t is discrete in time, $\dot{A}_t$ is the time-dependent derivative of A, σ_t^2 is the variance of the Wiener process dW_t at timestep t.

Thus, we can formalize the process of pansharpening as an SDE. We refer to this process as "degradation-SDE" as shown in the upper panel in Fig. 2, analogous to the forward process of diffusion in Eq. (1). Leveraging the FP equation, we can derive the reverse SDE formula:

$$d\hat{Y}_t = \left[\dot{A}_t(X_0) - (\frac{d}{dt}\sigma_t^2)\nabla_{Y_t}\log p_t(Y_t)\right]dt + \sqrt{\frac{d}{dt}\sigma_t^2}d\bar{W}_t, \tag{15}$$

where $d\bar{W}_t$ is the reverse process of the Wiener process [4]. The hat on a symbol represents the random variable when backwarding. This reverse SDE can be discretized using Euler-Maruyama discretization,

$$\begin{aligned} Y_{t-\Delta t} = Y_t + \underbrace{A_{t-\Delta t}(X_0) - A_t(X_0)}_{\text{recovery step}} \\ - \underbrace{(\sigma_{t-\Delta_t}^2 - \sigma_t^2)\nabla_{Y_t}\log p_t(Y_t)}_{\text{denoising}} + \sqrt{\sigma_t^2 - \sigma_{t-\Delta_t}^2}Z_t. \end{aligned} \tag{16}$$

Similarly, we term it "recovery-SDE". Naturally, we can derive the ODE forms for both the forward and backward processes:

$$dY_t = \dot{A}_t(X_0)dt, \tag{17a}$$

$$d\hat{Y}_t = \dot{A}_t(X_0)dt - \frac{1}{2}(\frac{d}{dt}\sigma_t^2)\nabla_{Y_t}\log p_t(Y_t). \tag{17b}$$

By parameterizing $\dot{A}_t$ with Gaussian transition kernels, we can directly train and sample the pansharpening task using Eqs. (14)-(17b).

However, on the one hand, despite being demonstrated to exhibit good performance in the pansharpening task [17], DPM often suffers from inefficient sampling due to the nature of Gaussian transition kernels. Especially when using specially customized DPM solvers [57], the generated quality tends to be low with a small number of sampling steps. In the following section, we demonstrate that this is attributed to the highly curved nature of DPM's sampling trajectory. Therefore, leveraging the properties of the Schrödinger Bridge (SB), we straighten the trajectory of DPM, reducing the training difficulty and simultaneously enhancing sampling efficiency. On the other hand, we can intuitively sense that as $t \to 1$ (sample large enough times e.g., 1000), Y_1 can be regarded as a pure Gaussian noise. This implies that during sampling, we also need to initiate sampling from a Gaussian noise (see one endpoint of SDE in Fig. 1), which is evidently unreasonable as we already possess prior information $\mathbf{M}$. Hence, initiating sampling from the known prior is a reasonable and efficient approach. Clearly, existing DPM frameworks ([31, 75, 87] and Definition 3.1) do not support it. A clear difference can be observed by comparing Fig. 1 and the lower panel in Fig. 2.

3.3. **Simulation-free Schrödinger Bridge SDE.** In Sect. 3.2, we elucidate that the pansharpening task can be fully expressed in the forms of SDE and ODE. We utilized existing DPM frameworks for training and sampling. However, due to the existing shortcomings of DPM, applying it directly to the pansharpening task is both unreasonable and inefficient. To address these issues, we leverage the properties of SB and introduce SB ODE and SDE.

Since SB is known as an entropy-regulized optimal transport model that can be formulated as the following SDEs:

$$dY_t = [f_t + g_t^2\nabla_{Y_t}\log p_t(Y_t)]dt + g_t dW_t, \tag{18a}$$

$$d\hat{Y}_t = [f_t - g_t^2 \nabla_{Y_t} \log p_t(\hat{Y}_t)]dt + g_t d\bar{W}_t, \tag{18b}$$

where the two end-point distributions on the SB become p_X and p_Y. For Eqs. (14) and (18a), it can be observed that if we replace $\dot{A}_t$ with $f_t + g_t^2 \nabla Y_t \log p_t(Y_t)$, we effortlessly transform the degradation-SDE into the form of SB SDE. Incorporating with Eq. (13), we can further derive the following proposition:

Proposition 3.2. *The pansharpening inverse problem can be represented as SB SDE forms*[1]*,*

$$\begin{aligned} f_t &= 1, \\ \dot{\mathcal{J}} &= g_t^2 \nabla_{Y_t} \log p_t(\hat{Y}_t), \end{aligned} \tag{19a}$$

which is an implicit learning objective by setting $s_\theta := \dot{\mathcal{J}}$. s_θ is a neural network, usually a U-Net [73].

Proof. The proof of the Proposition 3.2 is sufficient. □

Remark 3.3. Up to this point, it can be realized that the introduced SB-SDE can be seamlessly applied to the inverse problem's core, as we can employ a neural network to parameterize the degradation operator determined in the inverse problem (which may not be known). Moreover, relying on the multi-step diagram of the SDE, we can use the time steps as input (or conditions) to decompose a challenging single-step inverse degenerate process into multiple steps. This can alleviate the learning difficulty for the network, aligning with the observation in [6].

However, in Eqs. (18a) and (18b), it is evident that these SDEs are non-linear. Moreover, the nonlinear term $\nabla_{Y_t} \log p_t(Y_t)$ has not been explicitly defined, making the solution of these SDEs challenging. Additionally, due to the presence of the nonlinear term, *the trajectories of the SDE are not straight*, posing a challenge for sampling. Recently, there has been research attention to the use of the Hopf-Cole transform [33], which enables the conversion of nonlinear terms into linear ones. However, these studies are confined to the sampling stage, transforming the PF ODE [85], and the effectiveness of diffusion SDE remains unexplored.

Different from the previous diffusion works, we express SB in the form of linear SDEs.

Theorem 3.4. *The Schrödinger Bridge problem in Eq. (10a) can be expressed as the following forward-backward SDE with a linear f_t, which shares a form similar to that of a score-based model,*

$$dY_t = f_t dt + g_t dW_t, \qquad Y_0 \sim \hat{\Psi}(\cdot, 0), \tag{20a}$$

$$d\hat{Y}_t = f_t dt + g_t d\bar{W}_t, \qquad Y_1 \sim \Psi(\cdot, 1). \tag{20b}$$

The nonlinear terms in forward and backward drifts $\nabla_t \log \Psi, \nabla_t \log \hat{\Psi}$ have been absorbed into the initial conditions $\Psi(Y_t, 0)$ and $\hat{\Psi}(Y_t, 1)$.

Proof. $dY_t = f_t dt + g_t dW_t$ shows that the probability change is an Itô process, which can be reformulated by the FP equation,

$$\frac{\partial p(Y,t)}{\partial t} = -\nabla \cdot (f_t p_t) + \frac{1}{2} g_t \Delta p_t. \tag{21}$$

By comparing (10a) and (21), and using

$$\mathrm{Tr}(\nabla_{Y_t} \Psi) = \Delta \Psi, \tag{22}$$

[1]For simplicity of the symbols, the condition **P** is omitted.

it shows that the $\frac{\partial(Y_t,t)}{\partial t}$ has the same form when taking $\hat{\Psi} = p_t$. When reverse the SB PDE (10a):

$$\begin{cases} \frac{\partial \Psi(Y,s)}{\partial s} = \nabla \Psi^\top f_s + \frac{1}{2}\mathrm{Tr}(g_s^2 \nabla^2_{Y_s} \Psi), \\ \frac{\partial \widehat{\Psi}(x,t)}{\partial s} = \nabla \cdot (\widehat{\Psi}^\top f_s) - \frac{1}{2}\mathrm{Tr}(g_s^2 \nabla^2_{Y_s} \widehat{\Psi}), \end{cases} \tag{23}$$

where $s := 1 - t$. This implies that $\Psi(Y, s)$ can be interpreted as the density of one SDE

$$dY_s = -f_s ds + g_s dW_s, \tag{24}$$

which is equal to Eq. (20b). In the rest of this paper, we set $t := 1 - s$ under the continuous time SB context[2]. □

It is easy to observe that we have directly removed the nonlinear term, absorbing it into the initial condition $\hat{\Psi}(\cdot, 0)$, leaving only completely linear terms and keeping the conclusion in Eqs. (19a) unchanged. Therefore, we can parameterize $\nabla_{Y_t} \log \hat{\Psi}(Y_t)$ as a network s_θ, similar to training and sampling in DPM.

It's worth noting that our approach has a more appealing characteristic, i.e., it is simulation-free. Reviewing previous methods to solve the SB problems, such as the Iterative Proportional Fitting (IPF) [82] methods (*a.k.a.* Sinkhorn algorithm), they require a simulation to regularize the entire trajectory of the SDE. Even recent deep transport methods (*e.g.*, FB-SDE [9], Action Matching [62]) still necessitate simulation, which is impractical for some high-dimensional data (*e.g.*, multispectral or hyperspectral images) to simulate the entire trajectory and cache it. Our method aligns with score-based models in being simulation-free, and this is also why our approach can adapt to high-dimensional data.

3.4. **Schrödinger Bridge Matching for Pansharpening and its ODE version.** Similar to Probability Flow (PF) [45,85] but distinct, we can derive an ODE with optimal transport properties from Eqs. (10a)-(10c). We designate this ODE as "SB ODE".

Corollary 3.5. *When $g_t \to 0$, the proposed SDEs between (X, Y) can be expressed by an ODE with the same posterior mean of SB SDE in Theorem 3.4 and exhibit the OT property:*

$$dY_t = v_t(Y_t|X_0)dt, \tag{25a}$$

$$dY_s = -v_t(Y_s|Y_t)ds, \tag{25b}$$

It is worth noting that this ODE is not a PF ODE, and it simulates the OT plan [26,89] when the corresponding stochastic terms in the SDE (20a) vanish. In this context, v_t determines the speed of transition from X_0 to Y_t. If v_t is simple and tractable, similar to the approach used in score-based models [85,87], we still can perform a effecient forward process and a stimulation-free train. This enables us to conveniently carry out the reverse ODE sampling process (sampling from Y_t all the way back to X_0), aligning with our initial motivations. We further elucidate in the Supplementary the relationship between our SB ODE and Flow Matching [49], as well as its ability to derive a similar form to the Classifier Guidance [32] in DPMs. This paves the way for controlling image generation.

Algorithm 1: Training scheme

Input: Undegraded samples p_X, degraded samples p_Y, other paired conditions p_c (optional) and SB matching network s_θ, learning rate η.

Output: Trained SB SDE/ODE matching parameter θ.

while *until convergence* **do**

▷ Construct time steps t and two bridge endpoints $Y_0 := X_0$ and Y_1.

$t \sim \mathcal{U}(0,1)$, $X_0 \sim p_X, Y_1 \sim p_Y, C \sim p_c$;

▷ Sample samples on SB bridge (SB SDE (26) or ODE (39)).

$Y_t \sim \mathcal{N}(Y_t; \mu_t(Y_0, Y_1), \Sigma_t)$ or $Y_t = \frac{\hat{\sigma}_t^2}{\hat{\sigma}_t^2+\sigma_t^2} X_0 + \frac{\sigma_t^2}{\hat{\sigma}_t^2+\sigma_t^2} Y_1$;

▷ Take gradient step using one of four proposed bridge matching objectives.

$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(s_\theta(Y_t, C, t), Y_0, Y_1)$.

end

Algorithm 2: Sampling scheme

Input: Sampling timesteps T, trained s_θ, degraded samples p_0, degraded samples p_Y, Conditional samples p_c (optional).

Output: Sampled data X_0.

$T_s = \texttt{linspace}(1, 0, T)$; ▷ Init timestep sequence.

$Y_1 \leftarrow p_Y, C \leftarrow p_c$;

for $t \leftarrow T_s$ **do**

$\hat{X}_0 \leftarrow s_\theta(Y_t, C, t)$; ▷ Predicted endpoint.

$Y_t \leftarrow p(Y_t | \hat{X}_0, Y_{t+\Delta t})$; ▷ Posterior sampling (31a).

end

$X_0 \leftarrow Y_0$.

3.5. **Bridge Matching Parameterization, Train/Sampling Algorithms and Training Objectives.** In this section, we perform necessary parameter design for SB SDE and SB ODE, including the design of f_t and g_t in the Eqs. (20a) and (20b), as well as σ_t in Eq. (38). We provide three feasible parameterization schemes, all of which can satisfy the assumptions of the SB system induced by Eqs. (10a)-(10c). Then, the different training objectives are derived and we further provide the training and sampling algorithms 1 and 2. We follow the notation of [31] by denoting the forward process as q and backward process as p.

Since the derived SB SDE/ODE results in a non-Gaussian endpoint, we set the drift term $f_t := 0$ and further derive a forward process with an analytical solution. This approach resolves the computational intractability issue of Eqs. (18a) and (18b), as well as the problem of not converging to regions of high probability density caused by Eqs. (20a) and (20b).

Proposition 3.6 (Analytic posterior when given bridge endpoints)**.** *When setting $f_t := 0$, $g_t := \sqrt{\beta_t}$ (β_t is a function that increases for $t \in [0, 1/2]$ and decrease $t \in (1/2, 1]$), and given paired bridge endpoints (X_0, Y_1), the posterior distribution*

[2]We simultaneously define X_s and Y_t for a better description of SB in the context of the inverse problem. Such a definition is reasonable.

has an analytic form:

$$q(Y_t|X_0, Y_1) = \mathcal{N}(Y_t; \mu_t(X_0, Y_1), \Sigma_t), \tag{26}$$

$$\mu_t = \frac{\hat{\sigma}_t^2}{\hat{\sigma}_t^2 + \sigma_t^2} X_0 + \frac{\sigma_t^2}{\hat{\sigma}_t^2 + \sigma_t^2} Y_1, \Sigma_t = \frac{\sigma_t^2 \hat{\sigma}_t^2}{\hat{\sigma}_t^2 + \sigma_t^2} \mathbf{I}, \tag{27}$$

where $\sigma_t^2 := \int_0^t \beta_\tau d_\tau$ *and* $\hat{\sigma}_t^2 := \int_t^1 \beta_\tau d\tau$ *are variances of Gaussians. This analytic form avoids recursively running the poterior sampling in the above forward SDE:*

$$q(Y_t|X_0, Y_1) = \int \prod_{k=n}^{N-1} p(Y_k|X_0, Y_{k+1}) dY_{k+1}, \tag{28}$$

making the forward sampling much more effecient.

Proof. Since we have $q(\cdot, t) = \Psi(\cdot, t)\hat{\Psi}(\cdot, t)$ contributed by Nelson's duality [63], one can write,

$$q(Y_t|X_0, Y_1) = \Psi(Y_t, t|X_0)\hat{\Psi}(Y_t, t|Y_1). \tag{29}$$

Due to fact that the solution of FP equations are actually $\Psi(Y_t, t|X_0)$ and $\hat{\Psi}(Y_t, t|Y_1)$, the posterior are two Guassian potential product:

$$\Psi(Y_t, t|X_0)\hat{\Psi}(Y_t, t|Y_1) \tag{30a}$$

$$= \exp\left(-\frac{1}{2}\left(\frac{\|Y_t - Y_0\|^2}{\sigma_t^2} + \frac{\|Y_t - Y_1\|^2}{\hat{\sigma}_t^2}\right)\right) \tag{30b}$$

$$= \mathcal{N}(Y_t; \frac{\hat{\sigma}_t^2}{\hat{\sigma}_t^2 + \sigma_t^2} X_0 + \frac{\sigma_t^2}{\hat{\sigma}_t^2 + \sigma_t^2} Y_1, \frac{\sigma_t^2 \hat{\sigma}_t^2}{\hat{\sigma}_t^2 + \sigma_t^2} \mathbf{I}), \tag{30c}$$

where $\sigma_t^2 := \int_0^t \beta_\tau d_\tau$ and $\hat{\sigma}_t^2 := \int_t^1 \beta_\tau d\tau$ are variances of Gaussians. Since the SDE should be discrelized, by slight abuse of notation as $Y_{t_n} = Y_n$, we prove $q(Y_t|X_0, Y_0)$ is the marginal probability density of the posterior $p(Y_n|X_0, Y_{n-1})$. When $f_t := 0$, $p(Y_n|X_0, Y_{n-1})$ has an analytic Gaussian form,

$$p(Y_n|X_0, Y_{n+1}) \tag{31a}$$

$$= \mathcal{N}(Y_n; \frac{\alpha_n^2}{\alpha_n^2 + \sigma_n^2} X_0 + \frac{\sigma_n^2}{\alpha_n^2 + \sigma_n^2} Y_{n+1}, \frac{\alpha_n^2 \sigma_n^2}{\alpha_n^2 + \sigma_n^2} \mathbf{I}), \tag{31b}$$

where $\alpha_n^2 := \int_{t_n}^{t_{n+1}} \beta_\tau d\tau$ is the accumulated Gassian variance. So, we can derive that,

$$q(Y_{N-1}|X_0, Y_N) = p(Y_{N-1}|X_0, Y_N), \tag{32}$$

because one fact exists that $\alpha_{N-1} = \int_{t_{N-1}}^{t_N} \beta_\tau d\tau = \hat{\sigma}_{N-1}^2$. We can deduce at any timestep t_n, the relation still holds by induction:

$$q(Y_n|X_0, Y_N) = \int p(Y_n|X_0, Y_{n+1}) q(Y_{n+1}|X_0, Y_{n+1}) dY_n \tag{33}$$

The RHS is a Gaussian that can be derived by using the Gaussian additive property:

$$\frac{\alpha_n^2}{\alpha_n^2 + \sigma_n^2} X_0 + \frac{\sigma_n^2}{\alpha_n^2 + \sigma_n^2} \left(\frac{\hat{\sigma}_{n+1}^2}{\hat{\sigma}_{n+1}^2 + \sigma_t^2} X_0 + \frac{\sigma_{n+1}^2}{\hat{\sigma}_{n+1}^2 + \sigma_{n+1}^2} Y_N\right) \tag{34a}$$

$$= \frac{\alpha_n^2(\hat{\sigma}_{n+1}^2 + \sigma_{n+1}^2) + \sigma_n^2 \hat{\sigma}_{n+1}^2}{\sigma_{n+1}^2(\hat{\sigma}_n^2 + \sigma_n^2)} X_0 + \frac{\sigma_n^2}{\hat{\sigma}_n^2 + \sigma_n^2} Y_N \tag{34b}$$

$$= \frac{\alpha_n^2 \sigma_{n+1}^2 + \hat{\sigma}_{n+1}^2(\alpha_n^2 + \sigma_n^2)}{\sigma_{n+1}^2(\hat{\sigma}_n^2 + \sigma_n^2)} X_0 + \frac{\sigma_n^2}{\hat{\sigma}_n^2 + \sigma_n^2} Y_N \tag{34c}$$

$$= \frac{\hat{\sigma}_n^2}{\hat{\sigma}_n^2 + \sigma_n^2} X_0 + \frac{\sigma_n^2}{\hat{\sigma}_n^2 + \sigma_n^2} Y_N, \tag{34d}$$

By utilizing $\hat{\sigma}_n^2 + \sigma_n^2$ is a const and

$$\alpha_t = \sigma_{n+1}^2 - \sigma_n^2 = \hat{\sigma}_n^2 - \hat{\sigma}_{n+1}^2. \tag{35}$$

The variance Σ_t of RHS is:

$$\frac{\alpha_n^2 \sigma_n^2}{\alpha_n^2 + \sigma_n^2} + \frac{\hat{\sigma}_{n+1}^2 \sigma_{n+1}^2}{\hat{\sigma}_{n+1}^2 + \sigma_{n+1}^2} \left(\frac{\sigma_n^2}{\alpha_n^2 + \sigma_n^2} \right)^2 \tag{36a}$$

$$= \frac{\alpha_n^2 \sigma_n^2 (\hat{\sigma}_n^2 + \sigma_n^2) + \hat{\sigma}_{n+1}^2 \sigma_n^4}{\sigma_{n+1}^2 (\hat{\sigma}_n^2 + \sigma_n^2)} \tag{36b}$$

$$= \frac{\sigma_n^2 \hat{\sigma}_n^2}{\hat{\sigma}_n^2 + \sigma_n^2}. \tag{36c}$$

Thus, the proof is concluded. □

In practice, we set β_t in the interval $[0, \frac{1}{2}]$ to be the quadratic function and flip it in $(\frac{1}{2}, 1]$,

$$\beta_t = \begin{cases} \left((\sqrt{\beta_{1/2}} - \sqrt{\beta_0})t + \sqrt{\beta_0} \right)^2, & t \in [0, \frac{1}{2}] \\ \left((\sqrt{\beta_{1/2}} - \sqrt{\beta_0})(1-t) + \sqrt{\beta_0} \right)^2, & t \in (\frac{1}{2}, 1] \end{cases} \tag{37}$$

where β_0 and $\beta_{1/2}$ are prefixed. Other SB hyperparameters are shown in Fig. 3.

Proposition 3.6 provides an explicit bridge process of SB SDE. Similarly, we can write the analytic posterior of v_t in Corollary 3.5.

Corollary 3.7 (Analytic posterior of SB ODE)**.** *The posterior of SB ODE in Corollary 3.5 has a closed-form posterior:*

$$v_t(Y_t|X_0) = \frac{\beta_t}{\sigma_t^2}(Y_t - X_0), \tag{38}$$

and the SB ODE can be discretized as,

$$Y_t = \frac{\hat{\sigma}_t^2}{\hat{\sigma}_t^2 + \sigma_t^2} X_0 + \frac{\sigma_t^2}{\hat{\sigma}_t^2 + \sigma_t^2} Y_1. \tag{39}$$

Proof. This proof start to take the infinitestimal limit of $g_t^2 := \beta_t \to 0$ in Eq. (20a), and the variance of q_t (*i.e.*, $\frac{\alpha_n^2 \sigma_n^2}{\alpha_n^2 + \sigma_n^2}$) also converge to 0 as the nominator becomes to 0 faster. The process become deterministic in Eq. (39) since the stochastic term is vanished. The mean u_t is unchanged. From Eq. (27), we can know the nonlinear term $\nabla_{Y_t} \log \hat{\Psi}(Y_t|X_0)$ absorbed in the drift term f_t becomes,

$$\nabla_{Y_t} \log \hat{\Psi}(Y_t|X_0) = \frac{1}{\sigma_t^2}(Y_t - X_0), \tag{40}$$

and we define v_t with the nonlinear term multiplied by β_t, which is Eq. (38). □

Clearly, from Eq. (38), it can be seen that the ODE drift v_t interpolates between Y_1 and X_0 We explicitly perform this interpolation process during the forward pass.

Remark 3.8. "SB ODE" corresponds to the ODE versions of Eqs. (20a) and (20b). Its rationale lies in removing the stochastic terms from the SDE, transforming it into an ODE. Additionally, a favorable OT property [69, 91] can be derived from Eq. (38), making sampling extremely convenient and practical (or, in other words, ensuring that the curvature is flat [46]).

Moreover, when β_t becomes a constant, SB ODE can have the same forward and backward process as Flow Matching [49], which is detailed in discussion.

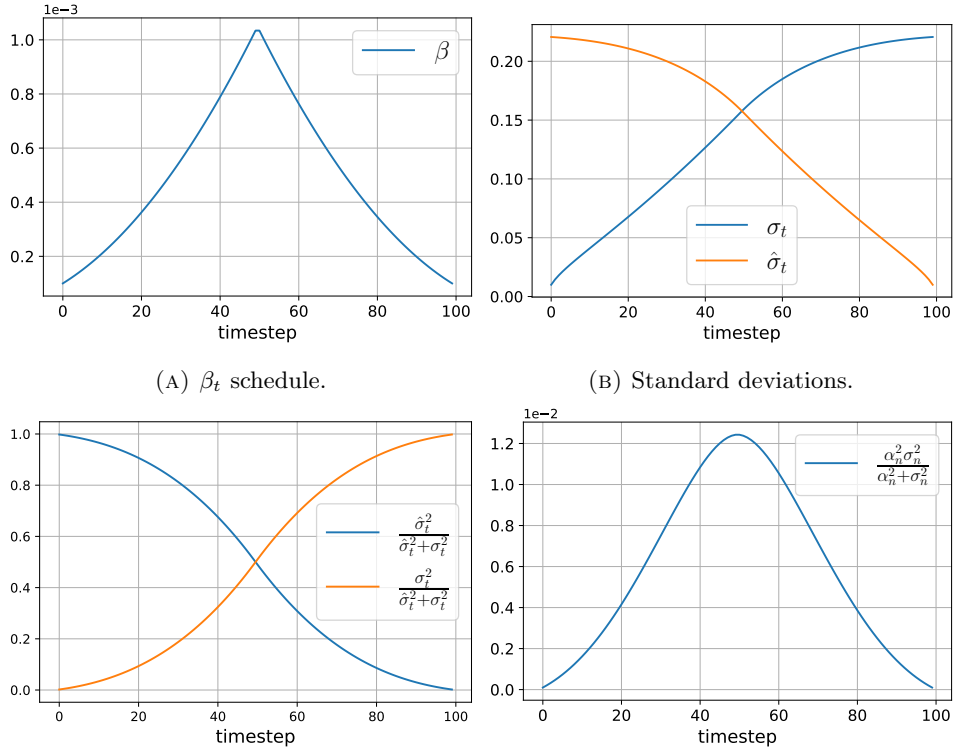
To reverse back the SB forward process q (*i.e.*, p), we need to learn a neural network s_θ trained on a specific objective. We provide four different learning objectives for SB SDE and ODE:

(a) bridge endpoint X_0;
(b) bridge length $Y_1 - X_0$;
(c) bridge posterior length $Y_t - X_0$;
(d) bridge endpoint with score $[X_0, \epsilon]$ (SDE only).

The first three objectives are suitable for both SB SDE and ODE but (d) only suits SB SDE as ODE does not involve a score function. To explain the rationale behind each objective, (a) directly predicts one bridge point which is similar to supervised learning, but the latter does not have a concept of timestep or dynamic; (b) lets the network learn the overall bridge length; while (c) straight-forwardly train the network by using the definition of v_t in Eq. (38); (d) incorporate score function in objective (a) which is related with the added Gaussian noise in q process [83] and we simplify it to ϵ inspired by [31].

4. **Bridge Matching Neural Architecture.** In the above sections, we elucidated the design paradigm, parameterization, training sampling strategy, and objectives of SB SDEs. However, we have not yet discussed how to design the network. In fact, the design of the network is a crucial aspect of SDE learning, and recent literature on score-based models and DPMs has also explored this topic [12, 41, 58, 67]. We design an efficient network architecture specifically for the pansharpening task, featuring high-speed training and inference. In comparison to architectures designed for image generation in DPMs (*e.g.*, U-Net), our designed network exhibits superior performance. Inspired by NAFNet [7] in image restoration and Metaformer [107] in architecture designing, we devise a NAFNet-like network named SBM-Net tailored for SB SDE learning.

4.1. **Overall Architecture.** As shown in Fig. 4, SBM-Net is an encoder-decoder architecture composed of several SBM blocks. The inputs LRMS and PAN are first concatenated along the channel dimension and then projected by a point-wise convolution into the latent space. Further, the projected feature is fed into the encoder layer by layer. Each of the SBM layers consists of several SBM blocks to extract spatial and channel representations. To incorporate the dependence of t, the timestep is embedded in the form of Sine-cosine [22] to feed into the block. After processing every layer, the feature is downsampled, and doubled the channel dimensions. Then, the feature after the encoder is input to the decoder. The decoder has a similar architecture to the encoder. Every decoder's input is the input feature from the last SBM block and the feature processed by the corresponding encoder layer. We simply concatenate them together to feed into the decoder SBM layer. Similarly, the timestep is also fed into the decoder layer. After every decoder layer, the feature dimension is halved and upsampled by the pixel-shuffle operation [79]. At last, another point-wise convolution is employed to project the feature back to the pixel space, forming the HRMS. In practice, we observed that forwarding the network twice can yield additional performance gains. Specifically, for each batch size of the input, we feed it into the network twice in a loop (i.e., the output of the first forward is fed back as input for the next forward). Note that objective (d) needs an additional predictive ϵ, we double the number of output channels.



(A) β_t schedule. (B) Standard deviations.

(C) Mean schedule. (D) Variance schedule.

FIGURE 3. β_t, μ_t, σ_t and variance schedules of the proposed SB matching.

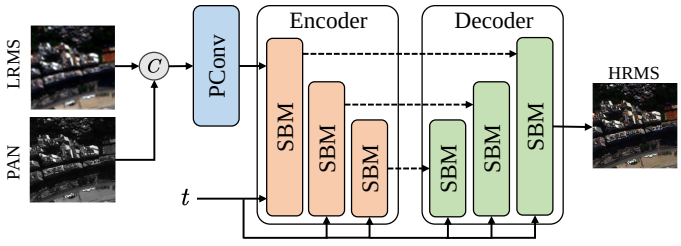


FIGURE 4. Overall architecture of the proposed SBM-Net for SB SDE learning. The SBM-Net is an encoder-decoder architecture. Timestep t is injected into every SBM blocks. Ⓒ denotes tensor concatenation. Dash lines represent the U-Net-like shortcut connection. Valid lines are data flows.

4.2. **SBM Block.** In the field of image generation, attention layers are often incorporated into U-Net architectures to facilitate additional inter-modal communication [72,75] (*e.g.*, between text and mask). In contrast, pansharpening, as a low-level image fusion technique in the domain of image fusion, lacks abstract high-level information such as text. Moreover, the quadratic memory overhead introduced by

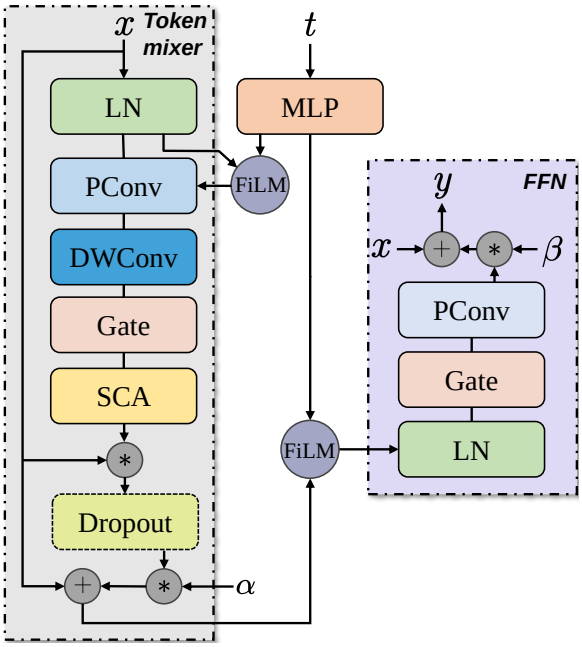


FIGURE 5. Illustration of the proposed efficient SBM block. SBM block shares a similar design scheme to the Metaformer block, composed of a token mixer and an FFN.

attention becomes unacceptable when dealing with high-resolution images under resource-constrained conditions. Due to the conditioning on t of $\nabla_{Y_t} \log p_t(Y_t)$, it is important to input timestep t into the model. We feed t into an MLP and adopt FiLM [68] to modulate feature x, formulated as,

$$a, b = MLP(t) \tag{41a}$$

$$y = (1 + a) \times x + b \tag{41b}$$

Specifically, we remove the commonly-used Attention layer and replace it with a Simple Channel Attention (CSA), which can formulated as,

$$y = PConv(AdapPool(x)), \tag{42}$$

where $PConv$ denotes point-wise convolution and $AdapPool$ is adaptive pooling the input x into $1 \times 1 \times c$ tensor. In order to reduce the number of parameters, we utilize a combination of point-wise convolution and depth-wise convoltuion [34] to replace the normal convolution. The pre-norm [64] scheme is adopted to stabilize the training. According to GELU [29] formulated, a simple gate is employed to replace the GELU activation function which is more computationally efficient and can be expressed as,

$$y = x_{[:c/2]} \times x_{[c/2:]}, \tag{43}$$

where the input x has the shape of $H \times W \times c$ (c is always divisible by 2.). To facilitate the adapbility of the SBM block, we introduce another two learnable parameters α, β to control the block feature z and shortcut feature x:

$$y = x + \alpha \times z, \tag{44a}$$

$$y = x + \beta \times z. \tag{44b}$$

The overall illustration of the SBM block is shown in Fig. 5.

5. **Experiments.** In this section, we first validate the designed training and sampling algorithms on a toy example, revealing some properties of SB SDEs and ODEs. Then we elaborate on the pansharpening experimental settings, datasets, and benchmarking. Performances of different traditional methods, regressive methods, and diffusion (or score)-based methods are compared on pansharpening reduced and full assessments. At last, some ablation studies are included to further elucidate some possible factors that can affect the proposed SB SDE learning.

5.1. **Toy Example.** We first examine the proposed SB SDE on a toy dataset. We set an 8 Gaussian mixture distribution as the SB start point P_X and a Swiss roll distribution as the SB endpoint P_Y. The learned SB can translate the Gaussian mixture to the Swiss Roll as shown in Fig. 6. The proposed SB SDE can *progressively* transport the 8 Gaussian mixture to the Swiss Roll distribution, which verifies the efficacy.

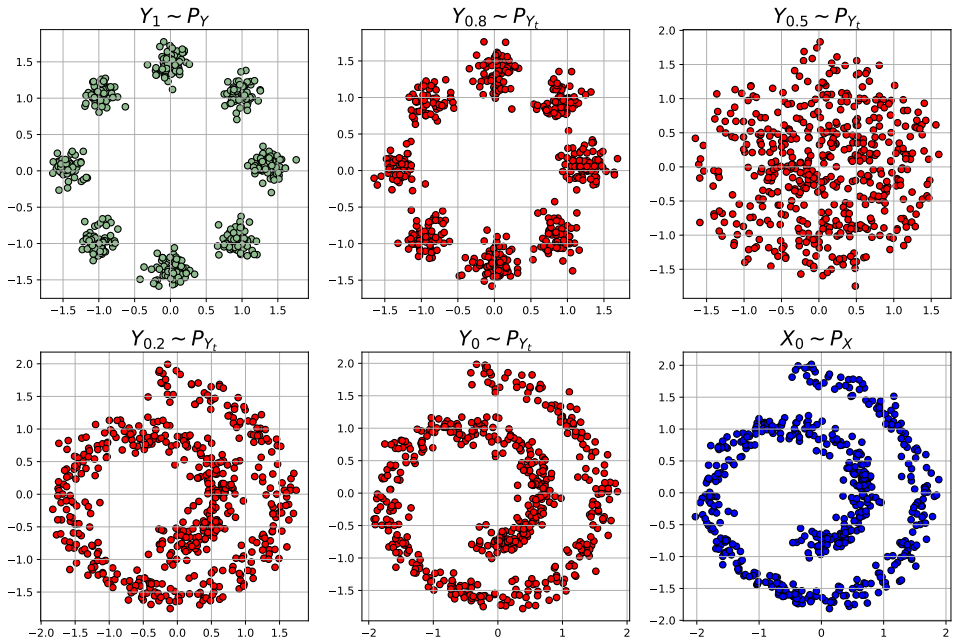


FIGURE 6. 2D toy example: 8 Gaussian Mixture translates to Swiss Roll.

5.2. **Experiment Setup & Implementation Details.** In the following experiment, we conduct pansharpening tasks using SB SDE and SB ODE and compare them with PanDiff [60], which is based on vanilla DPM [31] as the baseline. We leverage the algorithm 1 and 2 to train and sample from the dataset. p_X is the GT (*i.e.,* HRMS), p_Y is the LRMS, and p_c is the PAN as the condition for the proposed SDE and ODE. During the sampling phase, we employ *5-step* Euler SDE sampling

and *1-step* ODE sampling for all datasets, in order to compare the sampling performance of other DPM methods. We will discuss the advantages and disadvantages of SB SDE and ODE and some of their features.

Regarding the network architecture proposed in Sect. 4, we set the number of channels for the first SBM layer to 32, and each SBM layer contains 3 SBM blocks. In the encoder, we use the 3×3 convolution for downsampling and apply pixel-shuffle [79] for upsampling in the decoder. The input LRMS and PAN images are both normalized to the range of 0 to 1.

TABLE 1. Quantitative results of all competing methods. The Q2n index is Q8/Q4 for 8-band/4-band data. The best results are in **red** and the second best results are in **blue**.

		Reduced Resolution (RR): Avg±std				Full Resolution (FR): Avg±std		
		SAM	ERGAS	Q2n	SCC	D_λ	D_s	HQNR
WorldView-3 (WV3, 8-band)	BDSD-PC [92]	5.47±1.72	4.65±1.47	0.812±0.106	0.905±0.042	0.063±0.024	0.073±0.036	0.870±0.053
	MTF-GLP-FS [93]	5.32±1.65	4.65±1.44	0.818±0.101	0.898±0.047	0.021±0.008	0.063±0.028	0.918±0.035
	BT-H [55]	4.90±1.30	4.52±1.33	0.818±0.102	0.924±0.024	0.057±0.023	0.081±0.037	0.867±0.054
	LRTCFPan [102]	4.74±1.41	4.32±1.44	0.846±0.091	0.927±0.023	0.018±0.007	0.053±0.026	0.931±0.031
	PNN [59]	3.68±0.76	2.68±0.65	0.893±0.092	0.976±0.008	0.021±0.008	0.043±0.015	0.937±0.021
	PanNet [104]	3.62±0.77	2.67±0.69	0.891±0.093	0.976±0.009	0.017±0.007	0.047±0.021	0.937±0.027
	MSDCNN [108]	3.78±0.80	2.76±0.69	0.890±0.090	0.974±0.008	0.023±0.009	0.047±0.020	0.932±0.027
	DiCNN [27]	3.59±0.76	2.67±0.66	0.900±0.087	0.976±0.007	0.036±0.011	0.046±0.018	0.920±0.026
	FusionNet [16]	3.33±0.70	2.47±0.64	0.904±0.090	0.981±0.007	0.024±0.009	0.036±0.014	0.941±0.020
	LAGConv [38]	3.10±0.56	2.30±0.61	0.910±0.091	0.984±0.007	0.037±0.015	0.042±0.015	0.923±0.025
	Invformer [112]	3.25±0.64	2.39±0.52	0.906±0.084	0.983±0.005	0.055±0.029	0.068±0.031	0.882±0.049
	DCFNet [101]	3.03±0.74	2.16±0.46	0.905±0.088	0.986±0.004	0.078±0.081	0.051±0.034	0.877±0.101
	HMPNet [88]	3.06±0.58	2.23±0.55	0.916±0.087	0.986±0.005	0.018±0.007	0.053±0.006	0.929±0.011
	PanDiff [60]	3.30±0.60	2.47±0.58	0.898±0.088	0.980±0.006	0.027±0.012	0.054±0.026	0.920±0.036
	DDIF [6]	**2.74±0.51**	**2.02±0.45**	**0.920±0.082**	**0.988±0.003**	0.026±0.018	**0.023±0.008**	**0.952±0.017**
	Proposed(*5-step SDE*)	**2.73±0.51**	**1.99±0.44**	**0.921±0.081**	**0.989±0.003**	**0.013±0.005**	0.039±0.005	0.949±0.009
	Proposed(*1-step ODE*)	3.67±0.81	2.74±0.74	0.886±0.096	0.974±0.009	**0.011±0.004**	**0.031±0.002**	**0.958±0.005**
GaoFen2 (GF2, 4-band)	BDSD-PC [92]	1.71±0.32	1.70±0.41	0.993±0.031	0.945±0.017	0.076±0.030	0.155±0.028	0.781±0.041
	MTF-GLP-FS [93]	1.68±0.35	1.60±0.35	0.891±0.026	0.939±0.020	0.035±0.014	0.143±0.028	0.823±0.035
	BT-H [55]	1.68±0.32	1.55±0.36	0.909±0.029	0.951±0.015	0.060±0.025	0.131±0.019	0.817±0.031
	LRTCFPan [102]	1.30±0.31	1.27±0.34	0.935±0.030	0.964±0.012	0.033±0.027	0.090±0.014	0.881±0.023
	PNN [59]	1.05±0.23	1.06±0.24	0.960±0.010	0.977±0.005	0.037±0.029	0.094±0.022	0.873±0.037
	PanNet [104]	1.00±0.21	0.92±0.19	0.967±0.010	0.983±0.004	0.021±0.011	0.080±0.018	0.901±0.020
	MSDCNN [108]	1.05±0.22	1.04±0.23	0.961±0.011	0.978±0.005	0.027±0.013	0.073±0.009	0.902±0.013
	DiCNN [27]	1.05±0.23	1.08±0.25	0.959±0.010	0.977±0.006	0.041±0.012	0.099±0.013	0.864±0.017
	FusionNet [16]	0.97±0.21	0.99±0.22	0.964±0.009	0.981±0.005	0.040±0.013	0.101±0.013	0.863±0.018
	LAGConv [38]	0.78±0.15	0.69±0.11	0.980±0.009	0.991±0.002	0.032±0.013	0.079±0.014	0.891±0.020
	Invformer [112]	0.83±0.14	0.70±0.11	0.977±0.012	0.980±0.002	0.059±0.026	0.110±0.015	0.838±0.024
	DCFNet [101]	0.89±0.16	0.81±0.14	0.973±0.010	0.985±0.002	0.023±0.012	0.066±0.010	0.912±0.012
	HMPNet [88]	0.80±0.14	**0.56±0.10**	0.981±0.030	**0.993±0.003**	0.080±0.050	0.115±0.012	0.815±0.049
	PanDiff [60]	0.89±0.12	0.75±0.10	0.979±0.010	0.989±0.002	0.027±0.020	0.073±0.010	0.903±0.021
	DDIF [6]	**0.64±0.12**	0.57±0.10	**0.986±0.008**	0.986±0.004	**0.020±0.011**	0.041±0.010	0.940±0.014
	Proposed(*5-step SDE*)	**0.59±0.11**	**0.53±0.10**	**0.987±0.007**	**0.994±0.002**	0.026±0.012	**0.028±0.010**	**0.947±0.013**
	Proposed(*1-step ODE*)	0.80±0.15	0.72±0.11	0.978±0.009	0.988±0.002	**0.016±0.012**	**0.017±0.004**	**0.967±0.010**
	Ideal value	**0**	**0**	**1**	**1**	**0**	**0**	**1**

5.3. **Datasets.** We conduct experiments on a standard pansharpening data-collection (*i.e.*, Pancollection[3]), which includes WorldView-3 (WV3, 8 bands) and GaoFen-2 (GF2, 4band) data. The reduced data are simulated from real-world images using Wald's protocol [96]. WV3 dataset contains 9714/1080 samples for training and validation. Each sample consists of a PAN/LRMS/GT image pair of size $64 \times 64 \times 1$, $16 \times 16 \times 8$, and $64 \times 64 \times 8$, respectively. PAN image has a spatial resolution of 0.3m, whereas the LRMS image has a spatial resolution of 1.2m. GF2 dataset contains 19809/2201 samples for training and validation. Each sample consists of a PAN/LRMS/GT image pair of sizes $64 \times 64 \times 1, 16 \times 16 \times 4$, and $64 \times 64 \times 4$, respectively. PAN images have a spatial resolution of 0.8m, while

[3] https://liangjiandeng.github.io/PanCollection.html

LRMS images have a spatial resolution of 3.2m. To evaluate the performance, we perform the reduced-resolution and full-resolution experiments to compute the reference and non-reference metrics, respectively. The WV3 reduced-resolution test set has 20 PAN/LRMS/GT image pairs of size $256 \times 256 \times 1, 64 \times 64 \times 8$ and $256 \times 256 \times 8$. The WV3 full-resolution test set consists of a PAN/LRMS image pair of size $512 \times 512 \times 1, 128 \times 128 \times 8$. The GF2 reduced and full-resolution test sets have the same samples as the WV3 test set but only differ in the number of bands (4 bands).

To verify the performance of each method on hyperspectral real remote sensing data, we utilize the GF5-GF1 public dataset [25]. 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 40×40 and 80×80 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 $\mathbf{M} = (\mathbf{M} - \mathbf{B} \cdot \mathbf{R})/\mathbf{A}$ (as proposed in [25]), where $\mathbf{A}$ and $\mathbf{B}$ are the correlated weight tensors, and $\mathbf{R}$ 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%.

5.4. **Benchmarking.** For the WV3 and GF2 multispectral datasets, we compare the proposed SB SDE/ODE with representative methods such as CS, MRA, VO, regression-based deep methods, and recent DPM-based methods as follows:

1. Model-based methods: BDSD-PC[TGRS, 2017] [92], MTF-GLP-FS[TIP, 2018] [93], BT-H[GRSL, 2017] [55], and LRTCFPan[TIP, 2023] [102];
2. Regressive DL-based methods: PNN[RS, 2016] [59], PanNet[ICCV, 2017] [104], MSDCNN[JSTAR, 2018] [108], DiCNN[JSTAR, 2019] [27], FusionNet[TGRS,2020] [16], LAGConv[AAAI,2022] [38], Invformer[AAAI, 2022] [113], DCFNet[ICCV, 2021] [101], HMPNet[TNNLS, 2023] [88];
3. DPM (score)-based methods: PanDiff[TGRS, 2023] [60], DDIF[InFus, 2024] [6].

For the hyperspectral real GF5-GF1 dataset, we choose widely-applied model-based methods, DL-based regressive methods, and DPM (score)-based methods to compare, listed as follows:

1. Model-based methods: CNMF[IGRSS,2011] [106], Hysure[TGRS, 2014] [81], GSA[TGRS, 2007] [3], LTTR[TNNLS, 2019] [21], LTMR[TIP, 2019] [20], MTF-GLP-HS[JSATR, 2015] [78];
2. Regessive DL-based methods: ResTFNet[InFus, 2020] [52], SSRNet[TGRS, 2020] [110], Fusformer[GRSL, 2022] [35], HSRNet[TNNLS, 2021] [36], DHIF[TCI, 2022] [37];
3. DPM (score)-based methods: PanDiff[TGRS, 2023] [60], DDIF[InFus, 2024] [6].

Following [17], we employ SAM [109], ERGAS [95], Q2n [24], and SCC [111] metrics to validate the effectiveness of different methods on the WV3 and GF2 reduced-resolution datasets. For the GF5-GF1 dataset with a greater number of spectral bands, we include PSNR and SSIM metrics. For the full-resolution experiments, we use D_λ, D_s, and HQNR [5] as the evaluation metrics. Note that the same training and data augmentation strategy is applied to the compared methods for a fair comparison.

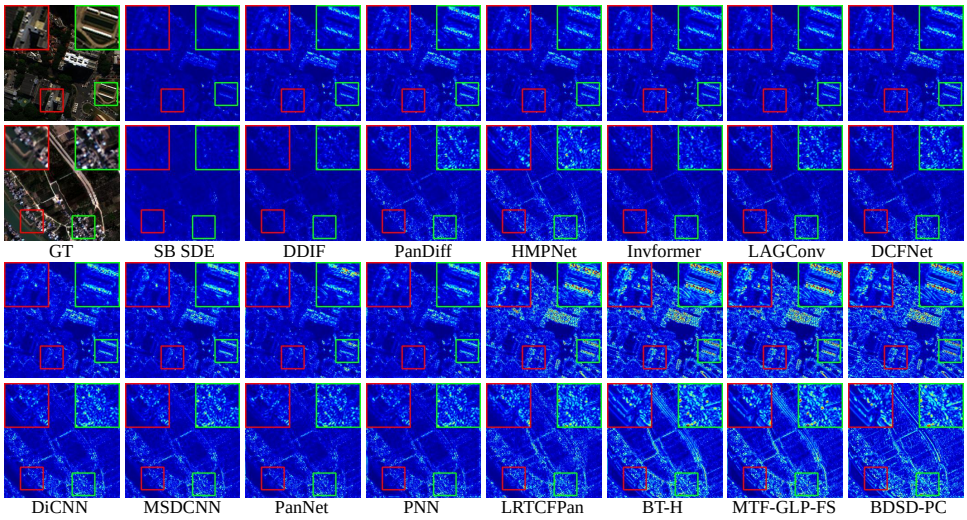


FIGURE 7. Illustration of ground truth and error maps with various pansharpening methods on WV3 (Row 1 and 3) and GF2 (Row 2 and 4) reduced-resolution test set.

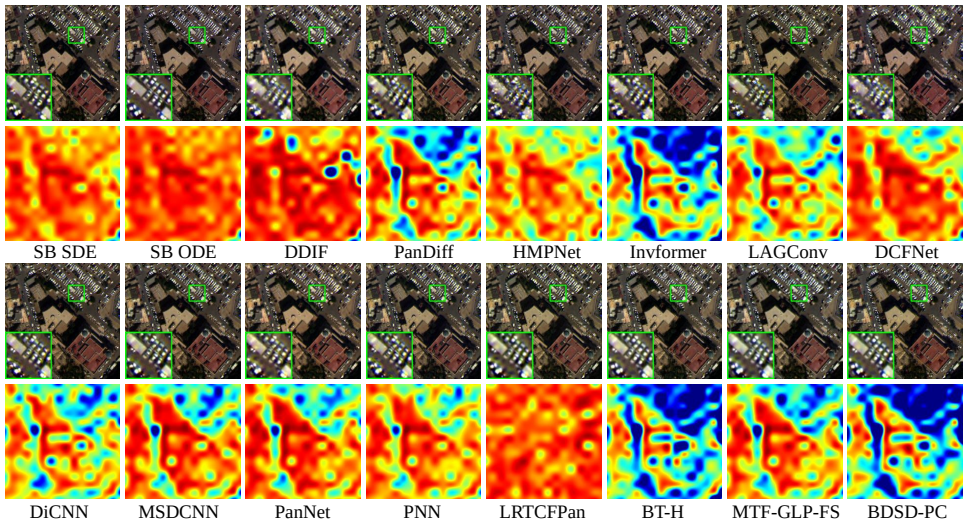


FIGURE 8. Illustration of fused full-resolution image and corresponding HQNR map on WV3 full-resolution test set. The color red in the HQNR map indicates a value close to 1, while the color blue represents a value close to 0. A higher value in the HQNR map indicates a better pansharpening performance.

5.5. **Reduced Assessments.** The reduced assessments of WV3, GF2, and GF5-GF1 datasets are provided at the left panel of Tabs. 1 and 2. Some error maps are shown in Fig. 7. We can see that all model-based methods still struggle to beat the first DL-based method PNN. With the continuous design and optimization of

network architectures, regressive deep learning-based methods have shown significant improvements in their performance on reduced-resolution data, achieving near state-of-the-art (SOTA) results.

The exploration of PanDiff for pansharpening is commendable. However, due to the choice of the ϵ-learning objective (readers are encouraged to refer to [65] for details about ϵ-learning), it has achieved only a relatively average fusion performance. It is worth noting that PanDiff still utilizes the DDPM ancestral sampling strategy [31], which significantly slows down the sampling process. It requires 2000 network forward steps to sample a single fused image. In contrast, DDIF has proposed a more efficient network architecture and alleviated the issues of DPM-based methods by using 25-step DDIM sampling [84], achieving state-of-the-art (SOTA) performance. However, due to the forward strategy of DPM, DDIF starts sampling from pure Gaussian noise, overlooking the inherent characteristics of the pansharpening task. Additionally, the adopted SDE-based sampling curvature in DDIF is curved, leading to larger discretization errors and making efficient sampling more challenging.

The proposed SB SDE outperforms DPM-based methods including PanDiff and DDIF. Alternatively, SB ODE demonstrates a more efficient sampling procedure (5-steps sampling *v.s.* only 1-step sampling) that achieves competitive pansharpening performance.

5.6. **Full Assessments.** The full assessments are shown in the right panel of Tabs. 1 and 2. SB SDE exhibits full-resolution metrics on the WV3 dataset that are second only to DDIF. Its comprehensive metric HQNR surpasses all model-based, DL-based regressive, and DPM-based methods on the GF2 dataset. Additionally, it demonstrates highly competitive performance on the GF5-GF1 dataset. For SB ODE, its highly efficient one-step sampling makes it an excellent pansharpening method compared to other DPM-based methods. As depicted in Fig. 8, both the proposed SB SDE and ODE demonstrate satisfying fusion results. The fused images exhibit the spectral characteristics of the LRMS and the spatial features of the PAN.

5.7. **Ablation Study.** In this subsection, we ablate the key factors in the SB SDE/ODE learning and sampling and provide experimental results on typical pansharpening WV3 dataset.

5.7.1. *Different Parameterization.* In Sect. 3.5, we provide four different learning objectives for SB SDE/ODE:

(a) bridge endpoint X_0;
(b) bridge length $Y_1 - X_0$;
(c) bridge posterior length $Y_t - X_0$;
(d) bridge endpoint with score $[X_0, s]$ (SB SDE only).

We design three ablations on the learning objectives of the WV3 dataset and the results are provided in the upper panel of Tab. 3.

We discovered that utilizing the score as the learning objective resulted in an unstable training procedure and slow convergence. However, when using the bridge length as the objective, we observed a more stable training loss and decent fusion performance. Although switching to bridge length with score improved the performance, it reintroduced training instability. Nevertheless, our default setting, which

TABLE 2. 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**.

Methods	Reduced Resolution (RR): Avg±std					Full Resolution (FR): Avg±std		
	PSNR	SSIM	Q2n	SAM	ERGAS	D_λ	D_s	HQNR
CNMF [106]	44.25±3.89	0.9823±0.0122	0.742±0.177	0.851±0.213	2.761±0.767	0.045±0.083	0.059±0.050	0.898±0.084
Hysure [81]	42.52±4.52	0.9723±0.0137	0.732±0.146	1.305±0.406	3.677±1.317	0.041±0.076	0.074±0.105	0.887±0.117
GSA [3]	44.99±5.34	0.9795±0.0119	0.754±0.131	1.200±0.332	2.898±0.956	0.053±0.103	0.067±0.062	0.882±0.101
LTTR [21]	47.15±2.91	0.9897±0.0028	0.844±0.098	2.159±0.286	5.808±2.732	0.099±0.123	0.047±0.023	0.860±0.106
LTMR [20]	45.52±2.42	0.9898±0.0030	0.849±0.108	1.595±0.344	2.720±1.277	0.057±0.103	0.036±0.018	0.910±0.105
MTF-GLP-HS [78]	45.60±5.82	0.9837±0.0120	0.777±0.146	0.856±0.265	2.848±1.154	0.030±0.048	0.075±0.105	0.897±0.106
ResTFNet [52]	46.98±2.11	0.9934±0.0022	0.850±0.100	0.906±0.130	3.323±3.123	0.042±0.078	0.088±0.059	0.874±0.099
SSRNet [110]	45.49±2.69	0.9880±0.0047	0.850±0.094	1.039±0.210	4.863±4.161	0.117±0.140	0.054±0.019	0.836±0.134
Fusformer [35]	49.74±4.64	0.9914±0.0031	0.891±0.076	0.638±0.155	4.761±0.592	0.030±0.056	0.040±0.025	0.931±0.064
HSRNet [36]	49.81±3.05	0.9964±0.0016	0.888±0.081	0.693±0.139	0.901±0.451	0.038±0.073	0.047±0.020	0.917±0.075
DHIF [37]	55.35±4.20	0.9982±0.0009	0.929±0.076	0.309±0.062	0.885±0.388	**0.031±0.057**	**0.034±0.022**	**0.937±0.062**
PanDiff [60]	50.43±2.89	0.9958±0.0017	0.903±0.080	0.633±0.108	1.877±1.393	0.033±0.059	0.041±0.027	0.928±0.063
DDIF [6]	**56.40±3.82**	**0.9984±0.0007**	**0.938±0.064**	**0.273±0.049**	**0.845±0.505**	**0.033±0.057**	**0.035±0.021**	**0.933±0.057**
Proposed(5-step SDE)	**60.94±2.03**	**0.9995±0.0006**	**0.964±2.031**	**0.233±0.031**	**0.679±0.541**	0.038±0.071	0.040±0.020	0.924±0.071
Proposed(1-step ODE)	50.64±2.62	0.9971±0.0012	0.909±0.081	0.743±0.066	1.721±1.265	0.036±0.066	0.040±0.021	0.926±0.069
Ideal value	$+\infty$	**1**	**1**	**0**	**0**	**0**	**0**	**1**

TABLE 3. Ablation studies on learning objectives and training timesteps. The best options are in **red**. The left-hand side of "/" represents the results of SB SDE, while the right-hand side represents SB ODE's.

Ablations	Metrics / Options	SAM	ERGAS	Q2n	SCC
Objectives	X_0	2.73/3.67	1.99/2.74	0.921/0.886	0.989/0.974
	$Y_1 - X_0$	2.82/3.69	2.16/2.78	0.918/0.880	0.987/0.973
	$Y_t - X_0$	2.76/3.67	2.04/2.76	0.919/0.885	0.986/0.974
	$[X_0, \epsilon]$(SDE only)	2.74	2.00	0.921	0.898
Training Timesteps	1000	2.72/3.63	1.98/2.70	0.920/0.889	0.990/0.976
	500	2.73/3.66	1.99/2.72	0.921/0.887	0.989/0.975

employs the bridge endpoint, outperforms other objectives and achieves the best fusion metrics.

5.7.2. *Timesteps of Training and Fast Sampling.* As demonstrated in DDPM [31] and iDDPM [65], the choice of training timestep has a significant impact on the final generation performance. Previous studies have acknowledged that larger timesteps can enhance the model's generation ability. To investigate the influence of the timestep on bridge matching, we conducted training with different timestep values: 1000 steps, 500 steps, and 100 steps. The corresponding metrics are listed in the lower panel of Tab. 3.

5.7.3. *Better Architecture for Schrödinger Matching.* To validate the suitability of the designed SBM-Net for SB matching training, we employ X_0 as the prediction objective, substituting various networks for validation. Additionally, we compare it with other architectures: 1) Vanilla U-Net [31], PanDiff [60], and DDIF [6]. The results are shown in Tab. 4. It can be seen that the proposed SBM-Net can outperform all previous DPM-based networks.

6. **Disscussion.**

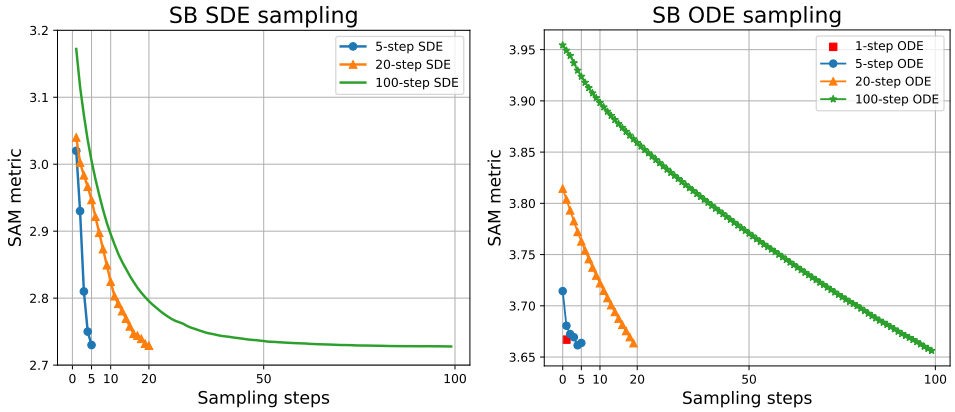


FIGURE 9. SB SDE/ODE performance of default setting with various sampling steps on WV3 dataset.

6.1. **Difference and Connections Between SB SDE/ODE, Diffusion, Flow Matching and Rectify Flow. Diffusion** is an SDE framework bridging the Gaussian distribution and the target distribution. Additionally, [85] removed the stochastic term from the SDE to construct the PF ODE, which can be expressed in discrete form:

$$Y_t = \alpha_t Y_0 + \sigma_t Z, Z \sim \mathcal{N}(0, \mathbf{I}). \tag{45}$$

Previous works [31, 85, 87] on diffusion propose various methods for constructing ODE formulations:

$$\begin{cases} \text{(sub-)VP ODE} : \alpha_t = \exp\left(-\frac{1}{4}a(1-t)^2 - \frac{1}{2}b(1-t)\right), \\ \text{VP ODE} : \sigma_t = \sqrt{1-\alpha_t^2}, \text{sub-VP ODE} : \sigma_t = 1-\alpha_t^2, \\ \text{where } a = 19.9, b = 0.1; \end{cases} \tag{46a}$$

$$\begin{cases} \text{VE ODE} : \alpha_t = 1, \sigma_t = \sigma_{\min}\sqrt{r^{2(1-t)}-1}, \\ \text{s.t., } \sigma_{\max} := r\sigma_{\min}, Y_1 = Y_0 + \beta_1 Z \approx \sigma_{\max} Z \end{cases} \tag{46b}$$

All of them can be regarded as interpolations between two distributions. In this view, the cost incurred by a sampler transferring one distribution to another depends on the distance and curvature of its trajectory. When the coupling (Y_0, Z) is given, the distance is prefixed, and the cost depends on the curvature. For different SDEs or ODEs, α_t and β_t determine the curvature of the trajectory. From the above expressions, it is evident that the conventional ODEs previously used do not yield straight trajectories due to the nonlinear relationship between α_t and σ_t. Consequently, these ODE methods require large sampling steps to achieve satisfactory sampling outcomes. In contrast, our method produces straight sampling trajectories and demonstrates excellent sampling efficiency and minimal truncation errors even under the coarsest discrete method (Euler method). On the other hand, as pointed out in Sect. 3.2, the Diffusion framework facilitates transport between Gaussian and target distributions, but it is not conducive to inverse problems with a known prior distribution. Additionally, fixing one end to be a Gaussian distribution is unnecessary. The proposed SB SDE/ODE addresses this issue. Moreover, one can notice that the SB SDE can be degraded to DPM when setting p_Y as Gaussian.

Flow Matching extends Countinous Normalizing Flow [8] and utilizing the change of varibles of [9] to establish an ODE generative framework. A vector field v_t can be used to construct a time-dependent diffeomorphic map $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, which is defined by an ODE:

$$\frac{d}{dt}\phi_t(y) = v_t(\phi_t(y)), \tag{47a}$$

$$\phi_0(y) = y. \tag{47b}$$

The proposed SB ODE can be degraded to the flow matching ODE.

Corollary 6.1. *When β_t is a constant β over t, the velocity $v_t = (Y_t - X_0)/t$ and the mean in Eq. (27) turns into $\mu_t = (1-t)X_0 + tY_1$, which returns to **Example II** in Flow Matching [49].*

Proof. When $\beta_t = \beta$ is a constant, the term $\frac{\beta_t}{\sigma_t^2}$ decays in $\mathcal{O}(1/t)$ because $\sigma_t^2 = \int_0^t \beta_\tau d\tau = \beta t$. Incorporating with Corollary 3.7, SB ODE velocity $v_t = (Y_t - X_0)/t$ and the posterior mean $\mu_t = (1-t)X_0 + tY_1$. □

Rectify Flow is derived from the perspective of the transport mapping problem, resulting in ordinary differential equation (ODE) formulations consistent with Flow Matching formulations. In order to further rectify the straightness of the flow, the Reflow [53] possesses the ability to reduce transport costs, which parallels the proposed SB ODE.

6.2. **Relation between SB SDE with OT Problems.** We first introduce entropic OT formulation, which is based on Kantorovich's OT problem [91]. We use $H(\pi)$ to express the entropy of a distribution π, and $D_{KL}(\pi_1, \pi_2)$ to denote the Kullback-Leibler divergence. One widely-used entropic OT formulation with $H(\pi)$ based on Kantorovich's OT (with quadratic cost) is:

$$\inf_{\pi \in \Pi(p_X, p_Y)} \int_{\mathcal{X}\times\mathcal{Y}} \frac{\|x-y\|^2}{2} d\pi(x,y) - \epsilon H(\pi), \tag{48}$$

where π is a transport plan between p_X and p_Y, $\mathcal{X}, \mathcal{Y}$ are D-dimensional Euclidean space, and $\Pi(p_X, p_Y) \subset \mathscr{P}_2(\mathcal{X}\times\mathcal{Y})$ is the set of probability distribution on $\mathcal{X}\times\mathcal{Y}$ with marginals p_X and p_Y. The minimizer is unique due to the convexity of $H(\pi)$.

We denote the path measures $\mathbb{Q}$ and $\mathbb{P}$ by the forward (20a) and backward SDEs (20b), respectively. Within the SB framework (8), the SB problem gives rise to a stochastic process $\mathbb{P}$ with marginal distributions p_X and p_Y at $t=0$ and $t=1$, respectively, which has minimal KL divergence with the prior path measure $\mathbb{Q}$.

Now, by using Problem 4.2 in [10], we denote OT plan $\pi^{\mathbb{P}}$ as the joint distribution of the stochastic process $\mathbb{P}$ whose marginal distrbutions are $\pi_0^{\mathbb{P}}$, $\pi_1^{\mathbb{P}}$. The coupling (X,Y) at times $t=0,1$ determines the process $\mathbb{Q}$ and further denoted as the conditional form $\mathbb{P}_{|X,Y}$. The KL term in Eq. (8) can be decomposed as [90],

$$\begin{aligned} D_{KL}(\mathbb{Q}\|\mathbb{P}) =& D_{KL}(\pi^{\mathbb{Q}}\|\pi^{\mathbb{P}}) \\ &+ \int_{\mathcal{X}\times\mathcal{Y}} D_{KL}(\mathbb{Q}_{|X,Y}\|\mathbb{P}_{|X,Y}) d\pi^{\mathbb{P}}(X,Y), \end{aligned} \tag{49}$$

the first term is the KL divergence at $t=0,1$ (*i.e.,* SDE start and end points), and the second term denotes the similarity of two processes during intermediate times.

TABLE 4. Ablation study on different architecture for SB SDE.

Arch.	SAM	ERGAS	Q2n	SCC
U-Net	3.24	2.43	0.907	0.982
PanDiff	3.10	2.28	0.912	0.985
DDIF	2.76	2.03	0.919	0.989
SBM-Net	2.73	1.99	0.921	0.989

Moreover, the first term can be rewritten as,

$$D_{KL}(\pi^{\mathbb{Q}}\|\pi^{\mathbb{P}}) = \int_{\mathcal{X}\times\mathcal{Y}} \frac{\|X-Y\|^2}{2\epsilon} d\pi^{\mathbb{P}}(X,Y) - H(\pi^{\mathbb{P}}) + const. \tag{50}$$

Proofs are in Appendix B. In Proposition 2.3 of [47], if $\mathbb{P}^*$ is the solution to Eq. (8), then $\mathbb{P}^*_{|X,Y} = \mathbb{Q}_{|X,Y}$, hence, $\forall(X,Y)$, one can set the second term in Eq. (49) to zero and optimize Eq. (8) over process $\mathbb{P}$, which means:

$$\begin{aligned} \inf_{\mathbb{P}\in\mathcal{F}(p_X,p_Y)} D_{KL}(\mathbb{Q}\|\mathbb{P}) &= \inf_{\mathbb{P}\in\mathcal{F}(p_X,p_Y)} D_{KL}(\pi^{\mathbb{Q}}\|\pi^{\mathbb{P}}) \\ &= \inf_{\pi^{\mathbb{P}}\in\Pi(p_X,p_Y)} D_{KL}(\pi^{\mathbb{Q}}\|\pi^{\mathbb{P}}), \end{aligned} \tag{51}$$

It implies the solutions of Eqs. (48) and (50) coincide and the SB problem can be simplified to the entrpoic OT problem.

6.3. **Why Stochasticity Can Help SB Matching.** Informally, SB SDE and SB ODE differ only by a stochastic Wiener term in their discrete formulations. This term introduces stochasticity along the trajectory in SB SDE. From the main results, it is evident that SB SDE yields better outcomes compared to SB ODE. This is attributed to the characteristic of pansharpening tasks requiring the restoration of high-frequency information. The trajectory of SB ODE is deterministic and do not introduce any Gaussian noise. Therefore, to recover the high-frequency information of HRMS, the network needs to directly learn to generate the high frequencies, which is challenging. In contrast, with SB SDE, the introduction of Gaussian noise containing high frequencies during the sampling process makes it easier for the network to restore.

6.4. **Curvature of SB ODE compared with Diffusion VP ODE.** For a generative process $\mathbf{Y} = \{Y_t\}_0^1$ with initial value $Y_0 = X_0$, we informally provide curvature definition which the generative trajectory from a straight path:

$$C(\mathbf{Y}) = \mathbb{E}\left\|Y_1 - X_0 - \frac{\partial}{\partial t}Y_t\right\|_2^2. \tag{52}$$

The mean curvature $\mathbb{E}_{Y\sim p(\mathbf{Y})}C(\mathbf{Y})$ is first considered and related to the truncation error of the ODE solver. Therefore, the zero curvature represents straight sampling trajectory. As a generative process is a time reversal of the forward process, the curvature of sampling process is determined by the forward process. In previous DPM works [31, 85], the forward process is prefixed, thus the optimal curvature of the sampling process can be derived. Consider the proposed SB ODE (38) for example, the average curvature can be derived,

$$C(\mathbf{Y})_{\text{SB ODE}} = \mathbb{E}_{\mathbf{Y},t}\left\|Y_1 - X_0 - \nabla_t \frac{\hat{\sigma}_t^2}{\hat{\sigma}_t^2+\sigma_t^2}X_0 - \nabla_t \frac{\sigma_t^2}{\hat{\sigma}_t^2+\sigma_t^2}Y_1\right\|_2^2 \tag{53a}$$

$$= \mathbb{E}_{\mathbf{Y},t}\|(1-M_t)(Y_1 - X_0)\|_2^2, \tag{53b}$$
$$\text{where } M_t = \frac{2\beta_t\sigma_t\hat{\sigma}_t(\sigma_t+\hat{\sigma}_t)}{(\sigma_t^2+\hat{\sigma}_t^2)^2},$$

and the average curvature of PF ODE (*e.g.*, VP SDE) is,

$$C(\mathbf{Y})_{\text{VP ODE}} = \mathbb{E}_{\mathbf{Y},t}\Big\|Y_1 - X_0 - \nabla_t(\alpha_t Y_0 + \beta_t Z)\Big\|_2^2 \tag{54a}$$
$$= \mathbb{E}_{\mathbf{Y},t}\Big\|(1+N_t)X_0\Big\|_2^2 + \mathbb{E}_t\|1+K_t\|_2^2\mathbf{I}_d, \tag{54b}$$
$$\text{where, } N_t = \frac{1}{8}[-2a(1-t)^2 - 4b(1-t)][a(1-t)+b]\alpha_t,$$
$$K_t = -\alpha_t(1-\alpha_t^2)^{-\frac{1}{2}}N_t.$$

Following this definition, we perform average curvature numerical check on WV3 test set, the results are only 0.09 and 2.79 of SB ODE and VP ODE (note that f_t is defined by more widely-used Variance Preserving (VP) parameterization [31, 87] as in Eq. (46a)). The curvature of the proposed SB-ODE is much smaller than that of the previous DPM ODE, indicating that we can use fewer sampling steps to sample along the trajectory and suffer less sampling truncation errors. The derivations of the curvatures are provided in Appendix A.

7. Conclusion.

This paper introduces an SDE/ODE method based on the Schrödinger Bridge (SB) to address the limitations of previous DPM-based approaches, such as being limited to transportation between Gaussian and data distributions, low sampling efficiency, and the complexity associated with training objectives and trajectory simulation in previous IPF and likelihood-based SB methods. Our approach achieves SOTA performance on three datasets for pansharpening, laying the groundwork for the application of SB methods in pansharpening tasks.

8. 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).

REFERENCES

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

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

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

[4] B. D. Anderson, Reverse-time diffusion equation models, *Stochastic Processes and their Applications*, **12** (1982), 313–326.

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

[6] Z. Cao, S. Cao, L.-J. Deng, X. Wu, J. Hou and G. Vivone, Diffusion model with disentangled modulations for sharpening multispectral and hyperspectral images, *Information Fusion*, **104** (2024), 102158.

[7] L. Chen, X. Chu, X. Zhang and J. Sun, Simple baselines for image restoration, in *European Conference on Computer Vision*, Springer, 2022, 17–33.
[8] R. T. Chen, Y. Rubanova, J. Bettencourt and D. K. Duvenaud, Neural ordinary differential equations, *Advances in neural information processing systems*, **31**.
[9] T. Chen, G.-H. Liu and E. Theodorou, Likelihood training of schrödinger bridge using forward-backward SDEs theory, in *International Conference on Learning Representations*, 2022.
[10] Y. Chen, T. T. Georgiou and M. Pavon, Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a schrödinger bridge, *SIAM Review*, **63** (2021), 249–313.
[11] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky and J. C. Ye, Diffusion posterior sampling for general noisy inverse problems, in *The Eleventh International Conference on Learning Representations*, 2023.
[12] K. Crowson, S. A. Baumann, A. Birch, T. M. Abraham, D. Z. Kaplan and E. Shippole, Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers, *arXiv preprint arXiv:2401.11605*.
[13] P. Dai Pra, A stochastic control approach to reciprocal diffusion processes, *Applied mathematics and Optimization*, **23** (1991), 313–329.
[14] V. De Bortoli, J. Thornton, J. Heng and A. Doucet, Diffusion schrödinger bridge with applications to score-based generative modeling, *Advances in Neural Information Processing Systems (Neurips)*, **34** (2021), 17695–17709.
[15] L.-J. Deng, G. Vivone, C. Jin and J. Chanussot, Detail injection-based deep convolutional neural networks for pansharpening, *IEEE Transactions on Geoscience and Remote Sensing*, **59** (2020), 6995–7010.
[16] L.-J. Deng, G. Vivone, C. Jin and J. Chanussot, Detail injection-based deep convolutional neural networks for pansharpening, *IEEE Transactions on Geoscience and Remote Sensing*, **59** (2021), 6995–7010.
[17] L.-J. Deng, G. Vivone, M. E. Paoletti, G. Scarpa, J. He, Y. Zhang, J. Chanussot and A. Plaza, Machine learning in pansharpening: A benchmark, from shallow to deep networks, *IEEE Geoscience and Remote Sensing Magazine*, **10** (2022), 279–315.
[18] S.-Q. Deng, L.-J. Deng, X. Wu, R. Ran, D. Hong and G. Vivone, Psrt: Pyramid shuffle-and-reshuffle transformer for multispectral and hyperspectral image fusion, *IEEE Transactions on Geoscience and Remote Sensing*.
[19] W. Deng, Y. Chen, N. T. Yang, H. Du, Q. Feng and R. T. Chen, Reflected schrödinger bridge for constrained generative modeling, *arXiv preprint arXiv:2401.03228*.
[20] R. Dian and S. Li, Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization, *IEEE Transactions on Image Processing*, **28** (2019), 5135–5146.
[21] R. Dian, S. Li and 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.
[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in *International Conference on Learning Representations*, 2021.
[23] T. D. Frank, *Nonlinear Fokker-Planck equations: fundamentals and applications*, Springer Science & Business Media, 2005.
[24] A. Garzelli and F. Nencini, Hypercomplex quality assessment of multi/hyperspectral images, *IEEE Geoscience and Remote Sensing Letters*, **6** (2009), 662–665.
[25] A. Guo, R. Dian and S. Li, A deep framework for hyperspectral image fusion between different satellites, *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
[26] N. Gushchin, A. Kolesov, A. Korotin, D. Vetrov and E. Burnaev, Entropic neural optimal transport via diffusion processes, *arXiv preprint arXiv:2211.01156*.
[27] L. He, Y. Rao, J. Li, J. Chanussot, A. Plaza, J. Zhu and 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.
[28] X. He, L. Condat, J. M. Bioucas-Dias, J. Chanussot and J. Xia, A new pansharpening method based on spatial and spectral sparsity priors, *IEEE Transactions on Image Processing*, **23** (2014), 4160–4174.
[29] D. Hendrycks and K. Gimpel, Gaussian error linear units (gelus), *arXiv preprint arXiv:1606.08415*.

[30] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet et al., Imagen video: High definition video generation with diffusion models, *arXiv preprint arXiv:2210.02303*.

[31] J. Ho, A. Jain and P. Abbeel, Denoising diffusion probabilistic models, in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, 6840–6851.

[32] J. Ho and T. Salimans, Classifier-free diffusion guidance, *arXiv preprint arXiv:2207.12598*.

[33] E. Hopf, The partial differential equation ut + uux = μxx, *Communications on Pure and Applied Mathematics*, **3** (1950), 201–230.

[34] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, *arXiv preprint arXiv:1704.04861*.

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

[36] J.-F. Hu, T.-Z. Huang, L.-J. Deng, T.-X. Jiang, G. Vivone and 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.

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

[38] Z.-R. Jin, T.-J. Zhang, T.-X. Jiang, G. Vivone and L.-J. Deng, Lagconv: Local-context adaptive convolution kernels with global harmonic bias for pansharpening, *Proceedings of the AAAI Conference on Artificial Intelligence*, **36** (2022), 1113–1121.

[39] I. Karatzas, Brownian motion and stochastic calculus, *Elearn*.

[40] T. Karras, M. Aittala, T. Aila and S. Laine, Elucidating the design space of diffusion-based generative models, in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, 26565–26577.

[41] T. Karras, M. Aittala, J. Lehtinen, J. Hellsten, T. Aila and S. Laine, Analyzing and improving the training dynamics of diffusion models, *arXiv preprint arXiv:2312.02696*.

[42] B. Kawar, M. Elad, S. Ermon and J. Song, Denoising diffusion restoration models, in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, 23593–23606.

[43] A. Korotin, D. Selikhanovych and E. Burnaev, Neural optimal transport, in *The Eleventh International Conference on Learning Representations*, 2023.

[44] P. Kwarteng and A. Chavez, Extracting spectral contrast in landsat thematic mapper image data using selective principal component analysis, *Photogramm. Eng. Remote Sens*, **55** (1989), 339–348.

[45] C.-H. Lai, Y. Takida, N. Murata, T. Uesaka, Y. Mitsufuji and S. Ermon, Fp-diffusion: Improving score-based diffusion models by enforcing the underlying score fokker-planck equation, in *International Conference on Machine Learning*, PMLR, 2023, 18365–18398.

[46] S. Lee, B. Kim and J. C. Ye, Minimizing trajectory curvature of ode-based generative models, *arXiv preprint arXiv:2301.12003*.

[47] C. Léonard, A survey of the schrödinger problem and some of its connections with optimal transport, *arXiv preprint arXiv:1308.0215*.

[48] H. Li, Y. Yang, M. Chang, S. Chen, H. Feng, Z. Xu, Q. Li and Y. Chen, Srdiff: Single image super-resolution with diffusion probabilistic models, *Neurocomputing*, **479** (2022), 47–59.

[49] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel and M. Le, Flow matching for generative modeling, in *The Eleventh International Conference on Learning Representations*, 2023.

[50] G.-H. Liu, T. Chen, E. Theodorou and M. Tao, Mirror diffusion models for constrained and watermarked generation, *Advances in Neural Information Processing Systems*, **36**.

[51] 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.

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

[53] X. Liu, X. Zhang, J. Ma, J. Peng and qiang liu, Instaflow: One step is enough for high-quality diffusion-based text-to-image generation, in *The Twelfth International Conference on Learning Representations*, 2024.

[54] Y. Liu, K. Zhang, Y. Li, Z. Yan, C. Gao, R. Chen, Z. Yuan, Y. Huang, H. Sun, J. Gao, L. He and L. Sun, Sora: A review on background, technology, limitations, and opportunities of large vision models, *arXiv preprint arXiv:2402.17177*.

[55] S. Lolli, L. Alparone, A. Garzelli and G. Vivone, Haze correction for contrast-based multispectral pansharpening, *IEEE Geoscience and Remote Sensing Letters*, **14** (2017), 2255–2259.

[56] A. Lou and S. Ermon, Reflected diffusion models, *arXiv preprint arXiv:2304.04740*.

[57] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li and 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)*, vol. 35, 2022, 5775–5787.

[58] N. Ma, M. Goldstein, M. S. Albergo, N. M. Boffi, E. Vanden-Eijnden and S. Xie, Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers, *arXiv preprint arXiv:2401.08740*.

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

[60] Q. Meng, W. Shi, S. Li and L. Zhang, Pandiff: A novel pansharpening method based on denoising diffusion probabilistic model, *IEEE Transactions on Geoscience and Remote Sensing*, **61** (2023), 1–17.

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

[62] K. Neklyudov, D. Severo and A. Makhzani, Action matching: A variational method for learning stochastic dynamics from samples, *arXiv preprint arXiv:2210.06662*.

[63] E. Nelson, *Dynamical theories of Brownian motion*, vol. 106, Princeton university press, 2020.

[64] T. Q. Nguyen and J. Salazar, Transformers without tears: Improving the normalization of self-attention, *arXiv preprint arXiv:1910.05895*.

[65] A. Q. Nichol and P. Dhariwal, Improved denoising diffusion probabilistic models, in *International Conference on Machine Learning (ICML)*, PMLR, 2021, 8162–8171.

[66] X. Otazu, M. González-Audícana, O. Fors and 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.

[67] W. Peebles and S. Xie, Scalable diffusion models with transformers, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 4195–4205.

[68] E. Perez, F. Strub, H. De Vries, V. Dumoulin and A. Courville, Film: Visual reasoning with a general conditioning layer, in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[69] G. Peyré, M. Cuturi et al., Computational optimal transport: With applications to data science, *Foundations and Trends® in Machine Learning*, **11** (2019), 355–607.

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

[71] R. Rombach, A. Blattmann, D. Lorenz, P. Esser and 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, 10684–10695.

[72] R. Rombach, A. Blattmann, D. Lorenz, P. Esser and B. Ommer, High-resolution image synthesis with latent diffusion models, 2021.

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

[74] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet and M. Norouzi, Palette: Image-to-image diffusion models, in *ACM SIGGRAPH 2022 Conference Proceedings (ACM SIGGRAPH)*, 2022, 1–10.

[75] 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, *Advances in Neural Information Processing Systems*, **35** (2022), 36479–36494.

[76] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet and M. Norouzi, Image super-resolution via iterative refinement, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **45** (2022), 4713–4726.

[77] E. Schrödinger, Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique, *Annales de l'institut Henri Poincaré*, **2** (1932), 269–310.
[78] M. Selva, B. Aiazzi, F. Butera, L. Chiarantini and 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.
[79] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert and Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 1874–1883.
[80] Y. Shi, V. De Bortoli, A. Campbell and A. Doucet, Diffusion schrödinger bridge matching, *Advances in Neural Information Processing Systems (Neurips)*, **36**.
[81] M. Simoes, J. Bioucas-Dias, L. B. Almeida and J. Chanussot, A convex formulation for hyperspectral image superresolution via subspace-based regularization, *IEEE Transactions on Geoscience and Remote Sensing*, **53** (2014), 3373–3388.
[82] R. Sinkhorn, A relationship between arbitrary positive matrices and doubly stochastic matrices, *The annals of mathematical statistics*, **35** (1964), 876–879.
[83] V. R. Somnath, M. Pariset, Y.-P. Hsieh, M. R. Martinez, A. Krause and C. Bunne, Aligned diffusion schrödinger bridges, *arXiv preprint arXiv:2302.11419*.
[84] J. Song, C. Meng and S. Ermon, Denoising diffusion implicit models, in *International Conference on Learning Representations (ICLR)*, 2021.
[85] Y. Song and S. Ermon, Generative modeling by estimating gradients of the data distribution, *Advances in neural information processing systems*, **32**.
[86] Y. Song, L. Shen, L. Xing and S. Ermon, Solving inverse problems in medical imaging with score-based generative models, in *International Conference on Learning Representations*, 2022.
[87] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon and B. Poole, Score-based generative modeling through stochastic differential equations, in *International Conference on Learning Representations (ICLR)*, 2021.
[88] X. Tian, K. Li, W. Zhang, Z. Wang and J. Ma, Interpretable model-driven deep network for hyperspectral, multispectral, and panchromatic image fusion, *IEEE Transactions on Neural Networks and Learning Systems*, 1–14.
[89] A. Tong, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, K. Fatras, G. Wolf and Y. Bengio, Improving and generalizing flow-based generative models with minibatch optimal transport, in *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023.
[90] F. Vargas, P. Thodoroff, A. Lamacraft and N. Lawrence, Solving schrödinger bridges via maximum likelihood, *Entropy*, **23** (2021), 1134.
[91] C. Villani et al., *Optimal transport: old and new*, vol. 338, Springer, 2009.
[92] G. Vivone, Robust band-dependent spatial-detail approaches for panchromatic sharpening, *IEEE Transactions on Geoscience and Remote Sensing*, **57** (2019), 6421–6433.
[93] G. Vivone, R. Restaino and J. Chanussot, Full scale regression-based injection coefficients for panchromatic sharpening, *IEEE Transactions on Image Processing*, **27** (2018), 3418–3431.
[94] G. Vivone, R. Restaino, M. Dalla Mura, G. Licciardi and J. Chanussot, Contrast and error-based fusion schemes for multispectral image pansharpening, *IEEE Geoscience and Remote Sensing Letters*, **11** (2013), 930–934.
[95] L. Wald, *Data fusion: definitions and architectures: fusion of images of different spatial resolutions*, Presses des MINES, 2002.
[96] L. Wald, T. Ranchin and M. Mangolini, Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images, *Photogrammetric engineering and remote sensing*, **63** (1997), 691–699.
[97] G. Wang, Y. Jiao, Q. Xu, Y. Wang and C. Yang, Deep generative learning via schrödinger bridge, in *International Conference on Machine Learning*, PMLR, 2021, 10794–10804.
[98] T. Wang, F. Fang, F. Li and G. Zhang, High-quality bayesian pansharpening, *IEEE Transactions on Image Processing*, **28** (2018), 227–239.
[99] T. Wang, F. Fang, F. Li and G. Zhang, High-quality bayesian pansharpening, *IEEE Transactions on Image Processing*, **28** (2018), 227–239.
[100] C. Wu, D. Wang, Y. Bai, H. Mao, Y. Li and Q. Shen, Hsr-diff: hyperspectral image super-resolution via conditional diffusion models, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 7083–7093.

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

[102] Z.-C. Wu, T.-Z. Huang, L.-J. Deng, J. Huang, J. Chanussot and G. Vivone, Lrtcfpan: Low-rank tensor completion based framework for pansharpening, *IEEE Transactions on Image Processing*, **32** (2023), 1640–1655.

[103] M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon and J. Tang, Geodiff: A geometric diffusion model for molecular conformation generation, *arXiv preprint arXiv:2203.02923*.

[104] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding and J. Paisley, Pannet: A deep network architecture for pan-sharpening, in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, 1753–1761.

[105] L. Yang, S. Ding, Y. Cai, J. Yu, J. Wang and Y. Shi, Guidance with spherical gaussian constraint for conditional diffusion, *arXiv preprint arXiv:2402.03201*.

[106] N. Yokoya, T. Yairi and 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, 1779–1782.

[107] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng and S. Yan, Metaformer is actually what you need for vision, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, 10819–10829.

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

[109] R. H. Yuhas, A. F. Goetz and 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.

[110] X. Zhang, W. Huang, Q. Wang and 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.

[111] J. Zhou, D. L. Civco and J. A. Silander, A wavelet transform method to merge landsat tm and spot panchromatic data, *International Journal of Remote Sensing*, **19** (1998), 743–757.

[112] M. Zhou, X. Fu, J. Huang, F. Zhao, A. Liu and R. Wang, Effective pan-sharpening with transformer and invertible neural network, *IEEE Transactions on Geoscience and Remote Sensing*, **60** (2022), 1–15.

[113] M. Zhou, J. Huang, Y. Fang, X. Fu and A. Liu, Pan-sharpening with customized transformer and invertible neural network, in *AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[114] Z. Zhu, H. Zhao, H. He, Y. Zhong, S. Zhang, Y. Yu and W. Zhang, Diffusion models for reinforcement learning: A survey, *arXiv preprint arXiv:2311.01223*.

Appendix A. **Derivations of Eqs.** (53b) **and** (54b)**.** We first derive the curvature of SB ODE. Using the definition:

$$C(\mathbf{Y})_{\text{SB ODE}} = \mathbb{E}_{\mathbf{Y},\mathbf{t}}\left\| Y_1 - X_0 - \nabla_t \frac{\hat{\sigma}_t^2}{\hat{\sigma}_t^2+\sigma_t^2} X_0 - \nabla_t \frac{\sigma_t^2}{\hat{\sigma}_t^2+\sigma_t^2} Y_1 \right\|_2^2, \tag{55}$$

we can deduce the first derivative:

$$\nabla_t \frac{\hat{\sigma}_t^2}{\hat{\sigma}_t^2+\sigma_t^2} = \frac{-2\hat{\sigma}_t^2\beta_t - \hat{\sigma}_t^2\nabla_t(\sigma_t^2+\sigma_t^2)}{(\hat{\sigma}_t^2+\sigma_t^2)^2}, \text{ where } Y_1 \sim p_Y, X_0 \sim p_X \tag{56a}$$

$$= -2\beta_t \frac{\hat{\sigma}_t^2(\hat{\sigma}_t^2+\sigma_t^2) + \hat{\sigma}_t^2(-\hat{\sigma}_t+\sigma_t)}{(\hat{\sigma}_t^2+\sigma_t^2)^2} \tag{56b}$$

$$= -2\beta_t \frac{\hat{\sigma}_t\sigma_t(\sigma_t+\hat{\sigma}_t)}{(\hat{\sigma}_t^2+\sigma_t^2)^2} =: -M_t. \tag{56c}$$

Similarly, we can write the second derivate:

$$\nabla_t \frac{\sigma_t^2}{\hat{\sigma}_t^2+\sigma_t^2} = 2\beta_t \frac{\hat{\sigma}_t\sigma_t(\sigma_t+\hat{\sigma}_t)}{(\hat{\sigma}_t^2+\sigma_t^2)^2} = M_t. \tag{57a}$$

By substituting back in to (55), we can derive Eq. (53b).

The curvature derivations of VP ODE is provided below. Using the definition:

$$C(\mathbf{Y})_{\text{VP ODE}} \tag{58a}$$

$$= \mathbb{E}_{\mathbf{Y},\mathbf{t}}\left\| Y_1 - X_0 - \nabla_t(\alpha_t Y_0 + \beta_t Z) \right\|_2^2, \qquad \text{where } Y_1, Z \sim \mathcal{N}(0, \mathbf{I_d}) \tag{58b}$$

$$= \mathbb{E}_{\mathbf{Y},\mathbf{t}}\left\| Y_1 - X_0 - \nabla_t(\alpha_t) Y_0 + \nabla_t(\beta_t) Z) \right\|_2^2 \tag{58c}$$

$$= \mathbb{E}_{\mathbf{Y},\mathbf{t}}\Bigg\| Y_1 - X_0 - \underbrace{\frac{1}{8}[-2a(1-t)^2 - 4b(1-t)][a(1-t)+b]\alpha_t}_{\nabla_t(\alpha_t)=:N_t} Y_0 \tag{58d}$$

$$\underbrace{-\alpha_t(1-\alpha_t^2)^{-\frac{1}{2}}\nabla_t(\alpha_t)}_{\nabla_t(\beta_t):=K_t} Z \Bigg\|_2^2 \tag{58e}$$

$$= \mathbb{E}_{\mathbf{Y},\mathbf{t}}\left\| (1+K_t)Y_1 - (1+N_t)X_0 \right\|_2^2 \tag{58f}$$

$$= \mathbb{E}_{\mathbf{Y},\mathbf{t}}\left\| (1+N_t)X_0 \right\|_2^2 + \mathbb{E}_t\|1+K_t\|_2^2 \mathbf{I_d} \tag{58g}$$

Appendix B. **Proof of Eq.** (50)**.**

Proof. Recall the prior process $\mathbb{Q} \in \mathcal{F}(p_X, p_Y)$ and let the $\mathbb{P}$ be a probility measure that define a process to recover the process $\mathbb{Q}$. As f_t is set to 0 in our proposition, hence, $\pi_0^{Y|X}$ is a Gaussian distribution $\mathcal{N}(Y|X, \epsilon\mathbf{I_d})$, where ϵ is defined by g_t in Eq. (20a). The KL term in Eq. (50) can be writed as,

$$D_{KL}(\pi^{\mathbb{Q}}\|\pi^{\mathbb{P}}) = -\int_{\mathcal{X}\times\mathcal{Y}} \log \frac{d\pi^{\mathbb{Q}}(X,Y)}{d[X,Y]} d\pi^{\mathbb{P}}(X,Y) + \int_{\mathcal{X}\times\mathcal{Y}} \log \frac{d\pi^{\mathbb{P}(X,Y)}}{d[X,Y]} d\pi^{\mathbb{P}}(X,Y) \tag{59a}$$

$$= -\int_{\mathcal{X}\times\mathcal{Y}} \log \frac{d\pi^{\mathbb{Q}}(X,Y)}{d[X,Y]} d\pi^{\mathbb{P}}(X,Y) - H(\pi^{\mathbb{P}}), \tag{59b}$$

where $\frac{d\pi(X,Y)}{d[X,Y]}$ denotes the joint density of distribution π. The first term can be derive:

$$\int_{\mathcal{X}\times\mathcal{Y}} \log \frac{d\pi^{\mathbb{Q}}(X,Y)}{d[X,Y]} d\pi^{\mathbb{P}}(X,Y) \tag{60a}$$

$$= -\int_{\mathcal{X}\times\mathcal{Y}} \log \frac{d\pi^{\mathbb{Q}}(Y|X)}{dY} \frac{d\pi^{\mathbb{Q}}(X)}{dX} d\pi^{\mathbb{P}}(X,Y) \tag{60b}$$

$$= -\int_{\mathcal{X}\times\mathcal{Y}} \log \frac{d\pi^{\mathbb{Q}}(Y|X)}{dY} d\pi^{\mathbb{P}}(X,Y) - \int_{\mathcal{X}}\int_{\mathcal{Y}} \log \frac{d\pi^{\mathbb{Q}}(X)}{dX} d\pi^{\mathbb{P}}(Y|X) d\pi_0^{\mathbb{P}}(X) \tag{60c}$$

$$= -\int_{\mathcal{X}\times\mathcal{Y}} \log \frac{d\pi^{\mathbb{Q}}(Y|X)}{dY} d\pi^{\mathbb{P}}(X,Y) - \int_{\mathcal{X}} \log \frac{d\pi^{\mathbb{Q}}(X)}{dX} \left[\int_{\mathcal{Y}} 1 d\pi^{\mathbb{P}}(Y|X)\right] d\pi_0^{\mathbb{P}}(X) \tag{60d}$$

$$= -\int_{\mathcal{X}}\int_{\mathcal{Y}} \log \frac{d\pi^{\mathbb{Q}}(Y|X)}{dY} d\pi^{\mathbb{P}}(X,Y) - \int_{\mathcal{X}} \log \frac{d\pi^{\mathbb{Q}}(X)}{dX} d\pi_0^{\mathbb{P}}(X) \tag{60e}$$

$$= -\int_{\mathcal{X}}\int_{\mathcal{Y}} \log \frac{d\pi^{\mathbb{Q}}(Y|X)}{dY} d\pi^{\mathbb{P}}(X,Y) - \int_{\mathcal{X}} \log \frac{d\pi_0^{\mathbb{P}}(X)}{dX} d\pi_0^{\mathbb{P}}(X) \tag{60f}$$

$$= -\int_{\mathcal{X}\times\mathcal{Y}} \log \frac{d\pi^{\mathbb{Q}}(Y|X)}{dY} d\pi^{\mathbb{P}}(X,Y) + H(\pi_0^{\mathbb{P}}) \tag{60g}$$

$$= -\int_{\mathcal{X}\times\mathcal{Y}} \log \left((2\pi\epsilon)^{-D/2} \exp\left(\frac{\|X-Y\|^2}{2\epsilon}\right)\right) d\pi^{\mathbb{P}}(X,Y) + H(\pi_0^{\mathbb{P}}) \tag{60h}$$

$$= \frac{D}{2}\log(2\pi\epsilon) + \int_{\mathcal{X}\times\mathcal{Y}} \frac{\|X-Y\|^2}{2\epsilon} d\pi^{\mathbb{P}}(X,Y) + H(\pi_0^{\mathbb{P}}) \tag{60i}$$

We can substitude back to Eq. (59b) and further obtains,

$$D_{KL}(\pi^{\mathbb{Q}}\|\pi^{\mathbb{P}}) = \int_{\mathcal{X}\times\mathcal{Y}} \frac{\|X-Y\|^2}{2\epsilon} d\pi^{\mathbb{P}}(X,Y) - H(\pi^{\mathbb{P}}) + \underbrace{\frac{D}{2}\log(2\pi\epsilon) + H(\pi_0^{\mathbb{P}})}_{const \text{ in Eq. (50)}}. \tag{61}$$

□

Appendix C. **SB ODE Reduce the Transport Cost.** Based on the relation of SB problem with OT problem, we conclude a corollary which claims the SB ODE can reduce the transport cost.

Corollary C.1. *Given coupling (X_0, Y_1), for any convex cost function $c : \mathbb{R}^d \to \mathbb{R}$, we have,*

$$\mathbb{E}[c(Y_1 - Y_0)] \le \mathbb{E}[c(Y_1 - X_0)]. \tag{62}$$

Proof. The proof is based on Jensen's inequality.

$$\mathbb{E}[c(Y_1 - Y_0)] = \mathbb{E}\left[c\left(\int_0^1 [Y_1 - p(\hat{Y}_0|Y_{t+\Delta t}, t)]dt\right)\right] \tag{63a}$$

$$\le \mathbb{E}\left[\int_0^1 c\left(Y_1 - p(\hat{Y}_0|Y_{t+\Delta t}, t)\right)\right] \tag{63b}$$

$$= \mathbb{E}\left[\int_0^1 c\left(\mathbb{E}[(Y_1 - \hat{Y}_0)|Y_t]\right) dt\right] \tag{63c}$$

$$\le \mathbb{E}\left[\int_0^1 \mathbb{E}[c(Y_1 - X_0)|Y_t]dt\right] \tag{63d}$$

$$= \int_0^1 \mathbb{E}\left[c(Y_1 - X_0)|Y_t)dt\right] \tag{63e}$$

$$= \mathbb{E}[c(Y_1 - X_0)]. \tag{63f}$$

Eq. (63a) is based on the integral of the sampling trajectory. Eq. (63b) is derived by using the convexity of the transport function c and Jensen's inequality. Eq. (63c) is held based on the definition of $p(\hat{Y}_0|Y_{t+\Delta t,t})$. The inequality of Eq. (63d) comes from the same marginal with p and q processes. Then, Eq. (63e) is held due to $\mathbb{E}[\mathbb{E}[Y_1 - X_0|Y_t]] = \mathbb{E}[Y_1 - X_0]$. □

Appendix D. **Constraint Sampling.** During the sampling process, we need to discretize the time steps for sampling. Despite the nearly straight trajectories of the proposed SB SDE/ODE, small sampling time steps can still introduce sampling errors. When sampling errors accumulate on manifolds not covered during training, it can lead to out-of-distribution (OOD) issues. Recently, several DPM and SB-related studies have discussed how to perform score matching or SB matching in constraint spaces [19,50,56,105]. They choose to train and sample under constraints such as hypercubes or simplexes, which incur additional costs during training (*e.g.*, involving trajectory reflections or dual space projection). Here, we propose imposing gradient constraints during sampling, utilizing Y_1 and c (optional) to solve gradients depending on the setting of the inverse problem:

$$\tilde{X}_0 = \hat{X}_0 - \psi \nabla_{\hat{Y}_0} \left(\|A_1(\hat{X}_0) - Y_1\|_2^2 + \|A_2(\hat{X}_0) - c\|_2^2 \right) \tag{64}$$

In the context of pansharpening, Y_1 is the LRMS and c is PAN. Thus, we can simulate the degradation process by implementing A_1 to be the blur kernel and A_2 to be the downsampling operator. This approach pulls the sampled $\hat{Y}_0$ back to the correct manifold, reducing accumulated truncation errors during small sampling steps. The gradient sampling scheme is summarized in Algo. 3.

Appendix E. **High-order Sampler for SB SDE and SB ODE.** Due to the larger sampling steps leading to increased truncation errors and accumulated errors, using a first-order Euler method can result in decreased sampling performance. Recent work has explored the use of higher-order SDE/ODE sampling methods to reduce truncation errors. For example, the RK method in DPM-solver, the Huen method in EDM, and the PC method in SGM. These methods can all be applied to the proposed SB SDE, maintaining sampling performance even with fewer sampling steps. We provide pseudocode for the corresponding methods in SB SDE/ODE using the two-order Heun method in Algo. 4.

Algorithm 3: Constraint sampling scheme

Input: Sampling timesteps T, trained s_θ, degraded samples p_0, degraded samples p_Y, Conditional samples p_c (optional), ψ is the gradient constraint rate.

Output: Sampled data X_0.

$T_s = \texttt{linspace}(1, 0, T)$; ▷ Init timestep sequence.

$Y_1 \leftarrow p_Y, C \leftarrow p_c$;

for $t \leftarrow T_s$ **do**

  $\hat{X}_0 \leftarrow s_\theta(Y_t, C, t)$; ▷ Predicted endpoint.

  ▷ Gradient constraint (64).

  $\tilde{X}_0 \leftarrow \hat{X}_0 - \psi \nabla_{\hat{Y}_0} \left(\|A_1(\hat{X}_0) - Y_1\|_2^2 + \|A_2(\hat{X}_0) - c\|_2^2 \right)$;

  $Y_t \leftarrow p(Y_t | \tilde{X}_0, Y_{t+\Delta t})$; ▷ Posterior sampling (31a).

end

$X_0 \leftarrow Y_0$.

Algorithm 4: Sampling scheme using Heun method

Input: Sampling timesteps T, trained s_θ, degraded samples p_0, degraded samples p_Y, Conditional samples p_c (optional).

Output: Sampled data X_0.

$T_s = \texttt{linspace}(1, 0, T)$; ▷ Init timestep sequence.

$Y_1 \leftarrow p_Y, C \leftarrow p_c$; $\Delta t \leftarrow 1/T$;

for $t \leftarrow T_s$ **do**

  $\hat{X}_0^{(1)} \leftarrow s_\theta(Y_t, C, t)$; ▷ Predicted endpoint.

  if $t \neq 0$ **then**

    $t' \leftarrow t + \Delta t$; ▷ Select a temporary timestep.

    $Y_{t'} \leftarrow q(Y_{t'} | \hat{X}_0^{(1)}, t')$; ▷ Move towards Y_1.

    $\hat{X}_0^{(2)} \leftarrow s_\theta(Y_{t'}, C, t')$; ▷ Predicted endpoint $Y_{t'}$.

    $Y_t \leftarrow p(Y_t | \frac{\hat{X}_0^{(1)} + \hat{X}_0^{(2)}}{2}, Y_{t+\Delta t})$;

  else

    $Y_t \leftarrow p(Y_t | \hat{X}_0^{(1)}, Y_{t+\Delta t})$;

  end

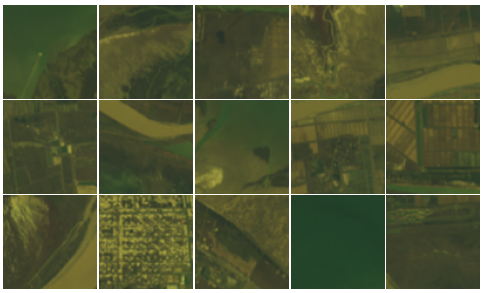
end

$X_0 \leftarrow Y_0$.

Appendix F. **More Visual Results.** In this section, we provide more visual results of WV3 and GF2 test set as shown in Figs. 10a and 10b.

Appendix G. **Different Parameterizations of f_t and g_t.** In Theroem 3.4, we set $f_t = 0$ and $g_t = \frac{d\sigma_t^2}{dt} = \sqrt{\beta_t}$ as a special parameterization for the SB SDE. In this section, we provide more parameterization for the bridge matching. Inspired by VP SDE [31] and sub-VP-SDE [87], we can derive (sub-)VP-like SB SDEs. We leave them as the feature work.

(A) WV3 reduced test set. (B) GF2 reduced test set.



(C) GF5-GF1 reduced test set.

FIGURE 10. Pansharpening results on WV3, GF2, and GF5-GF1 reduced test sets.

TABLE 5. Different parameterization of SB SDE of f_t and g_t.

Param.	f_t	g_t^2	$q(Y_t\|X_0, Y_1)$
VP	$\frac{\log \alpha_t}{dt} Y_t$	$\frac{d\sigma_t^2}{dt} - 2\frac{d \log \alpha_t}{dt}\sigma_t^2$	$\mathcal{N}(\alpha_t X_0 + \sigma_t Y_1, \sigma_t^2 \mathbf{I_d})$
sub-VP	$\frac{\log \alpha_t}{dt} Y_t$	$-2\frac{d \log \alpha_t}{dt}(1+\alpha_t^2)\sigma_t$	$N(\alpha_t X_0 + \sigma_t Y_1, \sigma_t^2 \mathbf{I_d})$
Default	0	$\sqrt{\beta_t}$	$\mathcal{N}(\frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2+\sigma_t^2} X_0 + \frac{\sigma_t^2}{\bar{\sigma}_t^2+\sigma_t^2} Y_1, \frac{\sigma_t^2 \bar{\sigma}_t^2}{\bar{\sigma}_t^2+\sigma_t^2} \mathbf{I}_d)$

Received xxxx 20xx; revised xxxx 20xx; early access xxxx 20xx.