Diff-DEM: A Diffusion Probabilistic Approach to Digital Elevation Model Void Filling

Kyle Shih-Huang Lo and Jörg Peters

***Abstract*—Digital Elevation Models (DEMs) are crucial for modeling and analyzing terrestrial environments, but voids in DEMs can compromise their downstream use. Diff-DEM is a self-supervised method for filling DEM voids that leverages a Denoising Diffusion Probabilistic Model (DDPM). Conditioned on a void-containing DEM, the DDPM acts as a transition kernel in the diffusion reversal, progressively reconstructing a sharp and accurate DEM. Both qualitative and quantitative assessments demonstrate Diff-DEM outperforms existing DEM inpainting, including Generative Adversarial Network (GAN) methods, Inverse Distance Weighting (IDW), Kriging, LR B-spline, and Perona-Malik diffusion. The comparison is on Gavriil's and on our benchmark that expands Gavriil's dataset from 63 to 217 full-size (5051×5051) 10-meter GeoTIFF images sourced from the Norwegian Mapping Authority; and from 50 DEMs to three groups of 1k each of increasing void size.**

Code and dataset: https://github.com/kylelo/Diff-DEM

***Index Terms*—Digital Elevation Model (DEM), inpainting, denoising diffusion probabilistic model, generative model.**

I. Introduction

DIGITAL Elevation Models (DEMs) are crucial for a wide range of applications, including geographic information systems [1], terrain monitoring [2], and disaster simulation [3]. However, DEM quality is frequently compromised by data voids. Data loss can stem from rugged terrain obstructing the back of a mountain, disparities in image content during stereophotogrammetry, or limitations in Light Detection and Ranging (LiDAR) data capture. Traditional mitigation utilizes techniques such as Inverse Distance Weighting (IDW) [4], Kriging [5], and Spline fitting [6] to fill DEM voids. These approaches do well when filling small voids but, due to a lack of prior knowledge of terrain's geometry, struggle with large and complex voids. To address this, Delta surface-based methods [7], [8] match and inpaint larger missing regions by inserting auxiliary DEMs.

Recent progress in DEM void-filling has been driven by methods based on Generative Adversarial Networks (GANs) [9]. GANs are based on the principle of adversarial learning, stemming from the competition between two neural networks: a generator and a classifier, which learn simultaneously. GANs that are effective for producing highly realistic images have also shown effective for image inpainting: [10], [11], [12], [13] employ Conditional GANs (CGANs) [14] for filling voids. These methods condition the model on void-containing DEM and predict the complete DEM. The studies [15], [16] additionally integrate attention mechanisms into CGANs. Shadow-Constrained GAN (SCGAN) [17] incorporates terrain shadow geometry into its loss function to improve restoration. Similarly, the Topographic Knowledge-constrained Conditional GAN (TKCGAN) [18] identifies and penalizes incorrect valley and ridge pixel predictions in DEM.

The strong image generation capabilities of diffusion models [19] have been summoned for the smaller-grained tasks of terrain data super-resolution [20], [21], [22], [23] and denoising [24], [25], and for support of 3D terrain generation (sketch-to-terrain [26] and text-to-satellite [27]) as well as cloud removal for satellite images [28], [29]. However, diffusion models have not been used to treat regions of incompleteness, i.e. DEM void filling.

Contributions.

- Diff-DEM is the first DEM void-filling technique based on conditional diffusion models.
- Diff-DEM improves on GAN-based [11] and traditional methods [4], [5], [6] for the test DEMs from [11].
- Diff-DEM is benchmarked on the entire Norway DEM dataset [30] with diverse void sizes (easy, medium, hard), underscoring effectiveness even with larger voids.

II. Methodology

This section introduces void filling using a conditional Denoising Diffusion Probabilistic Model (DDPM). The first two subsections explain the forward and reverse processes of DDPMs. The third shows how to adapt this model to DEM data, including specifics on training and inference.

A. Forward Diffusion Process

The forward process operates as a discrete-time Markov chain [31] that transforms a complex data distribution into a simpler one. Given data $\boldsymbol{x}_{t-1}$, the forward transition kernel perturbs the data by injecting Gaussian noise, with a variance β_t, thereby generating the data in the subsequent state, $\boldsymbol{x}_t$. The forward transition kernel [19] is defined as

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

Iterating Eq. 1 transforms $\boldsymbol{x}_0$, sampled from a complex distribution $q(\boldsymbol{x}_0)$, into $\boldsymbol{x}_T \sim q(\boldsymbol{x}_0) \approx \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ with $t \in \{0, \dots, T\}$. For efficient computation, Ho *et al.* [19] demonstrate that the reparametrization trick can be applied to directly compute $\boldsymbol{x}_t$ without sequential noise injection as

$$\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right), \qquad (2)$$

where $\bar{\alpha}_t = \prod_{s=0}^{t}(1-\beta_s)$.

Kyle Shih-Huang Lo and Jörg Peters are with Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL, 32611 USA. (e-mails: kyleshihhuanglo@ufl.edu, jorg@cise.ufl.edu)

Author K.L. was supported by Informatics Institute, University of Florida, UF Graduate School Preeminence Awards, and access to HiPerGator.

B. Reverse Diffusion Process

The reverse process transforms a simple distribution back into a complex one. Given $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, the reverse transition kernel $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ can be used to recover $\boldsymbol{x}_0$ from $\boldsymbol{x}_T$. According to [19], the reverse transition kernel can be regarded as a parameterized model, $\boldsymbol{\epsilon}_\theta$, that estimates the Gaussian noise in the current state, $\boldsymbol{x}_t$, and then subtracts the predicted noise from $\boldsymbol{x}_t$ to reconstruct the previous state $\boldsymbol{x}_{t-1}$. At present, we can ensure that the recovered $\hat{\boldsymbol{x}}_0$ bears similarity to data sampled from $q(\boldsymbol{x}_0)$. To exert control over this process, [32] shows that one can condition the model, $\boldsymbol{\epsilon}_\theta$, on supplementary information $\boldsymbol{y}$, formulated as

$$\hat{\boldsymbol{x}}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, \boldsymbol{y}, \bar{\alpha}_t)\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t \tag{3}$$

where $\hat{\boldsymbol{x}}_{t-1}$ denotes the estimated state at time step $t-1$, $\alpha_t = 1 - \beta_t$ and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

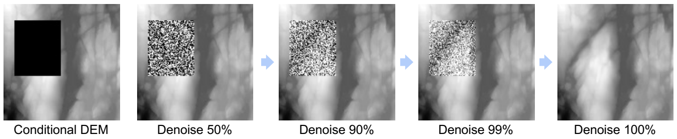

Fig. 1. Diffusion reversal in DEM void inpainting.

C. DEM Diffusion

Fig. 1 shows Diff-DEM progressively recovering missing terrain details. Conditioning on the void-containing DEM ensures that the inpainted details are coherently integrated with the existing data.

```
Algorithm 1 Diff-DEM: Inference
1: Given $\boldsymbol{y}$, $\boldsymbol{m}$
2: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
3: $\hat{\boldsymbol{x}}_T = \boldsymbol{m} \odot \boldsymbol{\epsilon} + \boldsymbol{y}$
4: for $t = T, \dots, 0$ do
5:     $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ if $t > 0$ else $\mathbf{0}$
6:     $\hat{\boldsymbol{x}}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\hat{\boldsymbol{x}}_t, \boldsymbol{y}, \bar{\alpha}_t)\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$
7:     $\hat{\boldsymbol{x}}_{t-1} \leftarrow \boldsymbol{m} \odot \hat{\boldsymbol{x}}_{t-1} + (\mathbf{1} - \boldsymbol{m}) \odot \hat{\boldsymbol{x}}_t$
8: end for
```

Inference. Algorithm 1 outlines the proposed inference method adopted from [19]. Given a void-containing DEM as a conditional image, represented by $\boldsymbol{y}$, and a binary void mask, $\boldsymbol{m}$, with pixel set to 1 for void, 0 otherwise, $\hat{\boldsymbol{x}}_T$, the DEM at time step T has voids filled with Gaussian noise, $\boldsymbol{\epsilon}$. Then, we repeatedly employ the trained model, $\boldsymbol{\epsilon}_\theta$, to estimate the noise at time step t, followed by subtracting noise from $\hat{\boldsymbol{x}}_t$ to generate $\hat{\boldsymbol{x}}_{t-1}$. Line 7 recovers any non-void pixels that were incorrectly predicted as noise and removed in line 6.

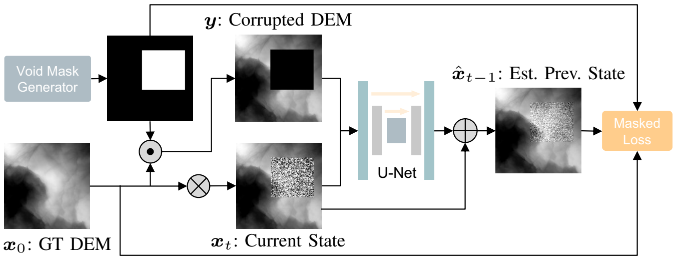
Void Mask Generator
$\boldsymbol{y}$: Corrupted DEM
$\hat{\boldsymbol{x}}_{t-1}$: Est. Prev. State
Masked Loss
U-Net
$\boldsymbol{x}_0$: GT DEM
$\boldsymbol{x}_t$: Current State

Fig. 2. The self-supervised pipeline for training transition kernel (U-net) of reverse diffusion process. $\odot$ denotes masking operation. $\bigotimes$ signifies transformation in Eq. 2, and $\bigoplus$ removes predicted noise from $\boldsymbol{x}_t$ Eq. 3.

```
Algorithm 2 Diff-DEM: Training
1: repeat
2:     $t \sim \mathcal{U}(\{1, \dots, T\})$
3:     $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$
4:     $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$
5:     $(h, w) \sim \mathcal{U}(\{h_{min}, \dots, h_{max}\} \times \{w_{min}, \dots, w_{max}\})$
6:     $(i, j) \sim \mathcal{U}(\{1, \dots, H-h\} \times \{1, \dots, W-w\})$
7:     $\boldsymbol{m} = \textsc{GenerateVoidMask}(i,\ j,\ w,\ h)$
8:     $\boldsymbol{y} = \boldsymbol{m} \odot \boldsymbol{x}_0$
9:     $\boldsymbol{x}_t = \boldsymbol{m} \odot \left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\right) + (\mathbf{1} - \boldsymbol{m}) \odot \boldsymbol{x}_0$
10:    Take a gradient descent step on
11:        $\nabla_\theta \| \boldsymbol{m} \odot (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, \boldsymbol{y}, \bar{\alpha}_t)) \|$
12: until converged
```

Training. Fig. 2 summarizes the proposed self-supervised pipeline for training the reverse transition kernel, $\boldsymbol{\epsilon}_\theta$, and Algorithm 2 details the training procedure. Initially, we randomly sample a time step t, Gaussian noise $\boldsymbol{\epsilon}$, and ground truth DEM, $\boldsymbol{x}_0$. Then, we create a mask, denoted as $\boldsymbol{m}$, of size $H \times W$ the same as $\boldsymbol{x}_0$. This mask contains a rectangle region set to 1 representing void, positioned at a random top-left corner with coordinates (i, j) and defined by random width w and height h. All pixels outside the void are set to 0, h_{min} and h_{max} are the minimum and maximum void mask heights, w_{min} and w_{max} the minimum and maximum widths. The void-containing DEM, $\boldsymbol{y}$, is synthesized by element-wise multiplication $\odot$ between $\boldsymbol{m}$ and $\boldsymbol{x}_0$. Next, we compute $\boldsymbol{x}_t$ via Eq. 2, where $\boldsymbol{m}$ prevents injecting noise outside of void. The gradient is computed exclusively within the void region.

Loss Function. Palette [32] demonstrates that training diffusion models with L1 loss tends to result in less hallucination compared to L2 loss, albeit with a slightly higher tendency for mode collapse. Given that precision is paramount in DEM inpainting, we opted for L1 loss in our training, see Line 11 of Algorithm 2.

Network Architecture. Diff-DEM uses the U-Net [33] architecture with attention mechanisms [34] in deeper layers, as described in Palette [32]. We process inputs as dual-channel $2 \times 128 \times 128$ images, incorporating the DEM containing voids, $\boldsymbol{y}$, and the step t approximation of the inpainted DEM, $\hat{\boldsymbol{x}}_t$. The output, $\boldsymbol{x}_t$, is formatted as a $1 \times 128 \times 128$ single-channel image. Structurally, our network includes a quartet of down-sampling and up-sampling stages, at 128×128, 64×64, 32×32, and 16×16 resolution. These stages are configured with channel dimensions of 64, 128, 256, and 512. Each module consists of a pair of residual blocks [35], integrating the attention mechanism [34] at the 32×32 and 16×16 resolution stages.

TABLE I
HEIGHT RESTORATION PERFORMANCE ON GAVRIIL'S DATASET [11]. **BOLD** DENOTES THE BEST.

Methods	MAE	RMSE	MS-SSIM	PSNR
IDW [4]	15.556	21.363	0.822	30.739
Kriging [5]	16.332	20.995	0.851	32.536
LR B-Spline [37]	13.437	18.361	**0.857**	32.633
PM-Diff [36]	16.178	21.176	0.847	31.525
Gavriil *et al.* [11]	14.477	20.229	0.819	30.691
Diff-DEM (Ours)	**9.698**	**14.575**	0.852	**34.222**

III. EXPERIMENTS

A. Quantitative Tests

Train and Test with Gavrill's Dataset and Benchmark. Diff-DEM is trained and its performance assessed using the train split and benchmark in [11]. For benchmarking, the pixels in ground truth DEM, identified by the void mask, are set to zero. The goal is to reconstruct these masked areas. Reconstruction is measured by Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) expressed in meters. Visual quality is measured by Multi-Scale Structural Similarity Index Measure (SSIM) and Peak Signal-to-Noise Ratio (PSNR) in dB. We compare to the leading GAN-based method with accessible source code [11], Perona-Malik diffusion (PM-Diff), a recent Digital Surface Model (DSM) inpainting method [36], and representative DEM inpainting techniques: Inverse Distance Weighting (IDW) [4], Kriging [5], and Local Refined B-Spline (LR B-Spline) [37].

Table I demonstrates that our approach not only achieves the highest accuracy in reconstruction, as indicated by the minimal MAE, but also exhibits the least significant error, highlighted by the minimal RMSE.

Train and Test with our Dataset and Benchmark. The dataset [11] provides only 63 DEMs (5051 × 5051) for training and 50 DEMs (256 × 256) for testing. We collect a richer dataset: 218 Norway nationwide 10-meter DEMs (5051×5051). To optimize data loading, we divided each DEM into 512 × 512 DEMs, enabling random 256 × 256 crops for augmentation. We split the data into 1,404 DEMs for testing and 12,640 for training, excluding any with a uniform zero height. For benchmarking, we randomly collect 1,000 DEMs from test split and generated void masks with different size. We classify difficulty based on the size of the mask: "easy" for 64-96 pixels, "medium" for 96-128 pixels, and "hard" for 128-160 pixels, both in height and width.

We compared our method to with IDW [4], Kriging [5], and PM-Diff [36]. Table II shows Diff-DEM surpassing the alternatives both in interpolation (for smaller voids) and inpainting (for larger voids).

B. Qualitative Tests

Fig. 3 compares Diff-DEM void inpainting to Gavriil *et al.* [11], LR B-spline [37], Kriging [5], IDW [4], and PM-Diff [36]. Five examples, arranged across two rows each, show the ground truth in the first column of the top row and the void mask on the bottom row. From the second column onward, the top row exhibits the inpainted outcome, with brighter pixel values denoting higher altitudes. The bottom row presents the Mean Absolute Error (MAE) in meters.

TABLE II
DEM RESTORATION PERFORMANCE ACROSS VARIOUS VOID SIZES ON OUR DATASET.

Methods	Easy		Medium		Hard	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
PM-Diff	11.357	14.692	19.243	24.939	44.288	56.188
IDW	11.981	16.310	16.739	22.999	21.474	29.860
Kriging	13.001	16.123	16.611	21.173	20.813	26.854
Diff-DEM (Ours)	**8.808**	**12.174**	**13.331**	**18.764**	**17.497**	**24.985**

TABLE III
IMPACT OF VARING DENOISING STEPS ON INPAINTING DEM WITH DIFFERENT VOID SIZE TESTED ON OUR DATASET.

Test Set	Steps					
	32	64	128	256	512	1024
Easy (64-96)	48.366	15.872	**8.407**	8.678	8.964	8.852
Hard (128-160)	64.440	41.419	18.106	17.852	**17.450**	17.414

Diff-DEM exhibits superior performance compared to other techniques across different types of terrain. In cases of extensive missing areas in complex terrains, Fig 3a, 3b, 3e, and 3f, LR B-spline [37], Kriging [5], and PM-Diff [36] tend to yield overly smooth reconstructions. This is can be attributed to the absence of prior knowledge about the terrain's characteristics. IDW generates a mosaic effect. Compared to non-learning-based methods, the learning-based model, proposed by Gavriil *et al.* [11], demonstrates an enhanced capability in restoring complex terrains. Diff-DEM not only improves on [11] in terms of accuracy but also exhibits reduced occurrences of hallucination, as depicted in Fig 3c, 3d, 3i, and 3j.

C. Implementation Details

To improve training and inference speeds, we reduced the resolution of the DEM from 256 × 256 to 128 × 128 for model input, then linearly upscaled the output back to the original size. The training used 8 NVIDIA Ampere A100 GPUs with a batch size of 512, completing in approximately 2 days. Variance scheduling and exponential moving average strategies were implemented following the guidelines in [32]. The learning rate was set to 7×10^{-5}, with a warm-up period during the first 10,000 steps, starting at a 0.2 scaling factor.

D. Sampling Steps & Inference Time

We tested the necessary denoising steps on our benchmark with easy tasks, where void sizes range from 64 to 96 pixels, and hard tasks of 128 to 160 pixels. Table III shows that, for an easy task, only 128 steps are required, and increasing the number of steps does not significantly enhance performance. For the harder task, 512 steps are necessary. We determined that performing inference with 512 steps is adequately effective for tasks involving various void sizes. On a single NVIDIA RTX 3090 GPU, using 512 steps, inference took an average of 2.5 seconds for one DEM (128 × 128 before up-scaling).

IV. Limitations

Learning-based methods are prone to hallucination, see e.g. [11]. Future work can adjust network architecture or apply multi-modal approaches to address this deficiency. Distillation methods, such as [38], can be used to improve the inference speed of the diffusion models.

V. Conclusions

Diff-DEM is the first denoising diffusion model for DEM void inpainting. Conditioning on the void-containing DEM, Diff-DEM progressively completes the DEM, capturing sharp and accurate terrain features. Diff-DEM quantitatively and qualitatively outperforms learning-based approaches, Gavriil *et al.* [11], as well as prominent non-learning-based methods, IDW [4], Kriging [5], LR B-Spline [37], and Perona-Malik diffusion [36]. Time step and inference speed analysis has revealed an optimal choice of inference steps that ensure high-quality reconstruction using only the necessary denoising steps. Lastly, we curated a complete DEM dataset to serve as foundational benchmark for future research and includes train-test splits, along with a categorization of voids mask based on their sizes.

References

[1] L. Fan and P. M. Atkinson, "Accuracy of digital elevation models derived from terrestrial laser scanning data," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1923–1927, 2015.

[2] G. Luzi, M. Pieraccini, D. Mecatti, L. Noferini, G. Macaluso, A. Tamburini, and C. Atzeni, "Monitoring of an alpine glacier by means of ground-based sar interferometry," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 3, pp. 495–499, 2007.

[3] N. Yokoya, K. Yamanoi, W. He, G. Baier, B. Adriano, H. Miura, and S. Oishi, "Breaking limits of remote sensing by deep learning from simulated data for flood and debris-flow mapping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2020.

[4] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*, 1968, pp. 517–524.

[5] H. I. Reuter, A. Nelson, and A. Jarvis, "An evaluation of void-filling interpolation methods for SRTM data," *International Journal of Geographical Information Science*, vol. 21, no. 9, pp. 983–1008, 2007.

[6] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE transactions on acoustics, speech, and signal processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[7] G. Grohman, G. Kroenung, J. Strebeck *et al.*, "Filling SRTM voids: The Delta surface fill method," *Photogrammetric Engineering and Remote Sensing*, vol. 72, no. 3, pp. 213–216, 2006.

[8] E. Luedeling, S. Siebert, and A. Buerkert, "Filling the voids in the SRTM elevation model — a TIN-based Delta surface approach," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, no. 4, pp. 283–294, 2007.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[10] G. Dong, F. Chen, and P. Ren, "Filling SRTM void data via conditional adversarial networks," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 7441–7443.

[11] K. Gavriil, G. Muntingh, and O. J. Barrowclough, "Void filling of digital elevation models with deep generative models," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 10, pp. 1645–1649, 2019.

[12] Z. Qiu, L. Yue, and X. Liu, "Void filling of digital elevation models with a terrain texture learning model based on generative adversarial networks," *Remote Sensing*, vol. 11, no. 23, p. 2829, 2019.

[13] D. Zhu, X. Cheng, F. Zhang, X. Yao, Y. Gao, and Y. Liu, "Spatial interpolation using conditional generative adversarial neural networks," *International Journal of Geographical Information Science*, vol. 34, no. 4, pp. 735–758, 2020.

[14] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[15] C. Zhang, S. Shi, Y. Ge, H. Liu, and W. Cui, "DEM void filling based on context attention generation model," *ISPRS International Journal of Geo-Information*, vol. 9, no. 12, p. 734, 2020.

[16] G. Zhou, B. Song, P. Liang, J. Xu, and T. Yue, "Voids filling of DEM with multiattention generative adversarial network model," *Remote Sensing*, vol. 14, no. 5, p. 1206, 2022.

[17] G. Dong, W. Huang, W. A. Smith, and P. Ren, "A shadow constrained conditional generative adversarial net for SRTM data restoration," *Remote Sensing of Environment*, vol. 237, p. 111602, 2020.

[18] S. Li, G. Hu, X. Cheng, L. Xiong, G. Tang, and J. Strobl, "Integrating topographic knowledge into deep learning for the void-filling of digital elevation models," *Remote Sensing of Environment*, vol. 269, p. 112818, 2022.

[19] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[20] Y. Xiao, Q. Yuan, K. Jiang, J. He, X. Jin, and L. Zhang, "Ediffsr: An efficient diffusion probabilistic model for remote sensing image super-resolution," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[21] J. Liu, Z. Yuan, Z. Pan, Y. Fu, L. Liu, and B. Lu, "Diffusion model with detail complement for super-resolution of remote sensing," *Remote Sensing*, vol. 14, no. 19, p. 4834, 2022.

[22] T. An, B. Xue, C. Huo, S. Xiang, and C. Pan, "Efficient remote sensing image super-resolution via lightweight diffusion models," *IEEE Geoscience and Remote Sensing Letters*, 2023.

[23] M. Xu, J. Ma, and Y. Zhu, "Dual-diffusion: Dual conditional denoising diffusion probabilistic models for blind super-resolution reconstruction in rsis," *IEEE Geoscience and Remote Sensing Letters*, 2023.

[24] M. V. Perera, N. G. Nair, W. G. C. Bandara, and V. M. Patel, "Sar despeckling using a denoising diffusion probabilistic model," *IEEE Geoscience and Remote Sensing Letters*, 2023.

[25] S. Xiao, L. Huang, and S. Zhang, "Unsupervised sar despeckling based on diffusion model," in *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2023, pp. 810–813.

[26] J. Lochner, J. Gain, S. Perche, A. Peytavie, E. Galin, and E. Guérin, "Interactive authoring of terrain using diffusion models," in *Computer Graphics Forum*, vol. 42, no. 7. Wiley Online Library, 2023, p. e14941.

[27] S. Khanna, P. Liu, L. Zhou, C. Meng, R. Rombach, M. Burke, D. Lobell, and S. Ermon, "Diffusionsat: A generative foundation model for satellite imagery," *arXiv preprint arXiv:2312.03606*, 2023.

[28] R. Jing, F. Duan, F. Lu, M. Zhang, and W. Zhao, "Denoising diffusion probabilistic feature-based network for cloud removal in sentinel-2 imagery," *Remote Sensing*, vol. 15, no. 9, p. 2217, 2023.

[29] X. Zhao and K. Jia, "Cloud removal in remote sensing using sequential-based diffusion models," *Remote Sensing*, vol. 15, no. 11, p. 2861, 2023.

[30] K. Norwegian Mapping Authority, "DTM 10 Terrain Model (UTM33)," https://hoydedata.no/LaserInnsyn2/, 2018, accessed: Dec. 1, 2023.

[31] K. L. Chung, "Markov chains," *Springer-Verlag, New York*, 1967.

[32] 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*, 2022, pp. 1–10.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] P. Biasutti, J.-F. Aujol, M. Brédif, and A. Bugeau, "Diffusion and inpainting of reflectance and height LiDAR orthoimages," *Computer Vision and Image Understanding*, vol. 179, pp. 31–40, 2019.

[37] V. Skytt, O. Barrowclough, and T. Dokken, "Locally refined spline surfaces for representation of terrain data," *Computers & Graphics*, vol. 49, pp. 58–68, 2015.

[38] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," *International Conference on Learning Representations*, 2022.

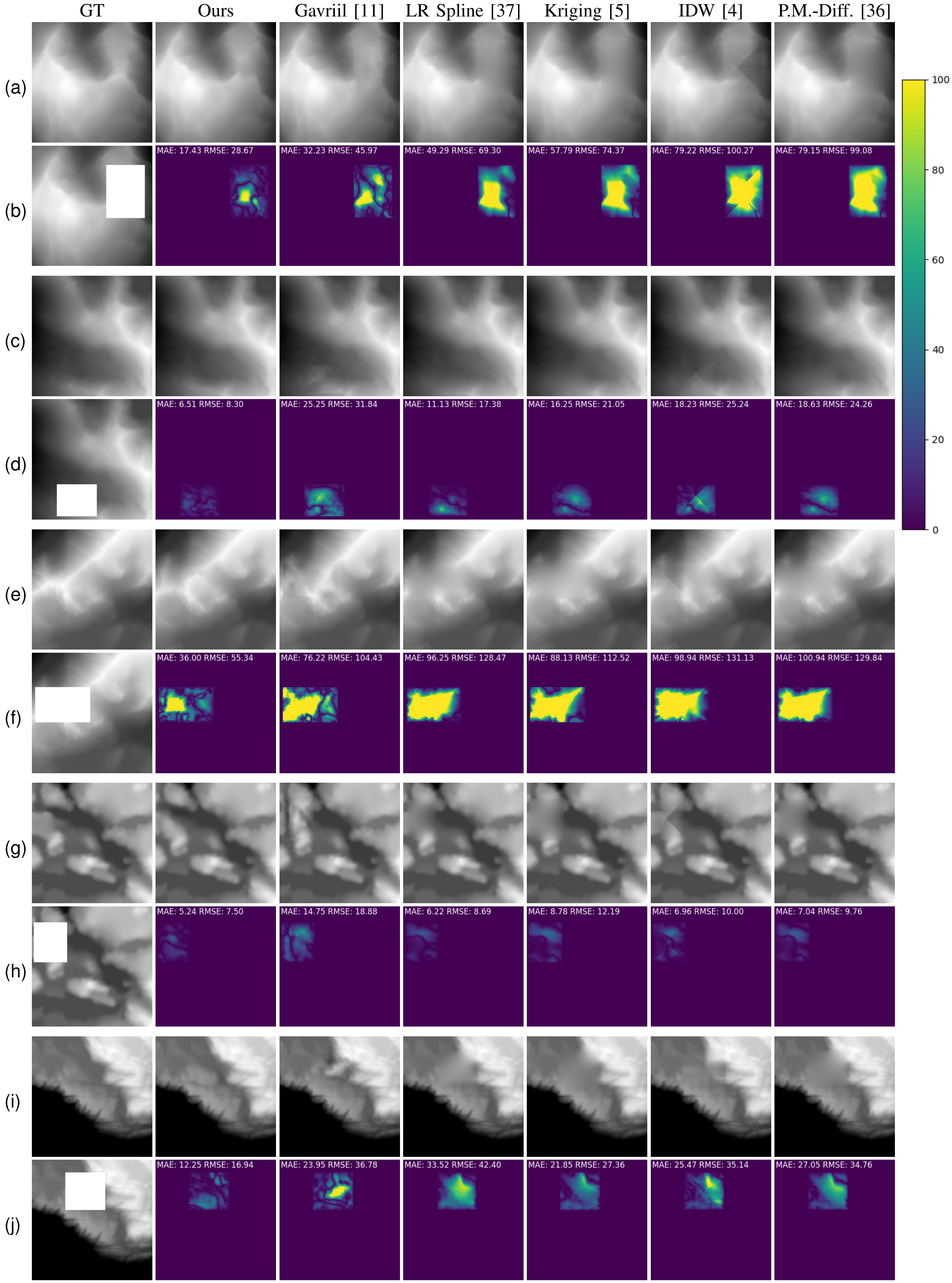

Fig. 3. The qualitative comparison of DEM void reconstructions on benchmark [11]. Images showing less yellow or green in in the white reconstruction window are better.