

# ***ORL-LDM: Offline Reinforcement Learning Guided Latent Diffusion Model Super-Resolution Reconstruction***

**Shijie Lyu<sup>1,a,\*</sup>**

<sup>1</sup>*Georgia institute of technology, North Avenue Atlanta, GA, 30332, United States of America*

*a. slyu41@gatech.edu*

*\*corresponding author*

**Abstract:** With the rapid advancement of remote sensing technology, super-resolution image reconstruction is of great research and practical significance. Existing deep learning methods have made progress but still face limitations in handling complex scenes and preserving image details. This paper proposes a reinforcement learning-based latent diffusion model (LDM) fine-tuning method for remote sensing image super-resolution. The method constructs a reinforcement learning environment with states, actions, and rewards, optimizing decision objectives through proximal policy optimization (PPO) during the reverse denoising process of the LDM model. Experiments on the RESISC45 dataset show significant improvements over the baseline model in PSNR, SSIM, and LPIPS, with PSNR increasing by 3-4dB, SSIM improving by 0.08-0.11, and LPIPS reducing by 0.06-0.10, particularly in structured and complex natural scenes. The results demonstrate the method's effectiveness in enhancing super-resolution quality and adaptability across scenes.

**Keywords:** Hyperresolution of remote sensing images, Potential diffusion model, reinforcement learning, Offline fine-tuning, PPO

## **1. Introduction**

Super-Resolution (SR) image reconstruction is essential for remote sensing applications like land monitoring and urban planning, where high-resolution images are crucial. Traditional SR methods, such as NEDI, SAI, and SR-NARM, enhance resolution through interpolation but are computationally complex. Deep learning techniques, including CNNs, DRCNs, and sparse coding models, have shown better performance, though they require large datasets and are computationally demanding. Generative models like GANs, VAEs, and Diffusion Models (e.g., SR3 and EDiffSR) provide promising alternatives, with Diffusion Models offering greater efficiency[1-2].

Pre-trained models for SR often require fine-tuning, and reinforcement learning (RL) can be used to optimize these models. This paper proposes an offline RL fine-tuning approach, ORL-LDM, which decouples RL from model training[3-4]. By creating an independent agent-environment setup, it reduces computational costs and improves the model's applicability to various SR tasks[5]. A comprehensive reward function that combines image quality metrics (PSNR, SSIM, LPIPS) and computational efficiency is used to guide optimization[6].

Experiments on the RESISC45 dataset demonstrate the effectiveness of ORL-LDM, showing significant improvements in SR performance, particularly in structured scenes[7]. The method

achieved up to a 3-4 dB increase in PSNR, proving its effectiveness and generalizability in real-world applications[8].

## 2. Principle of LDM model

### 2.1. Reverse sampling for denoising

This paper introduces the ORL-LDM method of offline reinforcement learning fine-tuning model from two parts: the principle of diffusion model LDM [9] and the construction of RL agent environment. The LDM model is used to receive the low-resolution image and generate the super-resolution image, and the RL agent is used to optimize the decision objective in the denoising stage to adjust the parameters of the LDM model.

LDM is a generative model of diffusion in latent space. The principle of the diffusion model is somewhat different from that of the previous diffusion model, so the introduction of LDM model needs to introduce the principle of diffusion model. The diffusion model is mainly composed of two parts, the forward noise-adding Markov process and the reverse sampling denoising two processes as shown in FIG. 1.

The forward noise-adding process of the diffusion model can be described as a gradual introduction of Gaussian noise, the purpose of which is to transform the original data distribution  $p(x_0)$  into a form close to the standard Gaussian distribution  $p(x_T) \approx N(x_T; 0, I)$ . This process is achieved by defining a series of Markov chains. At each time step  $t \in \{1, \dots, T\}$ , given the state  $x_{t-1}$  at the previous time step, we model the distribution at the current time step  $x_t$  as the following Gaussian distribution.

The nature of the forward noise-adding process is as follows: as the time step  $t$  increases,  $t$  gradually decreases,  $(1 - t)$  gradually increases, and the original data is covered by noise step by step. For  $t = T$ , the distribution of data  $x_T$  is close to the standard Gaussian distribution  $q(x_T) \approx N(x_T; 0, I)$ . The state  $x_t$  at any time step can be directly generated by the above formula.

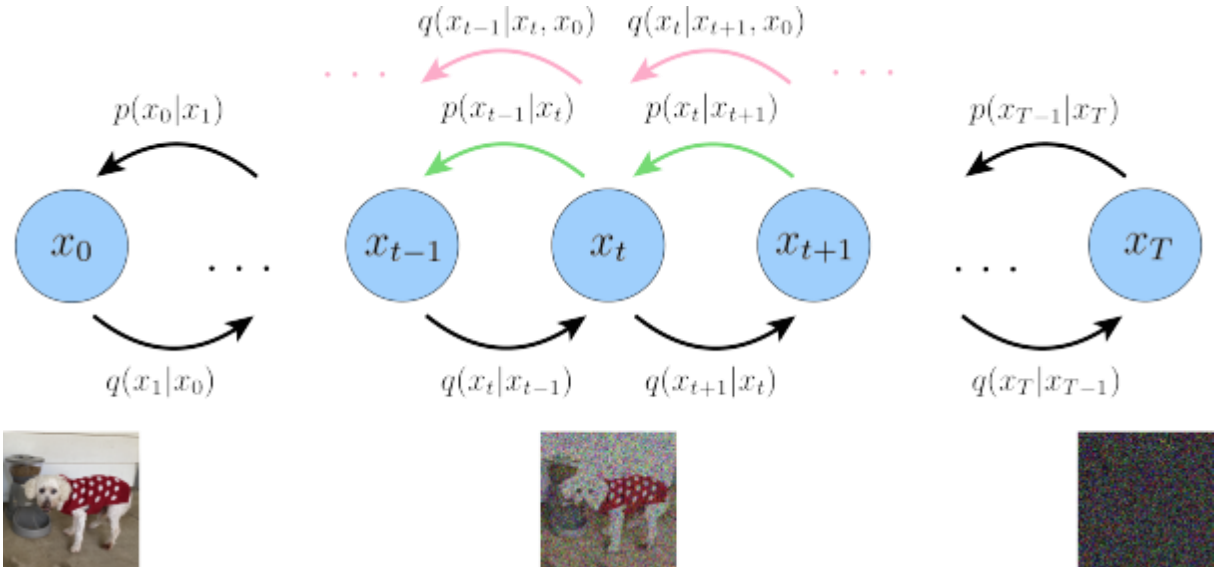


Figure 1: Diffusion model process

### 2.2. Reverse sampling for denoising

In the process of forward noise addition, the original data  $x_0$  is gradually noised until it becomes pure noise  $x_T$ . However, the goal of the reverse denoising process is to reconstruct the original data  $x_0$

starting from Gaussian noise  $x_T$  and gradually denoising it. This process is modeled as a backward Markov chain as shown in FIG. 2.

To optimize the model parameters  $\theta$ , we minimize the KL divergence between the posterior distribution  $p_\theta(x_{t-1}|x_t)$  and the true distribution  $q(x_{t-1}|x_t)$  as follows. During inference, we sample an initial point from the Gaussian noise distribution  $p(x_T) = N(x_T; 0, I)$ , and then generate  $x_{T1}$ ,  $x_{T2}$ , and  $x_0$  through the trained denoising network. By combining forward and backward denoising, the diffusion model can efficiently generate realistic data.

Although traditional diffusion models perform well in generating high-quality data samples, the computational overhead of directly performing the diffusion process in high-dimensional data Spaces is extremely high, which limits their application in high-resolution data generation. In order to solve this problem, Latent Diffusion Models (LDM) are proposed. LDM drastically reduces the computational complexity by performing the diffusion process in a low-dimensional latent space, while maintaining the high quality of the generated data. The core idea of LDM is to first use a pre-trained Encoder (Encoder) to map the high-dimensional data  $x_0$  to the low-dimensional latent space  $z_0$ , and then carry out the forward noise addition and reverse denoising process in the latent space. Finally, a pre-trained Decoder is used to restore the generated latent variable  $z_0$  to the high-dimensional data space.

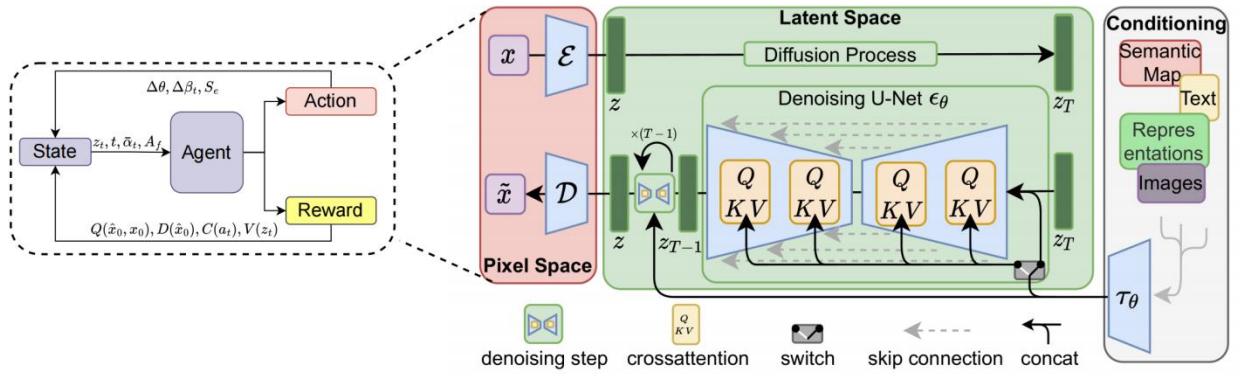


Figure 2: Schematic of the overall architecture of LDM integrated with PPO agent

### 2.3. Latent Diffusion Models

First, the original data  $x_0$  is mapped to the latent space using the encoder  $\epsilon$ . In the latent space, the forward noise addition process is similar to the traditional diffusion model, but the computational efficiency is significantly improved due to the lower dimensionality of the latent space. Here,  $\alpha_t$  is the same as in the traditional diffusion model and controls the noise intensity added at each step. The inverse denoising process aims to gradually recover the initial latent variable  $z_0$  from the Gaussian noise  $z_t$  in the latent space. To optimize the model parameters  $\theta$ , the KL divergence between the backward distribution  $p_\theta(z_{t-1}|z_t)$  and the true distribution  $q(z_{t-1}|z_t, z_0)$  is minimized. In particular, the last step exploits the property of the KL divergence of the Gaussian distribution to reduce the optimization objective to the mean squared error between the conditional mean  $\mu_\theta(z_t, t)$  and the true mean  $t(z_t, z_0)$ . To further simplify the optimization process, LDM usually represents  $t(z_t, z_0)$  as the noise  $\epsilon_\theta(z_t, t)$  predicted by the denoising network. Therefore, the optimization objective can be transformed into minimizing the difference between the predicted noise  $\epsilon_\theta(z_t, t)$  and the true noise  $\epsilon$ . Finally, the generated latent variable  $z_0$  is reduced to high-dimensional data through the decoder  $D$ . LDM significantly reduces the requirement of computational resources by conducting the diffusion process in the low-dimensional latent space, making it more efficient in the task of high-resolution data generation. Furthermore, LDM is able to combine pre-trained latent representations such as VAE

or CLIP to further improve generation quality and diversity. To further enhance the generation ability, LDM can achieve conditional generation by introducing conditional information such as text, category labels, etc. This extension is called Conditional Latent Diffusion Models (CLDM). The Condition information is usually embedded in the latent space through a Condition Encoder that guides the generation process to satisfy specific condition constraints. By introducing conditional information, the model can be controlled by specific conditions in the generation process, such as text description to generate images, image generation of specific categories and other tasks.

#### 2.4. An Environment for RL Agents

A reinforcement learning environment consists of states, actions, and rewards, where State: represents specific information or features of the current environment, which can be discrete or continuous and is denoted by  $s \in S$ , where  $S$  is the state space. Action: An action that the agent can choose to perform in each state, denoted by  $a \in A(s)$ , where  $A(s)$  is the set of actions available in state  $s$ . Reward: is a feedback signal returned by the environment after the agent performs an action, denoted by  $r \in R$ . The goal of the reward is to guide the agent to learn an optimal policy. During reinforcement learning, the agent, through interaction with the environment, observes a state  $s$ , selects an action  $a$ , and receives a reward  $r$  and the next state  $s'$ . This interactive loop is often modeled by a Markov Decision process (MDP), where the goal is to maximize the cumulative reward (often referred to as the payoff).

In order to effectively train reinforcement learning agents, we choose to use the Proximal Policy Optimization (PPO) algorithm. PPO is a policy gradient-based method, which is simple, stable and efficient, and suitable for tasks with continuous action space and high-dimensional state space. By limiting the step size of policy update, PPO avoids the instability caused by too large policy update. The key idea is to use the clipping method to limit the differences between the old and new policies during the policy update process, so as to achieve stable policy optimization.

### 3. Dataset and experiment

The dataset used in the experiments is RESISC45, which is provided by [10] and is specialized for remote sensing image classification tasks. RESISC45 contains 45 different scene categories, and each category contains 700 high-resolution remote sensing images, for a total of 31500 images. The size of the image is  $244 \times 244$  pixels, which covers a variety of geographical scenes such as city, forest, ocean, and desert, and has high diversity and complexity. In this study, the RESISC45 dataset is loaded by the load dataset function in the datasets library, and the dataset is divided according to the standard training set and test set ratio to ensure the fairness and effectiveness of the model training and evaluation.

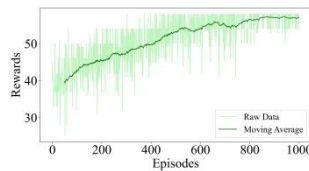


Figure 3: ORL-LDM Training Reward

The training process of the model was carried out on the Google Colab platform with NVIDIA T4 GPU (15 GB memory), Ubuntu 18.04, Python 3.8 as the programming language, and PyTorch 2.1 as the deep learning framework. And use the following libraries: diffusers, transformers, datasets, accelerate, lpips, torchmetrics, pandas, and piqa. The specific hyperparameter configuration of the PPO algorithm is shown in Table 1. To ensure reproducibility of the experiment, all random seeds have been fixed. The gradual convergence of the model is shown in FIG. 3.

Table 1: THE MAIN HYPERPARAMETER CONFIGURATION OF PPO ALPGRITHM

Parameter	Value	Parameter	Value
Learning Rate	3.00E-04	Training Epochs	1000
Discount Factor ( $\gamma$ )	0.99	Steps per Epoch	1000
Generalized Advantage Estimation	0.95	Batch Size	64
Policy Clipping Range	0.2	Policy Update Frequency	4
Value Function Coefficient	0.5	Random Seed	42
Reward Function Weight - PSNR	0.4	Reward Function Weight - SSIM	0.3
Reward Function Weight - LPIPS	0.2	Reward Function Weight - Computational Efficiency	0.1

This study uses Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learning-aware Image Patch Similarity (LPIPS) to evaluate model performance. Table 2 compares results with and without the RL model in various scenarios. The results show significant improvement with the RL model across all metrics. Specifically, PSNR improves by 3-4 dB, with the runway scene showing the most prominent gain (from 30.6392 dB to 34.4581 dB). SSIM also increases, with the industrial area SSIM rising from 0.7076 to 0.8176. For LPIPS, a lower value indicates better perceptual quality, and all scenes show a decrease in LPIPS, with the runway scenario improving from 0.4252 to 0.3252. The RL model demonstrates more significant improvements in structured scenes (e.g., runway, industrial area) and moderate gains in natural scenes (e.g., forests), indicating its ability to adapt to scene features and optimize accordingly. Figure 4 compares the super-resolved images with the original ones.

Table 2: THE MAIN HYPERPARAMETER CONFIGURATION OF PPO ALPGRITHM

Model	Categories	PSNR	SSIM	LPIPS
No RL model is used	Business district	24.8506	0.6224	0.5504
	Dense residential area	24.0548	0.5512	0.6235
	desert	27.4842	0.5825	0.6364
	forest	21.8969	0.4360	0.7824
	Industrial area	27.2189	0.7076	0.5414
	Train station	26.9435	0.6841	0.5245
	river	28.6742	0.6835	0.5651
	runway	30.6392	0.7714	0.4252
RL model is used	Business district	27.9821	0.7012	0.4892
	Dense residential area	27.2365	0.6723	0.5135
	desert	31.2947	0.7125	0.5164
	forest	25.8672	0.5860	0.6224
	Industrial area	30.8934	0.8176	0.4614
	Train station	29.9523	0.7841	0.4245
	river	31.9247	0.7635	0.4751
	runway	34.4581	0.8114	0.3252



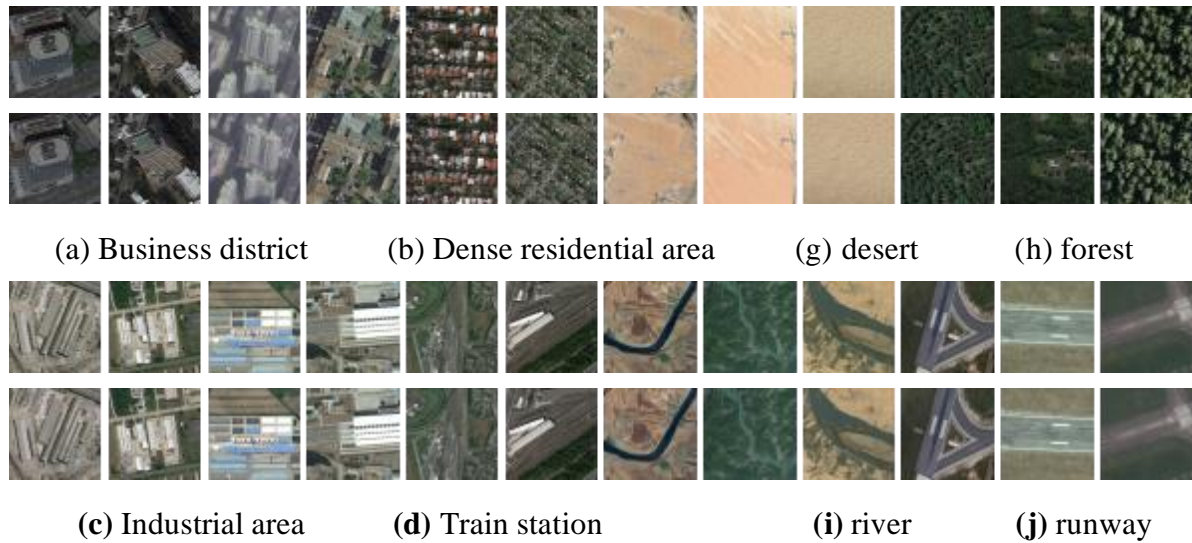


Figure 4: Comparison of different regions

#### 4. Conclusion

In this paper, we propose a reinforcement learning-based LDM model fine-tuning method for remote sensing image super-resolution reconstruction task. The experimental results show that the proposed method has the following advantages: (1) the three evaluation indicators of PSNR, SSIM and LPIPS are significantly improved, especially in scenes with regular structure; (2) It has good adaptability to different types of scenes, even in natural scenes with complex textures, it can also achieve significant improvement; (3) The fine-tuning strategy of offline reinforcement learning is simple, effective and easy to implement. These results show that the application of reinforcement learning to LDM model fine-tuning is an effective technical route, which provides a new solution for remote sensing image super-resolution reconstruction. Future work will further explore better reward function design and agent training strategy to further improve the model performance.

#### References

- [1] Sun Y, Ortiz J. An ai-based system utilizing iot-enabled ambient sensors and llms for complex activity tracking[J]. *arXiv preprint arXiv:2407.02606*, 2024.
- [2] Lin W. A Review of Multimodal Interaction Technologies in Virtual Meetings[J]. *Journal of Computer Technology and Applied Mathematics*, 2024, 1(4): 60-68.
- [3] Lyu S. The Application of Generative AI in Virtual Reality and Augmented Reality[J]. *Journal of Industrial Engineering and Applied Science*, 2024, 2(6): 1-9.
- [4] Sun, Y., & Ortiz, J. (2024). Machine Learning-Driven Pedestrian Recognition and Behavior Prediction for Enhancing Public Safety in Smart Cities. *Journal of Artificial Intelligence and Information*, 1, 51-57.
- [5] Lyu S. The Technology of Face Synthesis and Editing Based on Generative Models[J]. *Journal of Computer Technology and Applied Mathematics*, 2024, 1(4): 21-27.
- [6] Lin W. A Systematic Review of Computer Vision-Based Virtual Conference Assistants and Gesture Recognition[J]. *Journal of Computer Technology and Applied Mathematics*, 2024, 1(4): 28-35.
- [7] Lyu S. Machine Vision-Based Automatic Detection for Electromechanical Equipment[J]. *Journal of Computer Technology and Applied Mathematics*, 2024, 1(4): 12-20.
- [8] SINGH A, SINGH J. Content adaptive single image interpolation based super resolution of compressed images[J/OL]. *International Journal of Electrical and Computer Engineering (IJECE)*, 2020. DOI: 10.11591/ijece.v10i3.pp3014-3021.
- [9] Wu B, Cai Z, Wu W, et al. Aol-aware resource management for smart health via deep reinforcement learning[J]. *IEEE Access*, 2023.

- [10] CHENG G, HAN J, LU X. *Remote sensing image scene classification: Benchmark and state of the art[J/OL]. Proceedings of the IEEE, 2017, 105(10):1865-1883. <http://dx.doi.org/10.1109/JPROC.2017.2675998>. DOI: 10.1109/jproc.2017.2675998.*