Removing adverse weather impacts from autonomous vehicle camera systems with generative techniques

Chunming Peng
cmpeng@stanford.edu
X629350

Yun Kun Zhang
wyzhang@stanford.edu
X494533

Jason Alexander Chan
jchan7@stanford.edu
X562837

1 Introduction

This project seeks to use generative algorithms to mitigate adverse weather effects in autonomous vehicle camera feeds to enhance object detection, classification, and localization. Autonomous systems combine various sensors, with cameras being cost-effective yet vulnerable to weather disturbances like light rain, heavy rain, smoke, fog, haze, snow, and contamination. Despite prior research in weather removal, modern generative techniques, such as denoising and in-painting, present a novel and promising alternative. This project will: (1) experiment with a generative technique, Denoising Diffusion Restoration Models (DDRM) for driving scenarios under rain; and (2) compare DDRM with other generative techniques that address the DDRM limitations identified. Although autonomous vehicle cameras record videos, this project limits the scope to images with adverse weather effects.

2 Related Work

Generative Diffusion Models Özdenizci et al.'s (2022) applied denoising diffusion models to adverse weather datasets that included desnowing, deraining, dehazing and raindrop removal and claim state of the art performance in weather-specific and multi-weather restoration [1]. Their technique can be applied to any arbitary sized image . However, their implementation takes 20 seconds to process a single 640 x 432 pixel image on one NVIDIA A40 GPU, which cannot be used on autonomous vehicles due to the real time constraints. Kawar et al. (2023), on the other hand, focused on general image restoration [2]. They suggest that many restoration tasks can be framed as linear inverse problems and introduce their Denoising Diffusion Restoration Models (DDRM) for super-resolution, deblurring, inpainting, and colorization in noisy conditions. Their technique is 5x faster than the nearest competitor. DDRM's speed could be promising for autonomous vehicle cameras systems. However, heavy rain or severe weather conditions, combining snow and fog, are expected to introduce non-linear distortions.

Generative Adversarial Networks Yang et al. (2023) developed ViWS-Net using GANs to mitigate weather effects in videos. Their videos are processed at 224 x 224 pixels at five frames per second [3]. Training employed two NVIDIA RTX 3090 GPUs, with an inference time of 0.46 seconds—though the specifics of what was inferred and on which hardware aren't detailed. They note ViWS-Net's computational efficiency is on par with other methods but excels in multi-weather removal. Their evaluation used Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) metrics. They do not assess whether their outputs improve downstream object detection or classification algorithms.

Transformer Models Valanarasu et al. (2022) introduced TransWeather, a transformer technique with an encoder-decoder structure, boasting state-of-the-art performance across weather conditions [4]. It processes 256 x 256 pixel images in one second using an NVIDIA RTX 8000 GPU. However, TransWeather struggles with high-intensity rain in real-world datasets, aligning with Zhang et al.

(2023)'s observation that most de-raining algorithms falter in dynamic scenes or with high rain rates [5].

3 Technical Approach

Dataset used The primary dataset for this project is the RainDrops [6] dataset, which comprises paired images with identical backgrounds; one image is marred by raindrops, while its pair remains unaffected. The IUPUI Driving Video/Image Benchmark [7] was used for one of the fine-tuning experiments. This dataset has in-car camera footage under varying illumination and road scenarios, that includes higher risk driving scenarios. It captures diverse adverse conditions like snow, rain, direct sunlight, dim light, reflections, and wet roads, among others.

Expected results, and Evaluation We expect that the non-linear effects introduced by rain or severe weather conditions, combining snow and fog, can be removed or reduced by DDRM (with our customization and enhancements). Evaluation of the feasibility will be both quantitative and qualitative: Quantitative metrics of model performance such as PSNR, SSIM, KID (Kernel Inception Distance) and NFEs (Number of Function Evaluations), which are commonly used to evaluate denoising effectiveness. Also, the subjective evaluation, which requires manual inspection of the adverse weather removal results.

Proposed Methods Three approaches were taken: (1) experiment with the vanilla DDRM codebase to baseline the performance of adverse weather removal from driving scenes; and (2) experiment with extensions/modifications to DDRM, which includes fine-tuning a diffusion model, and (3) comparing DDRM to two other models that sought to address limitations identified from the experiments with DDRM: (3a) conditional diffusion (Özdenizci et al.'s (2022) [1]), (3b) Plug and Play Image Restoration (Zhu et al (2023) [8]).

3.1 Mathematics description of DDRM

Denoising Diffusion Restoration Models (DDRM), takes advantage of a pre-trained denoising diffusion generative model for solving a linear inverse problem.

3.1.1 Pre-Trained Diffusion Model

A typical pre-trained denoising diffusion generative model has a Markov chain structure in generating process, $X_T \to X_{T-1} \to ... \to X_1 \to X_0$, where X_t are the t-th step in the generating process. The joint distribution is denoted by [2] :

$$p_\theta(x_{0:T}) = p_\theta^{(T)}(x_T) \prod_{t=0}^{T-1} p_\theta^{(t)}(x_t|x_{t+1}).$$

After drawing $x_{0:T}$, only x_0 is kept as the sample of the generative model. To train a diffusion model, a fixed, factorized variational inference distribution is introduced [2]:

$$q(x_{1:T}|x_0) = q^{(T)}(x_T|x_0) \prod_{t=0}^{T-1} q^{(t)}(x_t|x_{t+1}, x_0),$$

which leads to an evidence lower bound (ELBO) on the maximum likelihood objective A special property of some diffusion models is that both $p_\theta^{(t)}$ and $q^{(t)}$ are chosen as conditional Gaussian distributions for all t < T, and that $q(x_t|x_0)$ is also a Gaussian with known mean and covariance, i.e., x_t can be treated as x_0 directly corrupted with Gaussian noise. Thus, the ELBO objective can be reduced into the following denoising autoencoder objective (please refer to [9] for derivations):

$$\sum_{t=0}^{T} \gamma_t \, \mathbb{E}_{(x_0,x_t)\sim q(x_0)q(x_t|x_0)} \left[\left\| x_0 - f_\theta^{(t)} \right\|_2^2 \right]$$

where $f_\theta^{(t)}$ is a $\theta-$ parameterized neural network that aims to recover a noiseless observation from a noisy x_t, and $\gamma_{1:T}$ are a set of positive coefficients that depend on $q(x_{1:T}|x_0)$

3.1.2 Linear Inverse Problems

A general linear inverse problem can be denoted as [2]

$$y = Hx + z, \tag{1}$$

where we aim to recover the signal $x \in R^n$ from measurements $y \in R^m$, where $H \in R^{m \times n}$is a known linear degradation matrix, and $z \sim N(\theta, \sigma_y^2 I)$ is an i.i.d. additive Gaussian noise with known variance. The underlying structure of x can be represented via a generative model, denoted as $p(x)$. Given y and H, a posterior over the signal can be posed as: $p_\theta(x|y) \propto p_\theta(x)p(y|x)$, where the "likelihood" term $p(y|x)$ is defined via Equation (1) and p_θ is the prior.

3.1.3 Denoising Diffusion Restoration Models

Combining the 3.2.1 and 3.2.2, We define joint distribution of x_t conditioned on y as [2]:

$$p_\theta(x_{0:T}|y) = p_\theta^{(T)}(x_T|y) \prod_{t=0}^{T-1} p_\theta^{(t)}(x_t|x_{t+1}, y).$$

and x_0 is the final diffusion output. In order to perform inference, we consider the following factorized variational distribution conditioned on y [2]:

$$q(x_{1:T}|x_0, y) = q^{(T)}(x_T|x_0, y)) \prod_{t=0}^{T-1} q^{(t)}(x_t|x_{t+1}, x_0, y),$$

which leads to an ELBO objective for diffusion models conditioned on y, which is described in section 3.2.

The degradation matrix H can be deconstructed using Singular Value Decomposition (SVD) [2]:

$$H = U\Sigma V^T,$$

We use the shorthand notations for values in the spectral space: $\bar{x_t}^{(i)}$ is the i-th index of the vector $\bar{x_t} = V^T X_t$, and $\bar{y}^{(i)}$ is the i-th index of the vector $\bar{y} = \Sigma \dagger \, Uy$ (where † denotes the Moore–Penrose pseudo-inverse). Because V is an orthogonal matrix, we can recover x^t from $\bar{x}^t$ exactly by left multiplying V. For each index i in x^t, we define the variational distribution as [2] :

$$q^{(T)}(\bar{\mathbf{x}}_T^{(i)}|\mathbf{x}_0, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{y}}^{(i)}, \sigma_T^2 - \frac{\sigma_{\mathbf{y}}^2}{s_i^2}) & \text{if } s_i > 0 \\ \mathcal{N}(\bar{\mathbf{x}}_0^{(i)}, \sigma_T^2) & \text{if } s_i = 0 \end{cases}$$

$$q^{(t)}(\bar{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t+1}, \mathbf{x}_0, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_0^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_0^{(i)}}{\sigma_{t+1}}, \eta^2\sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\mathbf{x}}_0^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{y}}^{(i)} - \bar{\mathbf{x}}_0^{(i)}}{\sigma_{\mathbf{y}}/s_i}, \eta^2\sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_{\mathbf{y}}}{s_i} \\ \mathcal{N}((1-\eta_b)\bar{\mathbf{x}}_0^{(i)} + \eta_b \bar{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_{\mathbf{y}}^2}{s_i^2}\eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_{\mathbf{y}}}{s_i} \end{cases}$$

where $\eta \in (0, 1]$ is a Parameter controlling the variance of the transitions, and η and η_β may depend on σ_t, s_i, σ_y . We further assume that $\sigma_T \geq \sigma_y/s_i$ for all positive s_i.

We define DDRM with trainable parameters θ as follows [2]:

$x_{\theta,t}$ to represent this prediction made by the model, and $\bar{x}_{\theta,t} = V^T x_{\theta,t}$

3.1.4 Loss Function

DDRM is a Markov chain conditioned on y, which would lead to the following ELBO objective [9]:

$$p_\theta^{(T)}(\bar{\mathbf{x}}_T^{(i)}|\mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{y}}^{(i)}, \sigma_T^2 - \frac{\sigma_\mathbf{y}^2}{s_i^2}) & \text{if } s_i > 0 \\ \mathcal{N}(0, \sigma_T^2) & \text{if } s_i = 0 \end{cases}$$

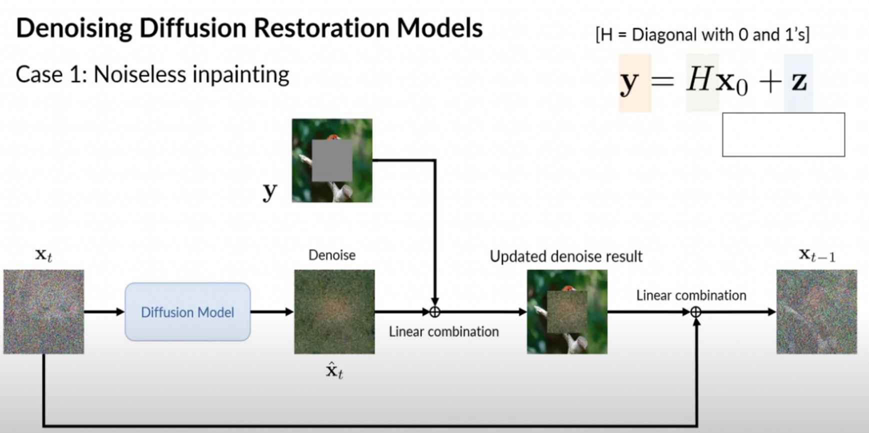
$$p_\theta^{(t)}(\bar{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t+1}, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}}, \eta^2\sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{y}}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_\mathbf{y}/s_i}, \eta^2\sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_\mathbf{y}}{s_i} \\ \mathcal{N}((1-\eta_b)\bar{\mathbf{x}}_{\theta,t}^{(i)} + \eta_b \bar{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_\mathbf{y}^2}{s_i^2}\eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_\mathbf{y}}{s_i}. \end{cases}$$

$$\begin{aligned} &\mathbb{E}_{x_0 \sim q(x_0), y \sim q(y|x_0)} \left[\log p_\theta(x_0|y)\right] \geq \\ &-\mathbb{E}\left[\sum_{t=1}^{T-1} D_{KL}(q^{(t)}(x_t|x_{t+1}, x_0, y) || p_\theta^{(t)}(x_t|x_{t+1}, y)\right] + \mathbb{E}\left[\log p_\theta^{(0)}(x_0|x_1, y)\right] \\ &-\mathbb{E}\left[\ D_{KL}(q^{(T)}(x_T|x_0, y) || p_\theta^{(T)}(x_T|y))\right] \end{aligned}$$

where $q(x_0)$ is the data distribution, $q(y|x_0)$ follows the degradation matrix equation (1) in section 3.1.2, the expectation on the right hand side is given by sampling $x_0 \sim q(x_0), y \sim q(y|x_0), x_T \sim q^{(T)}(x_T|x_0, y)$, and $x_t \sim q^{(t)}(x_t|x_t + 1, x_0, y)$ for $t \in [1, T-1]$.

3.2 Model Architecture and DDRM Algorithm

The main idea of DDRM is to leverage a diffusion prior and a degraded y, and try to optimize the posterior $\log p_\theta(x_0|y)$. Specifically, as shown in Figure 1, the pre-trained model output $\hat{x}_t$, is combined linearly with the degraded y in each time step to derive x_{t-1}. After several time steps, we can get a good restoration image x_0. The overall model structure is illustrated as in following two figures.



Denoising Diffusion Restoration Models

Case 1: Noiseless inpainting

[H = Diagonal with 0 and 1's]

$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$

Figure 1: DDRM Inference Process for Inpainting Example

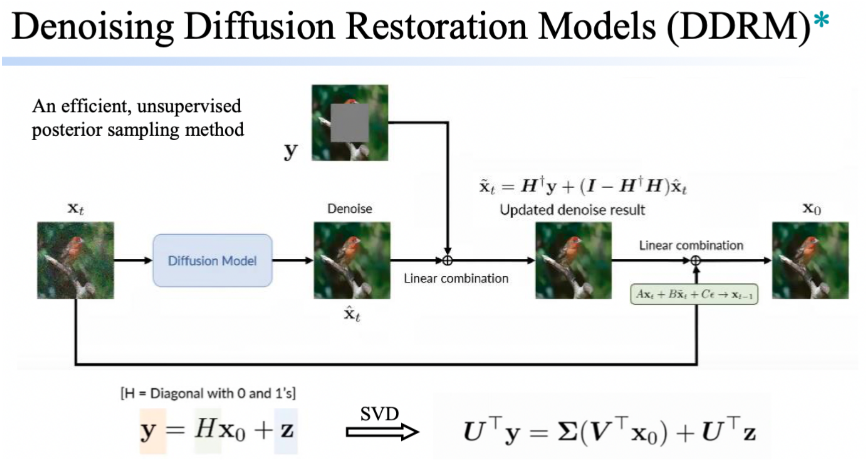

Denoising Diffusion Restoration Models (DDRM)*
An efficient, unsupervised posterior sampling method
$\mathbf{y}$
$\mathbf{x}_t$
Diffusion Model
Denoise
$\hat{\mathbf{x}}_t$
$\tilde{\mathbf{x}}_t = \boldsymbol{H}^\dagger \mathbf{y} + (\boldsymbol{I} - \boldsymbol{H}^\dagger \boldsymbol{H})\hat{\mathbf{x}}_t$
Updated denoise result
Linear combination
Linear combination
$A\mathbf{x}_t + B\tilde{\mathbf{x}}_t + C\epsilon \rightarrow \mathbf{x}_{t-1}$
$\mathbf{x}_0$
[H = Diagonal with 0 and 1's]
$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$
SVD
$\boldsymbol{U}^\top \mathbf{y} = \boldsymbol{\Sigma}(\boldsymbol{V}^\top \mathbf{x}_0) + \boldsymbol{U}^\top \mathbf{z}$

Figure 2: DDRM Inference Process for Inpainting Example Continued

The pseudo-code below describes the sampling algorithm used for our initial experiments with vanilla DDRM.

Algorithm 1: Sampling

```
Sample y from data distribution
Sample $\bar{x}_T$ using $P_\theta^{(T)}(\bar{x}_T|y)$
Set $x_T = V\bar{x}_T$
for t = T-1, ... , 1, 0 do
    Sample $\bar{x}_t$ using $P_\theta^{(t)}(\bar{x}_t|x_{t+1}, y)$
    Set $x_t = V\bar{x}_t$
End for
return $x_0$
```

3.3 Fine Tuning Diffusion Models

DDRM relies on a range of pretrained diffusion models. Since DDRM is only a sampling technique, we hypothesised that a diffusion model trained with driving scenes should improve the reconstruction accuracy. If the dataset of driving scenes excludes adverse weather then perhaps then perhaps this could also remove adverse weather during sampling.

The LSUN bedrom diffusion model selected and fine tuned for further experiments with DDRM. For more details please see our repository [1].

The pretrained LSUN bedroom model is based on the the pytorch re-implementation of Denoising Diffusion Probabilistic Mode (DDPM)[2] and instantiated with the same model configuration parameters used in DDRM. The model checkpoints were finetuned for 28,000 additional steps on a dataset of 1,000 clean images (see figure 4) from the RainDrop dataset on an AWS instance with a single NVIDIA A10G GPU. The training scheme was OpenAI's guided diffusion repo [3] with a learning rate of 2e-5, 1000 diffusion steps, linear noise schedule, and a batch size of 4 due to GPU memory limitations. The resultant EMA models were used per OpenAI's suggestion: *You will likely want to sample from the Exponential Moving Average (EMA) models, since those produce much better samples*.

[1] https://github.com/hifrickenfive/guided-diffusion-ft-ddpm

[2] https://github.com/pesser/pytorch_diffusion

[3] https://github.com/openai/guided-diffusion/tree/main/guided_diffusion

3.4 Experiment with Conditional Diffusion Model

Özdenizci et al.'s method divides the picture into patches and then utilizes conditional diffusion method. Since it's difficult to derive a degradation matrix H for adverse weather, we attempted to use the degraded picture as y below instead of having the linear transformation from x_0.

$$p_\theta(x_{0:T}|y) = p_\theta^{(T)}(x_T|y)\prod_{t=0}^{T-1} p_\theta^{(t)}(x_t|x_{t+1}, y).$$

The training data is the rain drop dataset with both original picture x_0 and degraded picture y. The model is trained with 140,000 iterations. For more details please see our repository[4].

3.5 Experiment with Plug and Play Image Restoration (DiffPIR)

Though DDRM endorsed a time-efficient approach which performs diffusion sampling to reconstruct the missing information in $\mathbf{y}$ in the spectral space of $\mathbf{H}$ with Singular Value Decomposition (SVD), it either needs to use hand-designed $\mathbf{H}$, or suffers from sampling speed to get favorable performance. The Plug-and-play posterior sampling methods (DiffPIR) proposed by Zhu et al. (2023) [8], could leverage the gradient of log posteriors to drive the samples to high-density regions. First, to separate the data term and prior term of the following optimization problem:

$$\hat{x} = \arg_x \min \frac{1}{2{\sigma_n}^2}||y - H(x)||^2 + \lambda P(x) \tag{2}$$

By introducing an auxiliary variable $\mathbf{z}$, equation 2 can be split into the following subproblems and be solved iteratively,

$$\begin{cases} z_k = \arg_z \min \frac{1}{2(\sqrt{\lambda/\mu})^2}||z - x_k||^2 + \lambda P(z) \\ x_{k-1} = \arg_x \min ||y - H(x)||^2 + \mu{\sigma_n}^2||x - z_k||^2 \end{cases} \tag{3}$$

Here the subproblem (equation 3a) with prior term is a Gaussian denoising problem, and the subproblem (equation 3b) with the data term is indeed approximal operator which usually has a closed-form solution that depends on $\mathbf{H}$. The transition workflow of DiffPIR can be found in Figure 3.

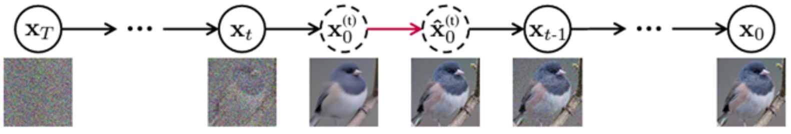


Figure 3: For every state x_t, following the prediction of the estimated ${x_0}^{(t)}$ by the diffusion model, the measurement y is incorporated by solving the data proximal subproblem (indicated by the red arrow). Subsequently, the next state x_{t-1} is derived by adding noise back [8].

Thus, completing one step of reverse diffusion sampling as shown in equation 4:

$$\begin{cases} x_0^{(t)} = \arg_x \min \frac{1}{2{\hat{\sigma}_t}^2}||z - x_t||^2 + P(z) \\ \hat{x}_0^{(t)} = \arg_x \min ||y - H(x)||^2 + \rho_t||x - x_o^{(t)}||^2 \\ x_{t-1} \leftarrow \hat{x}_0^{(t)} \end{cases} \tag{4}$$

As the data term and prior term are decoupled, the degradation model for the observed measurement y is linked solely to (equation 4b). In cases without an analytical solution to (equation 4b), it can be approximated using a first-order proximal operator, yielding the following solution:

$$\hat{x}_0^{(t)} \approx x_0^{(t)} - \frac{{\hat{\sigma}_t}^2}{2\lambda\sigma_n^2}\nabla_{x_0^{(t)}}||y - H(x_0^{(t)})||^2 \tag{5}$$

[4] https://github.com/WilsonZhang1913/ConditionalDiffusion

This represents a single numerical gradient descent step. In DDRM, Kawar et. al. [2] introduced variational distribution of variables in the spectral space of a linear operator H. While DDRM and DiffPIR share a structural similarity in predicting x_0 first and adding noise to forward sample x_{t-1}, DDRM is limited to linear H and may lack efficiency without fast SVD feasibility. In contrast, DiffPIR can handle arbitrary degradation operator H with equation 5. For more details, please see our repository [5].

4 Results

Evaluation/Experimental procedures The results in Section 4 experimented with de-raining of out-of-distribution driving images from the RainDrop dataset on two pretrained diffusion models (LSUN bedroom and Imagenet). The DDRM codebase was modified to not add additional degradation on the original RainDrop images. These experiments tested the hypothesis that de-raining is feasible despite using models not trained on driving images using the denoise degradation operator. Subsequent experiments were conducted to address the limitations found in DDRM for the use case of autonomous vehicles.

4.1 Experimenting with Vanilla DDRM

Initial experiments with vanilla DDRM revealed that adverse weather removal with diffusion models is possible. Figure 5 demonstrates modest de-raining effects on a single RainDrop dataset image. Notably, these effects were achieved by reducing the timestep parameter (5 or 10), despite the pretrained model being intended for different purposes (LSUN bedroom). However, the image quality suffered significantly, evident both visually and through a decrease in PSNR from 33.05 (timestep 1000) to 25.66 (timestep 5). For more details see our repository [6].

Inference Time In terms of inference time, processing a single image took four seconds, while a batch of 10 images required 12 seconds (0.8s per image), and a batch of 20 images took 17 seconds (1.2s per image). These tests utilized a timestep of 10 and were performed on a cloud instance with 16 vCPUs, 64GB of RAM, and an Nvidia T4 GPU, and a sampling batch size of 6.

Poor Reconstruction Accuracy The reconstruction accuracy is poor as shown in 3.1.3: (1) Deraining effects are limited to sections of the images where the road serves as the background, with rain effects persisting if the background is the sky, buildings, or objects; (2) where rain is removed, the road image loses significant quality, leading to the loss of lane markings, which is dangerous and undesirable; (3) distant objects, such as cars on the horizon, become blurred, which is also dangerous and undesirable.

The Effect of Changing Pretrained Models Results sampled using the ImageNet diffusion model in figure 7 were better than those sampled from the LSUN model in figure 6. It's hypothesised that ImageNet's pretraining data is more diverse than LSUN bedroom's and therefore better at handling out-of-distribution samples like our driving images. The deraining effect was subjectively equivalent to the LSUN results. However, the ImageNet results was capable of retaining original images' details better than LSUN.

4.2 Experiment with DDRM and a Finetuned Diffusion Model

Disappointingly, the reconstruction accuracy was worse when the pretrained diffusion model was swapped for the finetuned LSUN bedroom model. Across twenty images compared in figure 8, they generally fared worse than the outputs from the pretrained model. The DDRM parameters were set to timestep 10 and sigma 0.05,

In autonomous driving three specific features are important: lane markings, objects in the horizon, and road sign lettering. The outputs from the fine tuned model were all worse than the outputs from the pretrained model as shown in figures 9, 10, 11. The fine tuned model was also evaluated with

[5] https://github.com/CMPeng/DiffPIR_DDRM

[6] https://github.com/hifrickenfive/ddrm_test

different DDRM parameters such as higher sigma 0.25 and number of timesteps but the sample qualities were still unremarkable.

4.3 Experiment with Conditional Diffusion Model

The model is trained with time-step 25, and linear β schedule starting from 0.0001. Images are resized to 256X256. While effectively mitigating rain effects, this approach introduces a notable alteration in the background color, as illustrated in Figures 12 and 13. Consequently, the evaluation metrics yield sub-optimal results, with a PSNR of 18.4445 and SSIM of 0.7600 based on 47 samples. Addressing the background color change is a topic slated for future research endeavors. The color changes might be due to the fact that the degraded image y is concatenated with output of prior time step x_t, resulting in a six dimensional input image channels.

4.4 Experiment with DiffPIR

Image Manifold On one hand, unconditional pre-trained diffusion models can be employed for conditional generation by integrating an additional classifier. To enhance accuracy, we propose fine-tuning the pre-trained models using both clean images and those affected by adverse weather conditions, thereby enabling the model to better learn the degradation matrix. On the other hand, only images with introduced noise are utilized during inference or the reverse diffusion process (as depicted in Appendix, Figure 14). Initial observations reveal that the analytical solution provides little assistance in the early stages, prompting us to consider skipping this phase. As discussed in Appendix section 6.4, through experimentation, we determine the optimal starting time step (t_{start}) for this phase.

Experiment Set-up This experiment utilized pre-trained ImageNet and FFHQ models with a consistent noise schedule β_t across all methods. NFEs are reported for comparison, and degradation models include: (i) Inpainting with a box-type mask (128×128 region), random-type mask (half pixels randomly masked), and prepared mask images. (ii) Gaussian blur (61×61, $\sigma = 3.0$) and motion blur (61×61, $intensity = 0.5$) with a uniform kernel for fair comparison. (iii) SR with bicubic down-sampling. Image inpainting focuses on the noiseless case, while deblurring and SR experiments consider both noisy and noiseless settings. All images are normalized to [0, 1].

Noisy FFHQ		Deblur (Gaussian)			Deblur (motion)			SR(x4)		
Method	NFE	PSNR	FID	LPIPS	PSNR	FID	LPIPS	PSNR	FID	LPIPS
DiffPIR	100	27.36	59.65	0.236	26.57	65.78	0.2555	26.64	65.77	0.26
DDRM	20	25.93	101.89	0.298	N/A	N/A	N/A	27.92	89.43	0.265
Noiseless FFHQ		Deblur (Gaussian)			Deblur (motion)			SR(x4)		
Method	NFE	PSNR	FID	LPIPS	PSNR	FID	LPIPS	PSNR	FID	LPIPS
DiffPIR	100	31	39.27	0.152	37.53	11.54	0.064	29.52	47.8	0.174
DDRM	20	28.4	67.99	0.238	N/A	N/A	N/A	30.09	68.59	0.188

Table 1: Noisy (top) and noiseless (bottom) quantitative results trained on FFHQ, being applied to RainDrop Dataset. We compute the average PSNR(dB), FID and LPIPS of different methods on Gaussian deblurring, motion deblurring and 4×SR.

FID assesses visual quality and distribution distance, while LPIPS gauges perceptual similarity. PSNR evaluates restoration faithfulness, though it's less crucial for IR tasks. DiffPIR offers pre-trained models for both FFHQ 256×256 and ImageNet 256×256 datasets, but Table 1 displays only the former. (1) For $\sigma_n = 0.05$ noise, all methods are evaluated on both datasets for 4×SR, Gaussian deblurring, and motion deblurring, excluding DDRM from motion deblurring due to its support only for separable kernels.

Table 1 highlights DiffPIR's superior FID and LPIPS performance on FFHQ compared to DDRM, with competitive PSNR scores - except for SR's LPIPS score, possibly affected by inaccuracies in the approximated bicubic kernels k, leading to accumulated errors during sampling. (2) For noiseless measurement with $\sigma_n = 0.0$, all methods are assessed on FFHQ 256×256 for image inpainting, deblurring, and SR. DiffPIR with 100 NFEs notably outperforms DDRM in FID and LPIPS.

While both DiffPIR and DDRM exhibit a PSNR advantage in noiseless tasks, the generated images lack high perceptual quality. Even with 20 NFEs, DiffPIR achieves competitive FID and LPIPS scores, though its visual quality, especially for tasks like inpainting, doesn't match methods like DDRM or DPS.

5 Analysis

DDRM Inference Time is Too Slow For Autonomous Vehicles The inference times on a per-image basis would be too slow for real-time autonomous vehicle navigation systems, with the ideal target being around 10Hz or 10 frames per second (0.1s per image). Object detection algorithms optimized for embedded systems can run between 10-30Hz. The baseline inference time is 0.8s per image. It's worth noting that these times include certain overheads, such as the DDRM codebase making three copies of the original image for analysis so there's many opportunities to reduce inference times. Even if these changes were explored, it would be unlikely to be performant on embedded devices in the case of autonomous vehicles, which are much more limited in compute than cloud instances.

DDRM Sample Quality is Unacceptable For Autonomous Vehicles The reconstruction quality with the pretrained models used by DDRM was predictably poor, as these models were not originally trained on data related to autonomous vehicles. Two hypotheses for the root cause are (a) the pretrained model needs to be based on driving scenes, and (b) the linear denoising degradation operator needs to better describe non-linear adverse weather effects.

An attempt to create our own pre-trained diffusion model based entirely on driving images was ultimately abandoned due to the prohibitive training time (5.5 days on an engineering spec laptop). For completeness of the report, this attempt took 1000 night time driving scenes from the IUPUI dataset.

Hypothesising Why DDRM Samples Are Worse with Finetuned Models There are two possible reasons to explain why the DDRM samples performed worse with the fine tuned model: (1) naive fine-tuning, (2) too few training images. Zhu et al. (2023) demonstrated that fine tuning diffusion models all the parameters, as was done for this project, easily results in overfitting [10]. The fine tuning methodology in our project simply continued on from the pretrained LSUN checkpoint. Alternative approaches include freezing the majority of the parameters and introducing adaptor modules to support domain adaptation. However, these efforts don't overcome fundamental limitations in DDRM for autonomous driving discussed below.

Limitations with DDRM's Codebase and Sampling Algorithm For Practical Applications DDRM faces three significant limitations. Firstly, DDRM's codebase is not suitable for our objective of denoising images already degraded. The codebase processes original images $x_{original}$ by applying a known degradation matrix H to create y_0 and then attempts to reverse this process. Secondly, even with modifications, DDRM's sampling algorithm still requires y_0 to be derived from $y_0 = Hx + z$, which necessitates knowledge about $x_{original}$, which is impractical in real-world applications. When we experimented with H such that $y_0 \sim x_{original}$ we discovered that the algorithm makes assumptions compatibility about the SVD components of degradation matrix H and the SVD components of y_0. Lastly, DDRM's requires hand curated degradation matrices, which isn't generalizable to the full domain of adverse weather conditions.

Conditional Diffusion Model vs. DDRM Conditional diffusion can remove most of the raindrops unlike DDRM. Unfortunately, the change in background color remains an improvement opportunity for conditional diffusion. Furthermore, conditional diffusion is approximately 3.5s second per image samples, which is slower than DDRM, which is approximately 1.2s per image sampled as mentioned in section 4.1.

Comparing DiffPIR to DDRM The integration of DiffPIR into DDRM, showcases its HQS-based diffusion sampling approach. DiffPIR incorporates off-the-shelf diffusion models as denoising priors and solves the data subproblem in the clean image manifold. Extensive experiments reveal DiffPIR's superior flexibility, efficiency, and generalizability over DDRM with non-linear noises.

While competitive in noisy settings (with comparable PSNR scores) and noiseless setups (with FID and LPIPS scores), its visual quality, particularly for tasks like inpainting, falls short of DDRM.

6 Conclusions and Next Steps

In conclusion, our work found that while promising, current generative diffusion models and techniques are not suitable for adverse weather removal for autonomous vehicle use cases. Fundamentally, they suffer from at least one of these three problems:

1. Don't remove adverse weather effectively;
2. When adverse weather is fully removed, the techniques fail to preserve the underlying image;
3. They are too slow to sample. Unacceptable for fast moving autonomous vehicles, like passenger cars.

We arrived at this conclusion by the experiments with vanilla DDRM, which concluded that it was incapable of fulfilling the objectives. We then experimented with conditional diffusion and DiffPIR to address the limitations identified in our experiments with DDRM.

Future work and improvements include:

1. Creating a pretrained model entirely based on car driving scenes. Which could improve sampling quality for out of distribution car scene images. But this is time consuming and expensive to train.
2. Investigating why, despite removing adverse rain effects, the conditional diffusion model altered the background color.
3. Enhancing the comprehension of both noiseless and noisy scenarios within DiffPIR and refining the encoding of non-linear transformations is imperative.
4. Investigating to what extent images sampled from diffusion modes affect downstream object detection algorithms. There remains a risk for autonomous vehicles whereby the performance of object detection algorithms used for navigation are degraded despite subjectively perfect weather removed images.

Acknowledgements

We express gratitude to Pratyush Agarwal for insightful discussions and extend our appreciation to Bahjat Kawar et. al. for the inspirations derived from their published work.

References

[1] Ozan Özdenizci and Robert Legenstein. "Restoring Vision in Adverse Weather Conditions with Patch-Based Denoising Diffusion Models". In: (2022). arXiv: 2207.14626 [cs.CV].

[2] Bahjat Kawar et al. *Denoising Diffusion Restoration Models*. 2022. arXiv: 2201.11793 [eess.IV].

[3] Yijun Yang et al. *Video Adverse-Weather-Component Suppression Network via Weather Messenger and Adversarial Backpropagation*. 2023. arXiv: 2309.13700 [cs.CV].

[4] Jeya Maria et. al. "TransWeather: Transformer-based Restoration of Images Degraded by Adverse Weather Conditions". In: (2022). arXiv: 2111.14813 [cs.CV].

[5] Yuxiao Zhang et al. "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 196 (2023), pp. 146–177. ISSN: 0924-2716.

[6] *traindrop*. Accessed: 2023-10-22. 2023. URL: https://paperswithcode.com/dataset/raindrop.

[7] Jiang Yu Zheng. "IUPUI driving videos and images in all weather and illumination conditions". In: *arXiv preprint arXiv:2104.08657* (2021).

[8] Yuanzhi Zhu et al. "Denoising Diffusion Models for Plug-and-Play Image Restoration". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (NTIRE)*. 2023.

[9] Chenlin Meng Jiaming Song and Stefano Ermon. "Denoising diffusion implicit models". In: *International Conference on Learning Representations* (2021).

[10] Taehong Moon Zhu et al. *Fine-tuning Diffusion Models with Limited Data*. 2022.