Research paper

Deep diffusion models for seismic processing

Ricard Durall [a],[*], Ammar Ghanim [a], Mario Ruben Fernandez [a],[b], Norman Ettrich [a], Janis Keuper [a],[c]

[a] *Fraunhofer ITWM, Germany*
[b] *École Normale Supérieure, France*
[c] *IMLA, Offenburg University, Germany*

ARTICLE INFO

ABSTRACT

Seismic data processing involves techniques to deal with undesired effects that occur during acquisition and pre-processing. These effects mainly comprise coherent artefacts such as multiples, non-coherent signals such as electrical noise, and loss of signal information at the receivers that leads to incomplete traces. In the past years, there has been a remarkable increase of machine-learning-based solutions that have addressed the aforementioned issues. In particular, deep-learning practitioners have usually relied on heavily fine-tuned, customized discriminative algorithms. Although, these methods can provide solid results, they seem to lack semantic understanding of the provided data. Motivated by this limitation, in this work, we employ a generative solution, as it can explicitly model complex data distributions and hence, yield to a better decision-making process. In particular, we introduce diffusion models for three seismic applications: demultiple, denoising and interpolation. To that end, we run experiments on synthetic and on real data, and we compare the diffusion performance with standardized algorithms. We believe that our pioneer study not only demonstrates the capability of diffusion models, but also opens the door to future research to integrate generative models in seismic workflows.

1. Introduction

Deep generative learning has become an important research area in the machine learning community, being more relevant in many applications. Namely, they are widely used for image synthesis and various image-processing tasks such as editing, interpolation, colorization, denoising, and super-resolution. Recently, diffusion probabilistic models (Sohl-Dickstein et al., 2015; Ho et al., 2020) have emerged as a novel, powerful class of generative learning methods. In a short period of time, these models have achieved surprisingly high performance (Dhariwal and Nichol, 2021; Saharia et al., 2021a; Ramesh et al., 2022; Ruiz et al., 2022), and have even surpassed state-of-the-art algorithms like generative adversarial networks (Goodfellow et al., 2014) (GANs) and variational autoencoders (Kingma and Welling, 2013) (VAEs).

At the same time, the geophysics community has been actively adopting deep-learning techniques to boost and automate numerous seismic interpretation tasks including fault picking (An et al., 2021; Wu et al., 2019), salt delineation (Oh et al., 2018; Shi et al., 2019), well-to-seismic tie (Nivlet et al., 2020; Tschannen et al., 2022), horizon tracking (Yang and Sun, 2020; Tschannen et al., 2020), multiple removal (Bugge et al., 2021; Durall et al., 2022), etc. Nonetheless, to the best of our knowledge, there has not been yet any work exploring the

application of diffusion models to seismic data and thus, studying their potential advantages to already established deep-learning approaches in this domain. Driven by this motivation, in this work, we study the applicability of diffusion models for seismic processing.

Seismic imaging is essential to discover and characterize economically worthwhile geological reservoirs, such as hydrocarbons accumulations, and to manage the extraction of the resources stored in them. Unfortunately, recorded seismic signals at the surface are inevitably contaminated by coherent and incoherent noise of various nature. The process of removing the noise, while retaining the primary signal, is called seismic processing. In this paper, we focus on three relevant, well-known seismic processing tasks: demultiple, denoising and interpolation. Demultiple and denoising are both removing unwanted signals from the seismic section; the first gets rid of coherent noise caused by reverberations of waves between strong reflectors, whereas the latter removes incoherent noise of miscellaneous causes. The goal of interpolation is to fill-in gaps in the image caused by limitations during acquisition. Although at the first glance the nature of these problems might look different or unrelated, it is possible to formulate a common framework, in which they can be solved. This is feasible, due to the fact that the diffusion models, like most of generative models, learn the density distribution of the input data. In other words, unlike

---

* Corresponding author.
  E-mail address: ricard.durall.lopez@itwm.fraunhofer.de (R. Durall).

discriminative approaches which draw boundaries in the data space, the generative approaches model how data is placed throughout the space (Tomczak, 2022). As a result, they are powerful algorithms that can be independently applied to a large diversity of problems.

2. Motivation

In this work, we have decided to implement a solution based on diffusion models to try to exploit the intrinsic properties that these generative models bring along. Among these properties, it should be noted the powerful capacity of learning any kind of data distribution. As a result, these models are able to synthesize new samples with high diversity, quality and accuracy, independently of the application domain, for example medical domain Kazerouni et al. (2022a) and art domain Ramesh et al. (2022), among others. For this reason, such a technology can have a beneficial impact in all sorts of computer vision tasks that require image manipulation. In comparison with previous generation techniques such as GANs, which suffer from mode collapse (Metz et al., 2016; Arjovsky et al., 2017; Durall et al., 2020a), diffusion models do not have this struggle, i.e., can easily learn multimodal distributions; they have also more stable training since no adversarial formulation is involved, i.e., the model directly optimizes the likelihood function (see Section 3). Recent seminal work (Lugmayr et al., 2022; Rombach et al., 2022) has shown state-of-the-art results on image translation tasks, including domain transfer and image inpainting. Inspired by the improvement that these implementations have brought to the computer vision domain, we propose to bring a deep diffusion-based approach to the geophysics community. To the extent of our knowledge, this is the first study that applies diffusion models to seismic processing tasks. In particular, we focus on demultiple, denoising and interpolation tasks, as they can be naturally formulated as image-to-image translation problems, where the input images lay in one domain, for example with multiples, and the outputs in another, for example without multiples. Furthermore, since deep diffusion models' optimization is fairly robust and stable, no need for dull fine-tuning is required, and hence resulting in universal models capable of training for different tasks without significant changes. In practice, this means redefinition of the seismic processing task at hand only involves substitution of the training and testing data. All in all, diffusion models have the potential to be a game-changer, helping to boost the performance of currently utilized deep-learning approaches for seismic processing in a user-friendly manner.

3. Background

Generative models for modeling estimate the marginal distribution, denoted as $p(x)$, over observable variables x, e.g., images. In the literature, we can find different formulations that tackle this problem such as autoregressive generative models, latent variable models, flow-based models, and energy-based models.

3.1. Latent variable models

The main idea of this type of models is to utilize latent variables z to formulate the joint distribution $p(x, z)$, which describes the marginal distribution as a function of learnable parameters θ (likelihood). Mathematically, it can be written as:

$$z \sim p_\theta(z) \tag{1}$$

$$x \sim p_\theta(x|z) \tag{2}$$

$$p_\theta(x) = \int_z p_\theta(x, z) dz = \int_z p_\theta(x|z) p_\theta(z) dz. \tag{3}$$

Unfortunately, for most of the problems we do not have access to the true distribution $p(x)$ and hence, we need to fit our model to
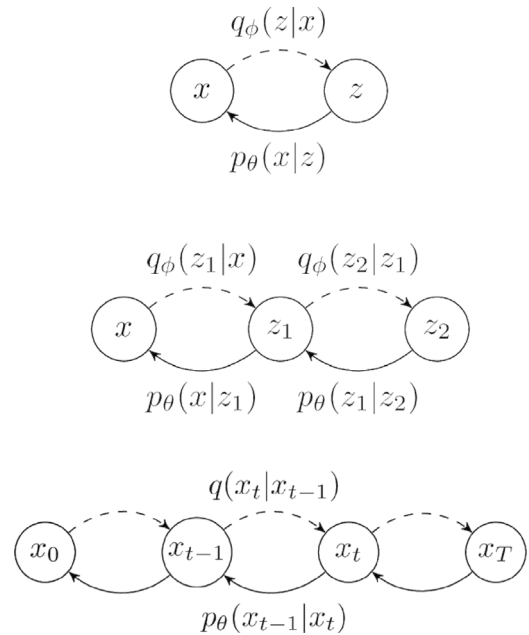

Fig. 1. Scheme of the different latent variable models. (Top) Single latent variable model. (Center) Hierarchical latent variable model. (Bottom) Diffusion model.

some empirically observed subset. One solution is to use Monte Carlo sampling to approximate the integral over z to try to estimate the model parameters θ. Nonetheless, this approach does not scale to high dimensions of z and consequently, we will suffer from issues associated with the curse of dimensionality. Another solution is to use variational inference, e.g., VAE (Kingma and Welling, 2013). In particular, the lower bound of the log-likelihood function, called the Evidence Lower BOund (ELBO). The ELBO provides a joint optimization objective, which simultaneously updates the variational posterior $q_\phi(z|x)$ and likelihood model $p_\theta(x|z)$. The objective is written as:

$$\log p(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - \text{KL}[q_\phi(z|x) \| p(z)], \tag{4}$$

where KL stands for the Kullback–Leibler divergence (see Fig. 1).

3.2. Hierarchical latent variable models

Once defined a single stochastic layer, it is straightforward to derive hierarchical extensions. For example, let us consider a latent variable model with two latent variables z_1 and z_2. We can define the joint distribution $p(x, z_1, z_2)$ and marginalizing out the latent variables:

$$p_\theta(x) = \int_{z_1} \int_{z_2} p_\theta(x, z_1, z_2) dz_2 dz_1 = \int_{z_1} \int_{z_2} p_\theta(x|z_1) p_\theta(z_1|z_2) p_\theta(z_2) dz_2 dz_1. \tag{5}$$

Similar to the single latent model, we can derive the variational approximation (ELBO) to the true posterior as:

$$\log p(x) \geq \mathbb{E}_{z_1 \sim q_\phi(z_1|z_2)}[\log p_\theta(x|z_1)]$$
$$- \text{KL}[q_\phi(z_1|x) \| p_\theta(z_1|x)] - \text{KL}[q_\phi(z_2|z_1) \| p(z_2)]. \tag{6}$$

3.3. Diffusion models

Diffusion models belong to the latent variable family as well. In fact, we can think of them as a specific realization of a hierarchical latent variable model, where the inference model[1] does not have learnable parameters. Instead, it is constructed so that the final latent distribution

---

[1] Remember that the inference model relates a set of observable variables to a set of latent variables, e.g., $q(z|x)$.
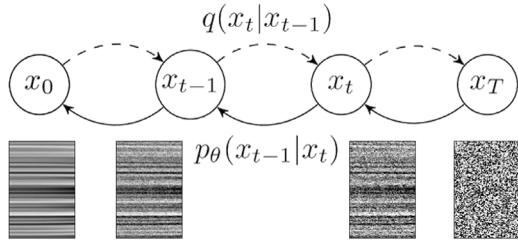
Fig. 2. Denoising diffusion process. While the Markov chain of the forward diffusion gradually adds noise to the input (dash arrows), the reverse process removes it stepwise (solid arrows).

Fig. 3. In each reverse step t, the model f_θ is fed with the semi-denoised multiple-free image x_t and the multiple-infested input. As an output, the network generates the image x_{t-1}, which should have less noise and no multiples.

$q(x_T)$ converges to a standard Gaussian (where T is the number of latent variables). The objective function of diffusion models is written as:

$$\log p(x) \geq \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)}[\text{KL}(q(x_T|x_0) \| p_\theta(x_T))$$
$$+ \sum_{t=2}^{T} \text{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1)]. \quad (7)$$

Under certain assumptions, this objective can be further simplified, leading to the following approximation:

$$\log p(x) \gtrsim \sum_{t=2}^{T} \text{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))$$
$$= \sum_{t=2}^{T} \| \epsilon - f_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) \|^2. \quad (8)$$

Note that we drop the expectation for clarity.

4. Methodology

In this section, we provide a brief overview of diffusion models framework. Note that we do not aim at covering the entire derivations. For a more in-depth, detailed mathematical description, we refer the reader to Ho et al. (2020).
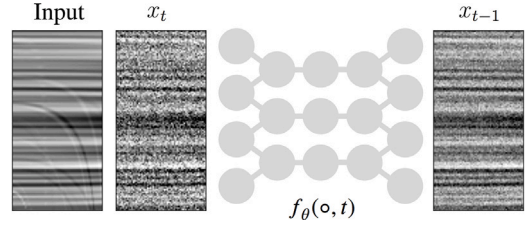
4.1. Framework

On a high level, diffusion models consist of two parts: forward diffusion and parametrized reverse. The forward diffusion part can be described as a process, where Gaussian noise ϵ is gradually applied to the input image x_0 until the image becomes entirely unrecognizable from a normal distribution $x_T \sim \mathcal{N}(0, \text{I})$ (T is the number of transformation steps). That is to say, at each step of this process, the noise is incrementally added to the data, $x_0 \xrightarrow{+\epsilon} x_1 \xrightarrow{+\epsilon} ... \xrightarrow{+\epsilon} x_T$. This procedure together with the Markov assumption[2] leads to a simple parameterization forward process expressed as:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) = \prod_{t=1}^{T} \mathcal{N}(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t \text{I}), \quad (9)$$

where the variable β defines a fixed variance schedule, chosen such that $q(x_T|x_0) \approx \mathcal{N}(0, \text{I})$ (see Fig. 2).

The second part, the parametrized reverse process, represents the data synthesis. Thus, it undoes the forward diffusion process and performs iterative denoising. To that end, the reverse process is trained to generate data by converting random noise into realistic data. Formally, this generative process is defined as a stochastic process, which iteratively removes noise from the input images using deep neural networks

f_θ. Starting with the pure Gaussian noise $p(x_T) = \mathcal{N}(x_T, 0, \text{I})$, the model learns the joint distribution $p_\theta(x_{0:T})$ as:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) = p(x_T) \prod_{t=1}^{T} \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)),$$
$$(10)$$

where the time-dependent parameters of the Gaussian transformations θ are learned. Note in particular that the Markov formulation asserts that a given reverse diffusion transformation distribution depends only on the previous timestep.

4.2. Training and inference

A diffusion model is trained by finding the reverse Markov transitions that maximize the likelihood of the training data. In practice, this process consists of optimizing the variational lower bound on the log likelihood. Hereunder the simplified expression derived by Ho et al. (2020):

$$\log p(x) \gtrsim \sum_{t=2}^{T} \| \epsilon - f_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) \|^2, \quad (11)$$

where

$$\alpha_t = 1 - \beta_t \quad \text{and} \quad \bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i. \quad (12)$$

Notice that, ultimately, the deep neural network learns to predict the noise component ϵ at any given timestep and therefore, at inference time (sampling) the algorithm will remove the noise in a step-wise fashion. Algorithm 1 describes the training procedure, and Algorithm 2 the inference.

Algorithm 1 Training the diffusion model f_θ.

1: **Input:** T, number of latent variables; m, training iterations.
   $x_0 \sim p_\theta(x_0)$
   $t \sim \text{Uniform}(\{1, ..., T\})$
2: $\epsilon \sim \mathcal{N}(0, \text{I})$
3: **for** t = 1,...,m **do**
4: $\quad \nabla_\theta \| \epsilon - f_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) \|^2$
5: **end for**

Algorithm 2 Inference of the diffusion model f_θ.

1: **Input:** T: number of latent variables.
   $x_T \sim \mathcal{N}(0, \text{I})$
2: **for** t = T,...,1 **do**
3: $\quad z \sim \mathcal{N}(0, \text{I})$ if $t > 1$, else $z = 0$
4: $\quad x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} f_\theta(x_t, t)) + \sqrt{1-\alpha_t} z$
5: **end for**
6: **return** x_0

---

[2] Markov assumption is used to describe a model that holds the memoryless property of a stochastic process.
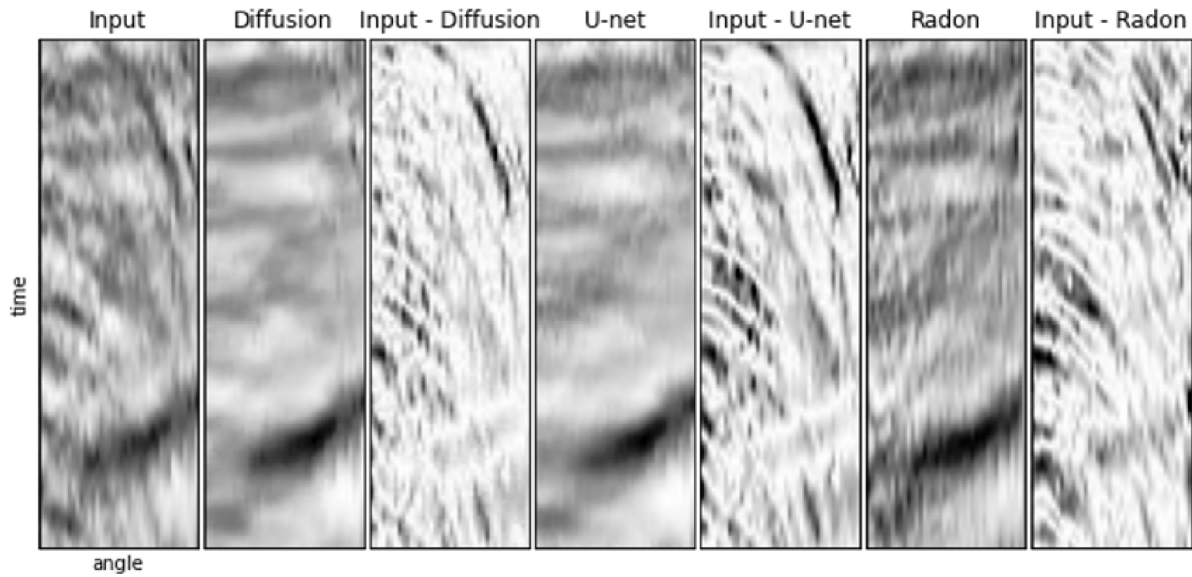
Fig. 4. This figure displays two cropped gathers that contain multiples (input), and the results after applying the demultiple algorithms. Moreover, we plot the difference between the input and the output to check the content that has been removed. Note that we apply a scaling factor of 3 in the differences to stress the changes.
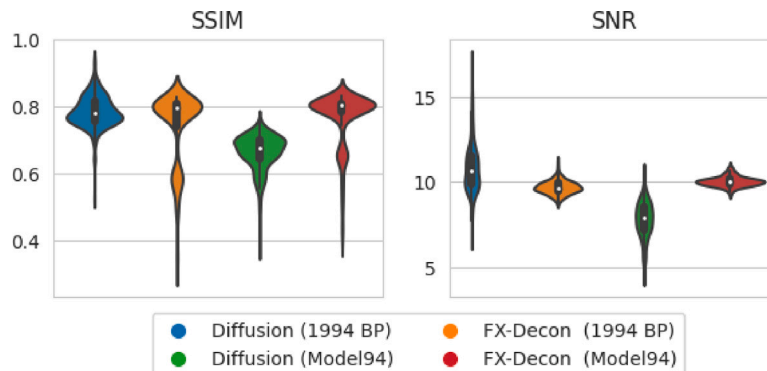

Fig. 5. Visualization (violin plot overlaid a box plot) of the SSIM and SNR metrics calculated on 500 random denoised images. Results from diffusion and FX-Decon on two datasets.

4.3. Conditional diffusion models

Generative models can be conditioned so that they also take advantage of labels during the training process. As a result, the outputs can be better controlled and can be steered to follow predetermined conditions, e.g., Mirza and Osindero (2014), Sohn et al. (2015), Saharia et al. (2021b). To achieve that, given a dataset of source–target image pairs that represent samples drawn from an unknown conditional distribution $p(x|y)$, we will learn a parametric approximation to $p(x|y)$ that maps between both domains. In practice, this means that our network will take an extra input y (label), that will condition the outcomes, i.e., $f_\theta(x_t|y, t)$.

5. Experiments

In this section, we validate the flexibility of diffusion models for different seismic tasks. In particular, we analyze three case studies: demultiple, denoising and interpolation. To do that, we present an end-to-end deep-learning approach that can deal (separately) with these scenarios. Furthermore, we benchmark the results with alternative paradigms that are currently employed in both academia and industry domains.

The implementation details are as following: In our main experiments, we train the diffusion model for 200,000 iterations with a batch size of 32; we set β to follow a linear schedule, and we use a depth T of 2000 timesteps for both the forward process (see Eq. (9)) and the reverse denoising process (see Eq. (10)).

5.1. Architecture

Image diffusion models commonly employ a time-conditional U-net (Ronneberger et al., 2015), parametrized as $f_\theta(\circ, t)$, as a neural backbone. This architecture was initially introduced in Ho et al. (2020), where the main motivation for this topology choice was the requirement for the model to have identical input and output dimensionality. The architecture consists of a stack of residual layers and downsampling convolutions, followed by a stack of residual layers with upsampling convolutions; skip connections connect the layers with the same spatial size. Furthermore, it uses a global attention layer with a single head to add a projection of the timestep embedding into each residual block.

5.2. Demultiple

Primary seismic reflections are events which have reflected only once, and they are employed to describe the subsurface interfaces. Multiples, on the contrary, are events which appear when the signal has not taken a direct path from the source to the receiver after reflecting on a subsurface boundary. The presence of multiples in a recorded

dataset can trigger erroneous interpretations, since they do not only interfere with the analysis in the post-stack domain, e.g., stratigraphic interpretation, but also with the analysis in the pre-stack domain, e.g., amplitude variation with offset inversion. Thereby, the demultiple process plays a crucial role in any seismic processing workflow.

In this first experiment, we follow the approach from Breuer et al. (2020), Durall et al. (2022), and generate synthetic pairs of multiple-infested and multiple-free gathers. This data setup allows us to train the model in a supervised manner and therefore, we can frame the demultiple problem as an image-to-image transformation task, where the network learns to remove the multiples without removing primary energy. As in Durall et al. (2022), the training dataset is designed to include a rich amount of features present in real datasets, to maximize transferability to real case uses. To that end, we employ as a baseline a conditional diffusion model proposed by Saharia et al. (2021b). More specifically, we condition our model by concatenating the semi-denoised multiple-free image x_t with the multiple-infested input (y) (see Fig. 3). Ideally, the network should return an improved semi-denoised multiple-free gather x_{t-1} that after T reverse steps should converge into a noise- and multiple-free gather x_0.

Once the model is trained, it is crucial to assess the inference capabilities of the network when working on real data, i.e., generalizability. Nonetheless, this is not a granted property in deep-learning models due to the distribution gap between different datasets, e.g., the gap between synthetic and real datasets (Durall et al., 2020b). In our experiments, we test the diffusion approach on the dataset from the Volve field made available under Equinor Open Data Licence. Furthermore, we compare the outcomes with two other multiple-attenuation methodologies: one based on Radon-transform (Hampson, 1986) and one based on deep learning (Durall et al., 2022). Fig. 4 shows an example of such a comparison, where we can observe how the diffusion solution offers competitive results, despite minimal hyperparameter tuning involved. For additional results, see Fig. 10 in Appendix.

5.3. Denoising

Incoherent noise can be caused by superposition of numerous unwanted signals from various sources such as ocean waves, wind and electrical instrument noise among others. Removing such incoherent noise can improve the overall signal-to-noise ratio and, consequently, increase the certainty of interpretation. Traditional approaches can be subdivided into two main categories: the prediction filtering methods and domain transform methods. The first type assumes linearity and predictability of the signal, and constructs a predictive filter to suppress the noise (Gulunay, 1986; Galbraith, 1991). These methods have been widely adopted by the industry due to their efficiency, although they tend to under-suppress noise and occasionally suffer from signal leakage (Gülünay, 2017). The second type of methods uses mathematical transformations, e.g., Fourier transform (Naghizadeh, 2012), wavelet transform (Mousavi et al., 2016), curvelet transform (Neelamani et al., 2008; Herrmann et al., 2008), to steer the seismic data into domains, where seismic signals and noise can be easier separated and then leverage the sparse characteristics of seismic data. This approach, however, often requires a time-consuming transform coefficient tuning. To cope with this drawback, a new trend based on deep-learning algorithms has emerged, resulting in optimized solutions that remove incoherent noise from seismic data as well as speed up the inference time (Yu et al., 2019; Saad and Chen, 2020).

Similar to the demultiple scenario, we create pairs of images to train our diffusion model. Nonetheless, this time, the objective is to eliminate undesired uncorrelated noise, while preserving the inherent characteristics of the data. To that end, the pairs of training data consist of a real image and their noisy version. To create the noisy images, we synthetically add Gaussian noise to the original real images with a variability of the 50% of their energy. For this second case of study, we train on 1994 BP (Gray and Marfurt, 1995) dataset, from which
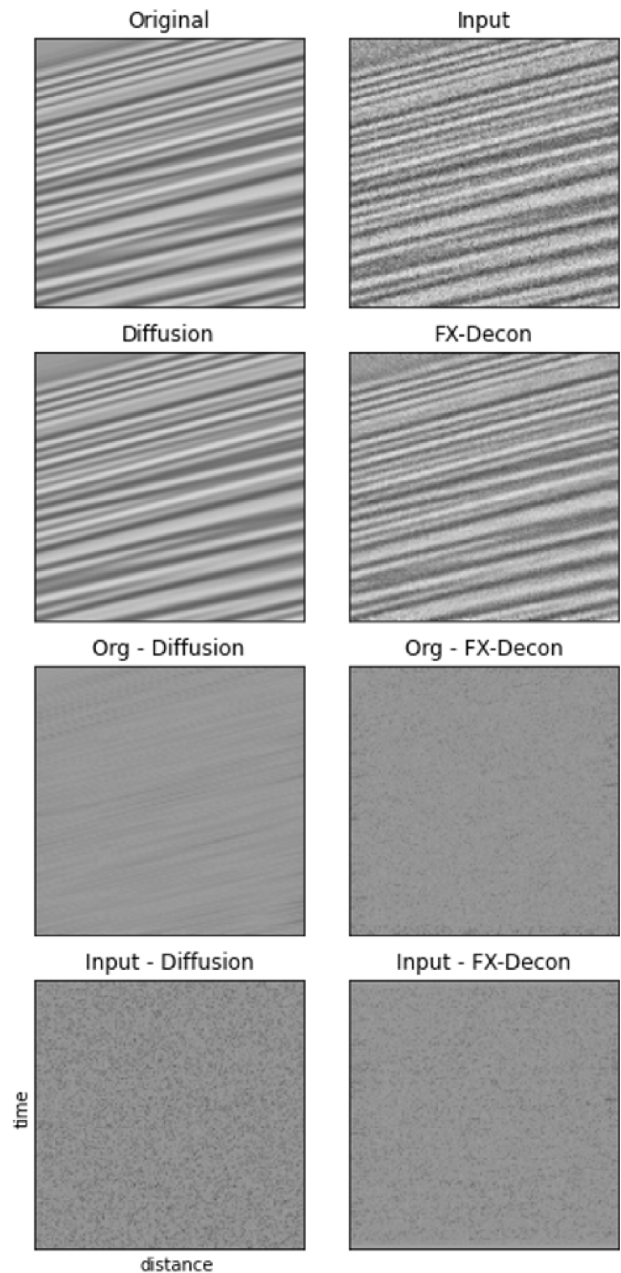

Fig. 6. This figure shows an example of denoising. The first row contains the original image and the input image (original with noise). The second row presents the diffusion and the FX-Decon results. Finally, the third and four rows display the difference between the results and the original and the input data, respectively.

we extract random patches (from different shot gathers) that neither overlap among each other, nor have more than 40% of their content equal to 0. In this fashion, we try to guarantee certain level of variety in the training data.

For the testing set, we apply the same conditions as for training. Additionally, we employ a second dataset (Model94 (Gray and Marfurt, 1995)) to evaluate the generalization capacity of our system. As for comparison, we use a spectral filtering technique based on the Fourier transform, namely a complex Wiener prediction filter called FX-Decon (Gulunay, 1986; Galbraith, 1991), which is dedicated for signal extraction and non-coherent noise suppression in the frequency domain. To assess the results, we use structural similarity index (SSIM) and signal-to-noise ratio (SNR) as quantitative metrics. Fig. 5 displays
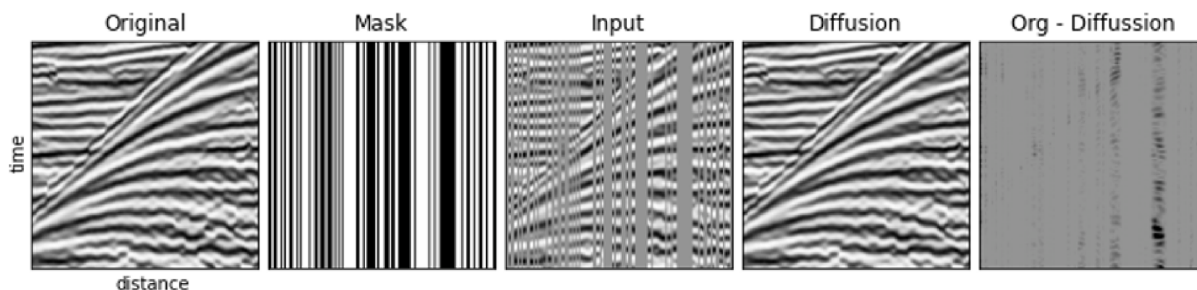
Fig. 7. This figure shows an example of interpolation. From left to right: the original image, the mask, the input image (original with mask), the diffusion result and its difference with respect to the original image. Note that we rescale the differences to stress the changes.
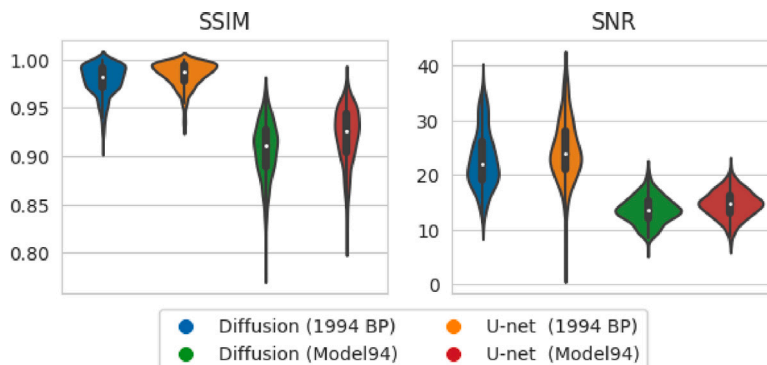

Fig. 8. Visualization (violin plot overlaid a box plot) of the SSIM and SNR metrics calculated on 500 random interpolated images. Results from diffusion and U-net on two datasets.

them for each configuration, i.e., different datasets and methods, and we can observe how the diffusion model provides the best scores when we test on data coming from the same dataset as the one used for training. However, as expected, it has a drastic drop when we test on a new dataset, e.g., Model94. This phenomenon is mainly caused by the distribution gap between different datasets. On the other hand, FX-Decon achieves similar performance on both datasets, as this method does not involve any learning, i.e., data fitting. Finally, Fig. 6 illustrates a denoising example for both algorithms. The difference between the outputs and the original data (third row in Fig. 6) allows us to see that diffusion model removes some coherent signal, while FX-Decon does not. Ideally, this should be corrected, but we leave this improvement for future work. Nevertheless, overall, the diffusion approach leads to less noisy outputs, as can be noticed in the output image. For additional results, see Fig. 11 in Appendix.

5.4. Interpolation

Seismic data processing algorithms greatly benefit from regularly sampled and reliable data. However, it is rarely the case where the acquired data is presented flawless, i.e., complete shot gathers without missing traces. Frequently, the reason for that are acquisition constraints such as geophones issues, topography, and economical limitations. As a consequence, interpolation techniques are a fundamental key for most seismic processing systems.

In this last case of study, we evaluate the capacity of our diffusion model to interpolate missing traces. To that end, we follow the evaluation methodology introduced by Fernandez et al. (2022), namely, we consider the scenario with irregular missing traces and with a level of decimation set to 50% (see Fig. 7). Regarding the data for this experiment, we repeat the setup presented in the denoising section, using 1994 BP dataset for training and testing, and Model94 for testing on a new dataset. Finally, to have a baseline to compare with, we implement the so-called "standard" topology from Fernandez et al. (2022), which is essentially a discriminative U-net-like network.

Fig. 8 shows the qualitative evaluation of the diffusion approach and of the U-net baseline. Although results from the latter are superior, the improvement could be considered marginal given the small metric differences. Moreover, specially in big gap scenarios, generative models can provide a wider range of alternatives. This ill-posed problem can be well handled by models that have learnt the inherent characteristics of the training data (i.e. data distribution), such as diffusion models, which can infill the gaps with more plausible solutions. Furthermore, both algorithms seem to struggle when inferring on unseen datasets. On the other hand, besides the quantitative results, the potential that diffusion models might bring is objectively higher than discriminative models. This is due to the fact that the former are generative models and, therefore, can capture more advanced data properties (Tomczak, 2022). For additional results, see Fig. 12 in Appendix.

5.5. Hyperparameter search

Hyperparameters are parameters that are set before the learning process begins. They are tunable and can directly affect how a model trains. In general, neural networks are notorious for being very sensitive to the choice of hyperparameters, such as optimizer configuration and loss function, resulting in fairly different outcomes when the parameters are slightly modified. In the case of diffusion models there are mainly two additional hyperparameters: the depth of the diffusion process T and the schedules β. While the depth T has a direct impact on the inference time, the schedules β can help to boost performance (Nichol and Dhariwal, 2021). Therefore, in this subsection, we investigate the empirical effects that these hyperparameters have on the diffusion model, testing different T (2000, 1000 and 100)[3] and β (linear and quadratic).

Fig. 9 displays the results for denoising (left) and for interpolation (right). We can observe how the schedules play a secondary role, as the metrics do not excessively fluctuate when moving from linear

---

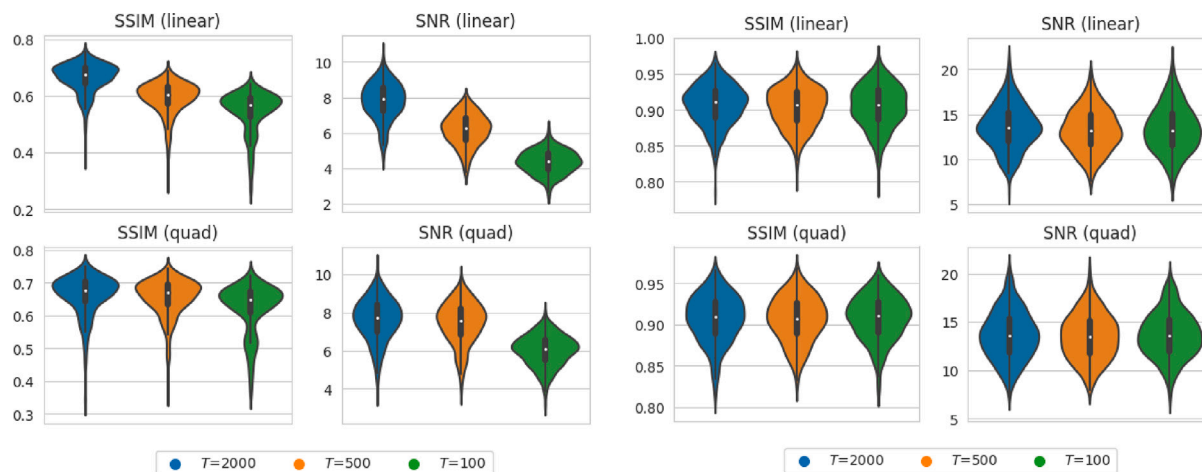[3] We also tested $T = 10$ but the model did not converge.

Fig. 9. Visualization (violin plot overlaid a box plot) of the SSIM and SNR metrics calculated on 500 random interpolated images. Results from different amount of timesteps *T* and schedules (linear and quadratic) on Model94. (Left): Denoising scenario. (Right): Interpolation scenario.

to quadratic schedules. On the other hand, the depth of timestep does have a major impact. As expected, the shallower the diffusion procedure is, the less accurate the reverse process becomes, and the less reliable results we can achieve, i.e., the performances gradually drop. However, depending on the nature of the problem, the number of timesteps might vary and thus, having more steps may not lead to significant improvements, i.e., the interpolation case. We hypothesize the main reason for that it is the complexity of the problem. While the interpolation task only needs to fit the missing gaps, the denoising has to deal with the whole input. For this reason, it is important to profile our task and system, to understand which needs they might have, and to choose the hyperparameters accordingly.

6. Discussion

In this work, we propose a generative framework based on diffusion models to address several seismic tasks. In particular, our case studies include demultiple, denoising and interpolation. To solve them, we define the problem as an image-to-image transformation, where we have an input image that requires certain modifications so that, the output result belongs to the target domain. For example, in the demultiple scenario, given a multiple-infested gather (input domain), our diffusion approach has to identify the multiples and cancel them out, leading to a multiple-free output gather (target domain).

The results of our experimental evaluations are fairly encouraging, as they show competitive performance, when comparing with standardized, customized algorithms. As we pointed out before, diffusion models for seismic data is an unexplored field to date and hence, the ultimate goal of this project is not to outperform these current algorithms in their respective areas, but to provide a solid analysis of the applicability and flexibility of this novel framework. Therefore, the main success of our implementation can be regarded as proof of concept that can be used to adopt generative models, namely diffusion models, in the geoscience community. We believe that our work can help to lay the foundation for future research that can benefit both academia and industry.

Code availability

Name of the code/library diffseis python package.
Contact: Ricard Durall, ricard.durall.lopez@itwm.fraunhofer.de
Hardware requirements: Modern laptop with GPU.
Program language: Python 3.9
Software required: No external software. Python dependencies are listed in the project.

Program size: Source files 1.3 Mb and pretrained model 1.2 Gb of memory.

The source codes are available for downloading at the link: https://github.com/dlseis/diffseis

CRediT authorship contribution statement

Ricard Durall: Conception of the original idea, Development of the source code, Conduction of the experiments, Writing of the manuscript. **Ammar Ghanim:** Conception of the original idea, Experiments for demultiple and denoising, Writing of the manuscript. **Mario Ruben Fernandez:** Experiments for interpolation, Writing of the manuscript. **Norman Ettrich:** Critical feedback, Supervision of the project. **Janis Keuper:** Critical feedback, Supervision of the project.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Demultiple — The real dataset (Volve) that was used, had been made available under Equinor Open License. Denoising and Interpolation — Publicly available on https://wiki.seg.org/wiki/Open_data.

Acknowledgments

Appendix

We provide additional results, where we can visualize the evolution of the reverse process for all the aforementioned case studies (see Fig. 10, Figs. 11 and 12). Note that the subindexes of the x indicate the output of an intermediate step during the inference process, being x_{1999} random noise and x_0 the final output of the model.
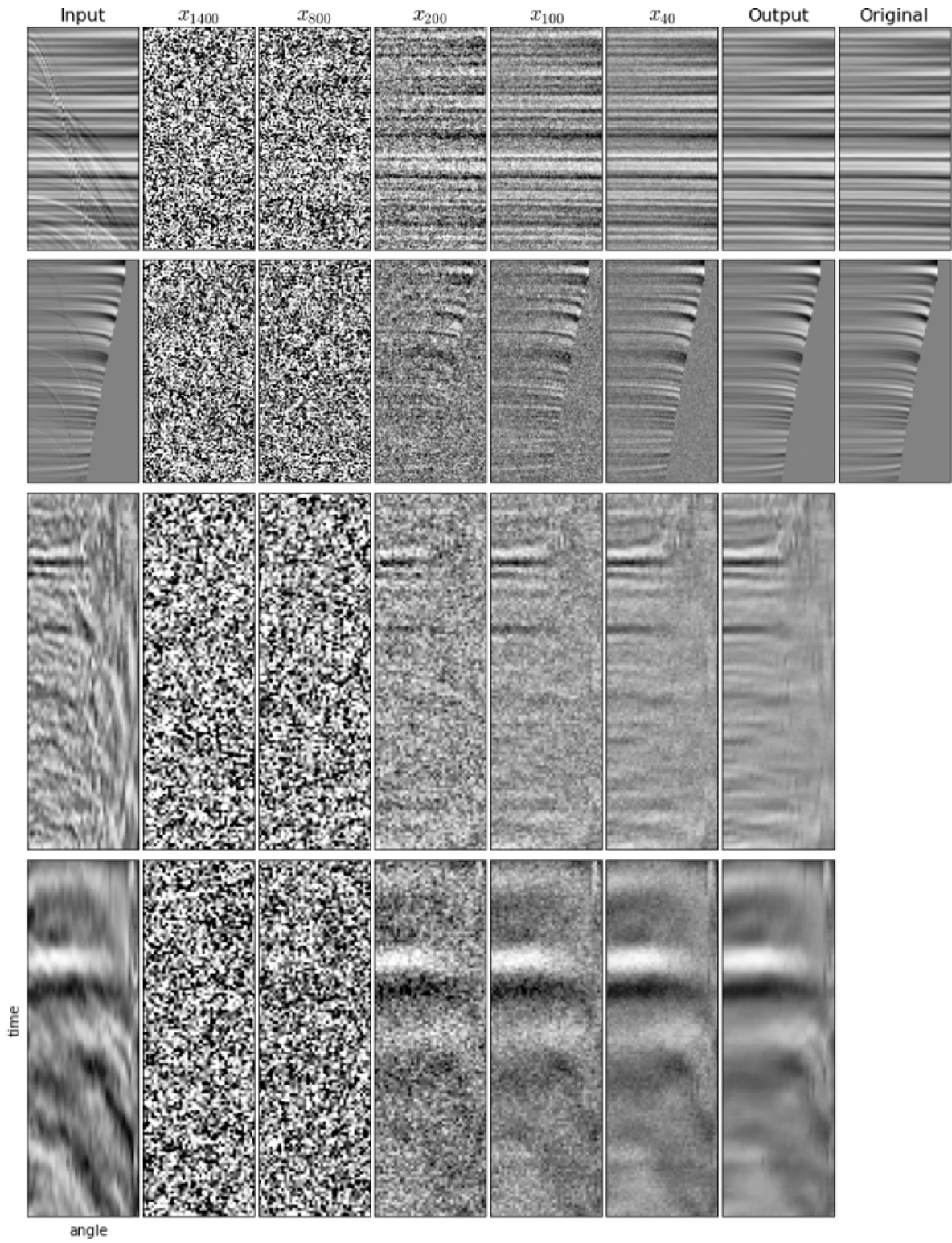
Fig. 10. This figure displays demultiple results at different intermediate steps for the reverse process. Note that the first two rows show synthetic data examples, while the last two from the Volve dataset.
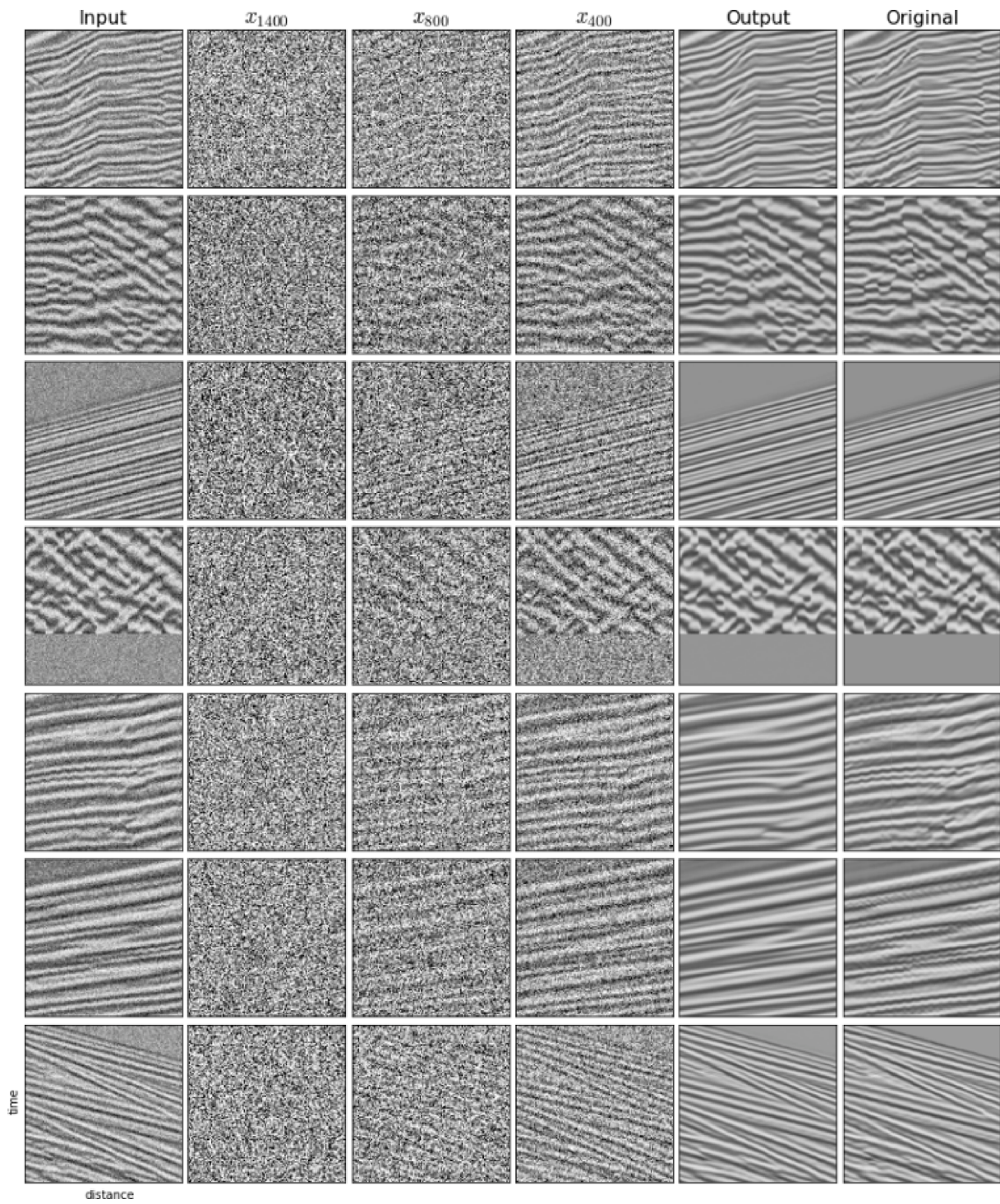
Fig. 11. This figure displays denoising results at different intermediate steps for the reverse process. Note that the examples belong to the Model94 dataset.
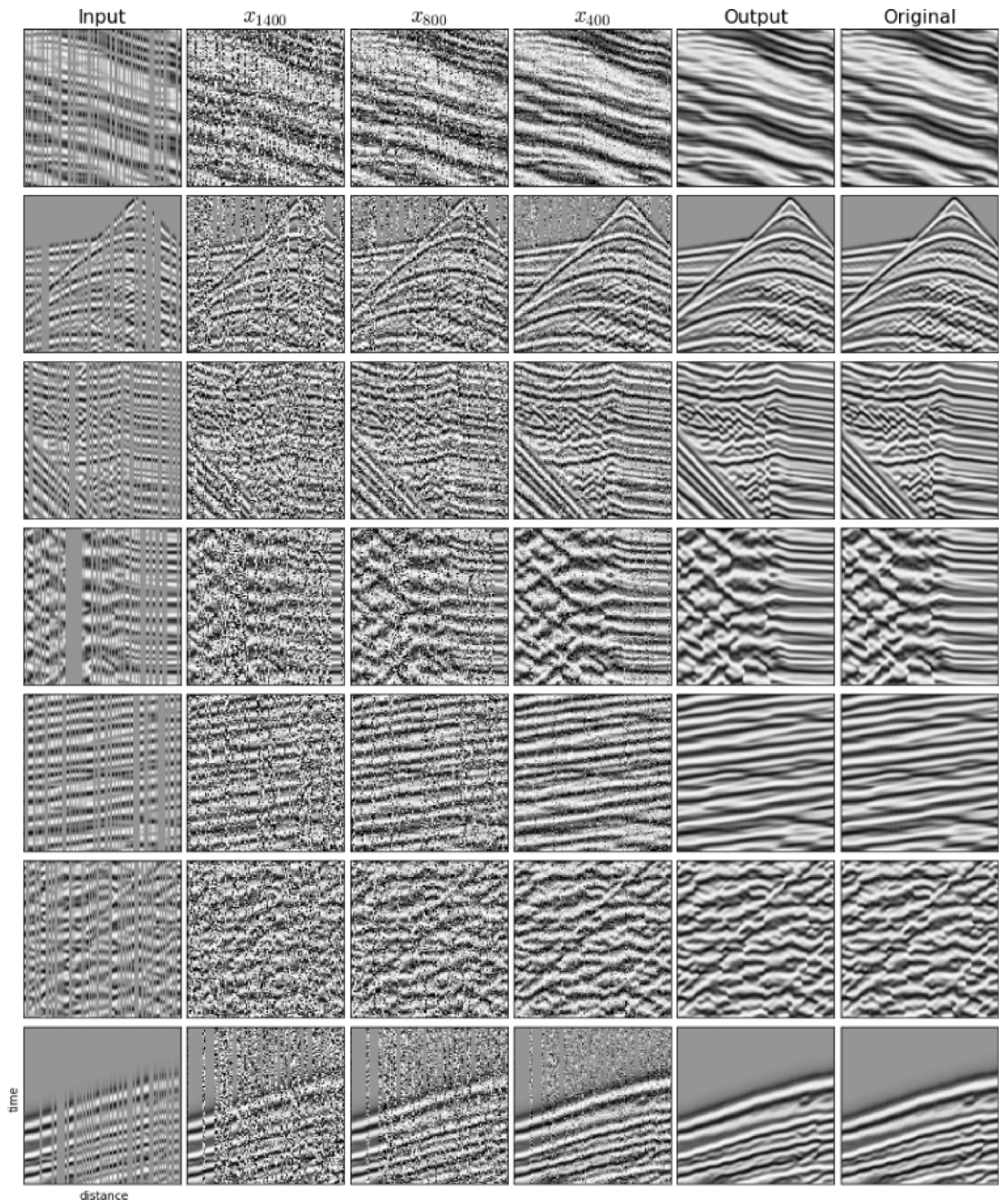
Fig. 12. This figure displays interpolation results at different intermediate steps for the reverse process. Note that the examples belong to the Model94 dataset.

References

An, Y., Guo, J., Ye, Q., Childs, C., Walsh, J., Dong, R., 2021. Deep convolutional neural network for automatic fault recognition from 3D seismic datasets. Comput. Geosci. 153, 104776.

Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein generative adversarial networks. In: International Conference on Machine Learning. PMLR, pp. 214–223.

Breuer, A., Ettrich, N., Habelitz, P., 2020. Deep learning in seismic processing: Trim statics and demultiple. In: SEG Technical Program Expanded Abstracts 2020. Society of Exploration Geophysicists, pp. 3199–3203.

Bugge, A.J., Evensen, A.K., Lie, J.E., Nilsen, E.H., 2021. Demonstrating multiple attenuation with model-driven processing using neural networks. Lead. Edge 40 (11), 831–836.

Dhariwal, P., Nichol, A., 2021. Diffusion models beat gans on image synthesis. Adv. Neural Inf. Process. Syst. 34, 8780–8794.

Durall, R., Chatzimichailidis, A., Labus, P., Keuper, J., 2020a. Combating mode collapse in gan training: An empirical analysis using hessian eigenvalues. ArXiv preprint arXiv:2012.09673.

Durall, R., Ghanim, A., Ettrich, N., Keuper, J., 2022. Dissecting U-net for seismic application: An in-depth study on deep learning multiple removal. ArXiv preprint arXiv:2206.12112.

Durall, R., Tschannen, V., Pfreundt, F.-J., Keuper, J., 2020b. Synthesizing seismic diffractions using a generative adversarial network. In: SEG Technical Program Expanded Abstracts 2020. Society of Exploration Geophysicists, pp. 1491–1495.

Equinor, 2018. Volve field dataset. https://cdn.equinor.com/files/h61q9gi9/global/de6532f6134b9a953f6c41bac47a0c055a3712d3.pdf?equinor-hrs-terms-and-conditions-for-licence-to-data-volve.pdf.

Fernandez, M., Durall, R., Ettrich, N., Delescluse, M., Rabaute, A., Keuper, J., 2022. A comparison of deep learning paradigms for seismic data interpolation. In: Second EAGE Digitalization Conference and Exhibition, vol. 2022, no. 1. European Association of Geoscientists & Engineers, pp. 1–5.

Galbraith, M., 1991. Random noise attenuation by fx prediction: A tutorial. In: SEG Technical Program Expanded Abstracts 1991. Society of Exploration Geophysicists, pp. 1428–1431.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Adv. Neural Inf. Process. Syst. 27.

Gray, S.H., Marfurt, K.J., 1995. Migration from topography: Improving the near-surface image. Canad. J. Explor. Geophys. 31 (1–2), 18–24.

Gulunay, N., 1986. FX decon and the complex Wiener prediction filter for random noise reduction on stacked data. In: 56th SEG Annual Meeting. pp. 2–6.

Gülünay, N., 2017. Signal leakage in f-x deconvolution algorithms. Geophysics 82 (5), W31–W45.

Hampson, D., 1986. Inverse velocity stacking for multiple elimination. In: SEG Technical Program Expanded Abstracts 1986. Society of Exploration Geophysicists, pp. 422–424.

Herrmann, F.J., Wang, D., Hennenfent, G., Moghaddam, P.P., 2008. Curvelet-based seismic data processing: A multiscale and nonlinear approach. Geophysics 73 (1), A1–A5.

Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models. Adv. Neural Inf. Process. Syst. 33, 6840–6851.

Kazerouni, A., Aghdam, E.K., Heidari, M., Azad, R., Fayyaz, M., Hacihaliloglu, I., Merhof, D., 2022a. Diffusion models for medical image analysis: A comprehensive survey. ArXiv preprint arXiv:2211.07804.

Kingma, D.P., Welling, M., 2013. Auto-encoding variational bayes. ArXiv preprint arXiv:1312.6114.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L., 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11461–11471.

Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J., 2016. Unrolled generative adversarial networks. ArXiv preprint arXiv:1611.02163.

Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets. ArXiv preprint arXiv:1411.1784.

Mousavi, S.M., Langston, C.A., Horton, S.P., 2016. Automatic microseismic denoising and onset detection using the synchrosqueezed continuous wavelet transform. Geophysics 81 (4), V341–V355.

Naghizadeh, M., 2012. Seismic data interpolation and denoising in the frequency-wavenumber domain. Geophysics 77 (2), V71–V80.

Neelamani, R., Baumstein, A.I., Gillard, D.G., Hadidi, M.T., Soroka, W.L., 2008. Coherent and random noise attenuation using the curvelet transform. Lead. Edge 27 (2), 240–248.

Nichol, A.Q., Dhariwal, P., 2021. Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning. PMLR, pp. 8162–8171.

Nivlet, P., Smith, R., AlBinHassan, N., 2020. Automated well-to-seismic tie using deep neural networks. In: SEG Technical Program Expanded Abstracts 2020. Society of Exploration Geophysicists, pp. 2156–2160.

Oh, S., Noh, K., Yoon, D., Seol, S.J., Byun, J., 2018. Salt delineation from electromagnetic data using convolutional neural networks. IEEE Geosci. Remote Sens. Lett. 16 (4), 519–523.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M., 2022. Hierarchical text-conditional image generation with clip latents. ArXiv preprint arXiv:2204.06125.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K., 2022. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. ArXiv preprint arXiv:2208.12242.

Saad, O.M., Chen, Y., 2020. Deep denoising autoencoder for seismic random noise attenuation. Geophysics 85 (4), V367–V376.

Saharia, C., Chan, W., Chang, H., Lee, C.A., Ho, J., Salimans, T., Fleet, D.J., Norouzi, M., 2021a. Palette: Image-to-image diffusion models. ArXiv preprint arXiv:2111.05826.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M., 2021b. Image super-resolution via iterative refinement. ArXiv preprint arXiv:2104.07636.

Shi, Y., Wu, X., Fomel, S., 2019. SaltSeg: Automatic 3D salt segmentation using a deep convolutional neural network. Interpretation 7 (3), SE113–SE122.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S., 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning. PMLR, pp. 2256–2265.

Sohn, K., Lee, H., Yan, X., 2015. Learning structured output representation using deep conditional generative models. Adv. Neural Inf. Process. Syst. 28.

Tomczak, J.M., 2022. Deep Generative Modeling. Springer Nature.

Tschannen, V., Delescluse, M., Ettrich, N., Keuper, J., 2020. Extracting horizon surfaces from 3D seismic data using deep learning. Geophysics 85 (3), N17–N26.

Tschannen, V., Ghanim, A., Ettrich, N., 2022. Partial automation of the seismic to well tie with deep learning and Bayesian optimization. Comput. Geosci. 164, 105120.

Wu, X., Liang, L., Shi, Y., Fomel, S., 2019. FaultSeg3D: using synthetic datasets to train an end-to-end convolutional neural network for 3D seismic fault segmentation. Geophysics 84 (3), IM35–IM45.

Yang, L., Sun, S.Z., 2020. Seismic horizon tracking using a deep convolutional neural network. J. Pet. Sci. Eng. 187, 106709.

Yu, S., Ma, J., Wang, W., 2019. Deep learning for denoising. Geophysics 84 (6), V333–V350.