FAULTDIFF: A CONDITIONAL DIFFUSION APPROACH TO MITIGATE SYNTHETIC-REAL DISPARITIES IN SEISMIC FAULT DETECTION

A PREPRINT

Soufiene Sellami

November 5, 2024

ABSTRACT

Despite the success of deep learning in seismic fault detection, the reliance on synthetic data limits the generalization capabilities of current methods. In this paper, we introduce Fault-Diff, a novel conditional denoising diffusion model designed to generate highly accurate and realistic synthetic seismic data. Fault-Diff generates synthetic 2D seismic images conditioned on faults and edge maps, effectively capturing complex seismic characteristics. Our approach outperforms state-of-the-art generative models across key metrics of synthetic data quality. We also demonstrate that models trained on our synthetic data achieve performance comparable to those trained on real data, effectively bridging the gap between synthetic and real-world applications.

Keywords Seismic Fault Detection · Diffusion · Synthetic

1 Introduction

Accurately identifying seismic faults is crucial for reservoir characterization and hydrocarbon exploration. Recent deep learning methods have shown promise in seismic interpretation tasks; yet, their effectiveness in fault segmentation remains limited due to the complexity of faults and the subtlety of their expressions in seismic images. The scarcity of annotated data further complicates this challenge, as labeling faults requires significant expertise and is time-consuming [1, 2]. While synthetic data has emerged as a practical alternative to overcome data limitations, it often fails to generalize well to field data due to discrepancies in statistical and structural properties [3, 4].

Recent advancements in generative models, such as Generative Adversarial Networks (GANs) and diffusion models, have shown promise in seismic applications like resolution enhancement, reconstruction, and inversion [5, 6, 7, 8]. Notably, Ferreira et al.[9] demonstrated the use of Pix2Pix for generating synthetic seismic images from sketches. However, their approach was constrained by the quality of the input sketches and did not investigate the potential of the generated data for specific seismic interpretation tasks . Diffusion models have outperformed GANs in terms of quality and training efficiency due to their simpler denoising process [**?** 10], yet their potential for generating synthetic seismic images remains underexplored. Leveraging diffusion models offers a unique opportunity to produce high-fidelity synthetic data that bridges the gap between synthetic and real seismic datasets.

Contribution In this paper, we introduce a novel conditional denoising diffusion model designed for generating synthetic seismic data. Our model is conditioned on fault segmentation maps and edge features extracted using Canny edge detection to enhance its ability to capture complex seismic characteristics.

We demonstrate that our approach outperforms state-of-the-art generative models in key quality metrics for synthetic data. Moreover, we validate the fidelity of the generated seismic images to their corresponding fault masks through segmentation, showing that models trained on our synthetic data perform comparably to those trained on real data.

To our knowledge, this is the first application of diffusion models to seismic data synthesis, achieving high quality and fidelity. We hope this work will encourage further research and help overcome the limitations of synthetic data in seismic interpretation. However, our current approach focuses on 2D seismic data and does not yet extend to 3D volumes or multi-attribute data, which could offer additional insights for seismic tasks.

2 Related Work

2.1 Seismic Fault Detection

The detection of faults in seismic data has long been a challenge in geophysics. Early approaches typically involved a two-stage process: first, computing seismic attributes to enhance discontinuities, such as coherence [11], dips and azimuth [12], and curvature [13]; and then applying specific algorithms to extract the faults. The Hough transform was initially introduced for seismic interpretation by [14] and later extended to 3D by [15]. Additionally, [16, 17] proposed the use of ant tracking algorithms, while other studies focused on filtering techniques [18, 19]. However, these methods are highly sensitive to noise, and their parameters are often chosen through trial and error.

With the rise of deep learning, various models have been employed for seismic fault detection, including CNNs [20, 21, 22], Transformers [23, 24, 25], and SAM [26]. However, the lack of annotated data remains a significant challenge [1]. To address this, Wu et al. [27] introduced FaultSeg3D, a 3D U-Net model trained on synthetic data. Their method involves generating a random horizontal reflectivity model, incorporating folds and faults, convolving the model with a Ricker wavelet, and adding noise to produce synthetic seismic images. This approach is efficient and memory-conserving due to its reliance on simulations. However, models trained on this synthetic data often underperform when applied to field seismic data [4]. For instance, FaultSeg3D, trained solely on synthetic data, struggled with the Thebe dataset—the largest annotated seismic fault dataset [3]—due to differences in fault characteristics and annotation thickness, resulting in high false-positive rates and reduced performance.

Several studies [28, 29, 30] have attempted to align synthetic data with real data by modifying the simulation process. Nevertheless, these approaches still face limitations, such as inadequate representation of complex fault shapes and imperfect noise modeling. According to a review by Lei et al. [2], over half of the studies on fault detection relied on synthetic data, underscoring the need for developing more realistic synthetic datasets to improve models performance.

2.2 Conditional Image Synthesis

Conditional image synthesis has emerged as a powerful technique in computer vision, enabling the generation of images based on specific constraints. Unlike unconditional image synthesis, which generates images from random noise that resembles its training data distribution, conditional methods allow for greater control over the output, producing images that adhere to given specifications such as text prompts[31], segmentation maps[32] or image references[33].

Among these conditioning approaches, using image references as input has given rise to a particularly important sub-field known as image-to-image translation. This sub-field has evolved alongside general generative AI, progressing from early approaches using conditional GANs, with Pix2Pix[34] being one of the most influential frameworks, to more recent advancements like ControlNet[35], adding spatial conditioning controls to large, pre-trained text-to-image diffusion models. However these methods primarily focus on natural images and are rarely adapted to the noisier conditions of seismic data, and Our work aims to bridge this gap by introducing a diffusion model tailored for seismic imaging.

3 Method

We begin by providing a brief overview of diffusion models, followed by a detailed explanation of the design choices behind FaultDiff, including condition creation, model architecture, and training.

3.1 Denoising Diffusion Probabilistic Models: A Background

Denoising Diffusion Probabilistic Models (DDPMs) [36] incrementally learn to generate by denoising. The learning process in DDPMs is framed as two Markov chain processes: a forward diffusion process and a reverse process1.

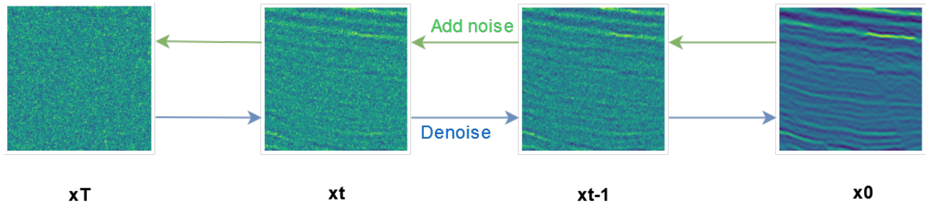

Figure 1: Diagram of forward and reverse processes in diffusion models.

Forward Process Random noise is added to the original image $\mathbf{x}_0$ over a series of T steps (with $T = 1000$ in our case). At each step, a small amount of noise is added according to a fixed schedule (we adopt a cosine scheduler), gradually transforming the image into pure noise xT. This process is defined by:

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

where $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ represents the Gaussian distribution with mean $\sqrt{1-\beta_t}\,\mathbf{x}_{t-1}$ and variance $\beta_t\mathbf{I}$.

This process can be reformulated to enable direct sampling of $\mathbf{x}_t$ from $\mathbf{x}_0$ with a single draw of Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$:

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

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$.

Reverse Process A neural network is trained to gradually denoise an image starting from pure noise. The equation for the reverse process is:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right), \sigma_t^2\mathbf{I}\right) \tag{3}$$

- $\epsilon_\theta(\mathbf{x}_t, t)$ estimate function to predict noise ϵ.
- σ_t^2 is the variance term .

The loss function used for training the neural network is:

$$L = \mathbb{E}_{\mathbf{x}_0,t,\epsilon}\left\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\right\|^2 \tag{4}$$

Adding Conditions to Diffusion Models To control the generation process, our approach is to condition the neural network. We implement this by having two separate encoder paths: one for the original input and one for the condition. We then fuse both of the encoded features. This does not alter the forward process, but the estimate function becomes:

$$\epsilon_\theta(\mathbf{x}_t, \mathbf{c}_t, t) = D\left((E_c(\mathbf{c}_t) + E_x(\mathbf{x}_t))\, t\right) \tag{5}$$

- $E_c(\mathbf{c}_t)$ represents the embedding of the conditional input $\mathbf{c}_t$.
- $E_x(\mathbf{x}_t)$ denotes the embedding of the original image $\mathbf{x}_t$.
- The decoder D generates the predicted noise ϵ_θ based on both the image and the condition.

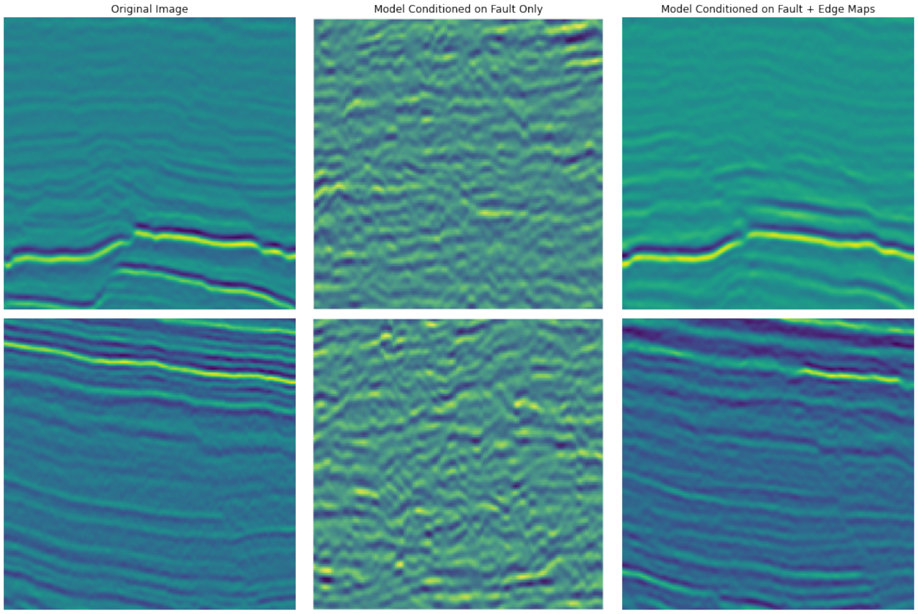
Original Image
Model Conditioned on Fault Only
Model Conditioned on Fault + Edge Maps

Figure 2: Comparison between models using fault segmentation as condition only and using both of fault and edge map.

3.2 Condition Design

In order to make a more realistic synthetic dataset for seismic fault detection, the core idea is to use existing fault segmentation from public datasets such as Thebe[3] to control the generation. To enhance the conditioning process, we use both fault and edge maps as input conditions, rather than fault maps alone. Fault maps capture critical information about seismic discontinuities, but they lack additional context provided by geological boundaries and fine-scale details that are crucial for generating realistic seismic images. Edge maps, on the other hand, capture these fine-grained details, such as layer boundaries and subtle changes within the seismic data. By conditioning the model on both fault and edge maps, we provide richer contextual information that improves the model's understanding of spatial relationships and patterns; this approach enhances the model's ability to generalize. In fact, when we trained a model using only fault maps as a condition (See figure2), it exhibited poor generalization leading to sub-optimal results. The inclusion of edge maps addresses this limitation, allowing the model to synthesize more realistic images and accurately represent diverse geological features.

To create the condition for our diffusion model, we apply the Canny edge detector to a real seismic image. The Canny parameters are set with σ=3, a low threshold of 0.25, and a high threshold of 0.75. These settings are chosen to capture prominent edges without introducing excessive noise or over-complicating the condition. The edge map obtained is then combined with the fault mask.

3.3 FaultDiff Architecture

The architecture is a ResUNet with two parallel ResNet encoders, a transformer block in the middle, and a ResNet decoder. The encoder consists of four downsampling stages, each with two ResNet blocks followed by one linear attention block. The ResNet blocks in each stage contain two convolutional blocks, each featuring a 3x3 convolution, group normalization, and SiLU activation. After the attention block, a downsampling operation is performed. After summing the encoded features from both paths, they are processed by a middle transformer block then fed into the mirrored decoder.

4 Experiments and Results

To evaluate the effectiveness of FaultDiff, we employed a two-part evaluation framework: (1) generative evaluation, where we compared FaultDiff against state-of-the-art models and assessed its generative fidelity,

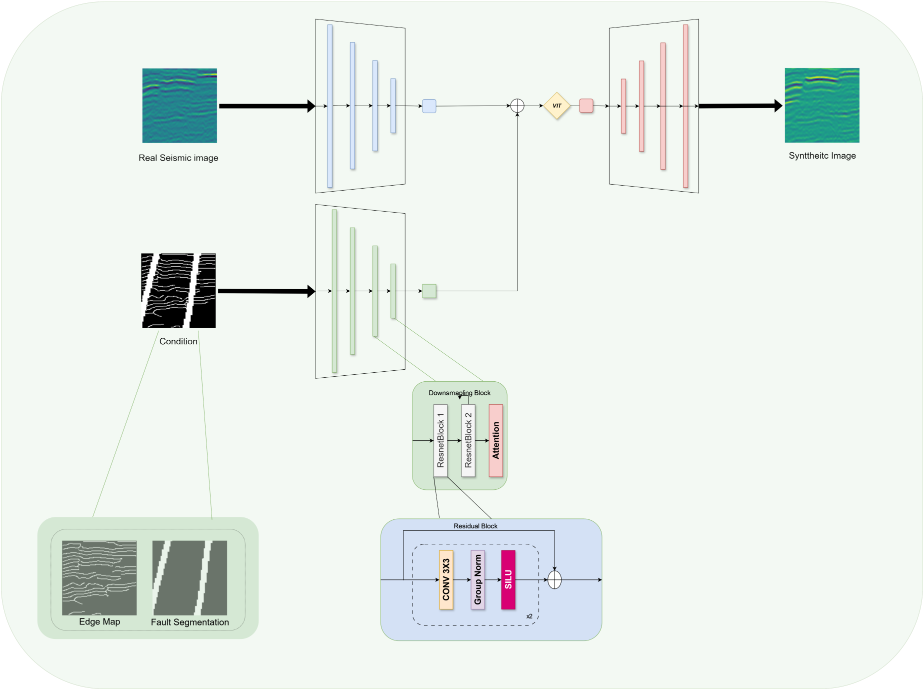

Figure 3: FaultDiff Architecture

and (2) segmentation evaluation, where we measured the impact of FaultDiff-generated data on fault detection accuracy compared to real and other synthetic data.

4.1 Comparisons with State-of-the-Art Methods

4.1.1 Dataset and Baselines

We use the Thebe dataset [3], which is partitioned into training, validation, and test sets, with dimensions of 1803 [crossline] x 3174 [inline] x 1537 [sample]. For training the diffusion model, we adopt a slicing window approach with a patch size of 128 x 128 and a step size of 64. To ensure high-quality samples, patches containing less than 3% faults are excluded. Specifically, we extract 128 x 128 patches from every 10th crossline of the training set, resulting in 12,133 images and 12,133 corresponding conditions as detailed in 3.2. This approach leverages only about 10% of the training set, significantly reducing computational load and training time while demonstrating that a smaller subset of data is sufficient to generate high-quality synthetic data effectively. For testing, we extract patches from the testing set at every crossline, yielding 143,560 images.We compare FaultDiff with Pix2Pix[34] and Controlnet[35].

4.1.2 Implementation

All experiments are implemented using PyTorch, with training and testing conducted on an NVIDIA A6000 GPU with 16 GB of RAM.

FaultDiff The hyperparameters are defined as follows: the learning rate is set to 0.0005, and the Adam optimizer is used with $\beta_1 = 0.95$, $\beta_2 = 0.999$, weight decay of 1×10^{-6}, and $\epsilon = 1 \times 10^{-8}$. We employ mixed precision with BF16 for enhanced computational efficiency. The batch size is 64, and the model converges after 200 epochs with $1{,}000$ diffusion timesteps. For sampling, we use the DDIM algorithm with 100 timesteps.

Pix2pix We train Pix2pix using the default settings [34] with a batch size of 64 for 200 epochs.

ControlNet Since seismic data differ significantly from the original large-scale natural image pre-training set used by ControlNet, we first fine-tuned the Variational Autoencoder (VAE) for 200 epochs. This was followed by fine-tuning the Stable Diffusion (SD) model for 400 epochs. Subsequently, we fine-tuned ControlNet using the images and their corresponding conditions for an additional 400 epochs. We utilized empty prompts for text inputs and employed Stable Diffusion 2 as our diffusion model. For all training, we used a batch size of 64 and a learning rate of 1×10^{-5}.

4.1.3 Generative Results

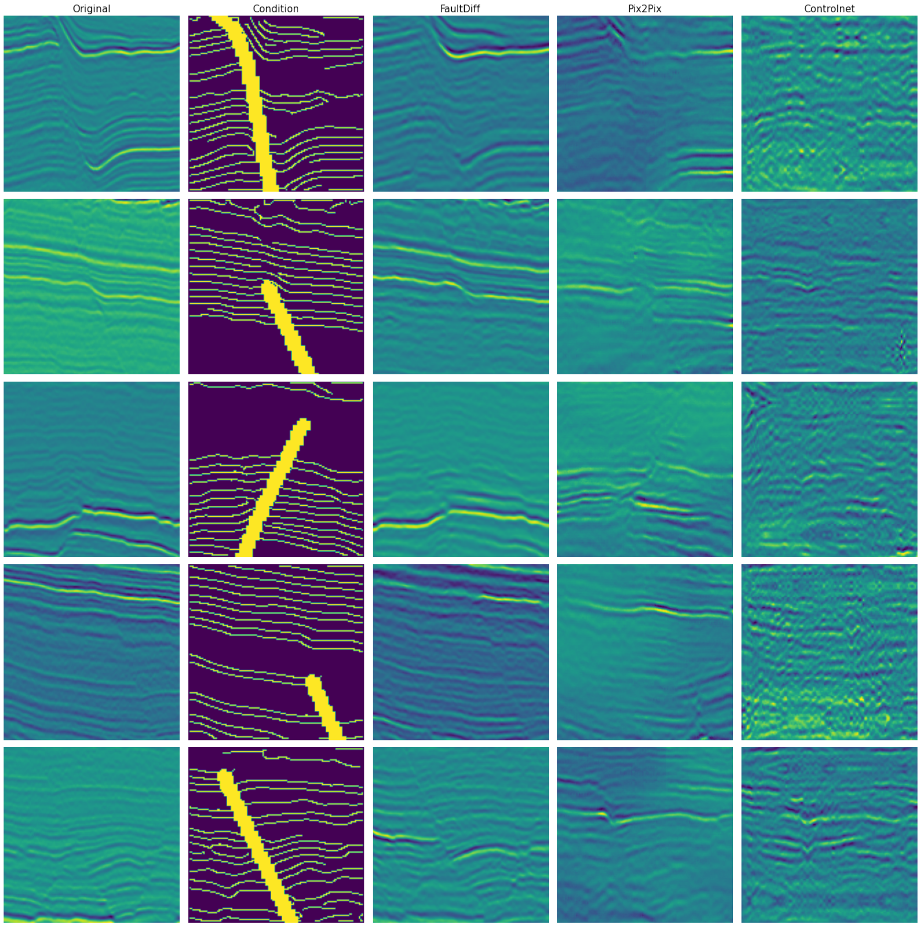

Figure 4: Visual Comparison of Seismic Image synthesis Across Different Models

Table 1: Generation Results for FaultDiff, Pix2pix, and ControlNet

Model	MSE	PSNR	DSSIM
FaultDiff	**0.0028**	**26.74**	**0.1905**
Pix2pix	0.0053	23.02	0.2143
ControlNet	0.0066	23.54	0.2218

Figure 4 presents a visual comparison of the generated images from FaultDiff, Pix2pix, and ControlNet, showcasing the results of each model under the same condition. From the generated seismic images, we observe that FaultDiff is most comparable to the Original image, with fewer discrepancies in the seismic reflections and fault structures.Additionally, the generated image by "FaultDiff" shows a smoother texture and more accurate replication of the seismic patterns, effectively preserving the original image's structure. In contrast, Pix2Pix, while competent in following the condition, introduces minor distortions in the reflections, and ControlNet struggles more with maintaining the geological continuity, displaying more artifacts and noise. This poor performance can be attributed to its limitations, including the need for a larger dataset, the lack of suitable text prompts, and its untested nature in adapting to the specific challenges of seismic data. Table 1 illustrates quantitative results in terms of generation metrics. FaultDiff outperforms the other models by up to 25% in MSE, achieving the lowest error rate 3.72 db in PSNR and 2%in less in DSSIM.

4.2 Generative Fidelity

To evaluate the generative fidelity of our synthetic seismic images, we assess how closely these images match the distribution of field seismic data. We use the Fréchet Inception Distance (FID)[37] to quantify this resemblance. We calculate the FID between features extracted from the last average pooling layer of the Inception-v3 model. we use the Thebe dataset [3] and the F3 dataset [38] as our real data. The F3 dataset serves as an unbiased benchmark, as it was not involved in the training of our diffusion model. This allows us to evaluate the performance of our synthetic images more impartially. We compare our synthetic data with those generated by the method introduced by Wu et al. [27].

Table 2: FID Scores between Synthetic Datasets and Real Datasets

Synthetic Dataset	**Real Dataset**	**FID Score**
FaultDiff	Thebe	6.47562
	F3	6.47568
FaultSeg3d	Thebe	130.12766
	F3	130.12789

Table 2 presents the Fréchet Inception Distance (FID) scores. Our approach, FaultDiff, achieves notably low FID scores for both the TheBe and F3 datasets, with values of 6.47562 and 6.47568, respectively. These results indicate that FaultDiff-generated images closely resemble real field data in terms of feature distribution, reflecting its high generative fidelity.

In contrast, the FaultSeg3D method [27] yields significantly higher FID scores, indicating that its synthetic images diverge considerably from the real data distributions. This high FID score is likely due to its simulation-based approach, explaining the poor generalization in models trained on such data.

4.3 From the prespective of Segmentation

4.4 Segmentation Evaluation

To evaluate the effectiveness of synthetic data in fault detection, we measure the ability of segmentation models trained solely on synthetic data to generalize to real field data. We compare the performance of segmentation models, based on a 2D U-Net architecture, trained on real data (TheBe) and synthetic data (FaultDiff and FaultSeg3D). For the TheBe dataset, we use the slicing window approach described in **??** to extract 120,976 images of size 128x128 from every crossline in the training set. For FaultDiff, we generate an equivalent number of 128x128 images from the respective conditions. Additionally, we experiment with training on a reduced dataset of synthetic images by generating images conditioned on patches from every 5th crossline, resulting in 24338 images less than 5x the original trainset size.

For FaultSeg3D [27], we convert every crossline and inline profile into 2D images, resulting in 51,200 images of size 128x128. The models are tested on the TheBe test set consisting of 143,560 images. We trained each model for 100 epochs with a batch size of 16 and a learning rate of 10^{-3}. Evaluation is performed using standard metrics, including Accuracy, Precision, Recall, Average Precision (AP), and Area Under the Curve (AUC). The results presented in Table 3 demonstrate the efficacy of the FaultDiff-generated synthetic data in training segmentation models. The model trained on FaultDiff-synthesized images exhibits performance metrics that are not only comparable but marginally superior to those of models trained on real data. Notably,

Table 3: Segmentation Performance Metrics for Real and Synthetic Data

Training Set	**Strategy**	**Accuracy**	**Precision**	**Recall**	**AUC**	**AP**
Thebe	Real	0.9316	0.6699	**0.5438**	**0.9291**	**0.6640**
FaultDiff	Synthetic	**0.9320**	**0.6873**	0.5142	0.9232	0.6430
FaultDiff	Synthetic (reduced set)	0.9315	0.6689	0.5122	0.9212	0.6330
FaultSeg3D	Synthetic	0.9043	0.4115	0.0322	0.6593	0.1933

the FaultDiff-trained model achieved an accuracy of 0.9322 and precision of 0.6879, slightly outperforming the model trained on original data (accuracy: 0.9316, precision: 0.6699). Besides, the segmentation model trained on a reduced subset of FaultDiff synthetic data, utilizing fewer crosslines, maintained comparable performance metrics. This reduced dataset model achieved an accuracy of 0.9315 and precision of 0.6689, closely mirroring the performance of models trained on both the full original and synthetic datasets. The consistency in recall measurements (0.5142 for the reduced set vs. 0.5244 for the full set) is particularly noteworthy. These results suggest that a small set of our synthetic data can yield performance comparable to larger datasets, while simultaneously offering advantages in computational efficiency during the training phase.

5 Conclusion

In this work, we introduced *FaultDiff*, a conditional denoising diffusion model designed to synthesize seismic images for fault detection. The core of our method is a conditional denoising process that incorporates fault segmentation and edge maps extracted from field data. Our evaluation confirms that FaultDiff generates synthetic seismic images that closely match real data distributions. Furthermore, segmentation models trained on FaultDiff data demonstrate competitive performance in fault detection.

Looking ahead, we plan to refine and expand FaultDiff by incorporating additional seismic attributes and exploring multi-modal generation. Future work will also focus on extending FaultDiff to 3D. This study lays a foundation for advancing synthetic seismic data generation and its integration into practical fault detection and analysis workflows.

References

[1] Yu An, Haiwen Du, Siteng Ma, Yingjie Niu, Dairui Liu, Jing Wang, Yuhan Du, Conrad Childs, John Walsh, and Ruihai Dong. Current state and future directions for deep learning based automatic seismic fault interpretation: A systematic review. *Earth-Science Reviews*, 243:104509, 2023.

[2] Lei Lin, Zhi Zhong, Chenglong Li, Andrew Gorman, Hao Wei, Yanbin Kuang, Shiqi Wen, Zhongxian Cai, and Fang Hao. Machine learning for subsurface geological feature identification from seismic data: Methods, datasets, challenges, and opportunities. *Earth-Science Reviews*, 257:104887, 2024.

[3] Yu An, Jiulin Guo, Qing Ye, Conrad Childs, John Walsh, and Ruihai Dong. Deep convolutional neural network for automatic fault recognition from 3d seismic datasets. *Computers Geosciences*, 153:104776, 2021.

[4] Yimin Dou, Kewen Li, Jianbing Zhu, Xiao Li, and Yingjie Xi. Attention-based 3-d seismic fault segmentation training by a few 2-d slice labels. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022.

[5] Dario A. B. Oliveira, Rodrigo S. Ferreira, Reinaldo Silva, and Emilio Vital Brazil. Improving seismic data resolution with deep generative networks. *IEEE Geoscience and Remote Sensing Letters*, 16(12):1929–1933, 2019.

[6] Haihang Zhang, Guangzhi Zhang, Jianhu Gao, Shengjun Li, Jinmiao Zhang, and Zhenyu Zhu. Seismic impedance inversion based on geophysical-guided cycle-consistent generative adversarial networks. *Journal of Petroleum Science and Engineering*, 218:111003, 2022.

[7] Hao-Ran Zhang, Yang Liu, Yu-Hang Sun, and Gui Chen. Seisresodiff: Seismic resolution enhancement based on a diffusion model. *Petroleum Science*, 2024.

[8] Paul Goyes-Peñafiel, León Suárez-Rodríguez, Claudia V. Correa, and Henry Arguello. Gan supervised seismic data reconstruction: An enhanced learning for improved generalization. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–10, 2024.

[9] Rodrigo S. Ferreira, Julia Noce, Dario A. B. Oliveira, and Emilio Vital Brazil. Generating sketch-based synthetic seismic images with generative adversarial networks. *IEEE Geoscience and Remote Sensing Letters*, 17(8):1460–1464, 2020.

[10] Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, 2020.

[11] Mike Bahorich and Steve Farmer. 3-d seismic discontinuity for faults and stratigraphic features: The coherence cube. *The Leading Edge*, 14(10):1053–1058, 1995.

[12] Arthur E. Barnes. Weighted average seismic attributes. *GEOPHYSICS*, 65(1):275–285, 2000.

[13] Andy Roberts. Curvature attributes and their application to 3d interpreted horizons. *First Break*, 19:85–100, 02 2001.

[14] Nasher M. AlBinHassan and Kurt Marfurt. *Fault detection using Hough transforms*, pages 1719–1721. 2005.

[15] ZhenWang* and GhassanAlRegib. *Automatic fault surface detection by using 3D Hough transform*, pages 1439–1444. 2014.

[16] Stein Inge Pedersen, Trygve Randen, Lars Sonneland, and Øyvind Steen. *Automatic fault extraction using artificial ants*, pages 512–515. 2005.

[17] D. Sun, Yibei Ling, Y. Bai, X. zhang, and X. Xi. Application of spectral decomposition and ant tracking to fractured carbonate reservoirs. 05 2011.

[18] Dave Hale. Structure-oriented smoothing and semblance. 01 2009.

[19] Ahmed Aqrawi and Trond Boe. Improved fault segmentation using a dip guided and modified 3d sobel filter. volume 30, pages 999–1003, 01 2011.

[20] Tao Zhao and Pradip Mukhopadhyay. *A fault-detection workflow using deep learning and image processing*, pages 1966–1970. 2018.

[21] Xinming Wu, Yunzhi Shi, Sergey Fomel, and Luming Liang. Convolutional neural networks for fault interpretation in seismic images. 08 2018.

[22] Bowen Guo, Lu Liu, and Yi Luo. *Automatic seismic fault detection with convolutional neural network*, pages 1786–1789. 2018.

[23] Zhanxin Tang, Bangyu Wu, Weihua Wu, and Debo Ma. Fault detection via 2.5d transformer u-net with seismic data pre-processing. *Remote Sensing*, 15(4), 2023.

[24] Z. Wang, J. You, W. Liu, and X. Wang. Transformer assisted dual u-net for seismic fault detection. *Frontiers in Earth Science*, 11:1047626, 2023.

[25] Zeren Zhang, Ran Chen, and Jinwen Ma. Improving seismic fault recognition with self-supervised pre-training: A study of 3d transformer-based with multi-scale decoding and fusion. *Remote Sensing*, 16:922, 03 2024.

[26] Ran Chen, Zeren Zhang, and Jinwen Ma. Seismic fault sam: Adapting sam with lightweight modules and 2.5d strategy for fault detection, 2024.

[27] Xinming Wu, Luming Liang, Yunzhi Shi, and Sergey Fomel. Faultseg3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation. *GEOPHYSICS*, 84(3):IM35–IM45, 2019.

[28] Yile Ao, Wenkai Lu, Bowu Jiang, and Patrice Monkam. Seismic structural curvature volume extraction with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 59(9):7370–7384, 2021.

[29] Jiankun Jing, Zhe Yan, Zheng Zhang, Hanming Gu, and Bingkai Han. Fault detection using a convolutional neural network trained with point-spread function-convolution-based samples. *GEOPHYSICS*, 88(1):IM1–IM14, 2023.

[30] Xinming Wu, Zhicheng Geng, Yunzhi Shi, Nam Pham, Sergey Fomel, and Guillaume Caumon. Building realistic structure models to train convolutional neural networks for seismic structural interpretation. *GEOPHYSICS*, 85(4):WA27–WA39, 2020.

[31] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models, 2023.

[32] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors, 2022.

[33] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation, 2023.

[34] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[35] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[36] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[37] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[38] Yazeed Alaudah, Patrycja Michałowicz, Motaz Alfarraj, and Ghassan AlRegib. A machine-learning benchmark for facies classification. *Interpretation*, 7(3):SE175–SE187, 2019.