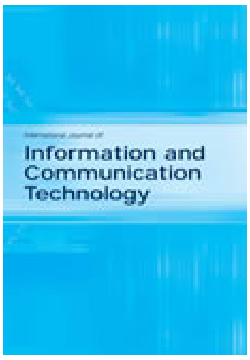



**International Journal of Information and Communication Technology**

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

---

# Obstacle-free robot path planning based on variational autoencoder and generative networks

Laide Guan, Bole Li

**Article History:**

| | |
|---|---|
| Received: | 10 February 2025 |
| Last revised: | 19 February 2025 |
| Accepted: | 19 February 2025 |
| Published online: | 15 April 2025 |

# Obstacle-free robot path planning based on variational autoencoder and generative networks

Laide Guan* and Bole Li

School of Mechanical and Electrical Engineering,
Liuzhou Vocational and Technical College,
Liuzhou 545006, China
Email: welongcom126@126.com
Email: libole520@163.com
*Corresponding author

**Abstract:** Robot route planning becomes crucial in intelligent navigation systems given the fast advancement of automation technologies. Concerning efficiency and resilience in handling challenging dynamic settings, traditional path-planning approaches have several restrictions. Based on variational autoencoder (VAE) and generative adversarial network (GAN), this work presents a path planning model, VAE-GAN PathNet, to handle this challenge. By combining the benefits of VAE in latent space modelling and the capacity of GAN in path optimisation, the model essentially increases the quality, smoothness and obstacle avoidance performance of path planning. This work uses the Stanford Drone Dataset and the ROS Path Planning Dataset to validate the efficacy of the model using trials. In terms of path length, obstacle avoidance performance, path smoothness and computing time, VAE-GAN PathNet beats conventional path planning algorithms experimental data demonstrate.

**Keywords:** variational autoencoder; VAE; generative adversarial network; GAN; path planning; obstacle avoidance; path smoothness.

**Reference** to this paper should be made as follows: Guan, L. and Li, B. (2025) ‘Obstacle-free robot path planning based on variational autoencoder and generative networks’, *Int. J. Information and Communication Technology*, Vol. 26, No. 7, pp.17–31.

**Biographical notes:** Laide Guan received his Master’s degree in Engineering from Guangxi University in 2009. He is currently an Associate Professor at the Liuzhou Vocational and Technical College. His research areas and directions include intelligent robots, intelligent equipment, machine vision and navigation path planning.

Bole Li graduated from Guangxi University of Science and Technology in 2010. He is currently a full-time teacher and Senior Engineer at the School of Mechanical and Electrical Engineering, Liuzhou Vocational and Technical College. His research areas and directions include intelligent manufacturing, intelligent equipment, and the application of machine vision technology.

# 1 Introduction

As artificial intelligence technology develops rapidly, robots are increasingly used in the field of automation and intelligence (Soori et al., 2023), particularly in the scenarios of automatic driving, logistics and distribution, unmanned aerial vehicles, service robots, etc., path planning plays a vital role as one of the fundamental tasks. The path planning challenge calls for the robot to be able to avoid obstacles, identify an ideal path from the beginning to the finish point, maximise the efficiency of the path as much as feasible in a complicated environment (Willms and Yang, 2006).

Graph search algorithms such A* and Dijkstra's algorithm have been extensively applied in conventional path planning techniques (Karur et al., 2021). Under the assumption of ensuring the shortest path, the A* algorithm leads the search process by a heuristic function and can thus increase the search efficiency (Yiu et al., 2019). But when the size of the environment grows, the computing complexity rises dramatically and A* method is less flexible to the changes in the obstacle in the dynamic environment (Zhang et al., 2019).

Researchers have started to try to apply neural networks to path planning activities, particularly the application of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in environment perception and path prediction, with the ongoing development of deep learning technology (Vulpi et al., 2021). Picture processing applications, which can anticipate pathways by extracting picture data and handle challenging visual inputs, make extensive use of CNNs (Han and Wang, 2021). Conversely, RNNs notably fit path generation in dynamic contexts since they employ its features of processing time series data to create future paths from historical path information. These techniques, meanwhile, have several flaws that make it challenging to satisfy the criteria of real-time and effective path planning: great computational volume, low adaptation to environmental changes, and inadequate model generalisation capacity.

Generative models are progressively starting to be used in path planning. As an unsupervised learning model, variational autoencoder (VAE) can generate pathways fitting to goal constraints by means of potential space modelling (Biswas et al., 2023). VAE uses an encoder to map the input paths to the potential space and decoder reconstruction produces a smooth and practical path (Yao and Bekhor, 2022). VAE is appropriate for a range of path planning activities including obstacle avoidance and path smoothing since it can create paths with variation. Nevertheless, the potential area limits the quality of the produced routes of VAE models, which could result in less variety and less resilience of the paths.

Applied to the subject of path planning recently, generative adversarial network (GAN) has shown amazing performance in tasks including creating visuals and texts (Shahriar, 2022). By means of adversarial training of generators and discriminators, GAN generates not only conformable pathways but also highly smooth and realistic paths, hence overcoming restrictions. In path planning, the generator creates potential paths and the discriminator assesses their quality, therefore always improving the efficiency of the produced paths. Particularly helpful in avoiding local optimums and enhancing path smoothness, GAN is able to obtain higher quality paths in the process of path generating than VAE. GAN's training method is more complex, nevertheless, and prone to issues including unstable training and mode collapse.

In this work, we offer VAE-GAN PathNet, a path planning model grounded on VAE and GAN combination. The model generates smooth, practical paths and optimises the quality of the paths by adversarial training by combining the benefits of VAE in latent space modelling with GAN in path optimisation.

This work presents the following innovations:

1 Path planning is done using VAE and GAN together. In this work, we creatively combine VAE with GAN to offer the VAE-GAN PathNet model. The quality and resilience of path planning can be sufficiently enhanced by employing VAE for learning the possible representation space of paths and concurrently optimising the generation of paths using adversarial training of GAN. This combination improves the feasibility and smoothness of paths and helps to offset the local optimum issue that could arise during VAE path generating.

2 Use several evaluation criteria simultaneously. To fully assess the performance of the VAE-GAN PathNet model, this work employs several evaluation criteria in the experimental design: path length, obstacle avoidance performance, path smoothness and computation time. These indicators taken together not only assess the path's overall quality but also facilitate the analysis of the model's performance in several settings. Comparative and ablation studies confirm many benefits of VAE-GAN PathNet over conventional path planning techniques in several spheres.

3 In path planning, smoothness optimisation and obstacle avoidance. Among the main criteria in path planning are path smoothness and obstacle avoidance. While the VAE-GAN PathNet model optimises the generated paths through adversarial learning, so that the planned paths not only avoid obstacles but also perform superiorly in terms of smoothness and naturalness. Typical shortcomings in obstacle avoidance and path smoothness are found in traditional path planning algorithms. The created path not only satisfies the restrictions but also helps to prevent sudden bends and significant changes, so enhancing the feasibility of the road and its efficiency in useful purposes.

4 Optimisation for dynamic path planning. While VAE-GAN PathNet is able to quickly modify the path planning in the presence of impediments through its generative model, hence boosting the real-time and dynamic responsiveness of path planning, traditional path planning algorithms have low adaptability in dynamic situations. In this work, we demonstrate in dynamic environments – particularly in terms of path smoothness and obstacle avoidance capacity – that the model beats conventional approaches.

## 2 Relevant technologies

### *2.1 Variational autoencoder*

By mapping the input data to a probability distribution in the latent space and sampling from it, VAE is a generative model creating fresh data. VAE generates different samples by approximating the posterior distribution of the data using variational inference techniques unlike those of conventional self-encoders. VAE may provide several candidate courses by sampling from the possible space and create obstacle avoidance paths depending on environmental information in the path planning problem.

VAE's central concept is to maximise the variational lower bound so approximating the marginal likelihood of the data (Zalman and Fine, 2023). One aims to maximise the following marginal log-likelihood:

$$\log p_\theta(x) = E_{q_\phi(z|x)}\left[\log p_\theta(x|z) - \mathrm{KL}q_\phi(z|x) \| p_\theta(z)\right] \tag{1}$$

where $x$ is the input data; $z$ is the latent variable; $p_\theta(x \mid z)$ is the conditional probability of the data produced by the decoder; $q_\phi(z \mid x)$ is the approximative posterior distribution of the latent variable produced by the encoder; KL is the Kullback-Leibler scatter, used as a metric of the difference between the approximative posterior distribution and the prior distribution.

The VAE is maximised by optimising the variational lower bound since direct computation of the marginal log-likelihood is not practical:

$$L(\theta, \phi; x) = -E_{q_\phi(z|x)}\left[\log p_\theta(x|z) + \mathrm{KL}q_\phi(z|x) \| p_\theta(z)\right] \tag{2}$$

Usually considered as Gaussian with mean $\mu$ and variance $\sigma^2$, the encoder in VAE maps the input data $x$ to a distribution $q_\phi(z \mid x)$ in the latent space:

$$q_\phi(z|x) = N\left(z; \mu(x), \sigma^2(x)\right) \tag{3}$$

In path planning, the encoder generates a distribution in the potential space from the input environmental data – that is, from barriers, start point, end point (Ding, 2022). We can get a latent variable $z$ and map it back to the data space via the decoder $p_\theta(x \mid z)$ by sampling from the potential space, hence creating a fresh path.

The VAE reduces the KL dispersion so that the distribution in the latent space approaches the standard normal distribution $N(0, 1)$:

$$\mathrm{KL}\left(q_\phi(z|x) \| p_\theta(z)\right) = -\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log\left(\sigma_j^2\right) - \mu_j^2 - \sigma_j^2\right) \tag{4}$$

where $J$ is the latent space's dimension; $\mu_j$ and $\sigma_j^2$ are the mean and variance of the latent variables.

In path planning, the VAE's generated pathways could have to be further refined to satisfy the robot's kinematic restrictions and smooth the paths. B-spline interpolation allows one to smooth the paths:

$$\hat{p}(t) = \sum_{i=0}^{n} B_i(t) p_i \tag{5}$$

where $\hat{p}(t)$ is the smoothed path; $B_i(t)$ is the B-spline basis function; $p_i$ is the path's control point. This allows the produced path to be polished such that it corresponds with the robot's real mobility capability.

VAE can be combined with other path planning techniques, including Algorithm A*, to guarantee that the path avoids obstacles and is efficient from start to finish. Algorithm A maximises the path's cost by means of a heuristic search:

$$f(n) = g(n) + h(n) \tag{6}$$

The total cost of the node is $f(n)$; the actual cost from the beginning point to the current node is $g(n)$; the projected cost from the current node to the target node is $h(n)$.

Combining the VAE and A* algorithms allows the produced pathways to not only avoid obstacles but also optimise the path quality using heuristic search.

## 2.2 Generative networks

Generative networks are deep learning models that learn their distribution and thereby create data (Harshvardhan et al., 2020). Training a generative model is meant to enable the creation of samples akin to the real data distribution. Generative networks can be applied in path planning activities to provide several paths that satisfy restrictions, so offering the robot more path choices.

Among the most classical generating networks is GAN. The generator and the discriminator comprise its two primary components (Zhou et al., 2023). While the discriminator aims to differentiate between genuine and produced samples, the generator seeks to produce as realistic samples as feasible. Eventually, by use of adversarial training of the generator and discriminator, the generator can produce samples approximating the genuine data distribution.

Usually, a GAN's loss function is described via a gaming process. Generating bogus samples is the aim of generator $G$, hence the discriminator $D$ finds it challenging to tell real from fake samples. Its loss function shows as:

$$L_{\text{GAN}} = E_{x \sim p_{\text{data}}(x)}\left[\log D(x)\right] + E_{z \sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \tag{7}$$

where $D(x)$ is the discerning of the true sample $x$ by the discriminator; $G(z)$ is the sample produced by the generator depending on the input noise $z$.

Generator $G$'s objective is to create samples that might fool the discriminator, so its loss function is:

$$L_G = -E_{z \sim p_z(z)}\left[\log D\left(G(z)\right)\right] \tag{8}$$

Conversely, the discriminator $D$ aims to maximise its capacity to distinguish between real and fake samples, so its loss function is:

$$L_D = -E_{x \sim p_{\text{data}}(x)}\left[\log D(x)\right] - E_{z \sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \tag{9}$$

Wasserstein GAN (WGAN) presents the Wasserstein distance to evaluate the difference between the produced samples and the real samples, therefore addressing the gradient vanishing issue that could arise in conventional GAN. One may formulate WGAN's loss function as follows:

$$L_{\text{WGAN}} = E_{x \sim p_{\text{data}}(x)}\left[D(x)\right] + E_{z \sim p_z(z)}\left[D\left(G(z)\right)\right] \tag{10}$$

WGAN is trained by directly optimising the Wasserstein distance instead of the $\log(1 - D(G(z)))$ form of classic GAN, therefore avoiding the problem of vanishing gradient.

Under specific conditions, particularly when the generators and discriminators have somewhat differing capacities, the training process of GANs may become unstable (Borji, 2019). WGAN-GP (WGAN with gradient penalty) enhances its stability even more by including a gradient penalty term to the discriminator's loss function, therefore stabilising the training process. WGAN-GP's loss function is one which may be stated as:

$$L_{\text{WGAN-GP}} = E_{x \sim p_{\text{data}}(x)}\left[D(x)\right] - E_{z \sim p_z(z)}\left[D\left(G(z)\right)\right] + \lambda E_{\hat{x} \sim p_{\hat{x}}(x)}\left[\left(\left\|\nabla_{\hat{x}} D(\hat{x})\right\|_2 - 1\right)^2\right] \tag{11}$$

The gradient penalty term aids to enhance the training stability of the model by $\hat{x}$, the linearly interpolated samples of the generator and the discriminator, and $\lambda$, a hyperparameter regulating the gradient penalty strength.

An expansion of conventional GAN, conditional generative adversarial networks (cGAN) use conditional information to direct generation. The generator generates compliant samples by means of the conditional information y, which serves as an input to the discriminator as well. cGAN have as its objective function:

$$L_{\text{cGAN}} = E_{x,y \sim p_{\text{data}}(x,y)}\left[\log D(x, y)\right] + E_{z \sim p_z(z)}\left[\log\left(1 - D\left(G(z, y)\right)\right)\right] \tag{12}$$

In path planning, the condition variable $y$ can represent the destination position of the robot, the distribution of obstacles, etc., thereby allowing the produced path to fit particular limitations.

Several optimisation strategies can help to raise the generator's performance even more. For instance, the generator $G$ minimises the difference between the produced path and the actual path, therefore optimising path planning. One may represent the generator's optimisation goal as:

$$L_G^{\text{opt}} = \min_G E_{z \sim p_z(z)}\left[\left\|G(z) - x_{\text{target}}\right\|_2\right] \tag{13}$$

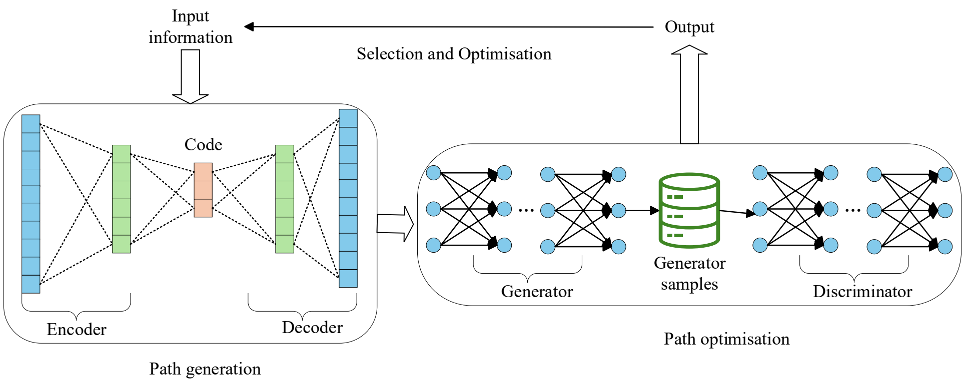
where $\| \bullet \|_2$ is the Euclidean distance, which shows the variation between the produced path and the target path; $x_{\text{target}}$ is the ideal or path of destination.

In this sense, the generative network can not only create paths that satisfy the restrictions but also constantly optimise the generating process to raise the quality and efficiency of path planning.

## 3 Path planning models for accessible robots based on VAE and GAN

This chapter proposes a path planning model VAE-GAN PathNet based on VAE and GAN combined approach. The model creates path candidates in the potential space by VAE and optimises the quality of the paths by GAN to guarantee that the paths satisfy the criteria bars such obstacle avoidance, smoothness and efficiency, see Figure 1.

**Figure 1** VAE-GAN PathNet model (see online version for colours)




## *3.1 Path generation*

This model starts with path generation, in which the VAE learns the latent representation of the environmental input to provide path candidates. First the encoder generates potential variables by mapping the input environment information $x$ to the potential space $z$. One can articulate the process by means of the following equation:

$$q(z|x) = N(\mu, \sigma^2) \tag{14}$$

where $N$ symbolises the normal distribution; $\mu$ and $\sigma^2$ are the mean and variance of the latent distribution learnt from the environmental data. We derive the latent space path representation with this encoder. The decoder then generates path candidates from the latent variable $z$ and the environmental information $y$:

$$q(x|z) = N(\hat{\mu}, \hat{\sigma}^2) \tag{15}$$

## *3.2 Path optimisation*

GAN helps to optimise the path candidates produced by the VAE in the path optimisation phase. Using $z$ and the environment information $y$, the generator $G$ produces the path candidate $x$. The discriminator $D$'s job is to assess the path's quality, therefore determining if it is a real path or not, and take path obstacle avoidance performance into account. The generator's goal in optimisation is as follows:

$$L_G = -E_{z,y}\left[\log D(G(z, y))\right] + \lambda_1 \cdot O(G(z, y)) \tag{16}$$

The obstacle avoidance cost of the path is denoted by $O(G(z, y))$; $\lambda_1$ is a hyperparameter controlling the obstacle avoidance cost. By means of adversarial training, the generator aims to create paths as 'real' as feasible and guarantee that they are safely apart from the environmental impediments.

The discriminator aims to analyse the feasibility of the produced paths and separate them from the actual ones. The discriminator's objective function for optimisation follows:

$$L_D = -E_{x,y}\left[\log D(z, y)\right] - E_{z,y}\left[\log\left(1 - D\left(G(z, y)\right)\right)\right] + \lambda_2 \cdot O\left(G(z, y)\right) \tag{17}$$

where the hyperparameter $\lambda_2$ controls obstacle avoidance's cost. By maximising this loss function, the discriminator detects the truthfulness of the path and guarantees that the produced path does not run across any obstacles.

### 3.3 Path selection with cost function

A cost function helps to assess the produced path candidates. Combining various elements, the path cost function considers path length, obstacle avoidance performance, and smoothness. One finds the path, $C(x)$,'s overall cost as follows:

$$C(x) = \alpha \cdot L(x) + \beta \cdot O(x) + \gamma \cdot R(x) \tag{18}$$

where $L(x)$ is the path's length; $O(x)$ is the cost of obstacle avoidance; $R(x)$ is the path's smoothness; $\alpha$, $\beta$, and $\gamma$ are hyperparameters adjusting the weight of every cost in the chosen path.

### 3.4 Optimisation of generated paths

We create an objective function combining the smoothness, obstacle avoidance, and efficiency of the pathways in the path optimisation process thereby enhancing the quality of the produced paths. The generator's goal is objective.

$$L_G = -E_{z,y}\left[\log D\left(G(z, y)\right)\right] + \lambda_1 \cdot O\left(G(z, y)\right) + \lambda_2 \cdot R\left(G(z, y)\right) \tag{19}$$

where respectively $\lambda_1$ and $\lambda_2$ are hyperparameters controlling the cost of path smoothness and obstacle avoidance respectively.

By means of adversarial training of GAN, the generator continuously optimises the produced pathways to progressively match the distribution of the real paths, so optimising the efficiency of the paths and so fulfilling path smoothness and obstacle avoidance.

### 3.5 Path assessment and selection

Four evaluation criteria help us to assess the quality of the path in the path evaluation and selection process. First, by adding the distances between the points on the path, one may get the path's path length $L_{\text{path}}$, therefore indicating its efficiency:

$$L_{\text{path}} = \sum_{i=1}^{N-1} \left\| p_i - p_{i+1} \right\| \tag{20}$$

where correspondingly $p_i$ and $p_{i+1}$ are two adjacent places on the path.

Avoidance of obstacles: cost $O_{\text{path}}$ makes sure the path does not go across an obstacle by computing the distance separating it from one. It comes from the formula:

$$O_{\text{path}} = \sum_{i=1}^{N-1} \max\left(0, \delta - d\left(p_i, O\right)\right) \tag{21}$$

Given $\delta$ as the minimum safe distance and $d(p_i, O)$ as the path point pi's distance from the obstacle $O$.

Smoothness of paths: the curvature of the path is determined using $k_{\text{path}}$, therefore guaranteeing path continuity and smooth turn alignment:

$$k_{\text{path}} = \frac{d^2 x}{ds^2} \tag{22}$$

where $x$ is the path coordinate; $s$ is the path's arc length. The smoothness of the path can be quantified by computing its curvature.

At last, the real-time performance of path planning – especially the real-time responsiveness in dynamic environments – is evaluated using the computation time $T_{\text{compute}}$.

These assessment criteria enable us to evaluate the produced paths in several dimensions, including the quality, safety, and computing efficiency of the paths, so covering all angles.

We present a new path planning paradigm for accessible robots by merging VAE with GAN, whereby VAE creates path candidates in the latent space and GAN optimises the paths to guarantee that they are optimal in terms of obstacle avoidance, safety, smoothness, and efficiency. Using several evaluation criteria helps us to fully assess the quality of the paths and finally choose the best one. This suggested approach offers fresh concepts and methods for easily available robot navigation in dynamic surroundings.

## 4 Experimental results and analyses

### *4.1 Datasets*

Two actual path planning datasets are chosen in this work to validate the VAE-GAN PathNet model. These datasets guarantee the generalising and adaptability of the model by covering various environmental complexity and path planning activities, so insuring its capacity. As shown in Table 1, the two datasets we selected – Stanford Drone Dataset and robot operating system (ROS) path planning data – are appropriate for path planning tasks in urban and simulated environments, respectively, and can thus help the training and testing of the models in this study rather effectively.

**Table 1** Dataset overview for path planning tasks

| *Dataset name* | *Description* | *Contents* | *Features* |
|---|---|---|---|
| Stanford Drone Dataset | This dataset contains a variety of complex urban environment scenarios, suitable for mobile robot and drone path planning tasks. | Environment maps, start points, target points, obstacle information, multiple environment types | Urban environment, dynamic obstacles, diverse scenes |
| Robot operating system (ROS) path planning data | The ROS dataset provides different types of path planning tasks, suitable for path generation and optimisation across various robotic platforms. | Environment maps, start points, target points, static obstacles, dynamic obstacles | Simulated environment, path generation and optimisation, support for various robotic platforms |

We preprocessed the data such that the VAE-GAN PathNet model could satisfy its input needs. First, all environment maps' obstacle and path data were normalised to guarantee that the path planning task is taught and tested at the same scale. Second, we performed data augmentation on the environment maps, including random rotations, translations, and scaling operations, so as to create a greater number of training samples and improve the model's capacity to generalise to unknown environments, so strengthening the robustness of the model, especially its adaptability in the face of diverse environments. At last, we split every dataset in a 7:3 ratio training set and a test set. Model training proceeds from the training set; final evaluation and comparison analysis come from the test set.

## *4.2 Comparison experiments*

We verify the relative benefits of VAE-GAN PathNet with numerous classical path planning methods in a comparison exercise. For comparison we selected Algorithm A, Dijkstra's algorithm, and deep learning based path planning models (DQN and DNN). Every experiment was carried out using Stanford Drone Dataset and ROS Path Planning Dataset; hence, the testing environment and assessment criteria were kept constant to guarantee comparability and fairness.

Figures 2 and 3 respectively exhibit the experimental outcomes.

**Figure 2** Experimental results on Stanford Drone Dataset (see online version for colours)

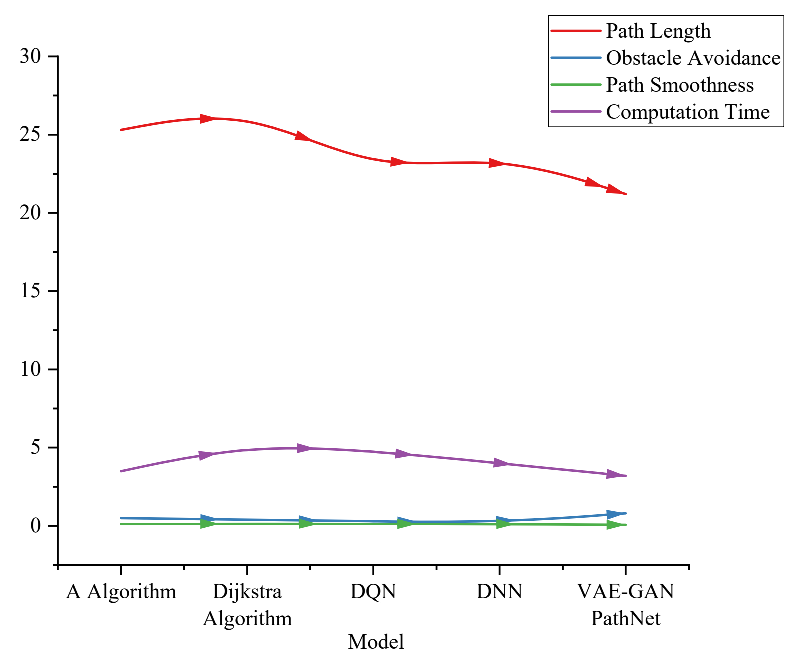
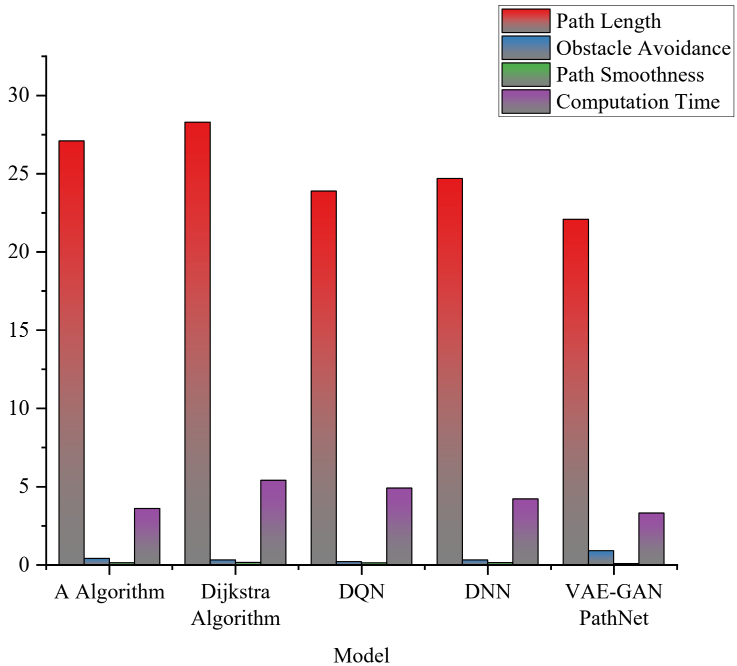


**Figure 3** Experimental results on ROS Path Planning Dataset (see online version for colours)




With a generated path length of 21.2 metres, which is considerably lower than A (25.3 meters) and Dijkstra (26.8 metres), VAE-GAN PathNet beats the other algorithms in terms of path length and obstacle avoidance performance in the experiments on Stanford Drone Dataset. < With the shortest distance between the path and the obstacle – 0.8 metres – which exceeds the other algorithms – it also boasts the best obstacle avoidance performance. VAE-GAN PathNet shows smoother pathways by having a curvature of 0.07, much lower than that of A (0.12) and Dijkstra (0.14). VAE-GAN PathNet runs 3.2 seconds, the shortest of all the algorithms in terms of computing time.

With a path length of 22.1 metres – shorter than A (27.1 metres) and Dijkstra (28.3 metres) – VAE-GAN PathNet also shines on the ROS Path Planning Dataset. Additionally leading with a shortest distance of 0.9 m, obstacle avoidance surpasses the other versions. VAE-GAN PathNet shows better path smoothness using a curvature of 0.08. With 3.3 seconds for computing, which is also favorable in terms of efficiency.

On both datasets, VAE-GAN PathNet beats the conventional A and Dijkstra techniques generally, particularly in terms of path quality, obstacle avoidance performance and computational economy. Furthermore proving its complete benefits in challenging settings, VAE-GAN PathNet beats deep learning-based path planning models DQN and DNN.

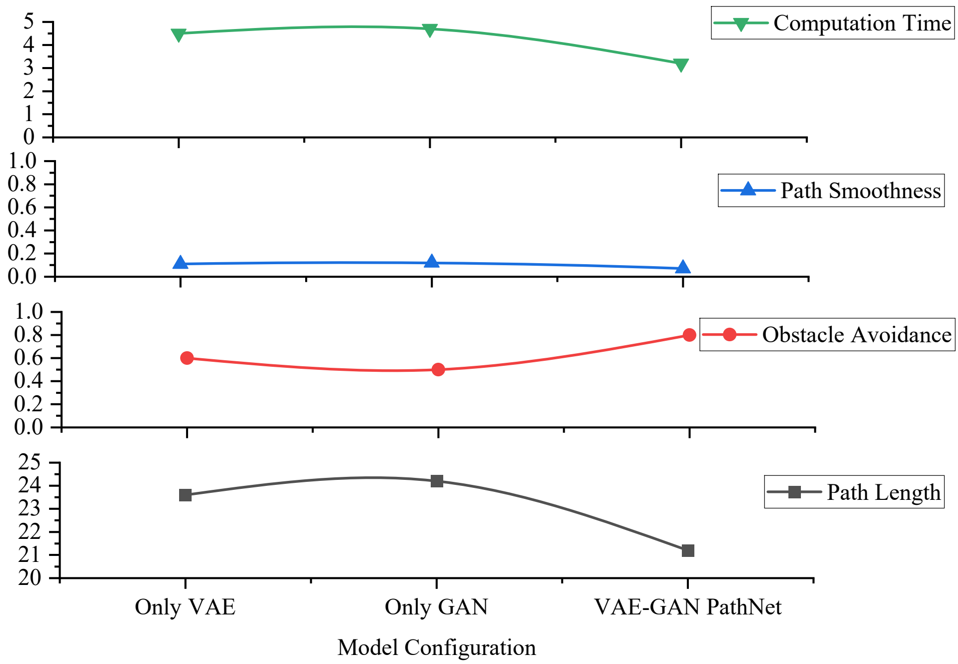
## *4.3 Ablation experiments*

We investigate the function of VAE and GAN modules in the VAE-GAN PathNet by deleting them independently in the ablation experiments. We devised three experimental setups: VAE module only; path generation and optimisation using variational autocoders;

GAN module only; path generation with cGANs; and the complete VAE-GAN PathNet model, so combining the strengths of variational autocoders and cGANs. We can understand the influence of every module on the path planning performance by means of comparison of the experimental findings under several setups.

Figures 4 and 5 respectively exhibit the outcomes for the two datasets.

**Figure 4** Ablation experiment results on Stanford Drone Dataset (see online version for colours)

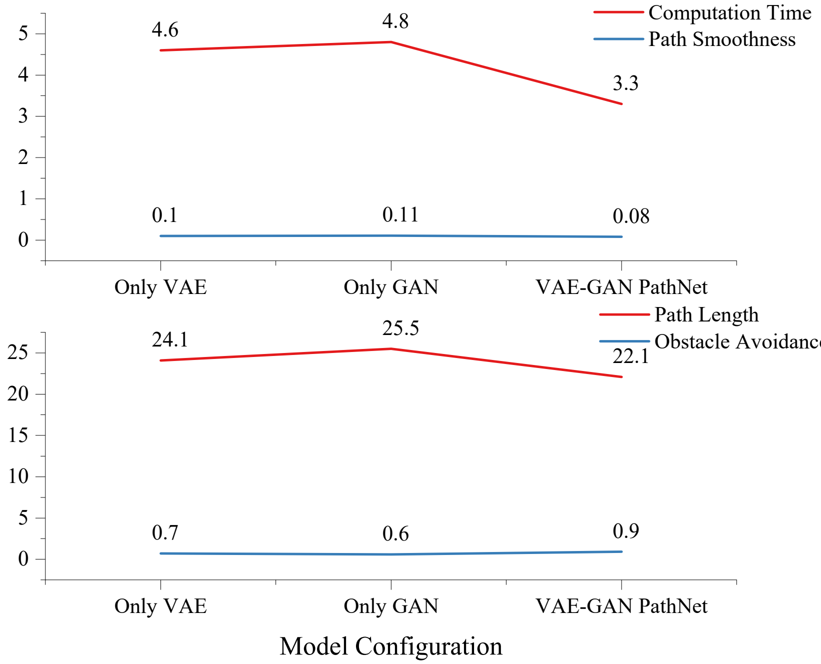



Using VAE and GAN alone, respectively, the experimental results on the Stanford Drone Dataset show path lengths of 23.6 m and 24.2 m respectively, both longer than the 21.2 m VAE-GAN PathNet. VAE-GAN PathNet has a shortest obstacle distance of 0.8 m, which is much more than VAE (0.6 m) and GAN (0.5 m). This relates to obstacle avoidance performance. With VAE-GAN PathNet's 0.07 curvature, which exceeds VAE and GAN employed alone, the path smoothing performance is likewise better.

With a path length of 22.1 metres – much less than the VAE (24.1 metres) and GAN (25.5 metres) – the VAE-GAN PathNet likewise performs effectively on the ROS Path Planning Dataset. Better than the other two configurations, the VAE-GAN PathNet has a shortest obstacle distance of 0.9 m in terms of obstacle avoidance performance. Furthermore improving in terms of path smoothness and calculation time is the VAE-GAN PathNet.

By use of VAE-GAN PathNet, which combines VAE and GAN modules in the path planning task, we can observe that it greatly enhances the path quality, obstacle avoidance effect, and computing efficiency above mentioned.

**Figure 5** Ablation experiment results on ROS Path Planning Dataset (see online version for colours)




## 5 Conclusions

Aiming to solve the path optimisation and obstacle avoidance challenges in robot path planning in complicated environments, this work proposes a combined VAE and GAN-based path planning model VAE-GAN PathNet for barrier-free robots. VAE-GAN PathNet shows the complete benefits in path planning by including VAE for path generating and combining with GAN to increase the authenticity and feasibility of paths.

pecially in the detection of small targets and complex backgrounds, demonstrating significant advantages.

Theoretically, the ideas of VAE and GAN approaches and their synergy to maximise the path planning problem are discussed in great length. After that, the VAE-GAN PathNet model is proposed and its developments in path generation, obstacle avoidance and path smoothness are shown. The efficacy of the model is validated by experiments employing the ROS Path Planning Dataset and the Stanford Drone Dataset. In terms of path length, obstacle avoidance performance, path smoothness and computing time, VAE-GAN PathNet greatly beats conventional path planning algorithms in the comparison studies. Furthermore proving the performance improvement of the merger of VAE and GAN modules are the ablation studies.

In path planning activities in complicated environments, VAE-GAN PathNet shows overall great efficiency and performance, therefore confirming the possibilities of VAE and GAN fusion technology.

Particularly when the quality of path generating varies or the training process is challenging to converge, there is a certain stability issue in the VAE-GAN PathNet model. Dealing with complex or irregular settings makes this instability especially clear and compromises the accuracy and dependability of the produced pathways. Future research can bring more advanced training strategies and simultaneously combine the Reinforcement Learning (RL) framework to optimise the adversarial training process, so strengthening the resilience of the model and improving the stability and quality of path generating.

Currently, the model is too dependent on a lot of labelled data, particularly in the VAE module's training phase where high quality path labelled data is needed. This makes the model less able to be successful in settings with limited data. Unsupervised or semi-supervised learning methods could be investigated in the future to solve this problem so enabling the model to be efficiently trained using unlabelled data, so lowering the dependency on hand labelling.

VAE-GAN PathNet has complex model structure and substantial computational overhead during training, which causes efficiency issues in real-time path planning even if it has outstanding performance in path generating accuracy. Model compression, pruning methods, or knowledge distillation helps to lower the computational complexity of the model thereby enhancing the inference speed and hence addressing this restriction. Furthermore, the combination of hardware acceleration (e.g., GPU/TPU) and parallel computing methods can sufficiently enhance the real-time reaction capacity of the model, thereby enabling path planning in a shorter amount of time to satisfy demand for effective computing.

Although VAE-GAN PathNet shows good performance on the present experimental dataset, its generalisation capacity still has to be strengthened. The model might suffer from overfitting in more complicated and varied settings, which would result in unsteady performance in certain surroundings. Future studies should thus enhance the generalising capacity of the model by increasing the dataset and applying more extensive and sophisticated environmental data for training. Furthermore, the combination of migration learning and multi-task learning approaches can improve the flexibility of the model so that it can offer dependable and effective path planning solutions in several application situations.

Particularly in path development and obstacle avoidance in complicated environments, the VAE-GAN PathNet model suggested in this work shows overall great performance in path planning tasks. By means of the efficient merging of VAE and GAN, the model is able to lower the calculation time and enhance the planning accuracy while guaranteeing the path smoothness and obstacle avoidance. Future studies will thus maximise the stability and computing efficiency of the model and investigate a greater spectrum of application scenarios, so addressing some of the still existing constraints.

## Declarations

All authors declare that they have no conflicts of interest.

## References


Biswas, A., Vasudevan, R., Ziatdinov, M. et al. (2023) 'Optimizing training trajectories in variational autoencoders via latent Bayesian optimization approach', *Machine Learning: Science and Technology*, Vol. 4 No. 1, p.015011.

Borji, A. (2019) 'Pros and cons of GAN evaluation measures', *Computer Vision and Image Understanding*, Vol. 179, pp.41–65.

Ding, M. (2022) 'The road from MLE to EM to VAE: A brief tutorial', *AI Open*, Vol. 3, pp.29–34.

Han, K. and Wang, Y. (2021) 'A review of artificial neural network techniques for environmental issues prediction', *Journal of Thermal Analysis and Calorimetry*, Vol. 145 No. 4, pp.2191–2207.

Harshvardhan, G., Gourisaria, M.K., Pandey, M. et al. (2020) 'A comprehensive survey and analysis of generative models in machine learning', *Computer Science Review*, Vol. 38, p.100285.

Karur, K., Sharma, N., Dharmatti, C. et al. (2021) 'A survey of path planning algorithms for mobile robots', *Vehicles*, Vol. 3 No. 3, pp.448–468.

Shahriar, S. (2022) 'GAN computers generate arts? A survey on visual arts, music, and literary text generation using generative adversarial network', *Displays*, Vol. 73, p.102237.

Soori, M., Arezoo, B. and Dastres, R. (2023) 'Artificial intelligence, machine learning and deep learning in advanced robotics, a review', *Cognitive Robotics*, Vol. 3, pp.54–70.

Vulpi, F., Milella, A., Marani, R. et al. (2021) 'Recurrent and convolutional neural networks for deep terrain classification by autonomous robots', *Journal of Terramechanics*, Vol. 96, pp.119–131.

Willms, A.R. and Yang, S.X. (2006) 'An efficient dynamic system for real-time robot-path planning', *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part B* (*Cybernetics*), Vol. 36, No. 4, pp.755–766.

Yao, R. and Bekhor, S. (2022) 'A variational autoencoder approach for choice set generation and implicit perception of alternatives in choice modeling', *Transportation Research Part B*: *Methodological*, Vol. 158, pp.273–294.

Yiu, Y.F., Du, J. and Mahapatra, R. (2019) 'Evolutionary heuristic A* search: Pathfinding algorithm with self-designed and optimized heuristic function', *International Journal of Semantic Computing*, Vol. 13, No. 1, pp.5–23.

Zalman, D. and Fine, S. (2023) 'Variational inference via Rényi bound optimization and multiple-source adaptation', *Entropy*, Vol. 25, No. 10, p.1468.

Zhang, H., Jin, H., Liu, Z. et al. (2019) 'Real-time kinematic control for redundant manipulators in a time-varying environment: multiple-dynamic obstacle avoidance and fast tracking of a moving object', *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 1, pp.28–41.

Zhou, T., Li, Q., Lu, H. et al. (2023) 'GAN review: models and medical image fusion applications', *Information Fusion*, Vol. 91, pp.134–148.