데이터 사이언스 연구실
GIST Data Science Lab

Solution Augmentation for ARC Problems Using GFlowNets: A Probabilistic Exploration Approach

Sanha Hwang

MS Candidate

DS LAB@GIST AIGS 2F Ted Hall
2024.11.29

hsh6449j@gm.gist.ac.kr

Thesis Committee

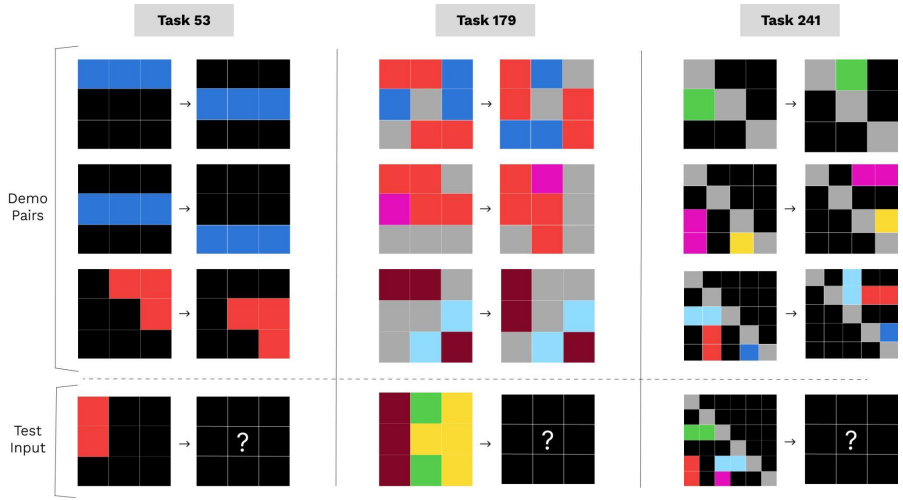
Prof. Sundong Kim (Chair)
Prof. Mansu Kim
Prof. Byungjun Lee (Korea University)

Contents

Introduction: Motivation

Abstraction and Reasoning Corpus (ARC)[1][2]

- IQ Test처럼 인공지능 모델의 지능을 평가하는 벤치마크 데이터셋
- 적은 개수의 예시(2~5)가 주어지고, 예시들 사이에 공유하는 규칙을 추론하여 맞추는 문제
- Multi task & OOD Problem



Introduction: Motivation

ARC를 풀기 위한 다양한 이전의 접근들

- LLM Approach: GPT [3], Code generation [4][5], MIND AI [25]…
- Program synthesis Approach: Dream coder [6], …
- Deep Learning: AE [7], MDL [8], …
- Hard-Coding: Icecuber [9]
- Test Time Adaptation : TTT [24]
- …

Introduction: Motivation

ARC를 풀기 위한 다양한 이전의 접근들

- LLM Approach: GPT [3], Code generation [4][5], MIND AI [25]…
- Program synthesis Approach: Dream coder [6], …
- Deep Learning: AE [7], MDL [8], …
- Hard-Coding: Icecuber [9]
- Test Time Adaptation : TTT [24]
- …

현재 ARC Prize 1등팀 (MIND AI)을 포함해서 다양한 딥러닝 기반 접근에 Data Augmentation이 활용되고 있다.

Motivation : Prior Works of Data Augmentation On ARC

ARC에서 데이터 증강의 역할

- 현재 연구 방향에서, 데이터 증강은 특히 ARC(Abstraction and Reasoning Corpus)와 같은 태스크에서 모델의 추론 능력을 향상시키는 데 필수

기존 데이터 증강 접근 방식

1. **Solution 증강**: **Rule-Based Approach** **[10]**
   - 사람이 직접 **하드코딩을 통해 데이터를 증강**하여, 특정 규칙 기반 데이터를 생성함

2. **Input 및 Output Pair 증강**: **In-context learning Approach** (MIND AI), **Rule Based Approach (RE-ARC)[26]**
   - MIND AI의 공개된 접근 방식은 방대한 데이터 증강 전략과 Test-Time Training (TTT)을 결합하여 성능을 크게 향상시킴.
     - **주어진 예시에서 하나의 예시를 빼고 나머지를 학습하는 leave-1-out 방식으로 태스크를 생성**
     - 즉, 모델이 한 가지 예시를 제외하고 나머지 예시를 학습 데이터로 사용하는 방식
     - 기하학적 변환을 부트스트랩핑에 적용
   - RE-ARC는 **Rule based로 input-output pair 증강**을 시도

Motivation : Why is Augmentation Beyond Input-Output Pairs Necessary?

기존 데이터 증강 접근 방식의 한계

1. **Input-Output 증강**
   - MIND AI의 접근 방식은 **Input-Output Pair 증강에 초점**을 맞추고 있음
     - Solution 증강을 통해 좀 더 다양한 분포에 대해 알려주면 성능이 올라가지 않을까?
   - 이 방식은 강화학습에 필요한 Solution을 다양화하지 못함
     - Offline RL모델을 학습시키기 위해서는 trajectory를 추가적으로 증강해야 함
     - Input-Output 방식만으로는 trajectory가 blackbox가 되어 설명 가능성을 잃는 문제가 발생

2. **Rule-Based 증강**
   - Solution / Input-Output pair를 직접 증강하기 위해 Rule-Based 방식을 사용할 경우:
     - 시간과 노력이 과도하게 소모됨
     - 규칙을 설계하고 디버깅하는 데 많은 리소스가 필요하며, 확장성이 낮음

Motivation : Proposed Approach

연구의 핵심 목표

- Solution Augmentation을 통한 강화 학습 혹은 ARC Solver 개선:
  - 기존의 데이터 증강 방식의 한계를 극복하기 위해 Solution 증강을 도입.
  - GFlowNet을 활용해서 trajectory를 다양화하고 학습 데이터의 구조적 복잡성을 증가시켜, ARC 문제에서 모델의 추론 및 일반화 성능을 강화.

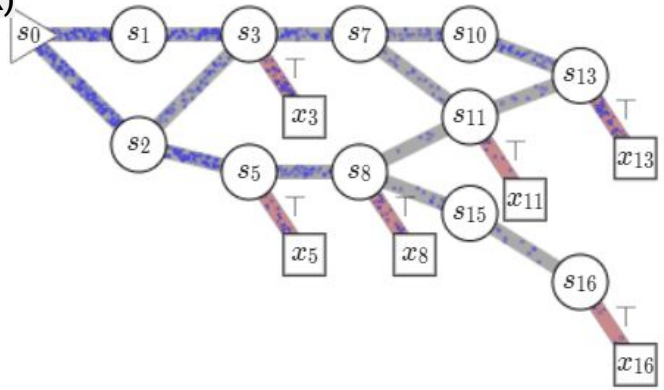
GFlowNet을 활용한 주요 전략

- **Probabilistic Solution Exploration**:
  - GFlowNet의 확률적 탐색 방식을 활용하여 다양한 솔루션(trajectory)을 생성.
  - 각 경로에 대한 보상 분포를 학습하여 모델이 ARC 문제에서 다양한 해결 방식을 학습 가능.
- **Sparse Reward 문제 해결**:
  - 강화 학습 환경에서의 sparse reward 문제를 GFlowNet의 분포 기반 학습을 통해 완화.
  - 이를 통해 ARC 문제에서 새로운 규칙을 발견하고 학습.
- **Off-Policy 학습**:
  - 효율적인 Off-Policy 전략을 도입하여, 다양한 경로를 병렬적으로 학습 가능.

What is Generative Flow Networks (GFlowNets)?

GFlowNets의 개념

- **GFlowNets**은 강화학습**(RL)**과 생성 모델**(Generative Model)**의 결합으로**, Flow Network** 개념에 기반한 모델
- **Terminal state**에서 받은 보상을 기반으로 **Flow**를 학습하는 강화학습 특성을 따름 $\pi(x) \propto R(x)$
- **보상 모델의 분포 자체를 학습 (RL**과의 차이점**)**
  - **Action Sequence**를 생성하는 생성 모델의 특성을 가짐
- **인간의 사고 방식을 모방: Stacking Thoughts**
  - 이는 복잡한 문제를 단계적으로 해결하는 과정을 학습함

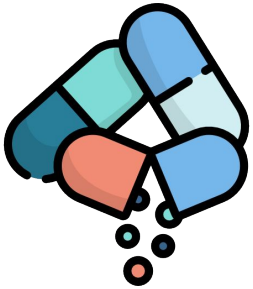


GFlowNets의 강점

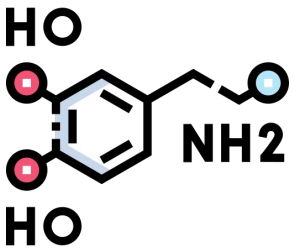
- 다양한 후보 탐색**: GFlowNet**은 높은 보상을 가진 다양한 후보 경로를 탐색하고 학습
  - 복잡한 문제에서 다수의 해답을 찾고**,** 일반화 성능을 향상시키는 데 효과적이라고 알려짐.

What is Generative Flow Networks (GFlowNet)?

기존 GFlowNet의 활용 사례



Drug Discovery
[12][13][14]



Molecular Discovery
[15][16][17]

Graph Combinatorial Optimization Problem
[18][19]

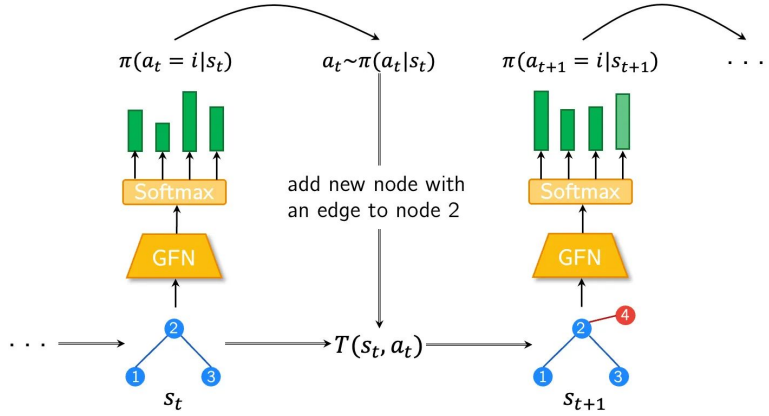
Biological Sequence Generation
[20]

What is Generative Flow Networks (GFlowNet)?

- Objective Function of GFlowNet
  - Trajectory Balanced Loss [21]

$$Z_\theta \prod_{t=1}^{n} P_F(s_t|s_{t-1}) = R(x) \prod_{t=1}^{n} P_B(s_{t-1}|s_t)$$

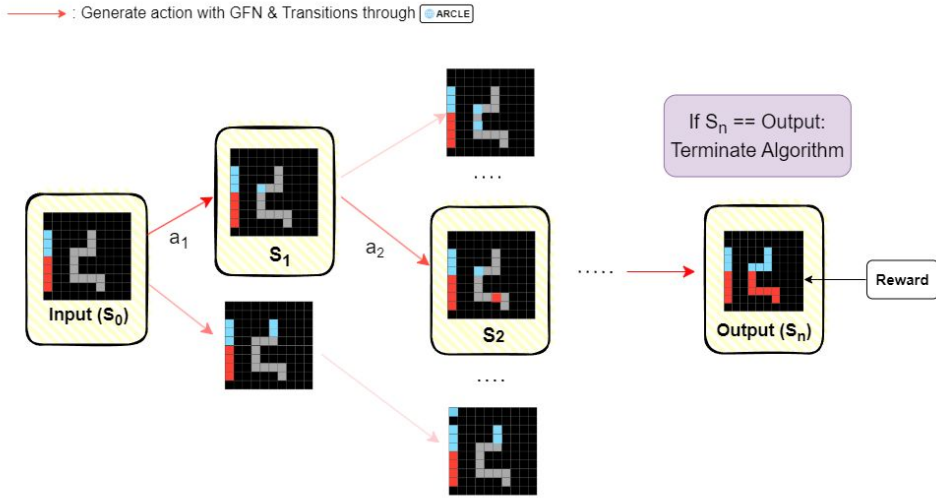
$$\mathcal{L}_{\text{TB}}(\theta) = \left(\log Z_\theta + \sum_{t=1}^{n} \log P_F(s_t|s_{t-1}) - \log R(x) - \sum_{t=1}^{n} \log P_B(s_{t-1}|s_t) \right)^2$$



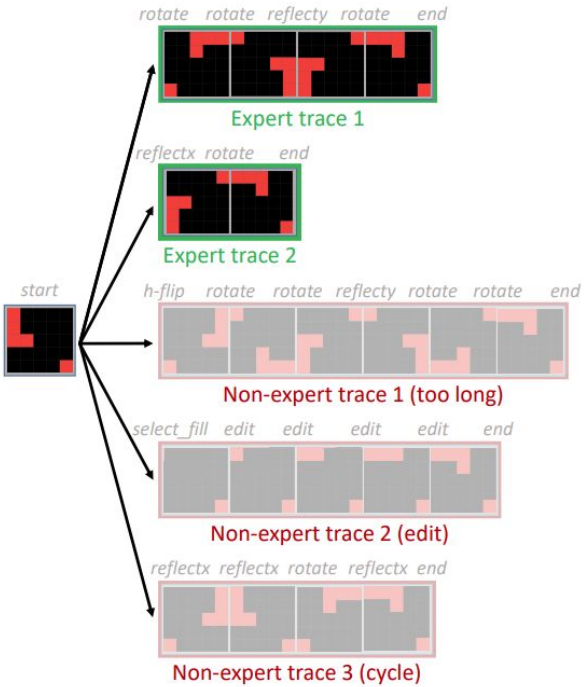
Purpose of Research Project

Why GFlowNet?

- **다양한 솔루션 탐색**: ARC는 여러 가지 가능한 해결 방식을 요구하며, GFlowNet은 다양한 확률적 경로를 탐색하고 높은 보상을 받는 경로를 학습할 수 있음.
- **인간 사고 모방**: GFlowNet은 step-by-step reasoning과 확률적 탐색을 통해 인간의 사고 방식을 흉내 내며, ARC와 같은 복잡한 문제 해결에 적합.
- **Sparse Reward 문제 해결**: GFlowNet은 보상 분포를 학습하여 ARC의 희소한 보상 환경에서도 효과적으로 솔루션을 탐색 가능.
- **효율적인 데이터 증강 및 학습**: 자동으로 다양한 경로를 생성하고 학습하여, 적은 데이터로도 일반화 성능을 향상.



Method 1: Reward Modeling



O2ARC [23] 툴로 수집한 사람의 풀이를 분석

- Expert Solution일 수록 짧음
- 반복이 없음

다양한 Solution이 존재!

[10] Park et al., Unraveling the ARC Puzzle: Mimicking Human Solutions with Object-Centric Decision Transformer, ICML Workshop, 2023

Method 1: Reward Modeling

- 방법 1) Cycle을 탐지해서 페널티
  - Cycle 탐지는 반복된 경로로 인한 학습 비효율성을 줄임
- 방법 2) 길이에 대한 규제항을 objective에 추가
  - 길이에 대한 규제는 짧고 간결한 경로 탐색을 유도
- 방법 3) Discount Factor 적용
  - Discount Factor는 미래 보상을 고려한 균형 학습 제공

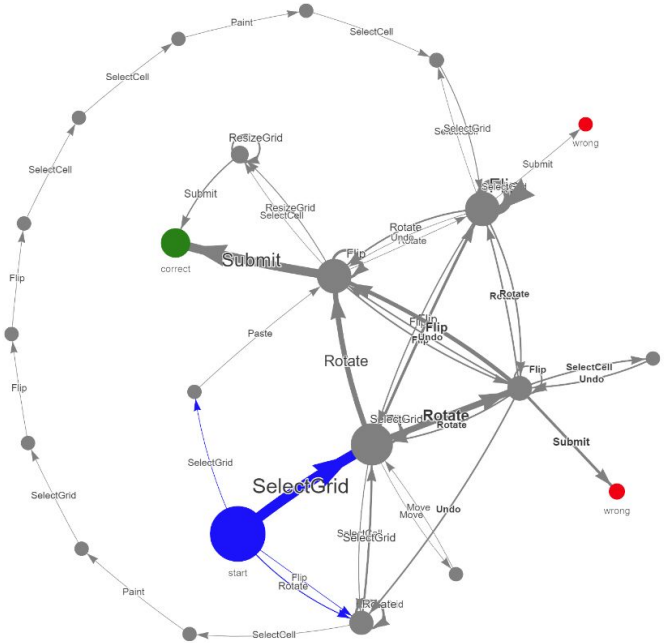


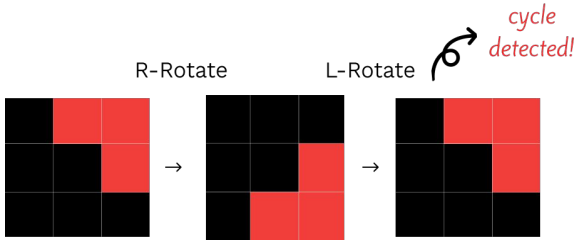
Figure 3.1: Trajectory Example

Method 1: Reward and Objective Function Modeling

**기존 Base Sparse Reward : 정답: 15, 오답: 0

- 방법 1) Cycle을 탐지해서 페널티

$$R(\tau) = \begin{cases} r(S_T) - \lambda \cdot C(\tau) & \text{if a cycle is detected,} \\ r(S_T) & \text{otherwise.} \end{cases}$$

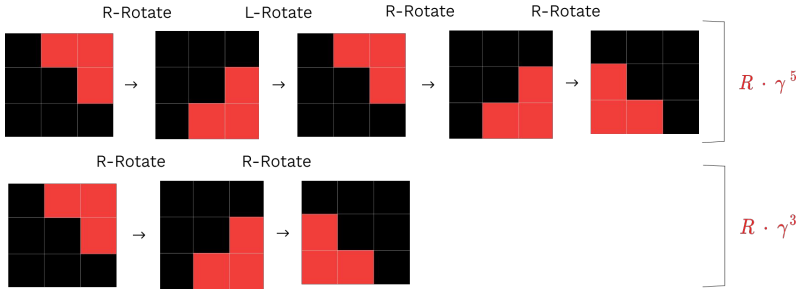


- 방법 2) 길이에 대한 규제항을 objective에 추가

$$L_{\text{tjr}} = \lambda_{\text{reg}} \cdot (L_\tau - L_{\text{target}})^2 \text{ , } L(\theta) = (1-\alpha) \cdot L_{\text{TB}} + \alpha \cdot L_{\text{tjr}}$$

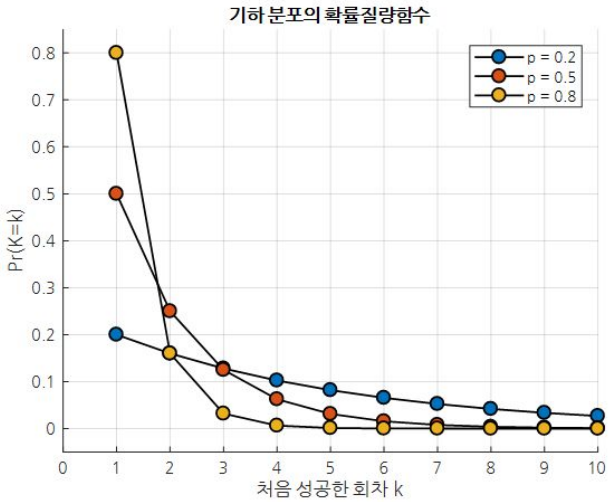
- 방법 3) Discount Factor

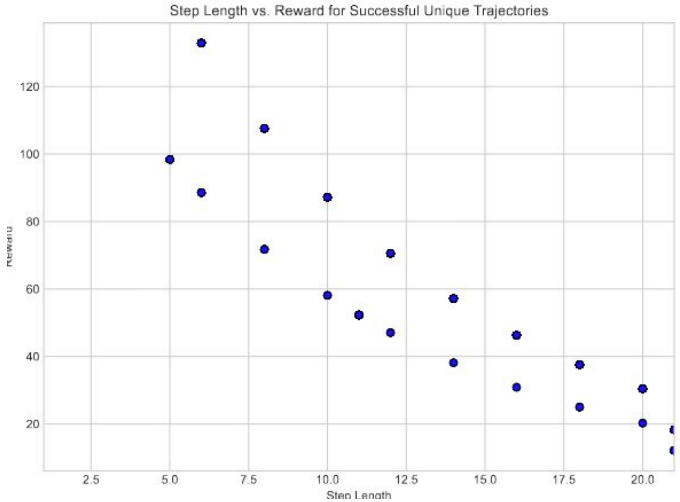
$$R_{\text{discount}} = R \cdot \gamma^t$$



Method 2: Geometric Distribution Action Sequence Modeling

- 설계된 리워드 함수의 분포는 기하분포의 분포를 따름
  - Step수를 기준으로 봤을 때, 길이가 길어질 수록 리워드가 적어짐



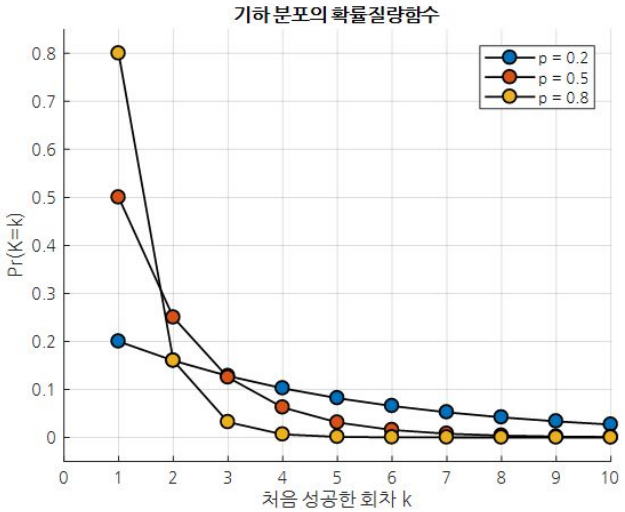


Method 2: Geometric Distribution Action Sequence Modeling

기하 분포란?

- 기하 분포(Geometric Distribution)는 성공이 처음 발생하는 시점까지의 시도 횟수를 모델링하는 확률 분포
- 확률 질량 함수(PMF): 여기서 p는 성공 확률, k는 성공이 처음 발생한 회차.

$$P(K = k) = (1 - p)^{k-1} \cdot p$$



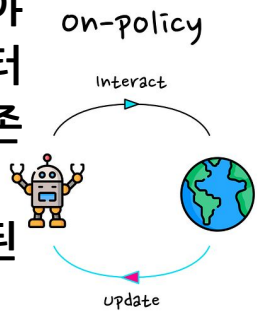
ARC 문제에서의 활용:

- 짧은 경로에 높은 보상을 부여: 성공 확률 p를 조정하여, 짧은 경로일수록 높은 보상을 받도록 설계.
- 긴 경로의 탐색 억제: 긴 경로는 낮은 확률로 모델링되어, 학습 효율성을 높임.

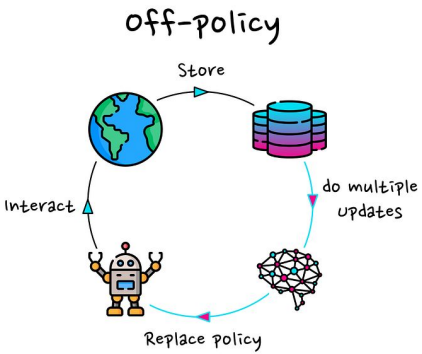
Method 3: Off-Policy Training Mechanism

On-policy VS Off-policy

- (On-policy) 학습 중 현재 정책(Current Policy)에 따라 생성된 데이터를 바로 사용하여 학습.
- 정책이 변화할 때마다 새로운 데이터를 생성해야 하므로, 데이터 효율성이 낮고 초기 학습의 데이터 품질에 크게 의존



- (Off-Policy) 다른 정책(Behavior Policy)으로 생성된 데이터를 재활용하여 학습
- Replay Buffer를 사용해 과거 데이터를 활용할 수 있어 데이터 효율성이 높음.



Method 3: Off-Policy Training Mechanism

On-policy로 학습 했을 때, GFlowNet

- 샘플된 에피소드를 바로 학습에 활용하기 때문에, 발견된 샘플에 따라 성능이 매우 달라짐
- 같은 세팅에 다른 Seed이더라도 성능이 매우 달라짐
- TB Loss의 특성상 더 문제는 심화 됨

$$Z_\theta \prod_{t=1}^{n} P_F(s_t|s_{t-1}) = R(x) \prod_{t=1}^{n} P_B(s_{t-1}|s_t)$$

$$\mathcal{L}_{\mathrm{TB}}(\theta) = \left(\log Z_\theta + \sum_{t=1}^{n} \log P_F(s_t|s_{t-1}) - \log R(x) - \sum_{t=1}^{n} \log P_B(s_{t-1}|s_t) \right)^2$$

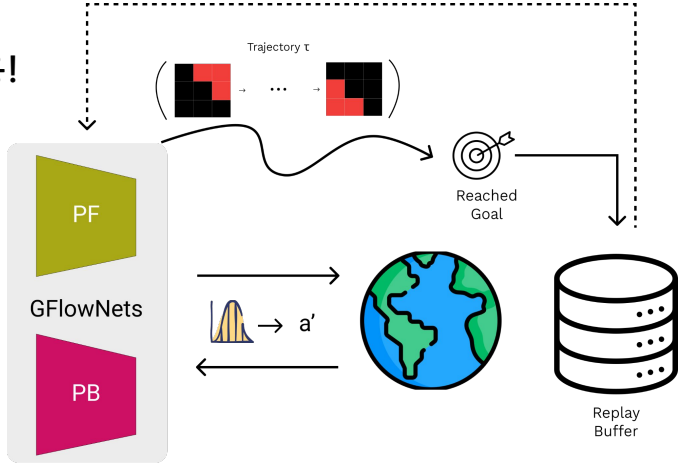
Method 3: Off-Policy Training Mechanism

Off-policy로 학습 했을 때, GFlowNet

- Replay Buffer에 좋은 경험을 수집해서 학습에 활용할 수 있음
- Offline Data 역시 replay buffer에 미리 넣어놓고 가이드하는 용도로 사용 가능
- 학습의 안정성을 높이기 위해서 GFlowNet 기존 Task에서도 많이 활용!

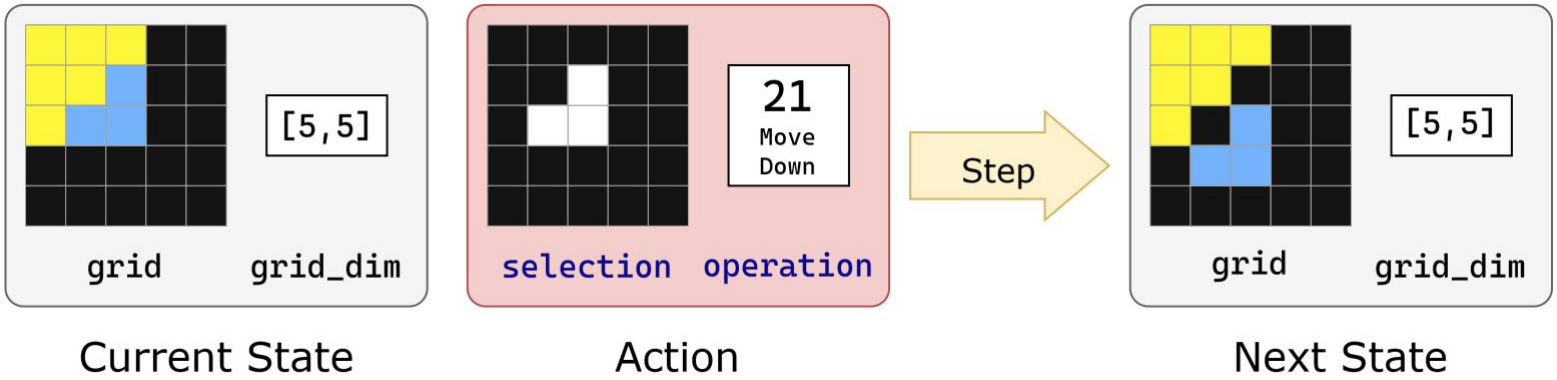
샘플링 방법

- Priotized Replay: 리워드의 분포에서 높은 리워드 위주로 샘플링
- Epsilon Greedy: 리워드 그룹을 나누어 높은 리워드 그룹에서 대부분 샘플링하고 엡실론 확률로 낮은 리워드 그룹에서 샘플링
- Fixed Ratio: 높은 리워드 0.8, 낮은리워드 0.2 확률로 샘플링



Experiments Setting: Environment - ARCLE

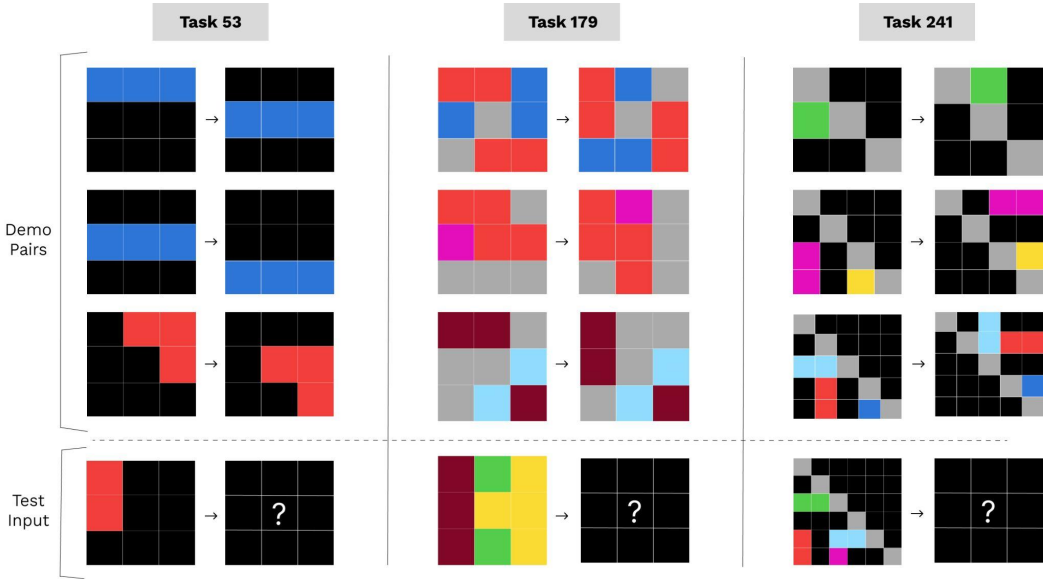
- ARCLE [13]을 활용
  - 강화학습용 ARC 환경
  - Transition을 해주기 때문에 agent는 action과 selection만 결정하면 됨



[22] Lee et al., ARCLE: The Abstraction and Reasoning Corpus Learning Environment for Reinforcement Learning, CoLLAs 2024

Experiments Setting: Problems

- 전체 그리드를 선택하는 문제로만 실험
  - Action Space가 방대해지기 때문 (Action 36 X selection 2^900)



Experiments Setting: Hyperparameters

- HyperParameters Setting

Hyperparameter	Value
Learning Rate (lr)	10^{-4}
Number of Actions	5
Episode Length	10
Base Reward	Correct: 15, Incorrect: 0
Replay Buffer Capacity	10000
Discount Factor (γ)	0.9
Loss Weight (α)	0.2
Trajectory Regularization Weight (λ_{reg})	0.01
Sampling Method	prioritized sampling

Table 4.1: Key Hyperparameter Settings for Experiment

Experiments Setting: Metrics

- Metric
  - Validation Accuracy (Val_ACC): 학습된 GFlowNet으로 100개의 샘플을 생성하게 한 뒤 정답에 도달한 Trajectory의 비율을 계산

$$\text{Val\_Acc} = \frac{\text{Correct}}{N} \times 100\%$$

  - Trajectory Diversity (D_traj): 생성된 trajectory가 다양한지를 평가하기 위해 Unique한 trajectory의 개수와 성공 비율을 계산

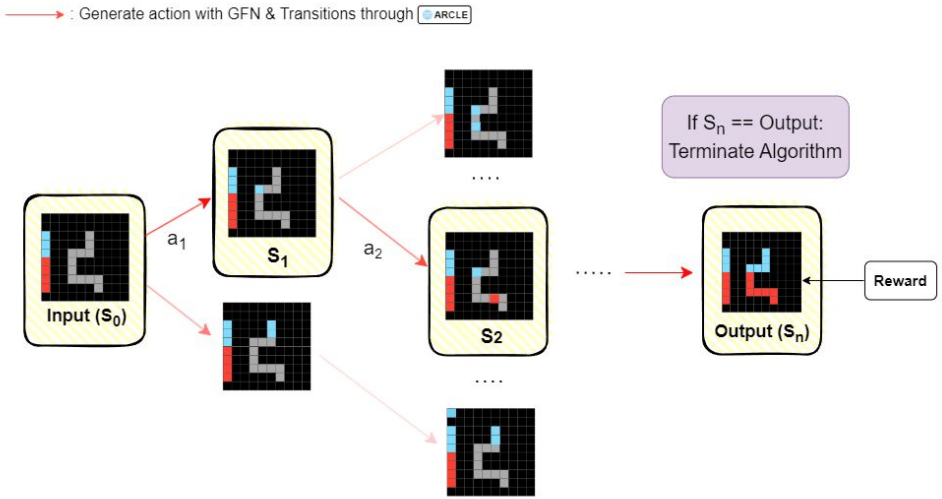
$$D_{\text{traj}} = \frac{|T_{\text{goal}}|}{|T_{\text{unique}}|}$$

  - Reward Distribution Diversity (D_reward): Shannon Index를 활용하여 엔트로피 기반 리워드분포의 다양성을 평가

$$D_{\text{reward}} = H(R) = -\sum_{i} p(r_i) \log p(r_i)$$

Experiments Setting

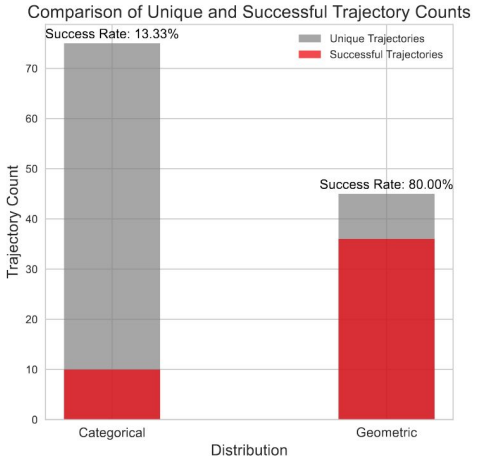
- Experiment4 (Off-policy 학습) 을 제외하고 전부 On-policy로 학습
  - 추가적인 data는 주어지지 않고 input state만 input으로 들어감
- Experiment 1~3는 Task 179번으로 수행



Experiments 1: Geometric Dist. VS Categorical Dist.

RQ1 : How does the type of action distribution impact the model's ability to find correct solutions?

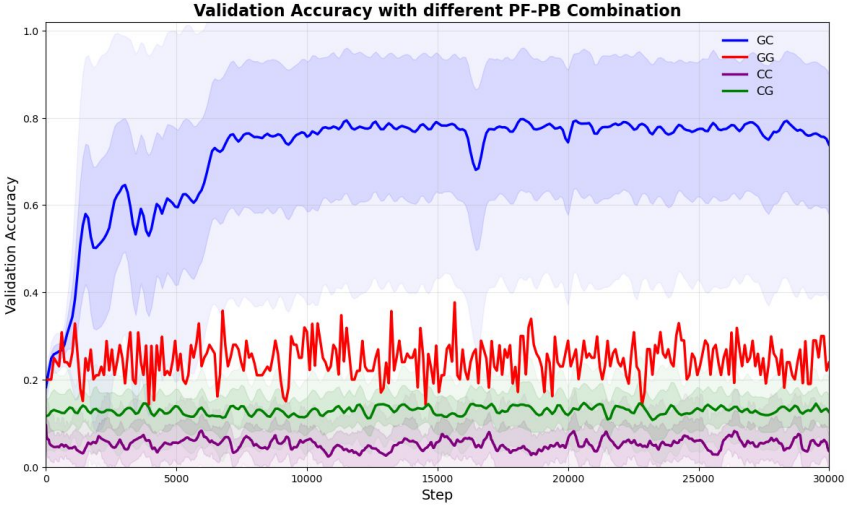
Figure 4.2: (Left) **Validation accuracy** comparison: all hyperparameters were kept constant, except for the action distribution, which was modeled using either a Categorical (green) or Geometric (blue) distribution. (Right) **Reward comparison**: both action distributions were compared in terms of reward values over training steps, showcasing the differences between Categorical and Geometric distributions.

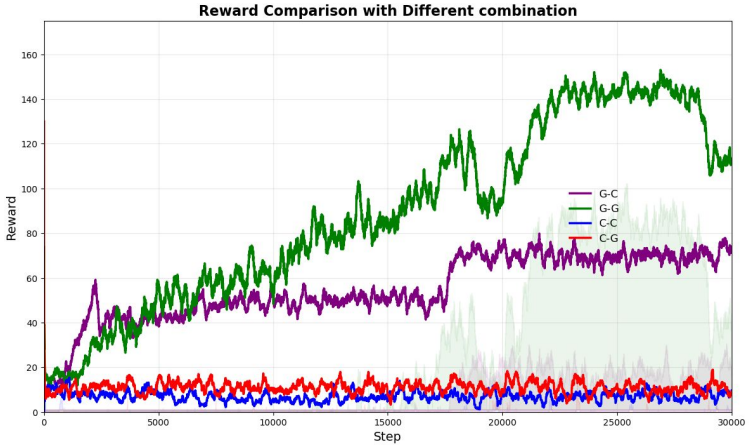


Experiments 2: Geometric & Categorical Combination

RQ2 : Why does learning occur effectively only when using the Geometric distribution for PF?

- Geometric - Categorical (GC)
- Geometric - Geometric (GG)
- Categorical - Categorical (CC)
- Categorical - Geometric (CG)





Experiments 2: Geometric & Categorical Combination

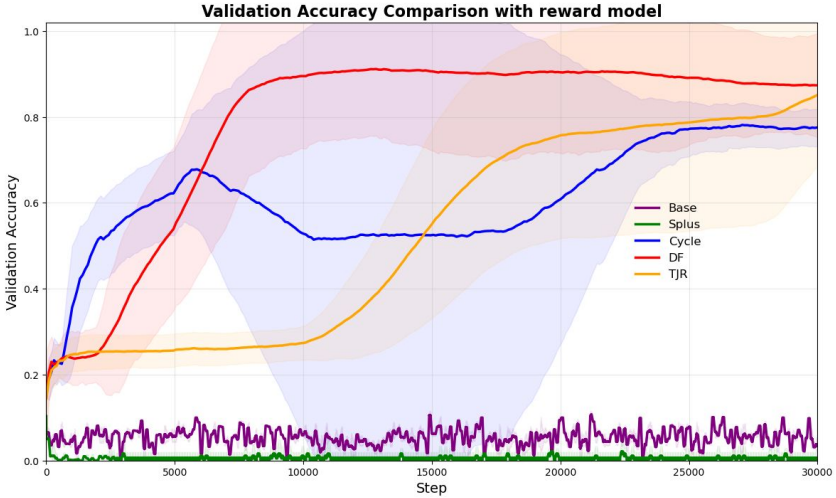
RQ2 : Why does learning occur effectively only when using the Geometric distribution for PF?

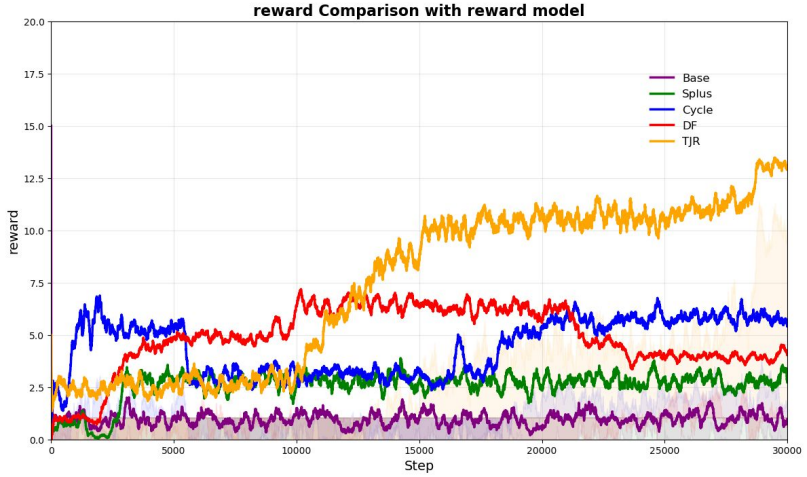
- Geometric - Categorical (GC)
- Geometric - Geometric (GG)
- Categorical - Categorical (CC)
- Categorical - Geometric (CG)

Experiments 3: Effectiveness Reward Modeling

RQ3: What is the effect of different reward models on GFlowNet's learning performance?

- Base: Correct - 15, Wrong - 0
- Splus: Submit Correct - 15, Correct - 10, wrong - 0
- Cycle: Cycle Penalty + Splus Reward
- DF: Discount Factor + Splus Reward
- TJR: Trajectory Regularization Objective + Splus Reward

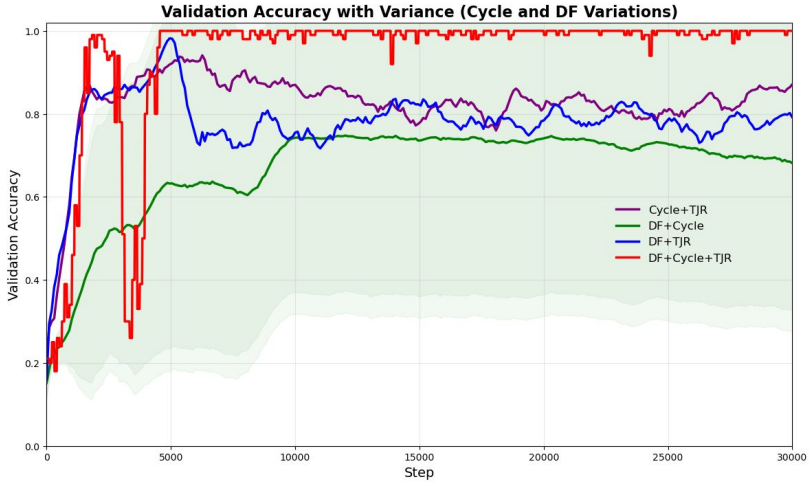




Experiments 3: Effectiveness Reward Modeling

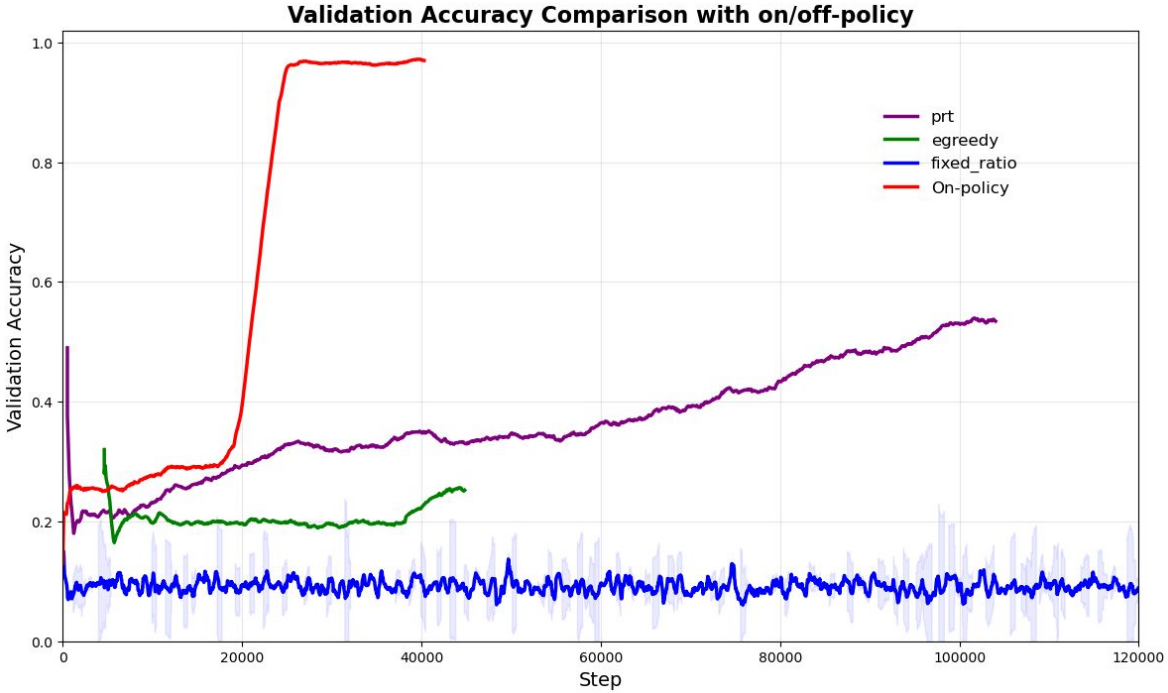
RQ3: What is the effect of different reward models on GFlowNet's learning performance?

Cycle, DF, TJR 조합



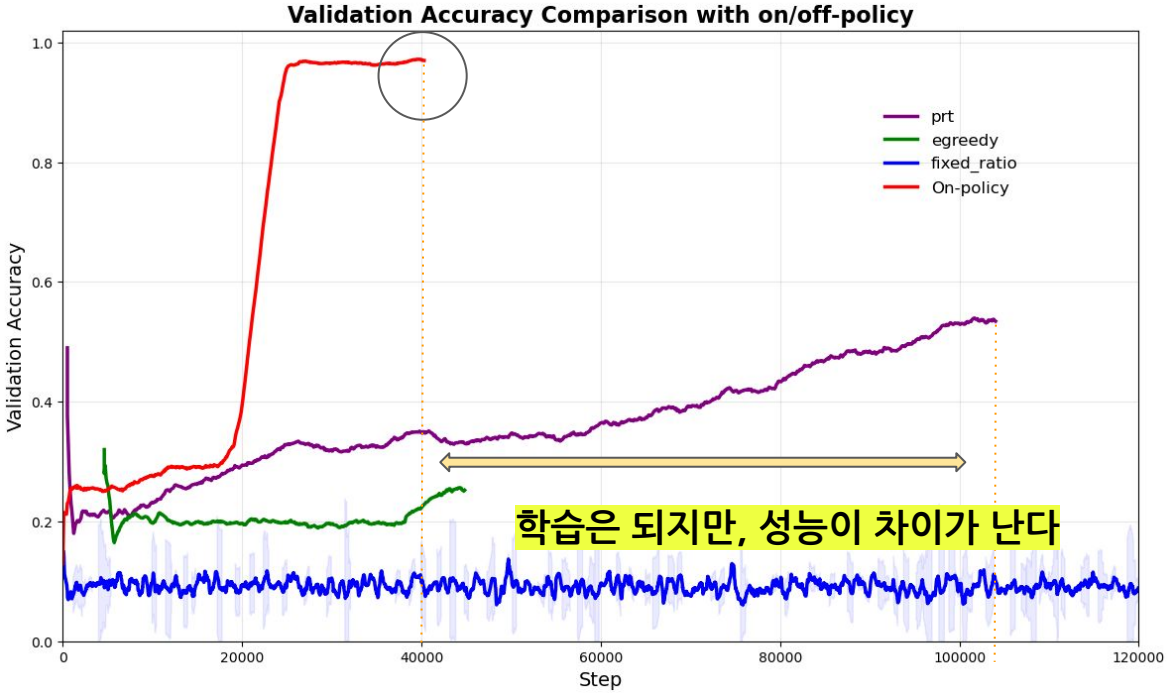
Experiments 4: Off policy Training

RQ4: How can off-policy training reduce model variance and improve stability?



Experiments 4: Off policy Training

RQ4: How can off-policy training reduce model variance and improve stability?



Result: Generated Trajectories on Task 179

Result: Generated Trajectories on other Task

Ablation Study: Reward Scale

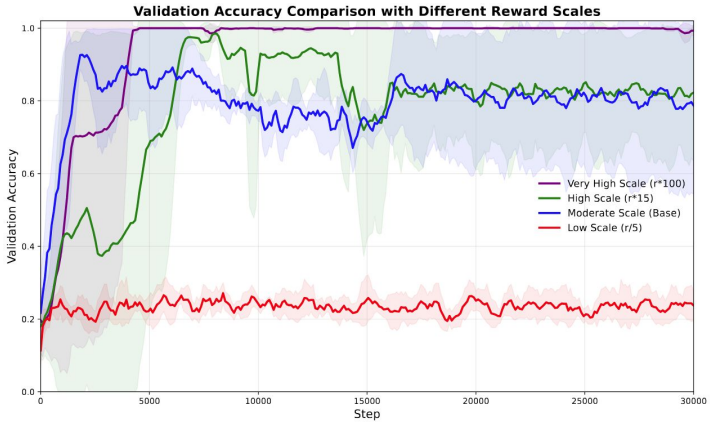


Figure 4.6: Performance comparison across different reward scales, showing the impact of reward values on validation accuracy over training steps.

Reward Scale	Unique Trajectories	Successful Unique Trajectories	Success Rate (Unique, %)	Reward Distribution Diversity (D_reward)
Low Scale (r/5)	100	0	0.00	0.5139
Moderate Scale (Base)	56	40	71.43	0.8817
High Scale (r*15)	74	53	71.62	0.6739
Very High Scale (r*100)	1	1	100.00	0.0

Table 4.4: Success rate and reward distribution diversity based on unique trajectories across different reward scales.

Ablation Study: Reward Scale

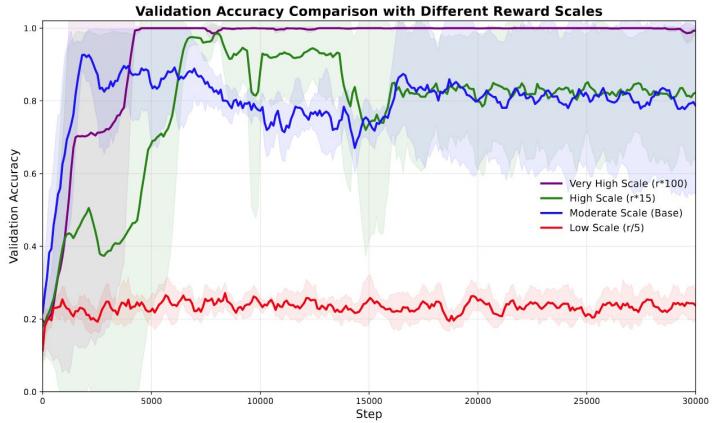


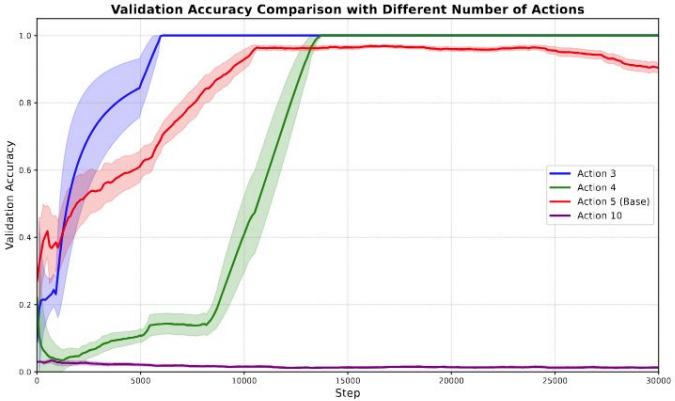
Figure 4.6: Performance comparison across different reward scales, showing the impact of reward values on validation accuracy over training steps.

Reward Scale	Unique Trajectories	Successful Unique Trajectories	Success Rate (Unique, %)	Reward Distribution Diversity (D_reward)
Low Scale (r/5)	100	0	0.00	0.5139
Moderate Scale (Base)	56	40	71.43	0.8817
High Scale (r*15)	74	53	71.62	0.6739
Very High Scale (r*100)	1	1	100.00	0.0

Table 4.4: S**Reward Scale이 커지면 성능에 긍정적 영향**unique trajectories across different reward scales.

Ablation Study: Number of Actions

- Action 3
- Action 4
- Action 5 (Base)
- Action 10



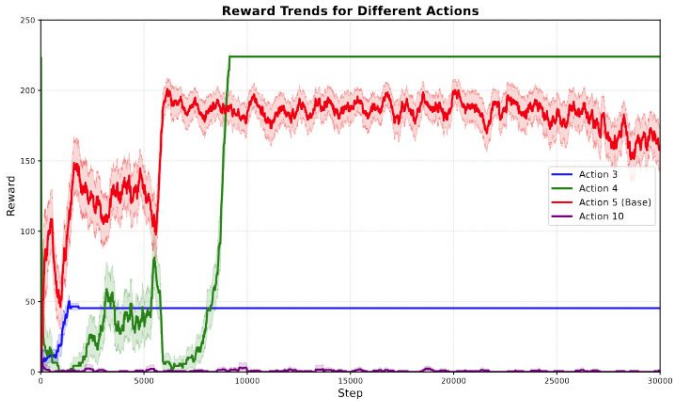


Figure 4.7: (Left) Performance comparison across number of actions, showing the impact of action count on validation accuracy over training steps. (Right) Reward comparison: comparison of reward values over training steps for each action configuration, highlighting differences in reward trends.

Ablation Study: Number of Actions

- Action 3
- Action 4
- Action 5 (Base)
- Action 10

File	Unique Trajectories	Successful Unique Trajectories	Success Rate (Unique, %)	Total Trajectories	Reward Distribution Diversity (D_reward)
a3	1	1	100.00	100	0.0
a4	1	1	100.00	100	0.0
a5 (Base)	41	26	63.41	100	0.6739
a10	95	11	11.58	100	0.4439

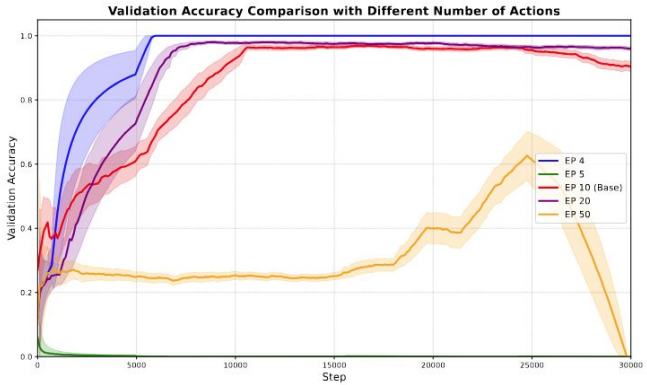
Table 4.6: Summary of unique and successful trajectories, success rates, total trajectories, and reward distribution diversity for each action configuration.

File	Successful Total Trajectories	Success Rate (Total, %)	Reward Distribution Diversity (D_reward)	Comments
a3	100	100.0	0.0	Minimal exploration with consistent success
a4	100	100.0	0.0	Limited diversity but high success rate
a5 (Base)	81	81.0	0.6739	Balanced exploration and success
a10	11	11.0	0.4439	High diversity, low success rate

Table 4.7: Total success rates, reward distribution diversity, and additional comments on trajectory diversity and performance for each action configuration.

Action space가 커지면 성능도 떨어진다

Ablation Study: Episode Length



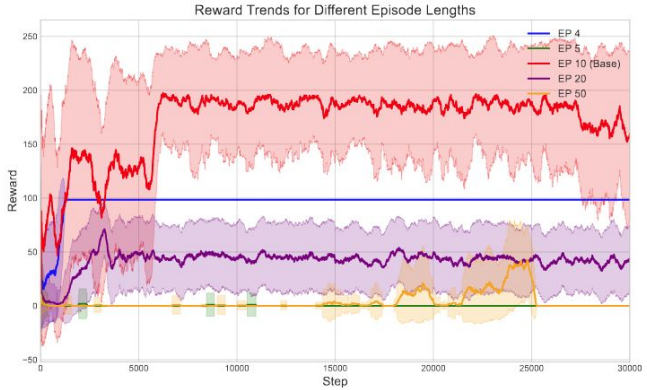


Figure 4.8: (Left) Performance comparison across episode length, showing the impact of action count on validation accuracy over training steps. (Right) Reward comparison: comparison of reward values over training steps, highlighting differences in reward trends.

Ablation Study: Episode Length

File	Unique Trajectories	Successful Unique Trajectories	Success Rate (Unique, %)	Reward Distribution Diversity (D_reward)
Episode Length 4	1	1	100.00	0.0
Episode Length 5	76	4	5.26	0.20
Episode Length 10	56	40	71.43	0.88
Episode Length 20	74	71	95.95	2.14
Episode Length 50	1	0	0.00	0.0

Table 4.8: Trajectory success rates and reward distribution diversity for various episode lengths and reward settings.

File	Total Trajectories	Successful Total Trajectories	Success Rate (Total, %)
Episode Length 4	100	100	100.0
Episode Length 5	100	5	5.0
Episode Length 10	100	59	59.0
Episode Length 20	100	97	97.0
Episode Length 50	100	0	0.0

Table 4.9: Total success rates for different episode lengths and reward configurations.

State space도 커지면 성능이 떨어진다

Limitation

- 제한된 Selection Operation
  - 현재 연구는 ARC 문제의 특정 카테고리(Selection Operation)에 초점.
  - 다양한 ARC 문제를 해결하기 위해 추가적인 범용성 검증 및 연구가 필요
- Action Space의 확장성 부족
  - 전체 Grid를 선택해서 풀 수 있는 문제도 action space가 방대해지면 풀기 어려워함
- Off-policy 학습의 효율성 문제
  - Onpolicy에 비해 매우 느림
- Solution Augmentation의 제한된 검증
  - 다른 강화학습 Task에서도 기하분포를 활용했을 때, 유의미한 성능을 보일 수 있는지?
  - ARC 문제와 유사하지 않은 데이터 분포에서 GFlowNet의 일반화 성능 검증 부족.

Future Work

1. Solution Augmentation 확장

- ARC 문제에서 제안한 Solution Augmentation 방식을 활용해서 ARC Solver에 학습 시켜 성능을 Test

2. 보상 함수 개선

- 현재 기하분포 기반 보상 모델을 다른 확률적 분포로 일반화

3. Action Space 최적화

- Action Space의 크기를 줄이기 위한 효율적인 탐색 방법 설계.
- Selection Operation을 최적화하여 현재 GFlowNet의 계산 비용을 낮추고 복잡성을 줄이는 방향 탐색.

4. Off-policy 학습 효율성 개선

- Off-policy 학습에서 효율성을 높이기 위한 새로운 Replay Buffer 전략 제안.
- Off-policy 학습이 느린 문제를 해결하기 위한 샘플 효율성 개선 연구.

5. ARC 문제의 다양한 카테고리로 확장

- 현재 연구는 특정 Selection Operation 카테고리에 집중했으므로, 다른 ARC 카테고리 문제에도 GFlowNet 적용.
- 다양한 데이터 분포에서 GFlowNet의 일반화 성능 평가.

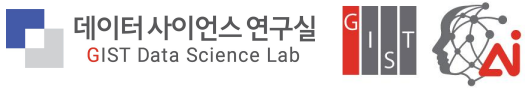
Conclusion

1. 사람의 분포를 분석하여 설계된 리워드 분포를 기하분포를 활용하여 효과적으로 모델링하고, 기존 모델 대비 다양한 솔루션의 증강에 성공함
2. 하지만, 직접 다른 Solver에 적용된 바가 없으며, 추가적인 Task와 Selection과 방대한 action space를 control 할 수 있는 추가적인 방법이 요구됨
3. 또한, 학습의 안정성을 높이기 위해 Off policy 방법의 학습 효율성과 안정성을 동시에 높일 수 있는 방법을 모색해봐야함

Thank you!

Sanha Hwang
MS Candidate @ GIST AIGS

hsh6449j@gm.gist.ac.kr



Reference

[1] F. chollet, On The Measure of Intelligence, 2019
[2] Lab42, https://lab42.global/arc/, 2024
[3] Mitchell et al, Comparing Humans, GPT-4, and GPT-4V On Abstraction and Reasoning Tasks, 2023
[4] Natasha et al, CodeIt: Abstract Reasoning with Iterative Policy- Guided Program Synthesis, 2024
[5] Camposampiero et al, Abstract Visual Reasoning Enabled by Language, 2023
[6] Simon et al, Neural-guided, Bidirectional Program Search for Abstraction and Reasoning, 2021
[7] Veldcamp et al. , Solving ARC Visual Analogies with Neural Embeddings and Vector Arithmetic: A Generalized Method, 2023
[8] ferre et al , Tackling the Abstraction and Reasoning Corpus (ARC) with Object-centric Models and the MDL Principle, 2023
[9] Kaggle, Abstraction and Reasoning Challenge competition, 2021
[10] Park et al., Unraveling the ARC Puzzle: Mimicking Human Solutions with Object-Centric Decision Transformer, ICML Workshop, 2023
[11] Yoshua Bengio, https://yoshuabengio.org/2022/03/05/generative-flow-networks/, 2022
[12] E. Bengio et al, flow network based generative models for non-iterative diverse candidate generation, 2021
[13] T. Shen et al., TacoGFN: Target-conditioned GFlowNet for Structure-based Drug Design, TMLR, 2024
[14] G.Lee et al., Geometric-informed GFlowNets for Structure-Based Drug Design, MOML, 2024
[15] M.Cretu et al., SynFlowNet: Towards Molecule Design with Guaranteed Synthesis Pathways, ICLR Workshop 2024
[16] E.Soares et al., A Framework for Toxic PFAS Replacement based on GFlowNet and Chemical Foundation Model, NeurIP Workshop 2023
[17] M.Koziarski, RGFN: Synthesizable Molecular Generation Using GFlowNets, NeurIPS 2024
[18] D. Zhang et al., Let the Flows Tell: Solving Graph Combinatorial Optimization Problems with GFlowNets, NeurIPS 2023
[19] Kim et al., Ant Colony Sampling with GFlowNets for Combinatorial Optimization, arXiv preprint arXiv:2403.07041, 2024
[20] Moksh Jain, et al., Biological Sequence Design with GFlowNets, ICML, 2022
[21] N. Malkin et al., Trajectory balance: Improved credit assignment in GFlowNets, NeurIPS 2022
[22] Lee et al., ARCLE: The Abstraction and Reasoning Corpus Learning Environment for Reinforcement Learning, CoLLAs 2024
[23] Shim et al., O2ARC 3.0: A Platform for Solving and Creating ARC Tasks, IJCAI, 2024
[24] Ekin Akyürek et al., The Surprising Effectiveness of Test-Time Training for Abstract Reasoning, arXiv:2411.07279, 2024
[25]Jack Cole, Mohamed Osman, Michael Hodel, Keith Duggar, and Tim Scarfe. Machine learning street talk, 2024.
[26]Hodel, Michael. re-arc. GitHub, 2024, https://github.com/michaelhodel/re-arc.

Motivation : Why is Augmentation Beyond Input-Output Pairs Necessary?

데이터 증강의 중요성

- 위에서 보았듯이 데이터를 다양하게 만들어주는 데이터 증강 기법은 모델의 성능을 높이는 데 효과적
- 예를 들어, MIND AI는 Input-Output 증강과 TTT를 활용하여 ARC 문제에서 큰 성과를 거둠

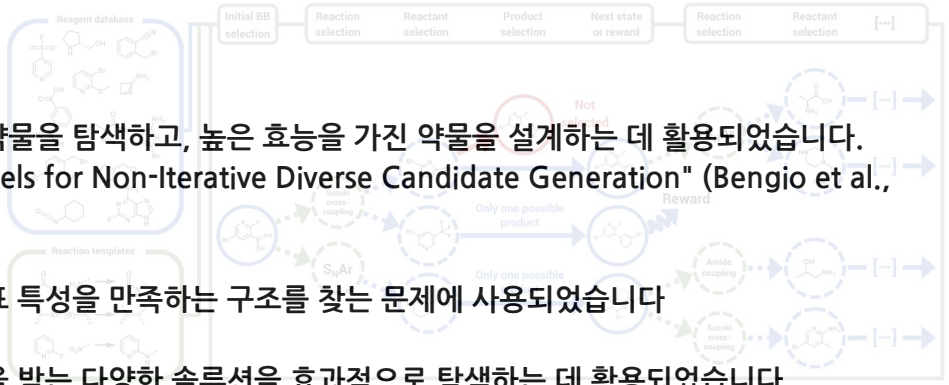
기존 데이터 증강 접근 방식의 한계

1. **Input-Output 증강**
   - MIND AI의 접근 방식은 Input-Output Pair 증강에 초점을 맞추고 있음
     - Solution 증강을 통해 좀 더 다양한 분포에 대해 알려주면 성능이 올라가지 않을까?
   - 이 방식은 강화학습에 필요한 Solution을 다양화하지 못함
     - Offline RL모델을 학습시키기 위해서는 trajectory를 추가적으로 증강해야 함
     - Input-Output 방식만으로는 trajectory가 blackbox가 되어 설명 가능성을 잃는 문제가 발생

2. **Rule-Based 증강**
   - Solution / Input-Output pair를 직접 증강하기 위해 Rule-Based 방식을 사용할 경우:
     - 시간과 노력이 과도하게 소모됨
     - 규칙을 설계하고 디버깅하는 데 많은 리소스가 필요하며, 확장성이 낮음

What is Generative Flow Networks (GFlowNet)?

기존 GFlowNet의 활용 사례



- 약물 발견(Drug Discovery):
  - GFlowNet은 화학 구조 공간에서 다양한 후보 약물을 탐색하고, 높은 효능을 가진 약물을 설계하는 데 활용되었습니다.
  - 논문: "Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation" (Bengio et al., 2021)
- 재료 과학(Material Science):
  - GFlowNet은 새로운 재료를 설계하고, 특정 목표 특성을 만족하는 구조를 찾는 문제에 사용되었습니다
- 조합 최적화(Combinatorial Optimization):
  - GFlowNet은 조합적 탐색 문제에서, 높은 보상을 받는 다양한 솔루션을 효과적으로 탐색하는 데 활용되었습니다.
  - 예: 경로 계획(Path Planning) 및 네트워크 최적화(Network Optimization).

[11] Yoshua Bengio, https://yoshuabengio.org/2022/03/05/generative-flow-networks/, 2022
[12] E. Bengio et al, flow network based generative models for non-iterative diverse candidate generation, 2021