Efficient Anomaly Detection using Diffusion Models

Sunshine Jiang[1] **Wenqi Ding**[1] **Xinran Wang**[1] **Stephen Yang**[2] **Qianru Lao**[2]

[1] MIT [2]Harvard University

Abstract

Anomaly detection plays a critical role in enabling robust navigation for autonomous robots in off-road environments. While diffusion models offer a powerful approach to anomaly detection, their slow inference time and high computational cost limit real-world applicability. This paper proposes a multi-faceted optimization strategy to improve the efficiency of diffusion-based anomaly detection. Our approach integrates three key techniques: (1) Denoising Diffusion Implicit Models (DDIM) to reduce the number of denoising steps, (2) latent space compression using Deep Compression Autoencoders (DC-AE) to operate within a reduced dimensional space, and (3) weight-only quantization to reduce model size and computational overhead. We evaluate these methods both individually and in combination, achieving substantial improvements in inference speed, GPU memory usage, and model size, with only a minor impact on detection accuracy. Our best approach regarding inference time, DDIM with 250-steps, achieves a 66.3% time reduction while preserving 98.8% of the baseline AUC-PR score representing anomaly detection performance. Quantized DCAE requires only 15.2% of the original GPU memory and still obtain 72% of baseline AUC-PR score. These results pave the way for real-time anomaly detection on resource-constrained edge devices, supporting the deployment of robust robotic navigation systems in challenging environments. Code can be found at https://github.com/xinyunsunshine/Tiny-Anomaly-by-Synthesis. The demo video can be found at https://youtu.be/cJTyEO5FPxU.

1 Introduction

Robust anomaly detection is essential for the navigation of autonomous robots in off-road and unstructured environments (1; 15; 7; 18). Perception systems used for off-road navigation, such as those relying on semantic segmentation (23; 24; 9), are typically trained on limited, domain-specific datasets, such as RUGD (25). When deployed, these systems often encounter environments where in the wild input images contain "out-of-distribution" (OOD) anomalies that are not adequately captured in the training data. Therefore,to ensure safe and reliable navigation in unfamiliar off-road environments, robots need to detect such anomalies to proactively address potential perception failures.

To this end, Jiang et al. (11) introduce an analysis-by-synthesis approach for anomaly detection using generative diffusion models. The pipeline of the method is shown in Fig. 1. In the *synthesis* step, this method uses a novel energy-guidance technique for diffusion models to edit an image by removing out-of-distribution (OOD) anomalies while keeping the remaining image unchanged. The new, *synthesized* image represents what the scene would have looked like had it not contained any anomalies. In order to detect anomalies, in the *analysis* step, this method analyzes which image segments were modified by the diffusion model using a combination of foundation vision models: MaskCLIP (4), FeatUp (8) and SAM (12).

The approach introduces several notable advantages. Unlike many existing methods, it avoids making any assumptions about the characteristics of anomalies that might be encountered during testing (11).

Preprint. Under review.

Furthermore, it eliminates the need for OOD data during the training phase. By utilizing diffusion guidance (22), the method performs input image editing as a post-hoc, test-time procedure, requiring neither re-training nor fine-tuning of the diffusion model. (11).

However, this approach is not without downsides. The inference phase of this method suffers from significant runtime inefficiencies, making it unsuitable for real-time applications. After simple profiling steps (see Fig. 1), we see that the Denoising Diffusion Probabilistic Models (10) used by this framework requires iterative denoising through up to 1,000 timesteps to generate high-quality images. This iterative process leads to substantial latency, with each image generation taking approximately 30 seconds. Such delays make deploying the method on real robotic systems, where real-time anomaly detection is critical, infeasible. Furthermore, the approach is hindered by extended training times, which poses a significant drawback. This limitation arises from the need for domain-specific training and overfitting to effectively distinguish in-distribution data from OOD anomalies.

To address these challenges, we propose **a multi-faceted optimization strategy tailored to the computational bottlenecks of the diffusion pipeline for anomaly detection**. First, we will integrate Denoising Diffusion Implicit Models (DDIM) (21), which decrease the number of required denoising steps. Additionally, we exploit model sparsity by quantizing diffusion U-Net weights using techniques like SVDQuant (13). To further reduce inference time, we apply Deep Compression Autoencoders (DC-AE) (2), which optimize the diffusion pipeline by operating within a compressed latent space, significantly reducing dimensionality and computational demands while maintaining accuracy. By integrating these techniques, we achieve **significant improvements in training speed, inference efficiency, and GPU memory utilization**. These advancements pave the way for **transforming diffusion-based anomaly detection into a practical, real-time solution, well-suited for robotic deployment in dynamic environments**.

2 Related Work

The primary objective of this project is to improve the efficiency of the diffusion model in our anomaly detection pipeline. We review relevant approaches and prior work in these areas. Specifically, we explore quantization techniques, diffusion model acceleration methods, and model compression.

2.1 Latent Space Diffusion

Recent works have shown that performing diffusion in a compressed latent space rather than the pixel space can significantly reduce computational requirements. Latent Diffusion Models (LDM) (17) was initially introduced to facilitate diffusion model training on limited computational resources while preserving quality and flexibility. By leveraging a rich latent space, LDM achieves impressive synthesis results on image data and beyond. Other works focus on enhancing the reconstruction accuracy of the autoencoder (5). Deep Compression Autoencoder (DCAE) (3) introduces residual autoencoding and decoupled high-resolution adaptation to achieve higher spatial compression ratios, allowing for a greater speed-up while maintaining reconstruction quality.

2.2 Quantization

Quantization has emerged as a key technique for reducing model size and computation cost by reducing the precision of model parameters such as weights and activations. The most naive form of quantization involves directly clamping the parameters values or linearly mapping floating-point values to lower-precision integer values with a uniform scaling factor. Depending on specific model use cases, various methods provides different levels of more sophisticated quantization while optimizing model performance. PTQ4DM (19) proposes 8-bit (W8A8) post-training quantization (PTQ) on diffusion models on smaller and lower resolutions datasets. Q-Diffusion (14) achieves W4A8 quantization by implementing timestep-aware calibration and split shortcut quantization. More aggressively, SVD-Quant (13) achieves a quantization of int 4 bit (W4A4) by consolidating outliers of weights and activation and integrating them into a high-precision low-rank branch.

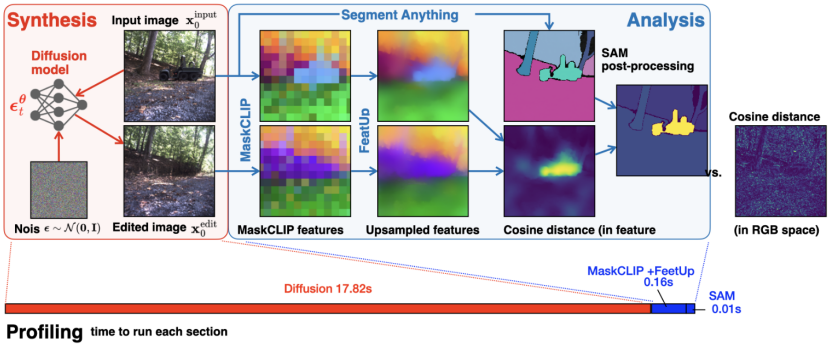


Figure 1: **Runtime Profiling of Anomalies by Synthesis.** The results highlight the runtime inefficiencies of the DDPM (10) within the synthesis pipeline, where the denoising process accounts for the majority of the computational time, rendering it unsuitable for real-time applications.

Figure 2: **Qualitative Performance Analysis of DC-AE (3) on the RUGD Dataset (25).** The pretrained DC-AE (3) demonstrates effective performance, as there is no noticeable degradation in picture quality between the original and reconstructed images.

2.3 Diffusion Model Acceleration

2.3.1 Reducing denoising steps

Our project focuses on speeding up the diffusion model of the OOD detection pipeline. One promising approach is Denoising Diffusion Implicit Models (DDIM) (20). It accelerates the diffusion process by reformulating the generative steps into a deterministic mapping that reduces the number of diffusion steps. Specifically, while DDPM uses a Markovian forward and reverse process, DDIM introduces a non-Markovian sampling mechanism that allows it to generate samples with fewer steps.

2.3.2 Distillation

Additionally, distillation methods have been applied to diffusion models to simplify the sampling process. Distillation creates smaller models that mimic the original diffusion model's behavior but operate with few steps. Progressive Distillation for Fast Sampling (Salimans, 2022) demonstrates the potential for enhancing generation speed without compromising accuracy, which could be critical for the deployment of diffusion models in TinyML contexts.

3 Methodology

In this section, we introduce three of our optimization techniques in details.

3.1 DDIM

We leverage Denoising Diffusion Implicit Models (DDIM) to reduce the number of diffusion steps and thus speed up the inference process. We adapt DDIM to our anomaly detection problem. Specifically, we perform a similarity-guided DDIM process, so that the generated image is sampled from training data distribution but also similar to the input OOD image. Jiang et al. (11) analyzes the theoretical

guidance gradient for energy-based conditional generation and derives a principled approximation. We combine their derived guidance gradient approximation and the DDIM formulation to perform guided DDIM.

3.1.1 Problem Formulation

Pixel-wise anomaly detection Task: We address pixel-wise anomaly detection in RGB images, where anomalies are implicitly defined as unlikely under the training distribution $q(x_0)$. Given training images $D_{\text{train}} = \{x_0^{(n)}\}_{n=1}^{N} \sim q(x_0)$, we train a generative model to approximate $q(x_0)$. At test time, given an input x_0^{input}, the goal is to detect anomalous pixels without prior assumptions about anomalies.

Synthesis formulation: The detection is structured in three stages: Training, Synthesis and Analysis. Here we focus on Synthesis. At test time, anomalies are identified by synthesizing an edited image x_0^{edit} via:

$$\mathbf{x}_0^{\text{edit}} \sim q(\mathbf{x}_0^{\text{edit}} \mid \mathbf{x}_0^{\text{input}}) \propto \underbrace{q(\mathbf{x}_0^{\text{edit}})}_{\text{likelihood under training distribution}} \underbrace{r_{\text{sim}}(\mathbf{x}_0^{\text{edit}}, \mathbf{x}_0^{\text{input}})}_{\text{similarity to input image}} \tag{1}$$

where $r_{\text{sim}}(\mathbf{x}_0^{\text{edit}}, \mathbf{x}_0^{\text{input}})$ ensures similarity to x_0^{input}. This process edits x_0^{input} to remove anomalies while retaining in-distribution pixels.

3.1.2 Generalized Guidance for DDIM

Jiang et al. (11) extends classifier guidance to a setting where the conditioner is any non-negative function. They prove that for a diffusion model trained on $q(x_{0:T})$, the conditional distribution $q(x_{0:T} \mid x_0^{\text{input}}) \propto q(x_{0:T})\, r_{\text{sim}}(x_0, x_0^{\text{input}})$ can be sampled using the guidance gradient:

$$g_t^*(x_t) = \nabla_{x_t} \log \mathbb{E}_{q(x_0|x_t)} \left[r_{\text{sim}}(x_0, x_0^{\text{input}}) \right] \tag{2}$$

And it can be approximated as

$$g_t^*(x_t) \approx \nabla_{x_t} \log r_{\text{sim}} \left(\mu_0(x_t), x_0^{\text{input}} \right) = g_t(x_t) \tag{3}$$

where $\mu_0(x_t)$ is the expected value of x_0 under $q(x_0 \mid x_t)$. This guides x_t to increase the similarity of the denoised image x_0 with the input x_0^{input}.

We combine the guidance gradient approximation (Eq. 3) with the DDIM generative process detailed in Eq. 4 to derive the Similarity-guided DDIM image editing algorithm (Alg. 1).

$$\mathbf{x}_{t-1} \leftarrow \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\, \epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{\alpha_t}} \right)}_{\text{“predicted } \mathbf{x}_0\text{”}} + \underbrace{\sqrt{1 - \alpha_{t-1} - \sigma_t^2}\, \epsilon_\theta^{(t)}(\mathbf{x}_t)}_{\text{“direction pointing to } \mathbf{x}_t\text{”}} + \underbrace{\sigma_t\, \epsilon_t}_{\text{random noise}} \tag{4}$$

In line 3 of Alg. 1, we calculate the expected value of x_0 given x_t by subtracting a scaled version of $\epsilon_t^\theta(x_t)$ from x_t, since $\epsilon_t^\theta(x_t)$ is trained to predict the expected noise $\boldsymbol{\epsilon}$ that produced x_t from x_0. In line 4 of Alg. 1, the log gradient of the similarity metric is computed to guide the generation closer to the input image.

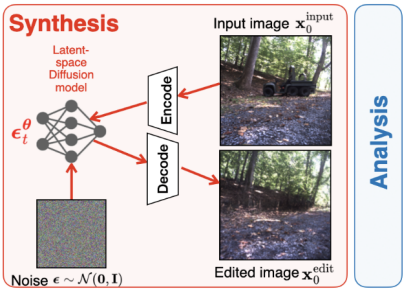


Figure 3: **Inference pipeline for latent compression optimization alone.** The DiT (16) diffusion model now operates in the latent space of the autoencoders.

Algorithm 1 Similarity-guided DDIM to edit input image and remove anomalies

procedure GUIDEDDIFFUSION(ϵ_t^θ, $\mathbf{x}_0^{\text{input}}$, r_{sim}, s)

$\mathbf{x}_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$

for t from T to 1 **do**

$\mu_0^\theta(\mathbf{x}_t) := \mathbb{E}_q[\mathbf{x}_0 \mid \mathbf{x}_t] \approx \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\alpha_t}}\epsilon_t^\theta(\mathbf{x}_t)$

$\mathbf{g}_t(\mathbf{x}_t) \leftarrow \nabla_{\mathbf{x}_t} \log r_{\text{sim}}(\mu_0^\theta(\mathbf{x}_t), \mathbf{x}_0^{\text{input}})$

$\hat{\epsilon} \leftarrow \epsilon_t^\theta(\mathbf{x}_t) - s\sqrt{1-\bar{\alpha}_t}\mathbf{g}_t(\mathbf{x}_t)$

$\mathbf{x}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon} + \sigma_t\,\epsilon_t$

end for

return $\mathbf{x}_0^{\text{edit}} := \mathbf{x}_0$

Algorithm 1: *Inputs:* (1) ϵ_t^θ: denoising diffusion model fit to training image distribution $q(\mathbf{x}_0)r_{\text{sim}}$. (2) $\mathbf{x}_0^{\text{input}}$: input image at test-time potentially containing anomaly segments. (3) r_{sim}: similarity metric between two images. We use L2 distance $r_{\text{sim}}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$) (4) s: guidance scale. The larger the guidance scale, the more important the similarity metric is in diffusion process.

Output: $\mathbf{x}_0^{\text{edit}}$: edited version of $\mathbf{x}_0^{\text{input}}$ sampled from $q(\mathbf{x}_0)r_{\text{sim}}$ that removes anomalies.

While standard reverse diffusion is designed to sample from $q(\mathbf{x}_0)$, the image similarity metric $r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})$ guides the generated image $\mathbf{x}_0$ to be close to $\mathbf{x}_0^{\text{input}}$ using guided diffusion. This is achieved by including the log-gradient of r_{sim}.

3.2 Latent Space Compression

Jiang et al.(11) employ a Denoising Diffusion Probabilistic Model (DDPM) as a subroutine for the diffusion model, operating directly in the image space. This design, however, incurs high computational costs during inference. To mitigate this, we propose a novel approach inspired by the Deep Compression Autoencoder (DC-AE)(3), and take advantage of the relatively low-rank nature of anomaly detection tasks. Specifically, we first evaluate the performance of the pretrained DC-AE on the RUGD dataset (25), demonstrating its strong performance with no additional fine-tuning, as illustrated in Fig.2. Building on this, we train an unconditional DiT(16) model in the latent space of DC-AE (3), significantly reducing computational costs. Utilizing an NVIDIA RTX 3090 GPU, our training process was completed in a matter of hours, in contrast to the substantially more resource-intensive setup required by the original Anomalies by Synthesis method (11).

The fine-tuned DiT (16) model operates within the latent space, leveraging classifier-free guidance in conjunction with our custom condition function. This condition function, which incorporates the gradient of similarity function, dynamically steers the generation process to ensure that the

synthesized output remains consistent with the training distribution of the RUGD dataset (25) while effectively removing undesired out-of-distribution (OOD) objects. Upon completing its edits in the latent space, the autoencoder's decoder reconstructs the edited representation back into the original image space. The reconstructed image is then compared with the original input image, with SAM (12) utilized to precisely identify and locate the OOD objects. This streamlined workflow not only facilitates efficient and high-quality image editing but also enables robust anomaly synthesis while maintaining computational feasibility, addressing key challenges in prior approaches.

3.3 Quantization

To optimize inference on resource-constrained edge devices, we aim to leverage model sparsity by quantizing the diffusion U-Net weights into lower-bit representations.

Our initial approach draw on concepts from SVDQuant, which enables aggressive quantization while preserving performance. However, our current model uses a DDPM implementation in PyTorch (denoising-diffusion-pytorch), the U-Net structure of which is not compatible with the SVDQuant open-source codebase. Even with moderate edits to the open-source codebase, only a few convolution layers could be recognized and quantized.

Given the time constraints of this project, we shift to a naive weight-only quantization approach, clamping weights of specific computational layers to lower precisions. We focused on quantizing key layers, such as QKV projections and convolutions, while avoiding auxiliary layers. Our methodology aims to identify the best precision levels and layers to quantize by evaluating three objectives: (1) determining the optimal precision level, (2) selecting layers for quantization, and (3) assessing the combined impact of quantization with other optimization techniques.

4 Experiment

4.1 Dataset

We use a off-road land navigation RUGD dataset (25), which contains real-world images collected in off-road environments using mobile robot platforms with manually labeled pixelwise class annotations. Same as the baseline (11), we split the semantic categories into in-distribution labels that contains mostly natural features., whereas classes like vehicle, building, person are defined as out-of-distribution.

4.2 Experiment Setup

We evaluate the individual and combined effect of the three optimization methods (**DDIM**, **Latent Compression**, **Quantization**) by applying them individually and combining **Quantization** with the other two methods.

4.2.1 DDIM

The advantage of DDIM is that it does not retrain the DDPM checkpoints. DDIM employs a non-Markovian diffusion process that maintains the same training objective as DDPMs (20). This alignment allows DDIMs to adopt the pre-trained noise-prediction model from DDPMs without necessitating additional training. Therefore, we directly apply the checkpoints in the DDPM baseline (11).

The guidance scale of the DDIM process is determined based on parameter sweep to be 0.6 so that the generated image is similar to the input image but also not too constrained to remove the anomaly region. The effect of different guidance scales are shown in Fig. 4.

DDIM with Quantization: Since DDIM only changes the inference time, we can directly use the quantized model with DDIM sampling for the experiment.

4.2.2 Latent Compression

Latent compression leverages the Deep Compression Autoencoder (DC-AE) (3) to reduce the dimensionality of input data before applying the diffusion model. By operating within this compressed

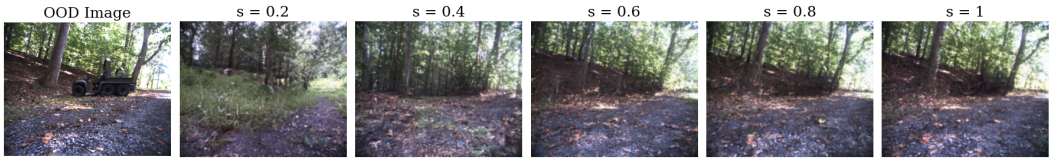


Figure 4: **Qualitative Analysis of DDIM with Different Guidance scales:** s refers to the guidance scale. As the guidance scale increases from 0.2 to 1, the generated image becomes more similar to the input OOD image. At $s = 0.6$, the OOD truck blends into the ground in the back.

latent space, the computational complexity of the diffusion process is significantly reduced without requiring extensive retraining of the model. For this experiment, we fine-tuned the diffusion model while keeping the DC-AE frozen on the RUGD dataset, allowing the diffusion model to adapt to compressed latent representations without compromising the robustness of the pretrained DC-AE. A guidance scale of 600 was determined through parameter sweeps, ensuring a balance between removing anomalies and preserving the fidelity of the in-distribution regions.

DC-AE with Quantization: To further optimize efficiency, DC-AE was integrated with quantization to 16 bits with selected layer (the quantization method with the best accuracy performance), wherein specific layers of the diffusion model were quantized to lower-precision formats.

4.2.3 Quantization

Due to the time constraints and the incompatibility of SVDQuant with current model's Unet, we implement a more straightforward weight-only quantization strategy, clamping weights of of selected layers to lower-precision. We experiment with precision levels (float 16 bit and int 8 bit) and explored which layers to quantize. We eventually combine quantization with other optimization methods.

We apply quantization to key computational layers. Specifically, we quantize the QKV projections, output projections, and ResNet convolution layers, which are responsible for the bulk of the model's computational workload. We avoid quantization of auxiliary components like bias terms, gates, down-sampling and up-sampling operations, normalization layers, and the first and last layers. These layers and generally known to be more sensitive to quantization.

To evaluate performance, we measured the impact of quantization on efficiency and accuracy metrics, comparing results across various experiments. We selected the optimal approach—quantizing only a subset of the U-Net's layers into float16 precision—as our final quantization strategy and proceeded to combine it with other optimization methods. Hereafter, if not specified otherwise, "quantization" refers to this selected-layer float16 quantization.

4.3 Results

Overall, DDIM maintains the highest accuracy with large inference time decrease (1/8 - 1/2 depending on number of time steps).

DC-AE achieves a substantial reduction in computational requirements with lower inference time and GPU memory usage while maintaining an acceptable level of reconstruction quality.

Quantization leads to significant reductions in model size but also causes non-negligible accuracy degradation. Out of the experiments, targeting selected layers with float16 quantization achieves a better balance between accuracy preservation (90% of baseline AUC-PR) and efficiency gains.

4.3.1 DDIM

We visualize the changes in accuracy and inference time with respect to the number of denoising steps in Fig. 5. The overall trend aligns with expectations: dereasing the number of steps decreases latency at the cost of accuracy.

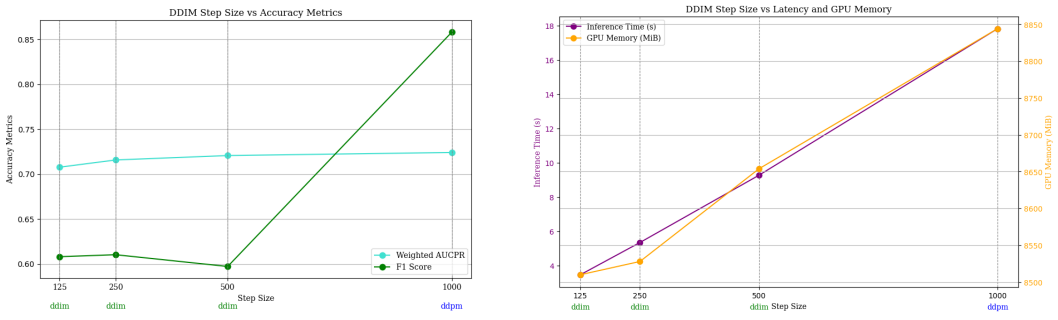
The AUPCR score, a pixel-wise accuracy metric, exhibits small amount of decline with fewer steps. In contrast, the F1 score, an object-wise accuracy metric, deteriorates more significantly as the step size decreases. This disparity is due to the fact that pixel-level evaluation metrics can often neglect

			Accuracy Metrics		Efficiency Metrics	
			AUC-PR (↑)	f* score (↑)	GPU mem (↓)	inference time (↓)
	1	DDPM Baseline	0.724	0.858	8844	17.82
DDIM	2	DDIM (125 steps)	0.707	0.607	8510	3.47
	3	DDIM (250 steps)	0.715	0.610	8528	5.35
	4	DDIM (500 steps)	0.721	0.600	8654	9.28
Quantization	5	Quantize all weights to int8	0.539	0.618	4380	16.83
	6	Quantize all weights to float16	0.547	0.647	6929	16.91
	7	Quantize selected layers to int8	0.517	0.633	5237	17.09
	8	Quantize selected layers to float16	0.672	0.672	7630	17.12
	9	DCAE	0.523	0.564	2696	12.52
Summary	10	**Best DDIM (250)**	0.715	0.610	8528	5.35
	11	**Best Quantization (selected float16)**	0.672	0.672	7630	17.12
	12	**DCAE**	0.523	0.564	2696	12.52
Combined	13	Quantized DDIM (125 steps)	0.595	0.560	7347	3.45
	14	Quantized DDIM (250 steps)	0.610	0.576	7399	5.36
	15	Quantized DDIM (500 steps)	0.621	0.574	7525	9.27
	16	Quantized DCAE	0.522	0.564	1348	12.52

Table 1: Anomaly detection accuracies and efficiencies across variation optimizations on RUGD dataset. ▭ represents our original baseline, and ▭ shows the individual optimization performance with different ablations, ▭ summarizes the best result for each optimization and ▭ combines DDIM and DCAE with quantization. The bottom section represents the final results with all the optimization methods. GPU memory is in MiB and inference time refers to the time needed to generate one image in seconds.

small anomalies and be biased towards anomalies with large sizes. This suggests that **DDIMs are effective at detecting large anomalies but less effective at identifying small ones**. The quanlitative result of DDIM wiht different step sizes are shown in Fig. 6.

As for latency metrics, the relationship between the number of steps and latency is nearly linear, as expected. GPU memory usage slightly decreases as the number of steps decreases. This reduction occurs because the number of parameters (e.g. $\bar{\alpha}_t$) stored during the inference process decreases with the number of steps.



(a) Accuracy metrics vs DDIM step size (b) Efficiency metrics vs DDIM step size

Figure 5: Analysis of DDIM Performance under Different Step Sizes

4.3.2 Latent Compression

The compressed latent space reduced dimensionality by a factor of 64, leading to significant computational savings. The results in Table 1 highlight the impact of latent compression on both accuracy

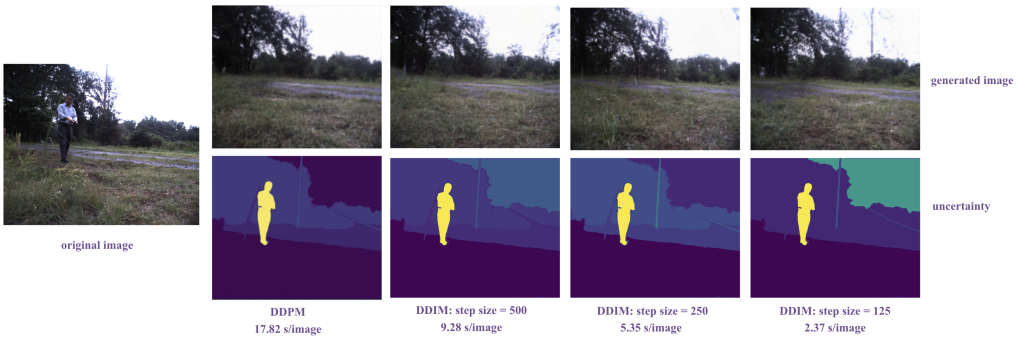


Figure 6: **Qualitative Analysis of DDIM with Different Step Sizes:** As the number of denoising steps decrease, the latency decreases, but the quality of the generated image and thus the uncertainty also decreases. In this example, while the anomaly is removed in all images, the images generated with more time keep the remaining parts more similar to the input image.

and efficiency metrics. By operating in the compressed latent space, the GPU memory usage dropped to 2696 MiB, and the inference time was reduced to 12.52 seconds—a substantial improvement over the 8844 MiB and 17.82 seconds required by the baseline DDPM model. Despite the compression, the anomaly detection accuracy was retained at a reasonable level, with an AUC-PR of 0.523 and an F* score of 0.564, making it a practical solution for resource-constrained applications.

Latent compression enables the diffusion model to balance efficiency and performance. Although there is a trade-off in terms of accuracy compared to DDIM, the approach achieves a good compromise by maintaining acceptable anomaly detection metrics while significantly reducing computational demands. This result underscores the value of latent compression for scalable, real-time deployment in edge environments.

4.3.3 Quantization

Model Type	**Baseline**	**all layers to int8**	**all layers to float16**	**selected to int8**	**selected to float16**
Model Size	136.3	34.08	68.26	67.26	90.28

Table 2: Model size (in MB) for different quantization strategies.

We evaluated the impact of quantization strategies on accuracy and efficiency, as shown in Table 2 and 1. Quantizing all layers to int8 achieved the smallest model size (34.08 MB) and fastest inference but caused significant accuracy loss (AUC-PR: 0.539). Float16 quantization for all layers slightly improved accuracy (AUC-PR: 0.547) but still considerably underperformed relative to the baseline AUC-PR of 0.724. Selective float16 quantization of key layers provided the best balance, preserving most of the baseline accuracy (AUC-PR: 0.672) while reducing model size (90.28 MB) and inference time. Selective int8 quantization was more efficient (model size: 67.26 MB) but less accurate (AUC-PR: 0.517).

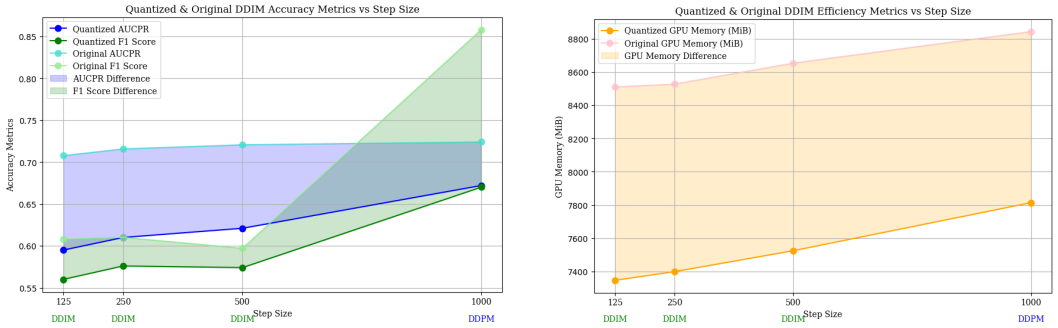
Overall, selective float16 quantization emerged as the most effective strategy, offering a strong balance between accuracy and efficiency for edge-device deployment.

4.3.4 DDIM with Quantization

The results of DDIM with quantization are visualized in Fig. 7. As expected, DDIM applied to the quantized model has lower accuracy compared to the original, unquantized model. Additionally, same as the unquantized case, the model's accuracy decreases as the number of denoising steps in DDIM is reduced. For AUCPR, we observe that the performance gap widens as the number of DDIM

timesteps is reduced. With fewer timesteps, the model has less opportunity to refine predictions, amplifying quantization-induced errors that result in noisier intermediate representations and a loss of fine-grained details crucial for maintaining a high AUCPR.

The memory required during inference is lower for the quantized model compared to the original. Furthermore, the trend of reduced memory usage with a smaller number of timesteps remains.



(a) Accuracy metrics vs DDIM step size

(b) Efficiency metrics vs DDIM step size

Figure 7: Analysis of quantized DDIM Performance under Different Step Sizes

4.3.5 DC-AE with Quantization

The combination of DC-AE with quantization further optimizes GPU memory usage while maintaining inference efficiency. Table 1 highlights the results for both the original and quantized versions of DC-AE.

By applying quantization, the GPU memory usage was reduced significantly from 2696 MiB in the original DC-AE to 1348 MiB—a 50% reduction—without impacting the inference time, which remained steady at 12.52 seconds. The anomaly detection performance, as measured by AUC-PR and F* score, showed negligible differences: 0.523 vs. 0.522 (AUC-PR) and 0.564 for both (F* score). This result demonstrates that quantization successfully enhances efficiency without a significant compromise in accuracy.

Limitation and Future Work

Jiang et al. (11) investigated various timestep matching methods for guided diffusion, including forward timestep matching and no timestep matching. In this study, we focused exclusively on the method with the best evaluation results: reverse timestep matching. It would be interesting to explore how different timestep matching methods perform in the context of guided DDIM. Additionally, several recent methods aim to accelerate the diffusion process by reducing the number of denoising steps, such as one-step diffusion via shortcut models(6), which could potentially be incorporated into our framework.

To avoid unacceptable degradation in performance, we adopted a conservative approach by applying float16 quantization to only selected layers. Future work could focus on developing a SVDQuant for the current model, enabling more aggressive quantization and significantly reducing model size. This approach could potentially better support deployment on resource-constrained edge devices.

While latent space compression significantly improves computational efficiency, it introduces challenges such as potential loss of fine-grained details critical for detecting subtle anomalies. Additionally, the pretraining of autoencoders requires careful alignment with the target distribution to avoid representation bias. Future work could focus on adaptive compression techniques that dynamically adjust the latent space resolution based on input complexity. Further exploration of hybrid autoencoders incorporating both global and local features may also enhance anomaly detection robustness without sacrificing efficiency.

5 Conclusion

This study introduced a multi-faceted optimization strategy to enhance the efficiency of diffusion-based anomaly detection, specifically targeting its application in resource-constrained environments like autonomous robotics. By integrating Denoising Diffusion Implicit Models (DDIM), latent space compression with Deep Compression Autoencoders (DC-AE), and weight quantization, the proposed methods achieved significant improvements in inference time, GPU memory usage, and computational overhead, while maintaining acceptable anomaly detection accuracy.

DDIM notably reduced inference latency with minimal accuracy loss, making it a viable solution for real-time applications. Latent space compression offered substantial computational savings, emphasizing its potential for scalable deployments on edge devices. Selective quantization helps slightly reduce GPU memory and inference time.

The combined application of these techniques demonstrated the practicality of transforming computationally intensive diffusion models into efficient, real-time solutions for off-road anomaly detection. Despite these advancements, challenges remain in preserving fine-grained anomaly details during compression and extending quantization methodologies for more aggressive optimizations without performance degradation. Future work could explore dynamic latent space resolution, more sophisticated quantization methods such as SVDQuant, and alternative timestep strategies for further accelerating the diffusion process to refine the balance between computational efficiency and anomaly detection robustness.

Contribution

- Sunshine is responsible for the DDIM and DDIM + quantization sections.
- Wenqi and Xinran are responsible for the quantization section.
- Stephen and Qianru are responsible for the latent compression and latent compression + quantization sections.

References

[1] John Bares, Martial Hebert, Takeo Kanade, Eric Krotkov, Tom Mitchell, Reid Simmons, and William Whittaker. Ambler: An autonomous rover for planetary exploration. *Computer*, 22(6):18–26, 1989.

[2] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models, 2024.

[3] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint arXiv:2410.10733*, 2024.

[4] Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10995–11005, 2023.

[5] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024.

[6] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models, 2024.

[7] Jonas Frey, David Hoeller, Shehryar Khattak, and Marco Hutter. Locomotion policy guided traversability learning using volumetric representations of complex environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5722–5729, 2022.

[8] Stephanie Fu, Mark Hamilton, Laura E. Brandt, Axel Feldmann, Zhoutong Zhang, and William T. Freeman. Featup: A model-agnostic framework for features at any resolution. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[9] Tianrui Guan, Divya Kothandaraman, Rohan Chandra, Adarsh Jagan Sathyamoorthy, Kasun Weerakoon, and Dinesh Manocha. Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments. *IEEE/RAS Robotics and Automation Letters (RAL)*, 7(3):8138–8145, 2022.

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

[11] Sunshine Jiang, Siddharth Ancha, Travis Manderson, Laura Brandt, Yilun Du, Philip R. Osteen, and Nicholas Roy. Anomaly detection using generative diffusion models for off-road navigation. *preprint*, 2024.

[12] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023.

[13] Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models, 2024.

[14] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17535–17545, 2023.

[15] Mauro Massari, Giovanni Giardini, Franco Bernelli-Zazzera, et al. Autonomous navigation system for planetary exploration rover based on artificial potential fields. In *Proceedings of the conference on Dynamics and Control of Systems and Structures in Space (DCSSS)*, pages 153–162, 2004.

[16] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023.

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[18] Amirreza Shaban, Xiangyun Meng, JoonHo Lee, Byron Boots, and Dieter Fox. Semantic terrain classification for off-road autonomous driving. In *Conference on Robot Learning*, pages 619–629. PMLR, 2022.

[19] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1972–1981, June 2023.

[20] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[21] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

[22] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[23] Abhinav Valada, Gabriel L Oliveira, Thomas Brox, and Wolfram Burgard. Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, pages 465–477, 2017.

[24] Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *Proceedings of the IEEE/RAS International Conference on Robotics and Automation (ICRA)*, pages 4644–4651, 2017.

[25] Maggie Wigness, Sungmin Eum, John G Rogers, David Han, and Heesung Kwon. A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5000–5007, 2019.