Initial assessment of a generative adversarial approach to rover path planning supported by space-based assets.

Guglielmo Daddi, Nicolaus Notaristefano, Fabrizio Stesina, Sabrina Corpino

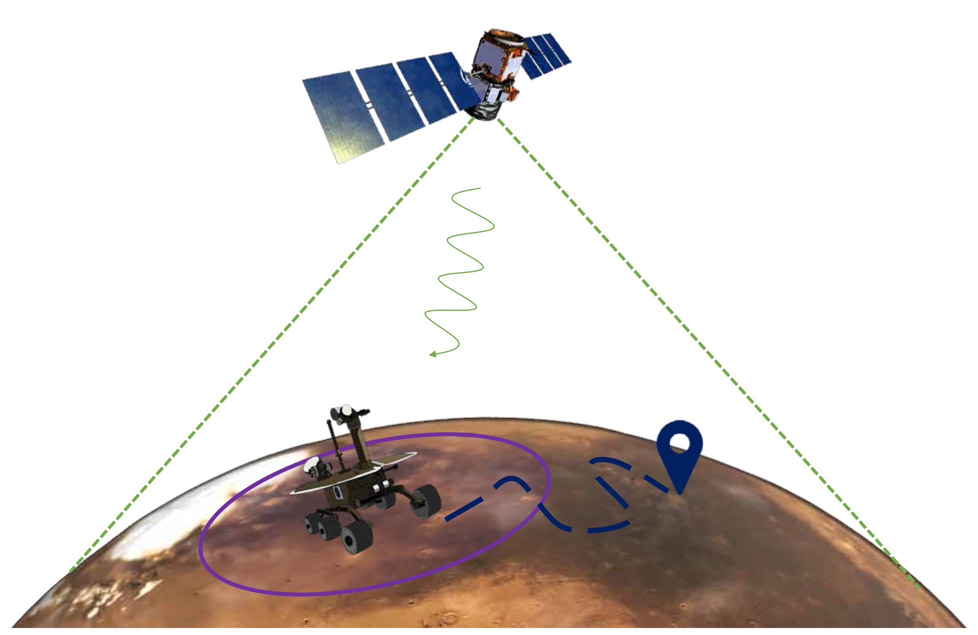
Contact. guglielmo.daddi@polito.it

Politecnico di Torino

Abstract

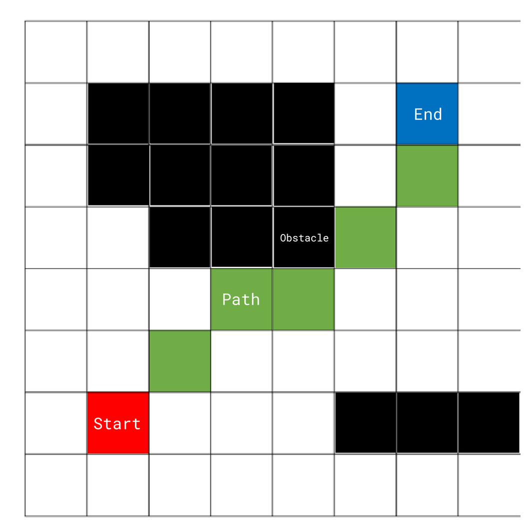
Increasing surface traversal speed for planetary rovers has been a long-standing topic of interest, spurred by the desire to increase science return given a fixed operational budget. In this work, we consider global path planning enabled by space-based assets. As a reference, we assume an architecture where space assets take images with the aim of generating a path suggestion that is communicated to a ground-based system. We focus solely on the path generation aspect; and, specifically, we look at the role that a pretrained generative adversarial network (GAN) may have in approximating efficiently any path planning algorithm on a 2D grid. Generative adversarial networks learn statistical relationships between obstacles, goals, states, and paths and can be asked to guess what a new path would look like given a previously unseen combination of obstacles, goals, and initial state. Given how efficient neural networks are at evaluation time, this approach could allow approximating the behavior of complex planning algorithms that would not run on a SmallSat's onboard computer due to computational limitations. We perform experiments on a 64x64 pixel grid generating a training set using randomly positioned obstacles and goals. The heuristic search algorithm A* is used to generate training paths, due to its abundance in literature and ease of implementation. We experimented with architectural elements and hyperparameters, converging to a pix2pix-based architecture where the generator is trained to generate plausible paths given obstacles and two points, and a discriminator tries to determine whether these maps are real or generated. Additionally, we define a qualitative path generation "success rate" metric derived from the Fréchet inception distance (FID) and optimize our architecture's parameters, ultimately reaching a 74% success rate on the validation set.

Introduction

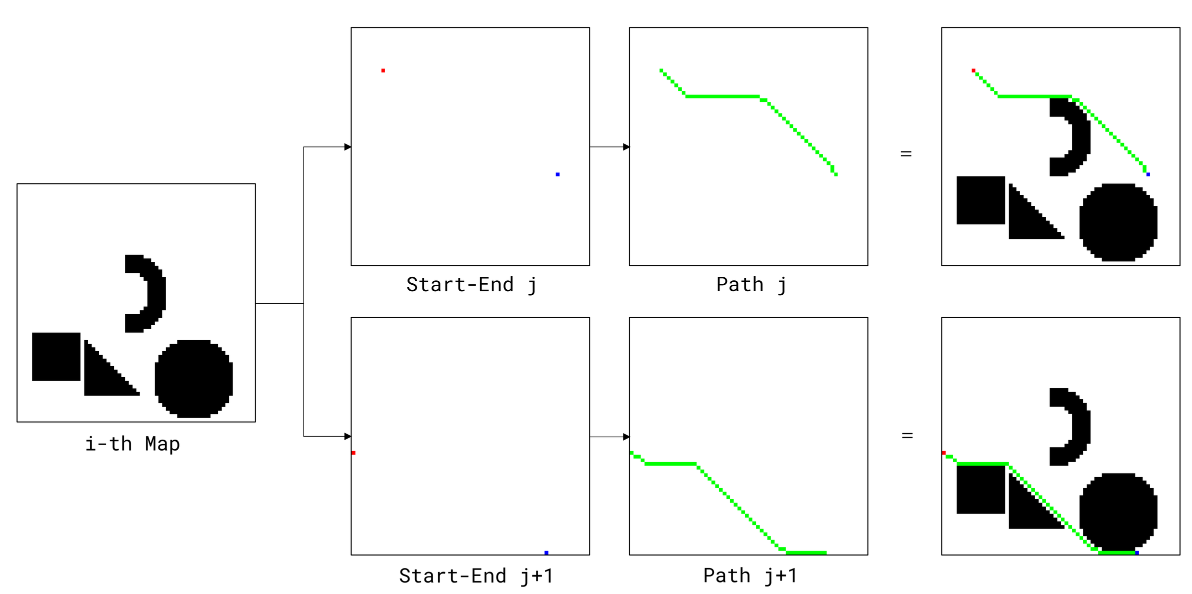


We consider mission architectures where an observer senses the environment, generates a path and subsequently communicates it to the mobility system.
Our approach focuses on solving fully-observable global path planning problems with generative adversarial networks (GAN). The main idea behind this approach is using a GAN to learn the relationship between environmental, goals, and path generated through a classical algorithm. This knowledge is embedded into the network's weights and allows to generate a plausible path given a new map by simple tensor operations, rather than solving a search problem.
All GANs share a common high-level architecture. a generator is trained to produce samples, while a discriminator tries to determine whether the samples are real or fake.
We found three works in literature presenting similar ideas. Ma et al. in 2021 present a method to speed up the RRT algorithm by learning ways to reduce the problem's search space. Zhang et al. propose an algorithm similar to Ma et al., but use a double disriminator architecture. Soboleva et al. in 2019 present an architecture that approximates the A* algorithm using a CGAN.

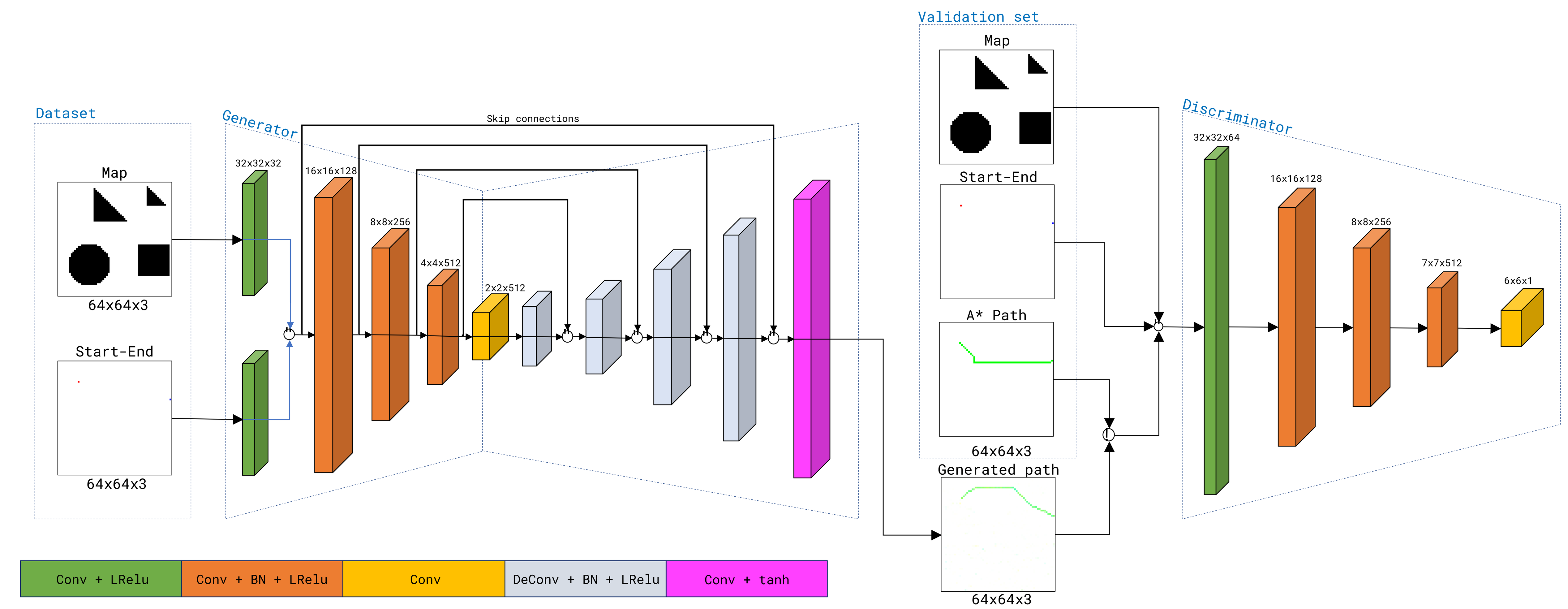
Maps and Dataset



We represent the map as a 64x64 grid of RGB pixels. The black pixels represent obstacles, white are empty space, blue and red are respectively end and start points and green is the path. This grid is represented by a 64x64x3 tensor. We generate training data data procedurally. We create obstacle maps by placing randomly rectangles, triangles, circles and semicircles throughout each map. Each map is populated by a set of goal pairs, and for each goal pair a path is generated by running the A* heuristic search algorithm. We then split the dataset and use 80% for training an 20% for validation



Architecture



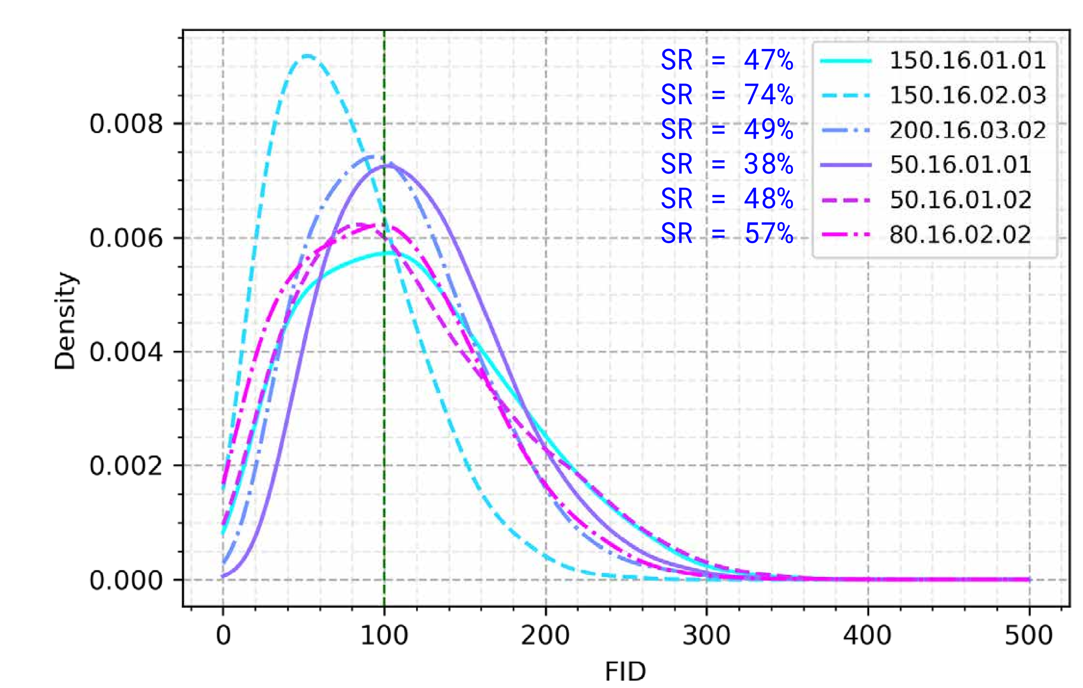
Our network's architecture is based on the pix2pix CGAN. The generator takes as input a map and a point image which are passed through a convolution layer with 32 channels. The results of this operation are concatenated into a single map with 32x32x64 dimension, and fed to a U-Net. the discriminator concatenates its inputs into single data structure that is passed through Convolution-leakyRelu, and convolution-batchnormalization-leakyRelu layers with 4x4 filters. The final layer is a pure convolution layer that generates a 6x6 truth map. Overall Truth is determined by acearaging these 36 values.

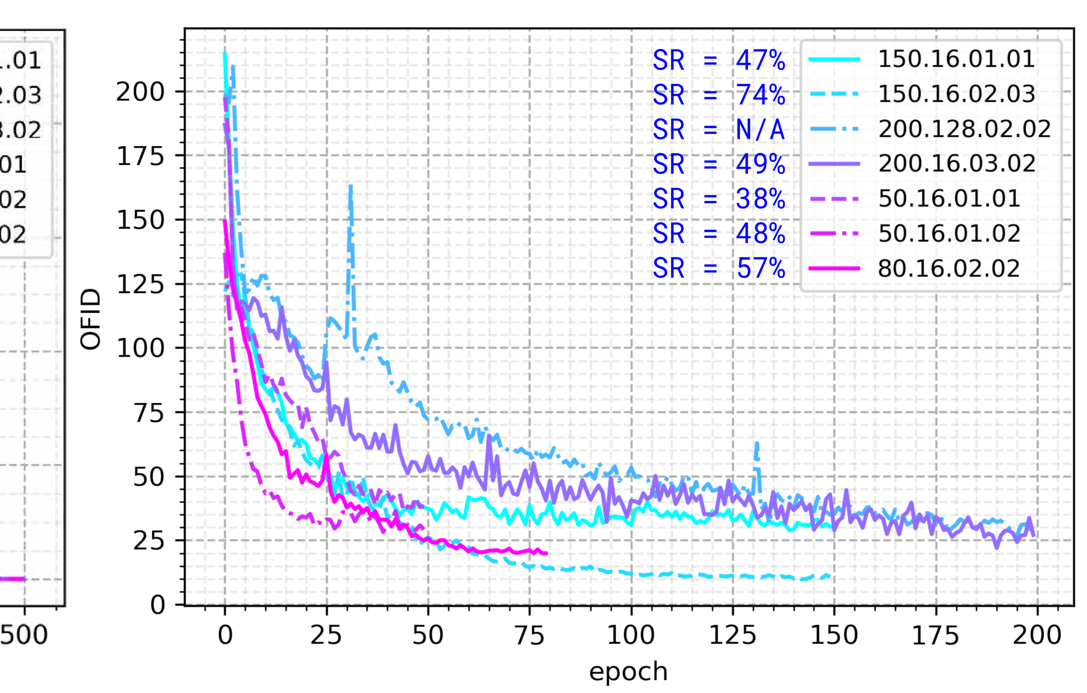
Results

We experimented with a number of architectural parameter and laid out the result with a naming convention seen below that summarizes each architecture. To increase readibility, we lay out the success rates (SR) in the FID image next to the legend. Note that the largest performance increase came from increasing the start and end point size.

number_of_epochs.batch_size.network_ID.dataset_ID

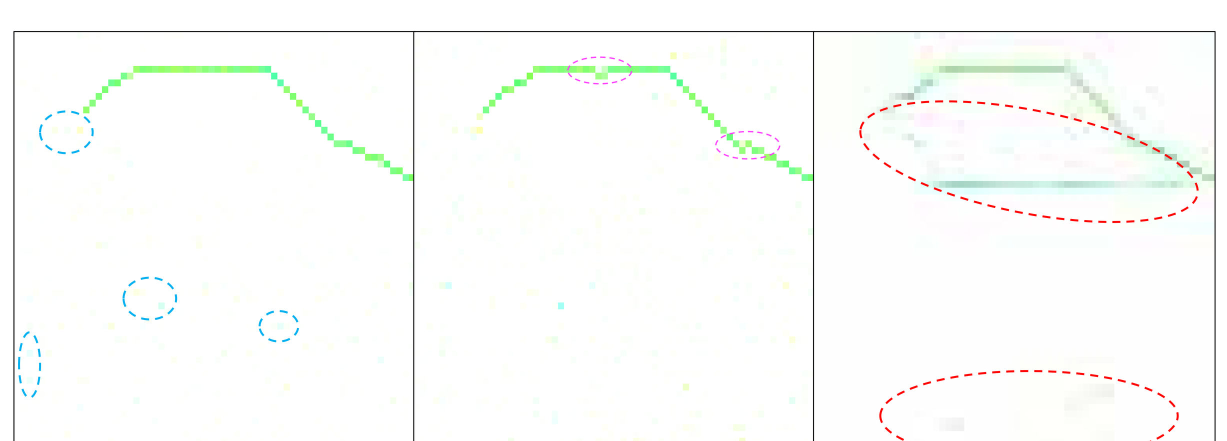
Network ID	Bottleneck size	Dataset ID	paths	goal size
01	2x2x512	01	44500	1x1x3
02	4x4x512	02	115000	1x1x3
03	1x1x512	03	115000	4x4x3





Success metrics

As the main success metric we chose the Fréchet Inception Distance (FID) due to its consistency with human judgement. We compute each of the network's maps FID distribution over the validation set to generate probability density distributions, and for a quicker assessment of network performance we aggregate ground truth maps and generated maps into single images and generate a FID assessment of those images (defined as OFID). For a heuristic look at network peformance, we note that FIDs below 50 lead to "correct" paths, while FIDs greater than 50 include errors or excessive noise. By taking the ratio of FID<50 over all validation maps we define a "success rate".



FID < 50	50 < FID < 100	FID > 100
Noise	Noise + Deviations + Missing pixels	Noise + Deviations + Missing pixels + Missing sections + wrong start or end point

Conclusions

We tested a pix2pix-based conditional generative adversarial network architecture, trained on paths generated on procedural obstacles maps of 64x64 RGB pixles via the A* heuristic search algorithm. We experimented with the effects of number of training epochs, and the size of goals, bottleneck and dataset. The most effective combination of architecture and hyperparameters we found was capable of successfully approximating the path generated with the A* algorithm with a rate of 74%. The main conclusion we reach is that, **with current technology, this approach to path planning is not suitable for the direct generation of paths for safety critical applications**. We recommend attempting future iterations of this approach, as generative network technology improves and success rates increase.