# SAR Data Generation Using Denoising Diffusion Probabilistic Models

Trey Crump
*Yale University*
New Haven, CT, USA
trey.crump@yale.edu

Sastry Kompella
*Nexcepta Inc.*
Gaithersburg, MD, USA
skompella@nexcepta.com

Clement Kam
*Naval Research Laboratory*
Washington, DC, USA
clement.k.kam@nrl.navy.mil

Leandros Tassiulas
*Yale University*
New Haven, CT, USA
leandros.tassiulas@yale.edu

***Abstract*—The lack of measured data is a common issue plaguing the synthetic aperture radar (SAR) automatic target recognition (ATR) community. For deep learning approaches such as convolutional neural networks (CNNs), a large amount of data is needed to achieve desirable recognition rates. Several solutions to synthetically generating SAR data have been proposed, including using radar signature simulators and generative AI (GenAI). Radar signature simulators are expensive, proprietary, and rely on the ability to correctly model target shape, pose, radar cross section (RCS), and background clutter. Generative adversarial networks (GANs) have been applied to the problem of synthetically generating SAR data but suffer from convergence issues and often have worse generative performance compared to denoising diffusion probabilistic models (DDPMs). In this paper, we propose a conditional DDPM for synthetically generating SAR data. We show the benefit of using conditional DDPMs to generate synthetic SAR data by both GenAI and ATR metrics. Moreover, we show that diffusion-generated data improves ATR performance and permits realistic dataset variability.**

***Index Terms*—Synthetic aperture radar, automatic target recognition, generative AI, denoising diffusion probabilistic models.**

## I. Introduction

Synthetic aperture radar (SAR) is an active sensing platform that allows for high-resolution imaging through virtual aperture synthesis. By emitting microwave signals and analyzing the returned echoes, SAR systems create detailed images of surfaces, regardless of lighting conditions. SAR maintains distinct advantages over optical systems, allowing for operation under unfavorable operating conditions such as poor weather, smoke, and dust. Moreover, SAR allows for the extraction of target pose, shape, and scattering characteristics. These desirable qualities lend use to multiple military and surveillance applications, such as detecting and tracking vehicles, ships, and other objects.

Classifying target class within SAR images, or SAR automatic target recognition (ATR), has been widely studied in the past few decades. More recently, deep learning has been used in ATR, using state-of-the-art computer vision classification networks to classify between target classes [1], [2]. To perform well on the classification task, deep learning models require large datasets with targets in various positions and backgrounds. However, SAR data is scarce due to the nature of employed use and collection time; requiring the user to radiate large amounts of electromagnetic energy to collect a single frame of a target at a given azimuth, elevation, and position. Given this limitation, SAR dataset collections can have a target class at only a few azimuth angles and a single target pose.

To remedy dataset scarcity, several solutions to synthetically generate SAR data have been proposed. These approaches can be grouped into two groups: using electromagnetic software and CAD models to generate scattering signatures and using generative AI (GenAI) to generate similar images to the training set. Radar signature simulators such as Xpatch [3], allow for the production of synthetic SAR images through electromagnetic simulations on CAD models of targets. These simulators are often expensive and proprietary, prohibiting open research. Moreover, dataset fidelity depends on the ability to correctly model target shape and pose using CAD, estimate the background clutter, and estimate the radar cross section (RCS) of the target [4]–[6]. Modeling assumptions can result in poor ATR performance when training on synthetic signatures and testing on measured data.

GenAI models such as generative adversarial networks (GANs) [7]–[9], have been used in the task of generating synthetic SAR image datasets. Conditional GANs (cGANs) use a conditional generator $G$ and a discriminator with classification ($DwC$) to generate labeled SAR data. Generator $G$ is conditioned on class embeddings, which guide it to generate SAR images corresponding to a predefined target class. The discriminator $DwC$ classifies whether the data, generated by $G$ or the original dataset, belongs to one of the target classes. After classifying the target class, the $DwC$ distinguishes between whether the data was produced by $G$ or is part of the original dataset. The $DwC$ ensures that the generated SAR images are indistinguishable from real images and are correctly labeled. While GANs can produce faithful synthetic images, it is well documented that they suffer from numerous issues such as mode collapse, instability, and non-convergence, requiring much more involved techniques to produce high-quality images [10].

We propose the use of denoising diffusion probabilistic models (DDPMs) to generate SAR data. DDPMs achieve better generative results than GANs, have improved training

This work was supported by the National Science Foundation SWIFT under Grant 114237, Office of Naval Research N0014-22-1-2147, and Army Research Office W911NF-23-1-0088.

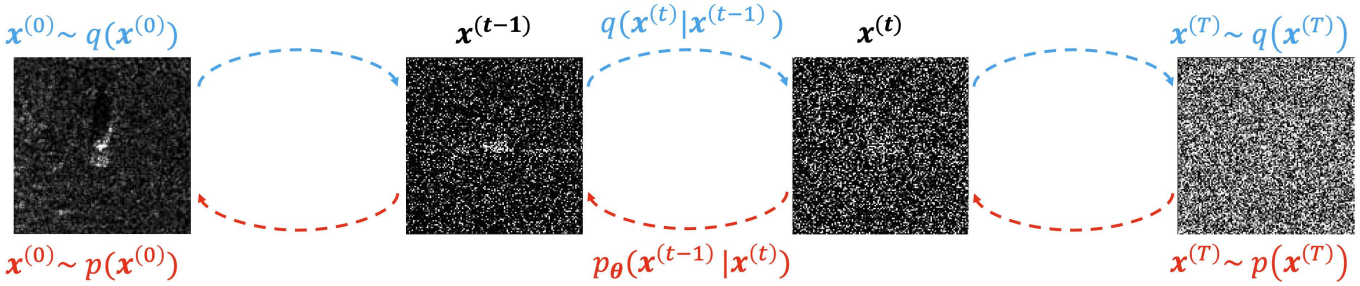



Fig. 1: Forward and reverse diffusion trajectories for SAR image generation.

stability, and promote better sample diversity [11]. Section II outlines the conditional DDPM architecture used for generating SAR images, conditioned on both target class and azimuth angle. Section III outlines the dataset used, training details, and evaluation criteria for the generated samples. Finally, we evaluate the diffusion-generated samples in Section IV, showing generative results and ATR accuracy.

## II. Diffusion Model

Diffusion models have received a large amount of attention in recent years, given their excellent generative capabilities. DDPMs learn the latent structure of data by adding Gaussian noise to the original base images in a process known as the forward trajectory. This process involves gradually corrupting the data with Gaussian noise over several time steps, creating a sequence of noisy images. Subsequently, a neural network is trained to denoise each image at each time step, a process referred to as the reverse trajectory [12]. During sampling, the reverse trajectory is used to generate new data. Starting from a randomly sampled noisy image, the neural network (trained on the reverse trajectory) iteratively removes the noise step-by-step, gradually transforming the noisy image back into a realistic image. Together, the forward and reverse trajectories form a Markov chain, as illustrated in Fig. 1.

### A. Diffusion Process

Assume we have a collection of SAR images $\mathbf{x}$ with label embeddings $\mathbf{y}_{\text{class}}$ and $\mathbf{y}_{\text{az}}$, where the embeddings correspond to target class and target azimuth, respectively. Let our collection of images and label embeddings be modeled by some distribution of data $q(\mathbf{x}^{(0)}, \mathbf{y}_{\text{class}}, \mathbf{y}_{\text{az}})$, where $q(\mathbf{x}^{(0)})$ denotes the original, clean images.

The forward diffusion trajectory transforms the original distribution $q(\mathbf{x}^{(0)})$ into a standard Gaussian distribution over a series of steps $T$ by applying a sequence of Gaussian transitions

$$q\left(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}\right) = \mathcal{N}\left(\mathbf{x}^{(t)}; \sqrt{1-\beta_t}\mathbf{x}^{(t-1)}, \beta_t\mathbf{I}\right), \quad (1)$$

where $\beta_t$ are the variances of the noise added at each step. The resulting process after $T$ steps is

$$q\left(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T)}\right) = q\left(\mathbf{x}^{(0)}\right) \prod_{t=1}^{T} q\left(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}\right). \quad (2)$$

In practice, we do not need to apply $q$ repeatedly to sample from $x^{(t)} \sim q(\mathbf{x}^{(t)}|\mathbf{x}^{(0)})$, we instead express $q(\mathbf{x}^{(t)}|\mathbf{x}^{(0)})$ as a Gaussian distribution [11]. With $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$, we have

$$q(\mathbf{x}^{(t)}|\mathbf{x}^{(0)}) = \sqrt{\bar{\alpha}_t}\mathbf{x}^{(0)} + \epsilon\sqrt{1-\bar{\alpha}_t}, \ \epsilon \sim \mathcal{N}(0, \mathbf{I}). \quad (3)$$

In the reverse diffusion trajectory, the model learns to reconstruct the data from the noisy distribution back to the original distribution (denoising) by estimating

$$p_\theta\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right) = \mathcal{N}\left(\mathbf{x}^{(t-1)}; \mu_\theta(\mathbf{x}^{(t)}, t), \Sigma_\theta(\mathbf{x}^{(t)}, t)\right), \quad (4)$$

where mean $\mu_\theta(\mathbf{x}^{(t)}, t)$ and diagonal covariance matrix $\Sigma_\theta(\mathbf{x}^{(t)}, t)$ are parameterized by a neural network. The reverse process begins from

$$p\left(\mathbf{x}^{(T)}\right) = \mathcal{N}\left(\mathbf{x}^{(T)}; \mathbf{0}, \mathbf{I}\right), \quad (5)$$

and aims to recover $\mathbf{x}^{(0)}$ from $\mathbf{x}^{(T)}$ through learned reverse transitions.

We train a model so that the distribution $p(\mathbf{x}^{(0)})$ learns the true data distribution $q(\mathbf{x}^{(0)})$. To do so, we train a model $\epsilon_\theta(\mathbf{x}^{(t)}, t)$ to predict $\epsilon$ from Eq. 3, with the objective function

$$L = \mathbb{E}_{t\sim[1,T], \mathbf{x}^{(0)}\sim q(\mathbf{x}^{(0)}), \epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[||\epsilon - \epsilon_\theta(\mathbf{x}^{(t)}, t)||^2\right]. \quad (6)$$

During the sampling process, the posterior mean $\mu_\theta(\mathbf{x}^{(t)}, t)$ is provided from $\epsilon_\theta(\mathbf{x}^{(t)}, t)$,

$$\mu_\theta(\mathbf{x}^{(t)}, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}^{(t)} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}^{(t)}, t)\right). \quad (7)$$

Instead of learning $\Sigma_\theta(\mathbf{x}^{(t)}, t)$, we instead fix the covariance as a constant $\tilde{\beta}_t\mathbf{I}$, where $\tilde{\beta}_t$ is the posterior variance [13]

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \quad (8)$$

### B. Framework

U-Net [14] is employed to model noise at each reverse timestep in the diffusion process. It consists of downsampling and upsampling blocks that encode images into compressed representations and reconstruct them. Residual blocks with linear attention refine feature maps after convolution, enhancing key features [15]. The bottleneck block applies general attention to capture long-range dependencies essential for high-quality reconstructions. We use dimension multipliers of

(1x, 2x, 4x, 8x) to progressively increase the channels at deeper layers. Sinusoidal positional embeddings encode the timestep $t$, while class and azimuth angle information are encoded via multi-layer perceptrons (MLPs).

The noise schedule determines how much noise is added at each step in the diffusion process. In [16], it was determined that while the linear scheduling performs well for high-resolution images, it is sub-optimal for lower-resolution images. As SAR images are typically lower-resolution images, we use a cosine noise schedule, defined by

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \; f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)^2, \tag{9}$$

with $\beta_t = 1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}$. This scheduler is designed to have a linear drop-off of $\bar{\alpha}_t$ in the middle of the process while changing very little near the extreme points of $t = 0$ and $t = T$. As in [16], we use an offset value of $s = 0.008$ and clip $\beta_t$ to be no larger than 0.999 to facilitate accurate learning of $\epsilon$ and to prevent collapse at $t = T$.

We implement classifier-free guidance to add the conditional information for both target label and azimuth [17]. Classifier-free guidance uses a single neural network to train an unconditional DDPM $p_\theta(\mathbf{x})$ parameterized through score estimator $\epsilon_\theta(\mathbf{x})$ together with a conditional model $p_\theta(\mathbf{x}|\mathbf{y}_{\text{class}}, \mathbf{y}_{\text{az}})$ parameterized through $\epsilon_\theta(\mathbf{x}|\mathbf{y}_{\text{class}}, \mathbf{y}_{\text{az}})$. The unconditional and conditional models are jointly trained, randomly setting the embeddings $\mathbf{y}_{\text{class}}, \mathbf{y}_{\text{az}}$ to the unconditional null embedding $\emptyset, \emptyset$ with probability $p_{\text{drop}}$. The sampling process is given by

$$\tilde{\epsilon}_\theta(\mathbf{x}, \mathbf{y}_{\text{class}}, \mathbf{y}_{\text{az}}) = (1+w)\epsilon_\theta(\mathbf{x}, \mathbf{y}_{\text{class}}, \mathbf{y}_{\text{az}}) - w\epsilon(\mathbf{x}), \tag{10}$$

where $w$ is the strength of the classifier-free guidance.

## III. Experiment

### A. Dataset Details

The Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset [18], collected by Sandia National Laboratory, is utilized for both training and testing. The collection platform is an X-band SAR sensor in one-foot resolution spotlight mode. The MSTAR training and testing datasets consist of 3 classes, `bmp2`, `btr70`, and `t72`, captured at azimuth angles of $0-360^\circ$ and elevation angles of $17^\circ$ and $15^\circ$ for the training and testing datasets respectively. The training dataset comprises 1,622 samples of SAR images, while the test dataset includes 1,351 samples, presenting a relatively small dataset with limited variation in target poses.

In this study, a subset of the original training data is also considered. Rather than using the complete set of azimuth angles for each target, we assume that only specific subsets of azimuth angles have been collected, mirroring real-world operating conditions. The azimuth angle ranges $0 - 120^\circ$, $120-240^\circ$, and $240-360^\circ$ correspond to the azimuth ranges of `bmp2`, `btr70`, and `t72` respectively. Table I shows a detailed overview of the datasets.

TABLE I: Training dataset details.

| Dataset | Target | Azimuth Range | Samples |
|---|---|---|---|
| Original Train | `bmp2` | $0-360^\circ$ | 698 |
| | `btr70` | $0-360^\circ$ | 233 |
| | `t72` | $0-360^\circ$ | 691 |
| Subset | `bmp2` | $0-120^\circ$ | 239 |
| | `btr70` | $120-240^\circ$ | 73 |
| | `t72` | $240-360^\circ$ | 244 |
| Test | `bmp2` | $0-360^\circ$ | 581 |
| | `btr70` | $0-360^\circ$ | 194 |
| | `t72` | $0-360^\circ$ | 576 |

### B. Evaluation Criterion

A ResNet34 convolutional neural network (CNN) [19] is trained to distinguish between the `bmp2`, `btr70`, and `t72` classes, allowing for evaluation of generated datasets in the task of ATR. On the original training and testing datasets, ResNet34 achieves a $94.4\%$ test accuracy using an Adam optimizer with a learning rate of 1e-4 and a cross-entropy loss function, making it well suited to evaluate the viability of using GenAI data in ATR training. Each dataset is trained for 30 epochs and three passes each, averaging the test accuracy between all passes.

Inception Score (IS) [20], Fréchet Inception Distance (FID) [21], and Kernel Inception Distance (KID) [22] are popular metrics for assessing generative image models, utilizing a pre-trained Inception-v3 model to evaluate how closely the generated images resemble the original dataset in terms of diversity and fidelity. IS evaluates the clarity and diversity of generated images by calculating the average Kullback–Leibler (KL) divergence between the conditional label distribution $p(y|x)$ and the marginal label distribution $p(y)$. FID assesses the similarity between the distributions of real and generated images, calculated using the Fréchet distance between two multivariate Gaussian distributions, defined by the means and covariances of the real and generated datasets. KID is an unbiased estimator that measures the squared maximum mean discrepancy (MMD) between Inception-v3 representations, providing a more robust measure than FID, which is sensitive to the number of samples.

Lower FID and KID scores typically indicate better sample variety and quality while higher values of IS score tend to have higher quality. Standard implementations of IS, FID, and KID rely on a pre-trained Inception-v3 model that was trained on natural images. To adapt these metrics to our dataset, we trained an Inception-v3 model on the entire MSTAR dataset. We scaled images from 128x128 to 299x299 and converted our single-channel SAR images into 3-channel RGB images to fit the Inception-v3 architecture. Using ATR accuracy, IS, FID, and KID scores, allows us to measure the quality, variability, and impact of diffusion-generated samples in ATR.

### C. Training and Sampling Details

In the training process, azimuth angles are binned into $5^\circ$ increments, given the minimal visual difference between

adjacent bins. Provided that the training dataset is small, it is necessary to bin the azimuth angles to allow the model to effectively learn the azimuth angle information without overfitting on specific pose angles and to increase the amount of data per azimuth embedding.

In classifier-free guidance, the probability of dropping the conditional embeddings is set to $p_{drop} = 0.5$, balancing the training between unconditional and conditional information. The conditional DDPMs are trained for 250k iterations, with a learning rate of 1e-4, and a batch size of 8. Each image is normalized to be in the range of $[-1, 1]$ and is resized to 128x128 before training the DDPM. During the sampling procedure, each class is sampled for azimuth angles of $0 - 360°$, generating images of targets at both seen and unseen azimuth angles. To establish a baseline, we also train a conditional deep convolutional GAN (cDCGAN) and a conditional self-attention GAN (cSAGAN).

The cDCGAN architecture follows the design described in [23], with the addition of class and azimuth conditioning in both the generator and discriminator. The generator employs convolutional blocks to progressively upsample the image, increasing its spatial dimensions. Conversely, the discriminator utilizes convolutional blocks to downsample the image. Additionally, the cSAGAN [24], is based on the DCGAN architecture but incorporates self-attention layers to focus on relevant spatial regions during generation and discrimination, resulting in improved generative performance.

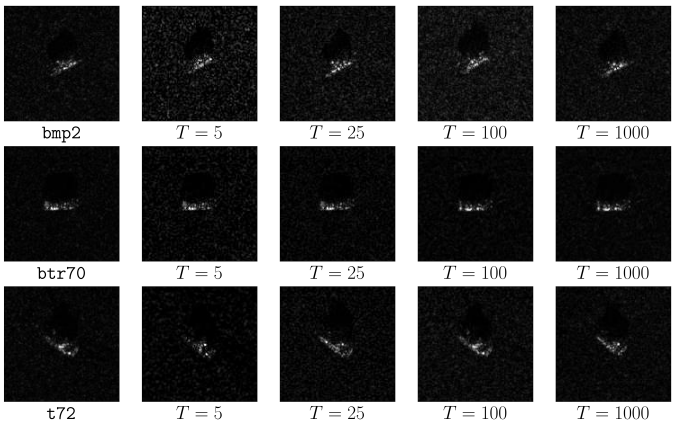
Both the cDCGAN and cSAGAN are trained using the same dataset and conditional information (class labels and azimuth angles) as the DDPM. The hyperparameters of the GAN models were optimized with respect to IS, FID, and KID to ensure stable training and high-quality image generation. Through this optimization process, we identified that a batch size of 64, a latent dimension of 100, and a learning rate of 2e-3, trained over 1000 epochs, yielded the best generative results for the MSTAR dataset. These GAN architectures serve as benchmarks to evaluate the relative performance of the DDPM in generating synthetic SAR images.

## IV. Results

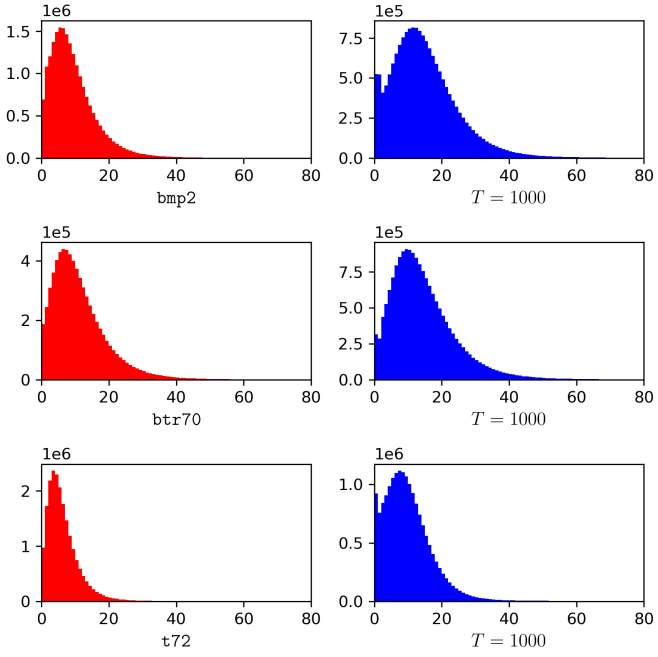
### A. Generative Results

In this section, we first evaluate how the number of timesteps $T$ in the diffusion process and the classifier-free guidance weight $w$ influence the quality and diversity of the generated samples. These factors are quantified using IS, FID, and KID which serve as our primary metrics for assessing image quality and variation. We vary the number of timesteps during the diffusion process and adjust the classifier-free guidance strength to explore their impact on generative performance.

In Fig. 2a, the impact of the number of timesteps $T$ in the diffusion process on generative quality in SAR images is illustrated. Shorter processes, such as $T = 5$ and $T = 25$, are more noisy compared to longer processes like $T = 1000$. Shorter diffusion processes do not provide sufficient steps for the neural network to accurately denoise the images,




(a)




(b)

Fig. 2: (a) Truth and diffusion generated images for $T = 5$, 10, 25, and 1000 with $w = 3$, (b) Histograms of pixel densities for truth and diffusion generated images with $T = 1000$ and $w = 3$.

leading to higher residual noise in the generated samples. Conversely, longer processes allow for more gradual and precise denoising, thereby improving image quality. Fig. 2b displays histograms of the pixel densities for both the ground truth images and those generated by the diffusion process for $T = 1000$ timesteps. The diffusion-generated images successfully replicate the distribution of the training set while also introducing some variation in pixel intensity, resulting in heavier tails in the pixel distributions.

The influence of classifier-free guidance strength on the diffusion process is shown in Fig. 3. For $w = 0$, the unconditional case is recovered, which reduces the quality

of the generated samples as it lacks the influence of class-specific conditioning. As $w$ increases, the model progressively leverages more of the conditional information, enhancing the fidelity of generated images to their target classes and azimuth angles. However, excessive guidance strength, such as $w = 7$, can introduce significant noise into the generated images. Overly strong guidance leads to overfitting on the guidance signal, causing the model to produce artifacts and reducing the overall quality of the images, as illustrated in the figure. Moreover, our experiments demonstrate that moderate guidance strengths, such as $w = 1$ or $w = 3$, provide a good balance between image fidelity and diversity. At these settings, the generated images effectively capture the desired attributes without introducing excessive noise.

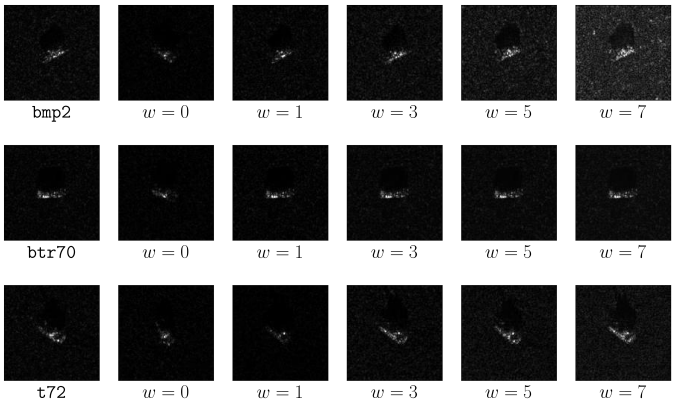



Fig. 3: Impact of guidance strength $w$ on generative results for $T = 1000$.

The interplay between $T$ and $w$ is crucial for optimizing generative performance. Our results indicate that using a higher number of timesteps in conjunction with a low classifier-free guidance weight $w$ yields the best performance in terms of IS, FID, and KID, as shown in Table II. This balance ensures that the diffusion process has enough steps for accurate denoising while the guidance strength effectively directs the generation process without overfitting on specific targets or clutter attributes. Furthermore, we observe that balanced DDPM models outperform both cDCGANs and cSAGANs in generating synthetic SAR samples.

### *B. Classification Results*

The classification results are provided in Table III. We observe that ATR accuracy significantly degrades when we only have a subset of the original training dataset, lowering from 94.4% total accuracy to 34.6%. When diffusion-generated samples are added to our initial subset, we observe a 28.4% boost in ATR accuracy, raising it to 63.0%. This improvement can be attributed to the increased diversity in the training data and the diffusion model filling in missing azimuth samples. The diffusion model effectively generates a wider array of synthetic samples that mimic varying operational conditions and target configurations, thereby enriching the data distribution. This added diversity strengthens the model's ability to generalize to new, unseen data during testing.

TABLE II: Performance metrics for varying models, timesteps $T$, and strengths $w$.

| **Model** | $T$ | $w$ | **IS** (↑) | **FID** (↓) | **KID** (↓) |
|---|---|---|---|---|---|
| cDCGAN | - | - | 1.193 ± 0.045 | 0.998 | 17.50 |
| cSAGAN | - | - | 1.283 ± 0.042 | 0.510 | 5.491 |
| cDDPM | 5 | 0 | 1.537 ± 0.025 | 1.177 | 14.69 |
| | 5 | 1 | 1.407 ± 0.135 | 0.210 | 3.034 |
| | 5 | 3 | 1.537 ± 0.069 | 3.411 | 116.0 |
| | 5 | 5 | 1.522 ± 0.066 | 6.905 | 237.0 |
| | 5 | 7 | 1.513 ± 0.058 | 11.12 | 356.0 |
| | 25 | 0 | **1.645 ± 0.023** | 17.16 | 443.0 |
| | 25 | 1 | 1.508 ± 0.119 | 1.791 | 52.96 |
| | 25 | 3 | 1.544 ± 0.077 | 2.400 | 73.54 |
| | 25 | 5 | 1.551 ± 0.066 | 2.846 | 93.24 |
| | 25 | 7 | 1.579 ± 0.062 | 4.670 | 155.0 |
| | 100 | 0 | 1.222 ± 0.031 | 0.744 | 19.89 |
| | 100 | 1 | 1.397 ± 0.140 | 0.161 | 3.251 |
| | 100 | 3 | 1.534 ± 0.102 | 2.167 | 67.43 |
| | 100 | 5 | 1.544 ± 0.100 | 2.905 | 102.9 |
| | 100 | 7 | 1.595 ± 0.103 | 8.235 | 279.8 |
| | 1000 | 0 | 1.125 ± 0.019 | 2.123 | 44.11 |
| | 1000 | 1 | 1.327 ± 0.152 | **0.004** | **0.259** |
| | 1000 | 3 | 1.515 ± 0.116 | 1.560 | 45.09 |
| | 1000 | 5 | 1.477 ± 0.117 | 0.940 | 19.62 |
| | 1000 | 7 | 1.603 ± 0.136 | 6.106 | 207.2 |

To further investigate the impact of diffusion-generated samples on ATR, we randomly sampled 80%, 60%, and 20% of the original dataset to create different training subsets with random gaps in the azimuth range of each target class. The results consistently show that the addition of diffusion-generated samples enhances accuracy across all levels of data reduction, indicating that diffusion-generated samples can effectively compensate for the reduced amount of real data, providing a valuable data augmentation strategy. The consistent improvement across various dataset sizes highlights the robustness of diffusion-generated samples in enhancing ATR training. The diffusion model effectively enriches the training set with diverse and realistic synthetic samples, improving the model's overall performance and generalization capability.

The inclusion of high-quality, diverse synthetic data helps to alleviate the challenge of overfitting to a limited number of samples and classes. This approach broadens the model's exposure to a wide range of target signatures and background noise conditions, which is crucial in SAR applications where even slight variations in angle, lighting, or environmental context can cause notable changes in target appearance. While both DDPM and GAN-based methods improve ATR accuracy, DDPM models consistently surpass GANs in generating synthetic samples. Additionally, when the dataset is limited to 20% or 60% of its original size, the integration of synthetic data leads to significant gains in ATR accuracy. This improvement can be attributed to the generative models' ability to approximate missing portions of the training distribution. However, in cases where the dataset is more complete, such as 80% of the original size, low-quality synthetic data tends

TABLE III: ATR results for datasets.

| Dataset | Accuracy |
|---|---|
| Training Baseline | 94.4% |
| Training Subset | 34.6% |
| Training Subset + cDDPM | (+28.4%) 63.0% |
| Training Subset + cDCGAN | 60.8% |
| Training Subset + cSAGAN | 60.2% |
| 80% Random Sample | 93.5% |
| 80% Random Sample + cDDPM | (+2.1%) 95.6% |
| 80% Random Sample + cDCGAN | 81.2% |
| 80% Random Sample + cSAGAN | 90.7% |
| 60% Random Sample | 49.3% |
| 60% Random Sample + cDDPM | (+39.8%) 89.1% |
| 60% Random Sample + cDCGAN | 78.6% |
| 60% Random Sample + cSAGAN | 79.8% |
| 20% Random Sample | 43.6% |
| 20% Random Sample + cDDPM | (+39.9%) 83.5% |
| 20% Random Sample + cDCGAN | 78.2% |
| 20% Random Sample + cSAGAN | 76.6% |

to negatively impact ATR accuracy, underscoring that data fidelity is more critical than approximating the training distribution.

The observed boost in ATR accuracy demonstrates the potential of incorporating diffusion-generated data into the ATR training process. Despite the MSTAR dataset presenting a relatively easy classification problem and providing a strong baseline, the presented improvement in accuracy highlights the effectiveness of synthetic data in enhancing model performance under constraints on measured data. The increase in accuracy suggests that diffusion-generated samples introduce valuable diversity, filling gaps and enriching the training set with variations that might be absent in the original dataset.

## V. Conclusion

In this work, we have demonstrated the effectiveness of denoising diffusion probabilistic models (DDPMs) in generating high-quality SAR images, conditioned on target class and azimuth angle. We showed that diffusion models enhance the fidelity and diversity of SAR datasets, improving ATR accuracy. Provided the extremely limited size of our training dataset, this highlights the robustness of conditional DDPMs in learning target properties in SAR images. These models successfully generate SAR images that faithfully model the measured background clutter and target scattering characteristics while introducing variability.

In future work, we aim to examine the potential of diffusion models in additional SAR problems. We plan to investigate specific aspects of super-resolution to enhance image detail at finer scales, apply target inpainting for reconstructing obscured or incomplete SAR imagery, and explore the integration of diffusion models in Video SAR for dynamic surveillance applications.

## References


[1] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmentation for sar target recognition," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, pp. 364–368, 2016.

[2] S. Chen and H. Wang, "Sar target recognition based on deep learning," in *2014 International Conference on Data Science and Advanced Analytics (DSAA)*, 2014, pp. 541–547.

[3] D. Andersh, J. Moore, S. Kosanovich, D. Kapp, R. Bhalla, R. Kipp, T. Courtney, A. Nolan, F. German, J. Cook, and J. Hughes, "Xpatch 4: the next generation in high frequency electromagnetic modeling and simulation software," in *Record of the IEEE 2000 International Radar Conference [Cat. No. 00CH37037]*, 2000, pp. 844–849.

[4] W. Huo, Y. Huang, J. Pei, Y. Zhang, and J. Yang, "A new sar image simulation method for sea-ship scene," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 721–724.

[5] C. L. Barbu, C. Cochin, and E. Everaere, "Improvement of sar/isar simulation in mocem v4.5 by computation of the ship bow wave," in *2019 International Radar Conference (RADAR)*, 2019, pp. 1–5.

[6] S. Kim, J. Yu, and M.-H. Ka, "Sar image synthesis with chirp scaling algorithm of 3d cad model using em simulator," in *2015 IEEE 5th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*, 2015, pp. 72–75.

[7] Z. Cui, M. Zhang, Z. Cao, and C. Cao, "Image data augmentation for sar sensor via generative adversarial nets," *IEEE Access*, vol. 7, pp. 42 255–42 268, 2019.

[8] P. Wang and V. Patel, "Generating high quality visible images from sar images using cnns," in *2018 IEEE Radar Conference (RadarConf18)*, 2018, pp. 0570–0575.

[9] G. Peng, M. Liu, S. Chen, Y. Li, and F. Lu, "Generation of sar images with features for target recognition," in *2022 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, 2022, pp. 1–4.

[10] H. Chen, "Challenges and corresponding solutions of generative adversarial networks (gans): A survey study," *Journal of Physics: Conference Series*, vol. 1827, p. 012066, 2021.

[11] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *ArXiv*, vol. abs/2105.05233, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:234357997

[12] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," 2015.

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[15] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," 2020.

[16] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," 2021.

[17] J. Ho and T. Salimans, "Classifier-free diffusion guidance," 2022.

[18] E. Keydel, S. Lee, and J. Moore, "Mstar extended operating conditions: a tutorial," in *Defense, Security, and Sensing*, 1996. [Online]. Available: https://api.semanticscholar.org/CorpusID:121242530

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

[20] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," 2016.

[21] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," 2018.

[22] M. Bińkowski, D. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," 2021.

[23] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016. [Online]. Available: https://arxiv.org/abs/1511.06434

[24] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2019. [Online]. Available: https://arxiv.org/abs/1805.08318