Probabilistic Generative Modeling for Procedural Roundabout Generation for Developing Countries

Zarif Ikram 1905111@ugrad.cse.buet.ac.bd
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

Ling Pan penny.ling.pan@gmail.com
Mila - Québec AI Institute

Dianbo Liu dianbo.liu@alumni.harvard.edu
Mila - Québec AI Institute and NUS, Singapore

Abstract

Due to limited resources and fast economic growth, designing optimal transportation road networks with traffic simulation and validation in a cost-effective manner is vital for developing countries, where extensive manual testing is expensive and often infeasible. Current rule-based road design generators lack diversity, a key feature for design robustness. Generative Flow Networks (GFlowNets) learn stochastic policies to sample from an unnormalized reward distribution, thus generating high-quality solutions while preserving their diversity. In this work, we formulate the problem of linking incident roads to the circular junction of a roundabout by a Markov decision process, and we leverage GFlowNets as the Junction-Art road generator. We compare our method with related methods and our empirical results show that our method achieves better diversity while preserving a high validity score.

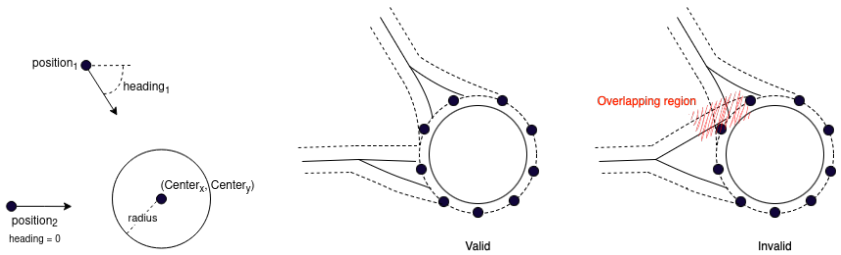
Keywords: Generative Flow Networks, Procedural Content Generation, Synthetic Map Generation, Road Generation

1 Introduction

The road transportation system plays a pivotal role in a nation's infrastructure development, as it fosters economic improvement by enhancing transportation. Effective planning and implementation of road designs are critical, as they not only shape and transform a country's landscape but also lay the foundation for potential adaptation for autonomous vehicles in the future. Nevertheless, in many developing countries, resource constraints impede their capacity to invest extensively in optimal road design. Extensive manual road design is often impractical due to the cost and time required, making road generators a more practical alternative in the real world.

Road design diversity is an important aspect of these generators, as it enables road designers to explore a wider range of options for optimal development. Recent advancements in road generators have enabled the creation of complex and diverse roads. However, many of these generators do not adequately address several intricate road structures commonly found in the real world. Among these structures, the generation of diverse roundabouts is particularly crucial, as they are one of the most common road configurations encountered in reality. Additionally, since roundabouts serve as modular components within other road networks, they need to exhibit diversity within a specific incident road network, providing multiple

options to enhance flexibility and adaptability in road design and planning. Consequently, both diversity and realism are equally important in the context of road design.



(a) Example problem (b) Solution

Figure 1: For any given incident road configurations, incident roads can connect to the circular roads in many valid ways.

In this paper, we propose to automatically generate such roundabouts with Generative Flow Networks (GFlowNets) (Bengio et al., 2021b), which is a generative model that can sample roundabouts proportional to a reward function. By employing this method, we can sample and discover roundabouts that cover the different modes of the reward function specified for realism. In this work, we formulate the problem of connecting incident roads in the circular roads of the roundabout by a Markov decision process and connect GFlowNets to the Junction-Art (Muktadir et al. (2022a)) road generator to sample diverse roundabouts. To the best of the authors' knowledge, this is the first work that incorporates GFlowNets for road generation. The key contributions are the following.

- A Markov Decision Process(MDP) formulation for connecting incident roads to the circular road junction of a roundabout
- A novel algorithm based on GFlowNets for diverse roundabout configuration generation with a carefully designed diversity measure in the context of road generation
- Strong empirical experimental results in terms of diversity and performance in roundabout generation

2 Problem Setup

The r road configurations are denoted by $P = \{(\text{position}_1, \text{heading}_1, n_{\text{leftlanes},1}, n_{\text{rightlanes},1}), \ldots, (\text{position}_r, \text{heading}_r, n_{\text{leftlanes},r}, n_{\text{rightlanes},r})\}$, where each tuple represents the position, heading of the starting point of an incident road and the number of incoming and outgoing lanes it has. We want to generate a roundabout by creating the circular road configuration with $(\text{Center}_x, \text{Center}_y, radius)$, where $(Center_x, Center_y)$ represents the center coordinate and $radius$ represents the radius of the circular shape of the roundabout. We build the circular road with C parametric cubic road segments and connect each N lanes $N = \sum_{i \in \{1,\ldots,r\}} n_{leftlanes,i} + n_{rightlanes,i}$ to one of the C circular road segments $s_1, \ldots, s_C$. We wish to create a roundabout for which the configuration of the connection is valid. Formally, we want to search over a (terminal) state space of $\mathcal{X}$ to find configurations $x \in \mathcal{X}$ that have a validity reward $f : x \to R^+$ which indicates design and construction feasibility of the roundabout where $x = \{s_{(x_1)}, \ldots, s_{(x_N)}\}$, while maintaining high diversity. For example, Figure 1 represents a problem where $C = 9$ and $N = 4$, and Figure 2 represents a problem where $C = 10$, $N = 4$, and $x = \{8, 6, 1, 9\}$.

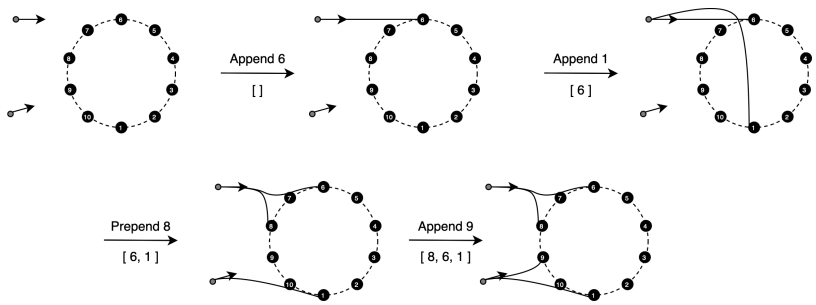


Figure 2: MDP design for incident road configuration generation. Number on dark filled circles on the dotted circle represent the endpoints of circle road segments. Small filled circle with arrow represent the road definition where direction of the arrow represent the heading of the initial road.

In this context of this problem, we follow the algorithm proposed in Ikram et al. (2023) to generate circular road segments. We want to generate diverse configurations for which the generated roundabouts are valid, i.e., there are no overlaps between any incident roads and circular road segments. Finally, we generate diverse roundabouts by connecting the incident lanes with circular segments according to the configuration.

2.1 Criteria for Roundabout Diversity

As the main goal of this work is the diversification of the roundabouts, we present a measure of diversity of the candidates. We adopt the diversity measure proposed in Jain et al. (2022) for our work, i.e., for a set of generated candidates $\mathcal{D}$,

$$Diversity(\mathcal{D}) = \frac{\sum_{x_i \in \mathcal{D}} \sum_{x_j \in \mathcal{D} \setminus \{x_i\}} d(x_i, x_j)}{|\mathcal{D}|(|\mathcal{D}| - 1)} \tag{1}$$

where d is the distance function. In our work, the diversity of the generated roundabouts emerges from the diversity of the connection roads between the incident and circular roads. As a result, we define the distance function d as the mean discrete Fréchet distance (Eiter and Mannila (1994)) between each connection road geometries. To discretize each geometry, we sample m equidistant coordinates from the geometry where $m = 10$.

3 Method

3.1 GFlowNets

GFlowNets (Bengio et al. (2021a), Bengio et al. (2021b)) are amortized samplers that can sample from high-dimensional state spaces. Given a state space $\mathcal{X}$, they can learn a stochastic policy π that can sample states $x \in \mathcal{X}$ using a non-negative reward function $R(x)$ such that $\pi(x) \propto R(x)$. This property of GFlowNets makes them the ideal candidate for generating diverse yet valid roundabouts.

We construct objects $x \in \mathcal{X}$ by sampling constructive, irreversible actions $a \in \mathcal{A}$ that transition s_t to s_{t+1}. Therefore, in graph terminology, GFlowNets generate a graph $(\mathcal{S}, \mathcal{E})$ where each node in $\mathcal{S}$ denotes a state in the Markov chain with an initial state s_0, and each edge in $\mathcal{E}$ denotes a transition $s_t \to s_{t+1}$ with a special terminal action indicating $s = x \in \mathcal{X}$. Moreover, the graph is an acyclic DAG. A trajectory $\tau \in \mathcal{T}$is a Markov chain $(s_0 \to s_1 \to \cdots \to x), x \in \mathcal{X}$ where $\mathcal{T}$ is the set of all trajectories.

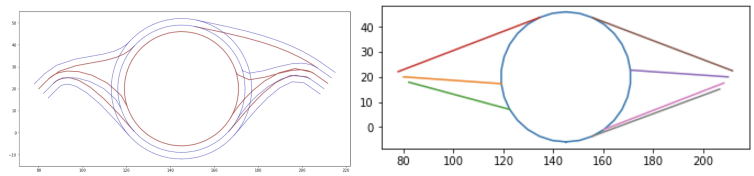


Figure 3: *Left*: Geometry of a generated roundabout. *Right*: Geometry generated by the proxy reward function.

By learning the stochastic forward and backward policies using a neural network, GFlownets learn to construct objects with the aforementioned properties. The forward policy $P_F(s_{t+1}|s_t;\theta)$ represents the probability distribution from s_t to s_{t+1}, where the neural network is parameterized by θ. The backward policy P_B can be chosen freely, and here we use a uniform P_B, which is uniformly distributed among allowed actions from s_t+1 to s_t. Therefore, the probability of sampling an object $x \in \mathcal{X}$ is:

$$\pi(x) = \sum_{\tau \in \mathcal{T}: x \in \tau} \prod_{t=0}^{|\tau-2|} P_F(s_{t+1}|s_t;\theta) \tag{2}$$

We use the trajectory balance objective (Malkin et al. (2022)) to train the GFlowNet, which is defined as in Eq. (3), where Z_θ is a learnable parameter that represents the total flow: $\sum_{x\in\mathcal{X}} R(x) = \sum_{s:s_0\to s\in\tau\forall\tau\in\mathcal{T}} P_F(s|s_0;\theta)$.

$$\mathcal{L}(\tau;\theta) = \big(\log Z_\theta + \log \sum_{s_t\to s_{t+1}\in\tau} P_F(s_{t+1}|s_t;\theta) - \log R(x) - \log \sum_{s_t\to s_{t+1}\in\tau} P_B(s_t, s_{t+1};\theta)\big)^2 \tag{3}$$

3.2 MDP For Incident Lane Connection Configuration

Here, we define the actions required to generate the MDP in the context of our problem. Starting from an empty state , we append or prepend a slot number $s_i, i \in 1,\ldots,C$ to the state. We, then, conclude the MDP with a special *terminate* action. Therefore, the action space is defined as $\mathcal{A} = \{a_1,\ldots,a_C,p_1,\ldots,p_C,TERMINATE\}$. Here a_i and p_i represent the actions of appending and prepending slot i to the current state, respectively. This choice of action space makes for a constant trajectory ($|\mathcal{T}| = N+1$) while keeping the action space relatively small. Figure 2 shows a detailed example of the generation of a valid configuration using this MDP.

3.3 GFlowNets for Roundabout Design

We begin by noting that the search space is C^N, which scales rapidly with N. To make matters worse, roundabout validation is costly, as each roundabout needs to be generated first to validate, which is time-intensive. On a desktop, one roundabout takes 200ms to generate, meaning we can validate only 18000 configurations in an hour, making GFlowNet training impractical with a reward function that depends on proper validation of generated configurations. To tackle this challenge, given $f(line_i, line_j)$ returns 1 or 0 depending on whether $line_i$ and $line_j$ intersects or not, we propose a proxy reward function that approximately validates the configuration by associating it with the following reward.

$$D(x) = \sum_{i<j}^{i,j\in N} (1 - f(line_i, line_j)) + \sum_{i}^{i\in N} (1 - f(line_i, circle)) \tag{4}$$

$$D_{normalized}(x) = \frac{D(x)}{\frac{N(N-1)}{2} + N} \tag{5}$$

$$R_{proxy}(x) = base^{D_{normalized}(x)} \tag{6}$$

We illustrate the workings of the proxy reward function using Figure 3. The proxy reward function creates straight lines $line_i$ between the point associated with $s_{(x_i)}$ and the endpoint of $lane_i$ $\forall i \in \{1, \ldots, N\}$. Using straight lines, we can approximately validate the configuration by associating a reward with each non-intersecting lines and circle. Besides, to sample from the high-reward region only, we use a reward exponent.

We train the GFlowNet for a specific input of N, C, and P. For each episode, we begin by generating m-sized minibatches of trajectories $\mathcal{T}$. Notably, we select actions by sampling using P_F from valid actions from s. Upon receiving $TERMINATE$, we calculate the proxy reward from $\mathcal{T}$ and forward and backward flows. Finally, we calculate $\mathcal{L}$ and update parameters θ. We detail the process in algorithm 1 in Appendix C.

For roundabout generation, we sample batch B according to GFlowNet policy π_θ and extract K configurations with the highest proxy reward. Initially, we create circular junctions using P and C, followed by the incident roads. Using $top - K$, we connect the circular junctions using cubic polymetric roads and evaluate the validity of the connections. Finally, we extract the distinct samples that generate valid roundabouts. Algorithm 2 from Appendix C shows the procedure.

4 Experiments

4.1 Results & Analysis

Experimental Setup We now analyze the effectiveness of our method for roundabout generation as detailed in Section 2. We follow Ikram et al. (2023) for generating input road configurations, which generates a random circle, followed by sampling points from the circle and adding noise to the sampled coordinates and their headings. We generate 20 road configurations for $N = 4, 6$, and 8. To ensure comparability, we use the same input for all methods. We compare our method with Ikram et al. (2023) and a strong reinforcement learning method — Soft Actor-Critic (SAC) (Haarnoja et al. (2018)) for comparison which is widely used to generate diverse states by probabilistically sampling the action space. We use a discrete SAC (Christodoulou (2019)) with a 64-dimentional hidden layer for both actor and critic networks, a replay buffer of 5000 capacity, as well as $\gamma = 0.99$, $\tau = 0.01$, and a parameterized α. We construct the GFlowNet agent with a 128-dimensional hidden layer for the forward layer and a uniform distribution for the backward layer. Finally, we train the GFlowNet and SAC agents for 1000 and 2000 iterations, respectively.

Analysis We begin the evaluation by sampling 10^4 samples, selecting top-K samples based on proxy rewards, and generating roundabouts from them. Then, we calculate the scores based on the number of non-overlapping lanes and the diversity based on Equation 1. We detail the results in Table 1. Our analysis show that both GFlowNets and SAC achieves comparable scores to the baseline. Due to maximizing proxy rewards, SAC achieves higher score than GFlowNets in small tasks ($N = 4$), but lacks diversity, falling short in case of bigger tasks or higher sampling.

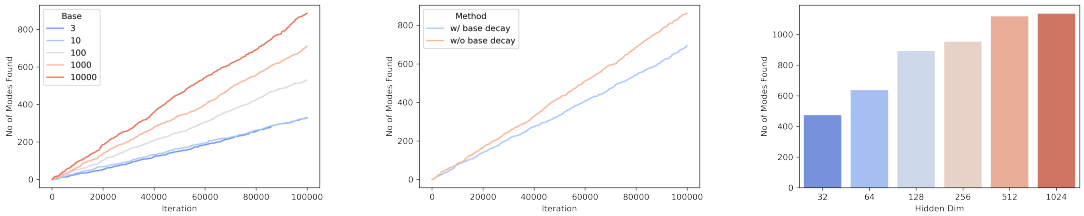
4.2 Ablation Study

To evaluate the design choices for our model, we conduct an in-depth ablation study next. First, we vary *base* from proxy reward R_{proxy} in Equation 6 and observe the number of

Table 1: Roundabout generation experimental results. Baseline uses rule-based method, and so cannot generate diverse roundabouts. Thus, "—" denotes its diversity.

N	METHOD	$K = 50$		$K = 200$	
		SCORE	DIVERSITY	SCORE	DIVERSITY
4	BASELINE	6.6 ± 1.3	—	6.6 ± 1.3	—
	SAC	6.8 ± 0.6	0.4 ± 0.5	6.7 ± 0.8	0.5 ± 0.8
	OURS	6.6 ± 0.7	4.7 ± 1.7	6.6 ± 0.7	4.8 ± 1.5
6	BASELINE	17.4 ± 0.8	—	17.4 ± 0.8	—
	SAC	15.0 ± 1.1	2.4 ± 1.5	14.8 ± 1.2	2.9 ± 1.7
	OURS	15.6 ± 0.9	6.3 ± 0.6	15.6 ± 1.0	7.0 ± 0.5
8	BASELINE	30.3 ± 1.2	—	30.3 ± 1.2	—
	SAC	25.8 ± 2.1	3.4 ± 2.0	25.1 ± 2.3	3.4 ± 2.0
	OURS	28.1 ± 1.4	7.0 ± 0.48	27.8 ± 1.5	7.9 ± 0.4

diverse modes sampled from each model. For the ablation study, we define a mode for which $D_{normalized}$ from Equation 5 is more than 0.85. We observe a positive correlation with the base and number of sampled modes. Furthermore, to experiment with the effect of decaying the base in training, we train two models with and without base decay. Figure 4b shows that decaying base, in fact, results in a smaller number of modes sampled. We also observe a trend with the number of hidden parameters and sampled modes, although the effect diminishes with increasing parameters (Figure 4c).



(a) (b) (c)

Figure 4: Ablation study on GFlowNet training.

5 Conclusion

Automatic road generation is important for intricate road design in developing nations for its cost-effectiveness. Diverse road generation can pave the way for better infrastructure development by providing designers and builders with more options. Roundabouts are notoriously underrepresented in road generators for their difficult intersection design, despite being an important component in road design. As the pluggable component within a set of already established road networks, generated roundabouts need to be diverse for the same set of incident roads. Based on Junction-Art, a high-definition (HD)-map generator, we provide a novel algorithm based on GFlowNets that can sample valid yet diverse configurations for roundabout generation. We also show that the roundabouts discovered by our algorithm are novel and diverse by comparing our method with other methods.

References

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *ArXiv*, abs/2106.04399, 2021a. URL `https://api.semanticscholar.org/CorpusID:235367990`.

Yoshua Bengio, Tristan Deleu, J. Edward Hu, Salem Lahlou, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *ArXiv*, abs/2111.09266, 2021b. URL `https://api.semanticscholar.org/CorpusID:244270393`.

Guoning Chen, Gregory Esch, Peter Wonka, Pascal Müller, and Eugene Zhang. Interactive procedural street modeling. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008.

Petros Christodoulou. Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*, 2019.

Rémi Cura, Julien Perret, and Nicolas Paparoditis. Streetgen: In-base procedural-based road generation. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W5:409–416, 08 2015. doi: 10.5194/isprsannals-II-3-W5-409-2015.

Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio M. López, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning*, 2017. URL `https://api.semanticscholar.org/CorpusID:5550767`.

Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. 1994.

A. Gambi, M. Mueller, and G. Fraser. Asfault: Testing self-driving car software using search-based procedural content generation. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 27–30, 2019. doi: 10.1109/ICSE-Companion.2019.00030.

Tianhan Gao, Qi Gao, and Tingting Yu. A procedural generation method of urban roads based on osm. *2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pages 242–248, 2022.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Zarif Ikram, Golam Md Muktadir, and Jim Whitehead. Procedural generation of complex roundabouts for autonomous vehicle testing. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6, 2023. doi: 10.1109/IV55152.2023.10186533.

Moksh Jain, Emmanuel Bengio, Alex Hernández-García, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Micheal Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with gflownets. *ArXiv*, abs/2203.04115, 2022. URL `https://api.semanticscholar.org/CorpusID:247315277`.

Moksh Jain, Tristan Deleu, Jason S. Hartford, Cheng-Hao Liu, Alex Hernández-García, and Yoshua Bengio. Gflownets for ai-driven scientific discovery. *ArXiv*, abs/2302.00615, 2023. URL `https://api.semanticscholar.org/CorpusID:256459319`.

Quanyi Li, Zhenghao Peng, Qihang Zhang, Cong Qiu, Chunxiao Liu, and Bolei Zhou. Improving the generalization of end-to-end driving through procedural generation. *arXiv preprint arXiv:2012.13681*, 2020.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *ArXiv*, abs/2201.13259, 2022. URL `https://api.semanticscholar.org/CorpusID:246430534`.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Golam Md Muktadir, Abdul Jawad, Ishaan Paranjape, Jim Whitehead, and Aleksey Shepelev. Procedural generation of high-definition road networks for autonomous vehicle testing and traffic simulations. *SAE International Journal of Connected and Automated Vehicles*, 6:22, 05 2022a. doi: 10.4271/12-06-01-0007.

Golam Md Muktadir, Abdul Jawad, Aleksey Shepelev, Ishaan Paranjape, and Jim Whitehead. P r e -p r i n t realistic road generation: Intersections. 05 2022b. doi: 10.13140/RG.2.2.30541.51683.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Yun Tang, Yuan Zhou, Kairui Yang, Ziyuan Zhong, Baishakhi Ray, Yang Liu, Ping jun Zhang, and Junbo Chen. Automatic map generation for autonomous driving system testing. *ArXiv*, abs/2206.09357, 2022.

Jie Wang, Gary Lawson, and Yuzhong Shen. Automatic high-fidelity 3d road network modeling based on 2d gis data. *Advances in Engineering Software*, 76:86–98, 10 2014. doi: 10.1016/j.advengsoft.2014.06.005.

Yahia A. Zakaria, Magda Bahaa Eldin Fayek, and Mayada M. Hadhoud. Start small: Training game level generators from nothing by learning at multiple sizes. *ArXiv*, abs/2209.15052, 2022. URL `https://api.semanticscholar.org/CorpusID:252668700`.

Chong Zhang. Generating a terrain-robustness benchmark for legged locomotion: A prototype via terrain authoring and active learning. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9190–9196, 2022. URL `https://api.semanticscholar.org/CorpusID:251594585`.

Haibin Zhou, Zichuan Lin, Junyou Li, Qiang Fu, Wei Yang, and Deheng Ye. Revisiting discrete soft actor-critic. *arXiv preprint arXiv:2209.10081*, 2022.

Supplementary material for

Probabilistic Generative Modeling for Procedural Roundabout Generation for Developing Countries

Zarif Ikram, Ling Pan & Dianbo Liu

Appendix A. Related Work

A.1 Automatic Road Generation

Due to the importance of HD maps in AV simulation testing, many works tackle the problem of road generation, most of which can be divided into two groups: (a) from real-world data (Cura et al. (2015), Tang et al. (2022), Wang et al. (2014), Chen et al. (2008)) and (b) from constructive generation (Gambi et al. (2019), Gao et al. (2022), Li et al. (2020), Ikram et al. (2023)). Recently, more advanced work such as Junction-Art (Muktadir et al., 2022a) proposes using procedural content generation (PCG) to produce road maps that are directly pluggable to existing simulation environments such as CARLA (Dosovitskiy et al., 2017) using OpenDrive format. Muktadir et al. (2022b) brings intersection generation to that work. However, current generation of road-generators lacked roundabout generation, which, more recently, Ikram et al. (2023) addressed, but it generated roundabouts using a rule-based method, producing a fixed set of roundabouts. As discussed in Section 1, we improve that aspect using GFlowNets sampling in this work.

A.2 GFlowNets for Generation

Many recent works show employing GFlowNets diverse candiate generation, such as for biological sequence design (Jain et al., 2022), scientific discovery (Jain et al., 2023), terrain generation (Zhang, 2022), and game level design (Zakaria et al., 2022). Notably, given the methods used, Zakaria et al. (2022) is the closest to our work.

A.3 Reinforcement Learning Methods for Discrete Action Space

Many reinforcement larning (RL) algorithms such as Mnih et al. (2013) and Schulman et al. (2017) promise proxy reward optimization for discrete action spaces. Soft actor critic algorithms (Haarnoja et al. (2018)) propose improved exploration by optimizing both its rewards as well as the entropy. Albeit originally proposed for continuous action space, the algorithm finds its application in discrete action space as well (Christodoulou (2019), Zhou et al. (2022)).

Appendix B. Other Experiments

In this section, we provide other experiments mentioned in Section 4. We study the effect of GFlowNets on two problems:

- 2 incident road roundabouts with a 2-2 lane split ($N = 4$)
- 2 incident road roundabouts with a 2-3 lane split ($N = 5$)

We pick $C = 15$ and 20 for the 2-2 and 2-3 split, respectively. We train the GFlowNets for 1000 episodes with a batch size $b = 16$. For 2-2 split, we sample 2000 configurations after training and find 107 modes. For 2-3 split, we sample 16000 configurations and find 157 distinct modes. It is worth noting that during training, 146 distinct modes were found, and after training, only 9 modes repeat again (Figure 5). Thus, the GFlowNets agent samples configurations not seen before. Figure 6 shows some of the outputs and the diversity in them.

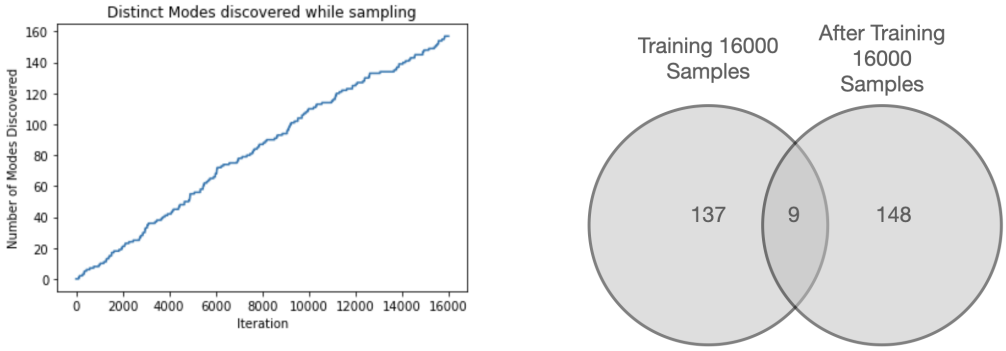


Figure 5: Left diagram shows the number of distinct modes discovered as a function of number of samples for $N = 5$ and $C = 20$. Right diagram shows that the agent discoverers configurations not seen previously in training.

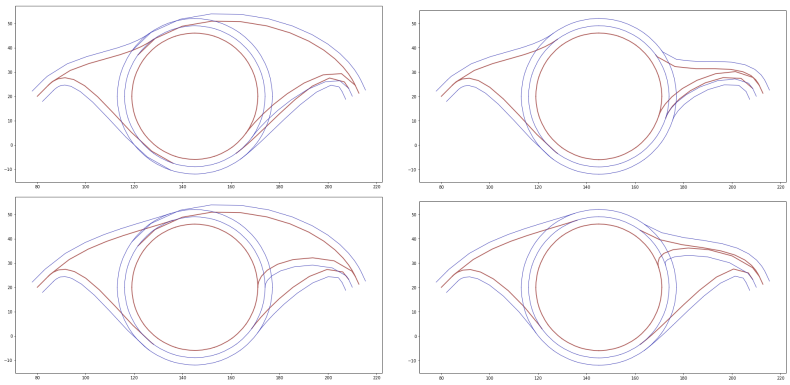


Figure 6: Diverse roundabouts generated by GFlowNets sampling

.

Appendix C. Algorithms

In this section, we provide the alogithms mentioned from Section 3.3.

Algorithm 1: GFlowNet Training

Input:
P_F: Forward flow
P_B: Backward flow (initialized as uniform flow)
Z_θ: Logarithmic total flow learned by neural network
N = Total number of lanes
m = Minibatch size
for *each episodes* **do**
  $s \leftarrow \{s_0^1, \ldots s_0^m\}$
  Initialize trajectory $\mathcal{T}$ to an empty list
  while *s not a terminal state* **do**
    Sample actions $A = \{a^1, \ldots a^m\}$ based on $P_F(s; \theta)$
    $s' \leftarrow transition(s, A)$
    Append (s', A) to the $\mathcal{T}$
    $s \leftarrow s'$
  Compute R_{proxy}(s) using reward function using equation 4 from $\mathcal{T}$ using the last state in $\mathcal{T}$
  Compute loss $\mathcal{L}$ using trajectory balance loss function in 3 from $\mathcal{T}$
  Update parameters θ using stochastic descent on loss $\mathcal{L}$

Algorithm 2: Roundabout Generation With GFlowNet

Input:
P: A set of road configurations $(x_i, y_i, h_i, nLeftLanes_i, nRightLanes_i)$
N = Total number of lanes
π_θ: Generative policy learned by GFlowNet trained using reward function R
C : Number of circular segments
b : Batch size
Result: O : Roundabout-like road geometries having the best rewards
Algorithm
Create circular road junction using C and P
Train π_θ using Proxy Reward function R_{proxy}
Sample batch $B = x_i, \ldots, x_b$ with $x_i \sim \pi_\theta$
Extract $top - K$ by taking distinct K configurations from B with highest proxy rewards
Connect incident roads using configurations from $BestEncodings$ and build $|BestEncodings|$ roundabouts