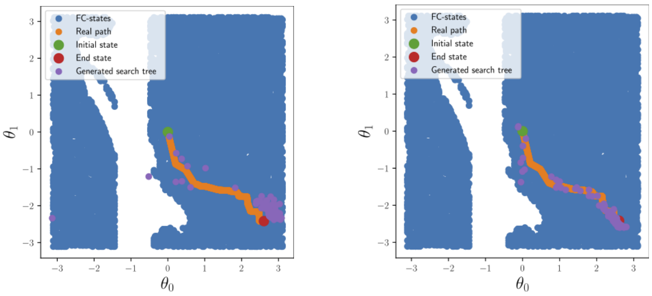
2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Learning Better Paths: Multimodal Generative Models Enhanced with Local Critics

Jorge Ocampo Jimenez and Wael Suleiman
Electrical and Computer Engineering Department, Université de Sherbrooke, Quebec, Canada
Emails: {Jorge.Ocampo-Jimenez; Wael.Suleiman}@USherbrooke.ca

***Abstract*—This work proposes a novel framework to accelerate motion planning in previously unseen environments with obstacles by modeling the distribution of the collision-free configuration space using Wasserstein Generative Adversarial Networks with Gradient Penalty (WGAN-GP). To effectively manage multimodal data, we condition the WGAN-GP using a Variational Autoencoder (VAE) embedded in a continuous latent space. The configuration space is approximated through a set of Gaussian distributions, allowing the dataset to be segmented into multiple localized models. This strategy not only enhances the learning efficiency but also reduces convergence time. We utilize the reconstructed configuration space to evaluate motion planning performance in previously unseen scenarios. Experimental results highlight the potential of our approach to significantly accelerate planning in unknown environments while maintaining the generation of near-optimal trajectories.**

***Index Terms*—Sampling-based path planning, Generative Adversarial Networks, Image-conditioned generative model, Variational Autoencoders**



(a) Approximation of the free-search tree of a path with WGAN-GP and a VAE for an unseen scenario

(b) Approximation of the free-search tree of a path with MultiWGAN-GP for an unseen scenario space

Fig. 1: Search tree for a path planning task of unseen scenarios.

I. Introduction

Sampling-based motion planning algorithms, such as the Rapidly-exploring Random Tree (RRT) [1] and its optimal variant RRT [2], are widely used to find collision-free trajectories for robots. Despite their generality and simplicity, these algorithms can be inefficient because they rely on unguided random exploration of the configuration space ($\mathcal{C}$s). This inefficiency becomes especially pronounced in high-dimensional or analytically intractable spaces, and in practical scenarios involving complex constraints or partially unknown environments. Consequently, there is a growing need for planning strategies that offer both greater speed and reliability.

To address these limitations, researchers have turned to learning-based methods to guide the sampling process. In particular, learning from demonstration has proven effective for improving sampling-based planners [3]. A common strategy is to train a generative model that maps the environment (e.g., an image or point cloud of the scene) to a distribution of promising states in the robot's configuration space, thereby biasing the planner toward fruitful regions with far fewer samples than a uniform random strategy would typically require. Modern deep neural networks are well suited to this task, as they can leverage large volumes of sensor data and capture complex relationships between the robot's possible configurations and the locations of obstacles in the workspace.

Generative models have been incorporated into RRT-based planners in two primary ways: (i) to bias the sampler towards more promising regions, and (ii) as a heuristic to guide the path cost evaluation. By conditioning on the scenario (such as the start and goal states and the obstacle layout), a learned model can guide the algorithm toward lower-cost, collision-free regions of the state space. One of the first works in this direction trained a conditional Variational Autoencoder (VAE) to learn a non-uniform sampling distribution from demonstrations [4]. Given the initial state, goal state, and positions of obstacles, this VAE model identified promising areas of the state space to sample, significantly improving planning efficiency by biasing the expansion of the RRT. Similarly, Qureshi et al. [5] encoded raw sensor data (or a voxelized 3D map of the environment) into a latent space, which a planning network then used—along with the current state and goal—to predict the next state. This approach effectively biases the sampling of an RRT planner [2], enabling motion planning to scale better in high-dimensional spaces.

Apart from learning sampling distributions, demonstration data can also be used to learn cost functions for planning. For example, inverse reinforcement learning has been applied to adjust the cost weights of RRT based on human-

978-1-6654-5784-2/26/$31.00 ©2026 IEEE

demonstrated trajectories [6], thus guiding the planner along more "natural" paths in environments previously navigated by humans. However, such cost-shaping approaches are less effective in dynamic environments, since the learned weights cannot be easily altered on the fly without compromising the asymptotic optimality of the planner.

Encouraged by the success of Generative Adversarial Networks (GANs) in modeling complex data distributions (e.g., for image synthesis), researchers have also applied GANs to motion planning. Several works have explored GAN-based approaches to bias the sampling process. For instance, one study employs a conditional GAN on 2D occupancy grid maps (with the start and goal locations encoded) and uses dual discriminators to enforce obstacle avoidance and path connectivity, achieving roughly a 90% success rate in generating a feasible path [7]. Another approach by Li et al. [8] treats path planning as an image-to-image translation problem: they embed an encoder–decoder network within a GAN generator that takes the initial state, an environment map, and a latent code as input and outputs a 2D representation of a path. This method was shown to require fewer iterations to find a low-cost path compared to standard RRT. In a different vein, Lembono et al. [9] trained a GAN to learn inverse kinematics, generating valid high-dimensional robot configurations for a given end-effector target (without explicitly conditioning on obstacle information).

Diffusion probabilistic models have also been explored for motion planning [10]. While diffusion models can produce high-quality results, they require many iterative refinement steps, making them impractically slow for on-line planning. Thus, despite these innovations, capturing the full multimodal distribution of collision-free states conditioned on complex sensory inputs (for instance, mapping raw RGB-D sensor images to feasible configurations) remains an open challenge.

To address this gap, we introduced in [11] a hybrid generative approach that combines a Variational Autoencoder with a Wasserstein GAN with Gradient Penalty (WGAN-GP) to learn the distribution of collision-free configurations conditioned on a given environment. Training such a VAE-conditioned WGAN-GP is non-trivial: naïvely combining them can lead the VAE's Kullback–Leibler term to converge toward an uninformative latent encoding. In [11], we alleviated this issue by reformulating the configuration space representation as a mixture of Gaussians and partitioning the training data into local clusters, each modeled by a dedicated sub-network. This hierarchical strategy improved training stability and convergence speed. The resulting model effectively approximated the feasible configuration space in previously unseen, obstacle-rich scenes.

Building on this foundation, the present work aims to further accelerate motion planning in unseen scenarios by leveraging such learned generative models. The main contribution of this paper is to demonstrate the feasibility of training and integrating a WGAN-GP–based sampler within a sampling-based planner. Unlike traditional planners that rely on a fixed ratio of uniform random samples to goal-biased samples, our approach directly adjusts the learned generative model—specifically by perturbing its encoder's latent input—to preferentially generate collision-free samples. This strategy biases the search toward valid regions of the state space without requiring any fallback to an alternative sampling distribution, thereby improving planning efficiency.

II. Overview of WGAN-GP–based Sampler

Wasserstein GAN model (WGAN) was proposed in [12] which employs the Earth-Mover distance to measure the similarity between p_g and p_r, which represent respectively the distributions over the multidimensional real data $\boldsymbol{x}$ and the noise input vector $\boldsymbol{z}$. This approach offers the benefit of providing smooth measures even in scenarios where the distributions are completely overlapping or disjoint.

In [11], instead of conditioning the GAN directly on the input image, we used a VAE [14]. This autoencoder is utilized to map the input image of the current scenario into a latent vector $\boldsymbol{z}$. By transforming the image space into a parametrized multivariate normal distribution with vector mean $\mathbf{0}$ and vector standard deviation $\mathbf{1}$, the autoencoder aims to represent the image as closely as possible to the Gaussian distribution while preserving enough information to reconstruct the image using the latent vector $\boldsymbol{z}$. This enables us to generate new obstacle configurations that interpolate between data points from the training data during inference. An overview of our WGAN-GP-based sampler is given in Fig. 2.

III. Motion Planning Problem Formulation

Mathematically, the motion planning problem is defined over a state space $\mathcal{C} = [0,1]^d$, where $d \in \mathbb{N}, d \geq 2$. The subset $\mathcal{C}_{obs} \subset \mathcal{C}$ denotes the set of obstacle regions corresponding to configurations that lead to collisions. The collision-free region is then defined as $\mathcal{C}_{free} = \mathcal{C} \backslash \mathcal{C}_{obs}$. Given an initial state $\boldsymbol{x}_0 \in \mathcal{C}_{free}$ and a set of goal states $\mathcal{C}_{goal} \subset \mathcal{C}_{free}$, a solution path is described by a continuous function $s : [0,1] \to \mathbb{R}^d$. The path is said to be collision-free if $s(\tau) \in \mathcal{C}_{free}$ for every $\tau \in [0,1]$. The path is considered feasible if, in addition, it satisfies the boundary conditions $s(0) = \boldsymbol{x}_0$ and $s(1) \in \mathcal{C}_{goal}$.

Identifying a feasible solution in the robot's configuration space is a PSPACE-complete problem [17], which makes it computationally intensive, especially in high-dimensional or complex environments. To overcome this, researchers have developed sampling-based motion planning algorithms, which approximate the solution by drawing random samples in the configuration space and connecting them to form potential paths to the goal. These algorithms guarantee probabilistic completeness, meaning they are capable of finding a valid solution with enough samples. Furthermore, asymptotic optimality—the convergence of the path cost

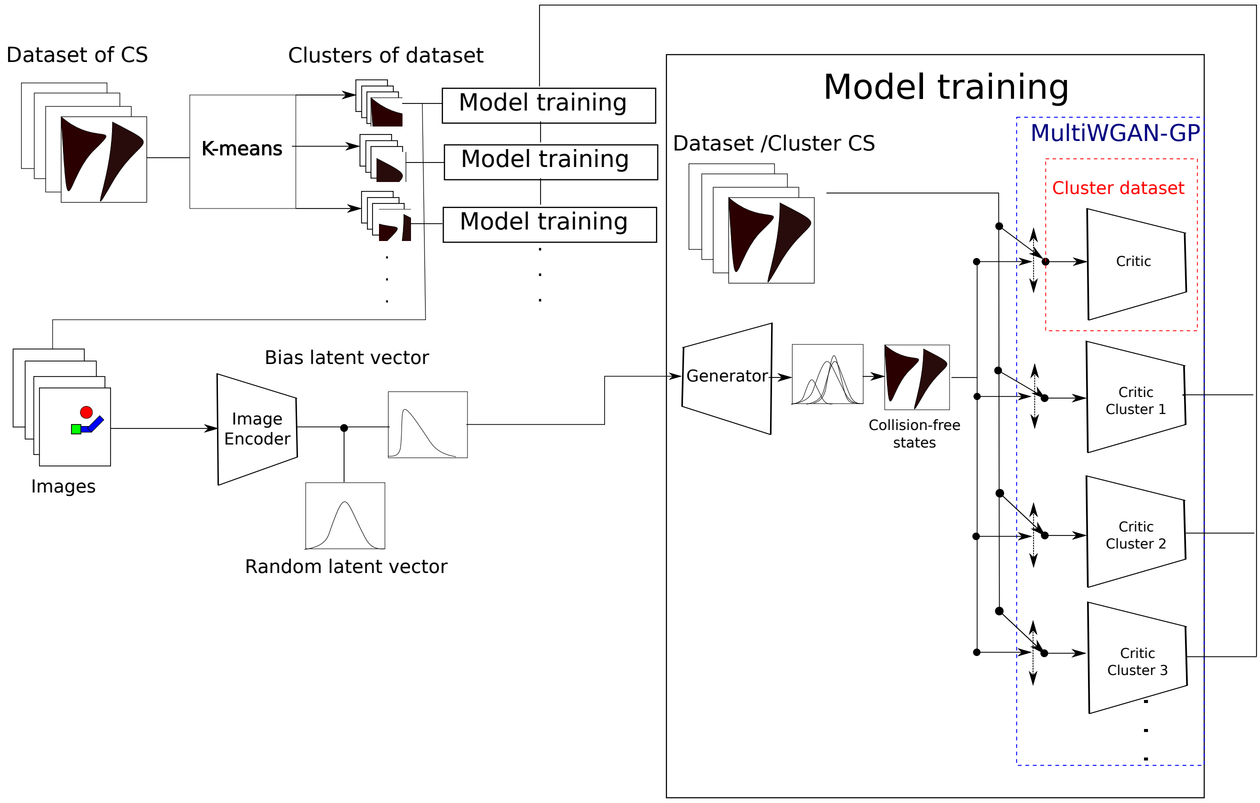


Fig. 2: Overview of WGAN-GP-based sampler architecture: the model generates the Gaussians of different regions of the CS of the robot. RGB images of the obstacles are used as input to bias the latent vector to the correct region of the $\mathcal{C}$.

toward the optimal value—can be achieved by carefully structuring the graph connections during planning [4].

To enhance the performance of sampling-based motion planning algorithms, several strategies have been introduced to steer the sampling process in the direction of the goal. A common approach involves learning a probabilistic model over $\mathcal{C}_{free}$, conditioned on the robot's environment. This learned distribution allows the planner to prioritize areas of the CS that are more likely to contain feasible paths to the target.

By concentrating sampling efforts on these promising regions, the planner reduces the time wasted on irrelevant areas of the $\mathcal{C}$. This targeted exploration significantly accelerates the overall planning procedure and improves the practicality of sampling-based algorithms for deployment in real-world robotic systems.

IV. Methodology

We introduce a novel method to accelerate sampling-based motion planning by producing samples within $\mathcal{C}_{free}$ that not only lie in the collision-free space but also exhibit desirable characteristics such as feasibility and connectivity to the current trajectory. Instead of relying on uniform sampling, our technique leverages a Generative Adversarial Network (GAN) trained to model a distribution over $\mathcal{C}_{free}$ that emphasizes areas of the configuration space likely to yield valid paths. More precisely, we adopt a Wasserstein GAN with Gradient Penalty (WGAN-GP) to synthesize high-quality, collision-free (CF)-states—eliminating the need for manual clipping of discriminator outputs. This targeted sampling approach effectively replaces uniform random sampling and results in reduced query times during planning.

To guide the path towards the desired region in the free configuration space, we utilized a technique inspired by [18]. Our approach involved using the learned generator as a sampler for RRT path planning algorithm. However, we made a modification in our implementation to specifically handle situations where our encoder failed to capture the complete encoding of a previously observed scenario. To this end, we increased the value of $\boldsymbol{\sigma}$ directly from the encoder's output, instead of trying to find a ratio between sampling from the uniform distribution and the generator. This idea is similar to having a forward trajectory of $k \in \mathbb{N}$ diffusion steps from the predicted distribution $f_w(\cdot)$ with fixed mean $\boldsymbol{\mu} \neq \mathbf{0}$ [19].

Our objective in taking this approach was to cover a more diverse set of latent vectors, thereby helping the generator to incorporate images that exhibit fewer similar features. This method can prove useful in situations where the encoder's trained query is unable to find any CF states within a

Algorithm 1: RRT with MultiWGAN-GP

Data: $x_{init}, x_{goal}, \boldsymbol{y} \subset I : U \rightarrow [-1,1]^{t\times 3\times 48\times 48}, t \in \mathbb{N}$ is the number of points to sample from MultiWGAN-GP with $t \geq n$ iterations,
$\boldsymbol{\epsilon_\sigma} = [0, \epsilon_1, ..., \epsilon_k], k \in \mathbb{N}, \epsilon_i \in \mathbb{R}^+$ and $\epsilon_i > \epsilon_{i-1}$ is the amount of the k perturbation of $\boldsymbol{\sigma}$.

Result: G

$V \leftarrow x_{init}, E \leftarrow \emptyset$;
$\boldsymbol{\mu}, \boldsymbol{\sigma} \leftarrow q(\boldsymbol{z}|\boldsymbol{y})$;
$BiasSampler = \cup_{i=0}^{k} f_w(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma} + \boldsymbol{\epsilon_\sigma}[\boldsymbol{i}]))$;
for $i = 1, ..., n$ **do**
  $x_{rand} \leftarrow BiasSampler[i]$;
  $x_{nearest} \leftarrow Nearest(G = (V, E), x_{rand})$;
  $x_{new} \leftarrow Steer(x_{nearest}, x_{rand})$;
  if $ObstacleFree(x_{nearest}, x_{new})$ **then**
    $V \leftarrow V \cup \{x_{new}\}$;
    $E \leftarrow E \cup \{(x_{nearest}, x_{new})\}$;
return $G = (V, E)$;

predetermined number of generator samples. By employing this method, it becomes possible to identify CF points, even in scenarios where no examples closely resemble the trained data by leveraging shared characteristics among multiple training data scenarios; this implementation is reflected in Algorithm 1.

V. Experimental results

In order to provide visual representation of the results, we have designed a setup that incorporates a two-degree-of-freedom manipulator robot. The purpose of this setup is to demonstrate, through graphical means, the effectiveness of our proposed architecture in learning a two-dimensional configuration space that is dependent on the position of an obstacle. Furthermore, we aim to demonstrate how our architecture can be applied to diverse problems that necessitate understanding of the configuration space.

A. Experimental Hardware Configuration and Reproducibility

All the models are trained on a system with 2 x Intel Gold 6148 Skylake, 16 GB of RAM and 2 x NVidia V100SXM2. For deployment, we use Ubuntu 22.04 running on a 3.60 GHz × 8 Intel Core i7-9700K processor, 16GB RAM on NVidia RTX 2070. Our code is openly available[1].

We employ an image to represent the conditioning factor in our experiment. This image encapsulates the obstacle's representation within the robot's operational space. To streamline the experimentation process, we opted for three circles of radius one unit to serve as the obstacle. Additionally, we include the robot's starting position, which is at the initial state of $\mathbf{0}$ before the path is planned.

B. Path planning with extra models

To train the generator for sampling paths in the $\mathbb{S}^1 \times \mathbb{S}^1$ configuration space, we first transformed the data by embedding it into a 4-dimensional Euclidean space, where we estimated the sine and cosine of each angle θ_i. This allowed us to represent each path as a continuous line and eliminate any discontinuities in the data.

To address the variability of the paths in the RRT algorithm, we used RRT* paths as training data, which provided a more stable distribution that did not fluctuate significantly when the configuration space was changed. In our experiments, we allowed the RRT* algorithm 10 seconds to find the path and rewire the tree. We used path length as the minimization objective for the generator training, as it provides a reliable measure of the quality of generated paths.

In the previous section, we generated 10 new different scenario/configuration spaces for testing, and in this section, we estimated paths in each of these spaces. Our objective was to combine the same critics used to learn the $\mathcal{C}$ reconstruction and improve the RRT planner's quality. To generate more data and overcome the curse of dimensionality, we interpolated all the paths.

In our proposed model, we used the same auto-encoder and parameters that were used in the $\mathcal{C}$ learning phase. To evaluate our approach's improvement compared to other methods, we trained two WGAN-GP paths: one using only the auto-encoder and another using the two critics from the previous section. Both models were trained for 600 epochs.

As expected, our proposed approach of incorporating critics to bias the training to the distribution yielded faster discovery of the $\mathcal{C}_{free}$ constrained to the path, compared to training the model with only one critic. This is evident from Fig. 3, where the partitioned critics captured the local properties of the $\mathcal{C}$ in greater detail.

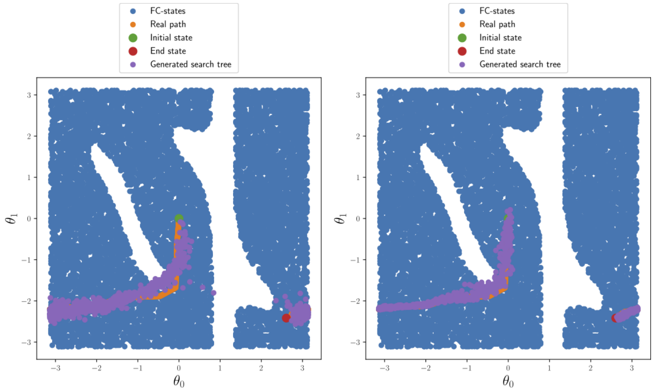
Furthermore, incorporating the extra critics helped to provide more information about the complete $\mathcal{C}$, making it easier to differentiate between CF-states with the image-scenario input, especially in non-previously seen instances. This is exemplified in Fig. 4, where our model with two critics was able to accurately reconstruct the path compared to the model with only one critic.

To implement the trained sampler, we utilized the Open Motion Planning Library (OMPL) [20] implementations of RRT and RRT*. Specifically, for our proposed MultiWGAN-GP path planner, we replaced the standard uniform sampler with our trained sampler to steer the planner towards the next CF-state.

The experimental results demonstrate that our method can generate a feasible CF-path for the given scenario. With more data, it may be able to achieve better path length than RRT* when the planning time is constrained.

Our proposed algorithm was able to find a CF-path in all the new cases in a timely manner compared with the running

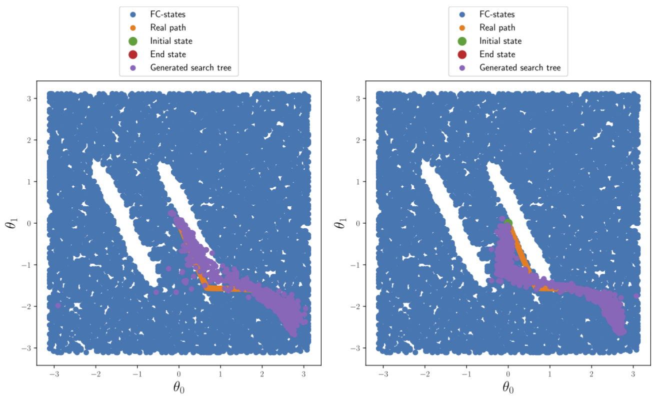
[1] https://bitbucket.org/joro3001/multiwgangp/



(a) Sampling using the model trained with only the critic for the path.

(b) Sampling using the model trained with the critics used to learn the $\mathcal{C}$ and path.

Fig. 3: Testing the sampling from the generators used for sampling.



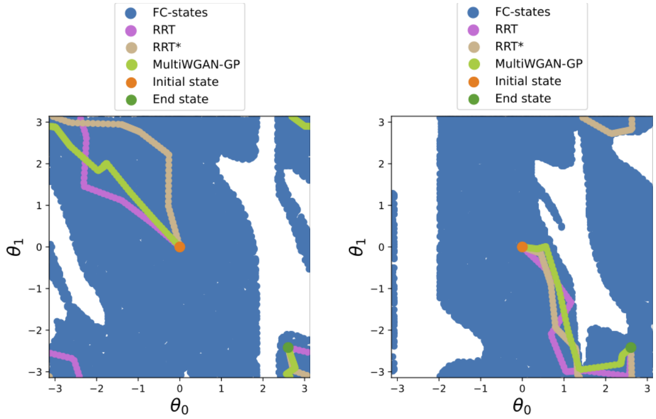
(a) Sampling using the model trained with only the critic for the path.

(b) Sampling using the model trained with the critics used to learn the $\mathcal{C}$ and path.

Fig. 4: Using the GAN model to estimate samples from the path in non-previously seen scenario and $\mathcal{C}$. The extra critics help to capture information from the whole $\mathcal{C}$.

time of RRT. During testing, we found that we needed to sample at least 300 states from the generated path, which takes approximately 0.18 seconds, one example is presented in Fig. 6. Since we used path interpolation as our training data, most of the states generated by the neural network are in close proximity to each other, which slows down the steering of the algorithm towards the closest neighbor in the graph.

To speed up our method, we need to reduce the number of sampling points and ensure that each point is as close as possible to a waypoint that can be accepted by the algorithm. This can be achieved by increasing the amount of training paths to cover a variety of scenarios, allowing the model to learn the minimum number of waypoints needed



(a) Our method produced longest path in this case. There is no sample in the training data that are close to this unseen scenario.

(b) Our method produces the shortest path. There are some samples of the training dataset that are similar to this unseen scenario.

Fig. 5: Extrapolation of the planners on non-previously seen scenarios. Our method is able to find the shortest path in the scenarios that are similar to the trained data.

to reduce the amount of required collision checking during interpolation between the new state and the current state. By improving the model's ability to learn these minimal waypoints, we can reduce the time taken to find a path and increase the efficiency of our proposed method.

There is still room for improvement in terms of the speed of querying the encoder and generator to achieve real-time performance. One approach to achieve this could be reducing the number of operations required by the neural network, for instance, by optimizing the architecture or using more efficient algorithms.

Overall, while our proposed method shows promising results, there is still further work to be done to improve its efficiency and real-time performance. By incorporating the aforementioned improvements, we believe that our algorithm can be further optimized to achieve better results and be applicable to more complex scenarios.

Regarding the success rate, our algorithm was able to generate feasible CF paths each time, even when faced with previously unknown scenarios, whereas RRT and RRT* struggled to find CF waypoints. This is especially important in tasks where low latency and safety are major concerns, such as human-robot interaction.

Although our planning scenarios were generated randomly, we recognize the possibility of our algorithm encountering challenges in generating a CF path within the time constraints of the query. Furthermore, if the dataset exhibits bias towards certain regions, the probability of sampling far from the mean vector ($\boldsymbol{\mu}$) diminishes exponentially, even with a large variance that approximates a uniform distribution. In such cases, it may be necessary to acquire

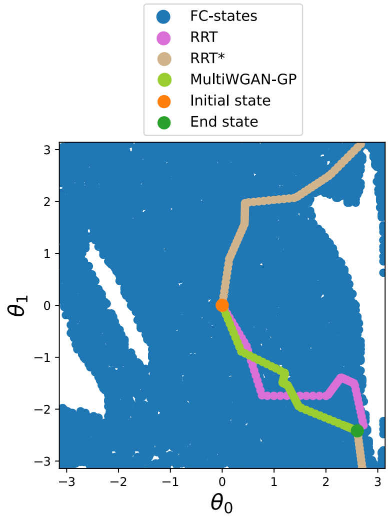


Fig. 6: Extrapolation of the path where our method achieves the same speed as the RRT algorithm, taking into account the overhead of calling both the encoder and generator.

additional training data that encompasses a broader range of scenarios to address these situations effectively.

VI. Conclusion and Future Work

In this work, we have demonstrated the effectiveness of WGAN-GP-based sampler in planning paths in a 4-dimensional space for a 2-dimensional 2-DOF simulated robot. The results of our experiments demonstrate that our proposed model is capable of generating CF-paths in unknown scenarios with improved success rate and reduced running time when compared to conventional path planning algorithms such as RRT and RRT*. However, to establish the broader applicability of our method, it is necessary to extend it to higher dimensional spaces for redundant articulated robots. This is critical to show its usefulness in solving high-dimensional problems for real-world applications and using real robots. Future work will focus on exploring the feasibility of this extension and evaluating the performance of our method on more complex tasks and scenarios.

Other problem to solve is the processing of the input, while we used a 2D representation of the configuration space, we need a way to combine the VAE now with noisy depth information to be able to give enough information to the GAN model to be able to reproduce the configuration space of the robot.

Acknowledgement

This work was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC, grant ALLRP 597914-24) and Fonds de recherche du Québec, Nature et technologies (https://doi.org/10.69777/328238). Additionally, we acknowledge the valuable assistance and resources provided by Calcul Québec and the Digital Research Alliance of Canada.

References

[1] J. Kuffner and S. LaValle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning." in *IEEE International Conference on Robotics and Automation*, vol. 2, 2000, pp. 995–1001.

[2] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning," *International Journal of Robotic Research - IJRR*, vol. 30, pp. 846–894, 06 2011.

[3] J. Wang, T. Li, B. Li, and M. Q.-H. Meng, "GMR-RRT*: Sampling-Based Path Planning Using Gaussian Mixture Regression," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 690–700, 2022.

[4] B. Ichter, J. Harrison, and M. Pavone, "Learning Sampling Distributions for Robot Motion Planning," in *2018 IEEE International Conference on Robotics and Automation*, 2018, pp. 7087–7094.

[5] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, "Motion Planning Networks: Bridging the Gap Between Learning-based and Classical Motion Planners," *IEEE Transactions on Robotics*, pp. 1–9, 2020.

[6] N. Pérez Higueras, F. Caballero, and L. Merino, "Teaching Robot Navigation Behaviors to Optimal RRT Planners," *International Journal of Social Robotics*, vol. 10, 04 2018.

[7] T. Zhang, J. Wang, and M. Q.-H. Meng, "Generative Adversarial Network Based Heuristics for Sampling-Based Path Planning," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. JAS-2021-0110, p. 64, 2022.

[8] Z. Li, J. Wang, and M. Q. Meng, "Efficient Heuristic Generation for Robot Path Planning with Recurrent Generative Model," *2021 IEEE International Conference on Robotics and Automation*, pp. 7386–7392, 2021.

[9] T. S. Lembono, E. Pignat, J. Jankowski, and S. Calinon, "Learning Constrained Distributions of Robot Configurations With Generative Adversarial Network," *IEEE Robotics and Automation Letters,*, vol. 6, no. 2, pp. 4233–4240, 2021.

[10] S. Huang, Z. Wang, P. Li, B. Jia, T. Liu, Y. Zhu, W. Liang, and S.-C. Zhu, "Diffusion-based Generation, Optimization, and Planning in 3D Scenes," *arXiv preprint arXiv:2301.06015*, 2023.

[11] J. O. Jimenez and W. Suleiman, "Improving configuration space reconstruction through multimodal generative models with local critics," in *IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO)*, 2025, pp. 245–252.

[12] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 214–223.

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5769–5779.

[14] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations*, 2014.

[15] L. P. Cinelli, M. A. Marins, E. A. B. da Silva, and S. L. Netto, *Variational Methods for Machine Learning with Applications to Deep Networks*. Springer International Publishing, 2021.

[16] K. Biswas, S. Kumar, S. Banerjee, and A. K. Pandey, "Smooth Maximum Unit: Smooth Activation Function for Deep Networks using Smoothing Maximum Technique," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 784–793.

[17] S. M. LaValle, *Planning Algorithms*. USA: Cambridge University Press, 2006.

[18] J. Wang, W. Chi, C. Li, C. Wang, and M. Q. Meng, "Neural RRT*: Learning-Based Optimal Path Planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, pp. 1748–1758, 2020.

[19] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep Unsupervised Learning Using Nonequilibrium Thermodynamics," in *32nd International Conference on International Conference on Machine Learning - Volume 37*, 2015, pp. 2256–2265.

[20] Z. Kingston, M. Moll, and L. E. Kavraki, "Exploring implicit spaces for constrained sampling-based planning," *International Journal of Robotics Research*, vol. 38, no. 10–11, pp. 1151–1178, 09 2019.