Discovering Latent Graphs with GFlowNets for Diverse Conditional Image Generation

Bailey Trang, Parham Saremi, Alan Q. Wang, Fangrui Huang, Zahra TehraniNasab,
Amar Kumar, Tal Arbel, Li Fei-Fei, and Ehsan Adeli

(Neurips 2025)

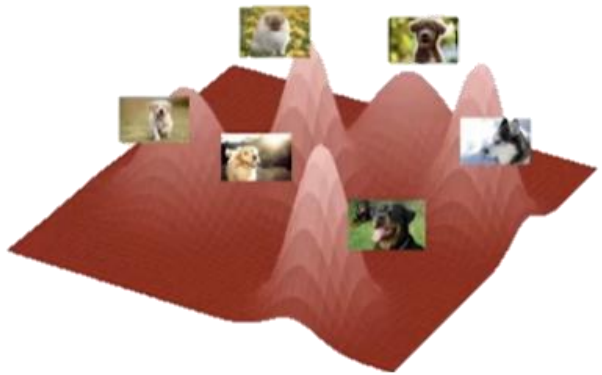
Multimodal Orchestration of Artificial Intelligence (MOAI) Lab

발표자 : 김명수

Introduction

Capturing Uncertainty in Generation

- In real-world scenarios, **input condition inherently** contain **ambiguity (uncertainty).**
  - **Text-to-image** : Same **prompt** "sunset scene" but can vary in season, lighting control, and overall ambiance.
  - **Medical** : Same **age and gender** but can vary in the structure of brain regions and their patterns.
- **Capturing uncertainty** refers to **generating diverse images** that reflect the **multiple possible interpretations** inherent **in an ambiguous condition**.
  - If a model fails to capture this uncertainty, it **cannot represent the data distribution**.



"The ability to explore multiple modes is required."

Diversify the condition through GFlowNets

- Previous approaches fail to properly capture the inherent uncertainty.
  - **Randomness based** : Change random seed or random sampling different latent.
  - **Prompt diversification** : Generate expand and enrich text prompts trough LLMs.

- **GFlowNets (Generative Flow Networks)** are designed to **capture uncertainty in tasks**.
  - **GFlowNets** can **generate diverse condition** that capture the uncertainty inherent in ambiguous conditions.

    Ex) A dog → (“A Poodle”, … , “A Husky”)

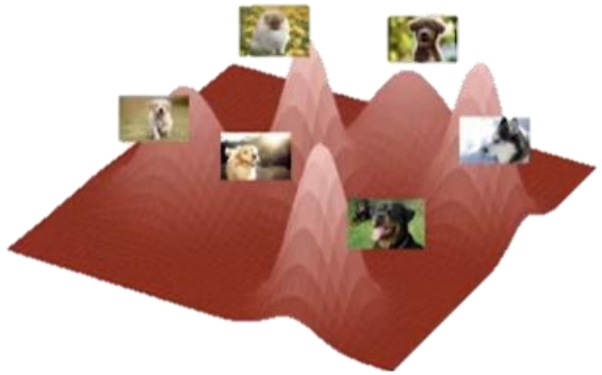
“A Dog”
(ambiguous prompt)

GFlowNets

- *“A Poodle”*
- *…*
- *“A Husky”*

(diverse condition)

Text-to-Image

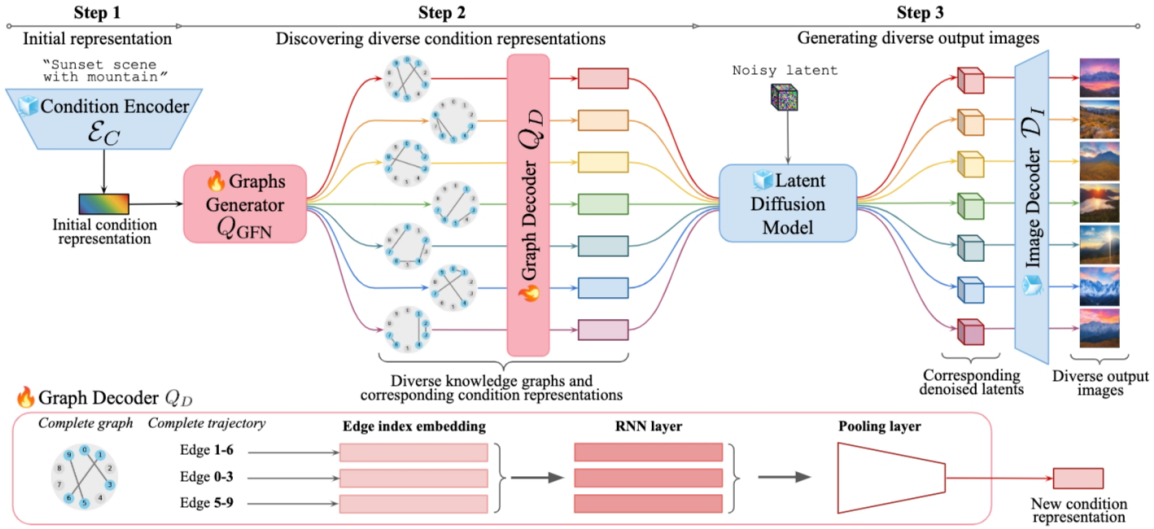


GFlowNets + Latent Diffusion Model

- In this paper, they models **uncertainty** as graph trajectories

  by **integrating GFlowNets and Latent Diffusion Models (LDMs).**

- Today, I will explain **GFlowNets** [1] and **Rainbows** (GFlowNets + LDMs) [2]



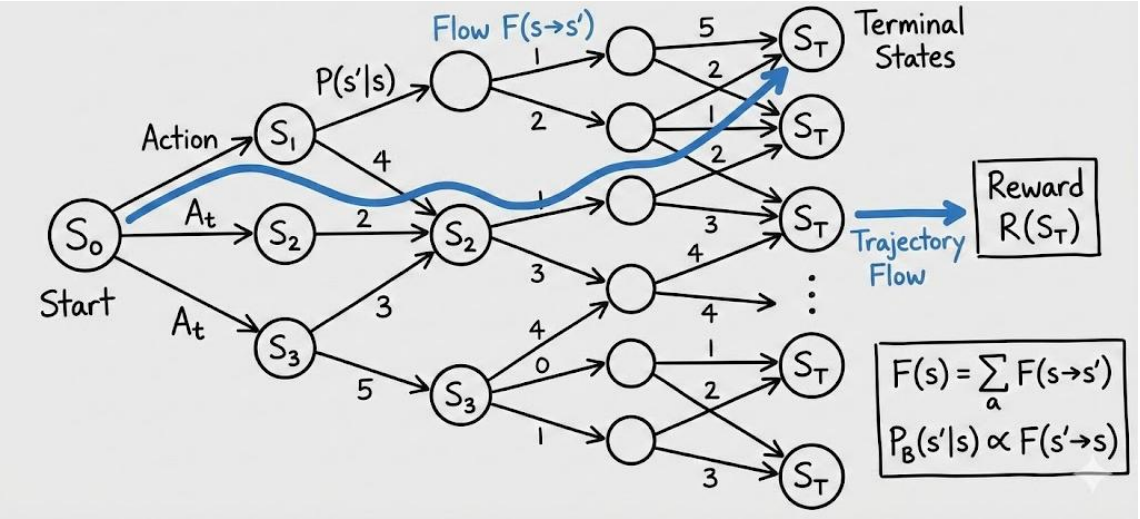
[1] The GFlowNet Tutorial, https://milayb.notion.site/The-GFlowNet-Tutorial-95434ef0e2d94c24aab90e69b30be9b3

[2] Trang, B., Saremi, P., Wang, A. Q., Huang, F., TehraniNasab, Z., Kumar, A., ... & Adeli, E. Discovering Latent Graphs with GFlowNets for Diverse Conditional Image Generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Generative Flow Networks (GFlowNets)

GFlowNets

- **Generative Flow Networks (GFlowNets)** generate samples through a **flow network** (**directed acyclic graph (DAG))** with probability proportional to a **reward function R(x).**
  - Notate dag $G = (S, A)$ where S is set of states and A is set of actions.
  - Generate **sample** s_T through **trajectory** $\tau = (s_0, \dots, s_T)$ by sampling **actions** a_i from **stochastic policy** $\pi(a_i|s_i)$.
    - Which can be thought of as **internal actions** (they are meant to construct things like thoughts or **causality** or explanations)



Flow Networks

- The flow network needs to satisfy the **two constraints** below.

  Denote nonnegative flow F (**$F(s \to s')$** : edge flow, $F(s)$: state flow)

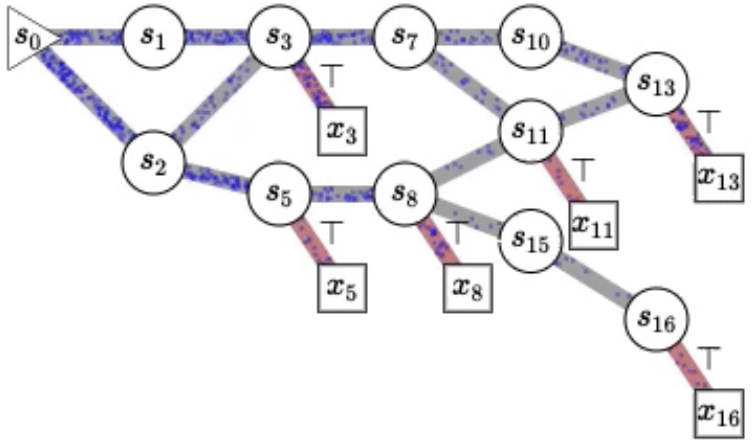
  1. For any state s, **incoming flow equals the outgoing flow**.

$$F(s) = \sum_{(s,s'') \in A} F(s'' \to s) = \sum_{(s',s) \in A} F(s' \to s)$$

  2. **Flow of final state** s_T is aligned to reward $F(s_T) \propto R(s_T)$

  - Then distribution of **generated samples** $p(s_T)$ proportional to **reward** $R(s_T)$

- **GFlowNets learn the flow** and use it to compute the policy.

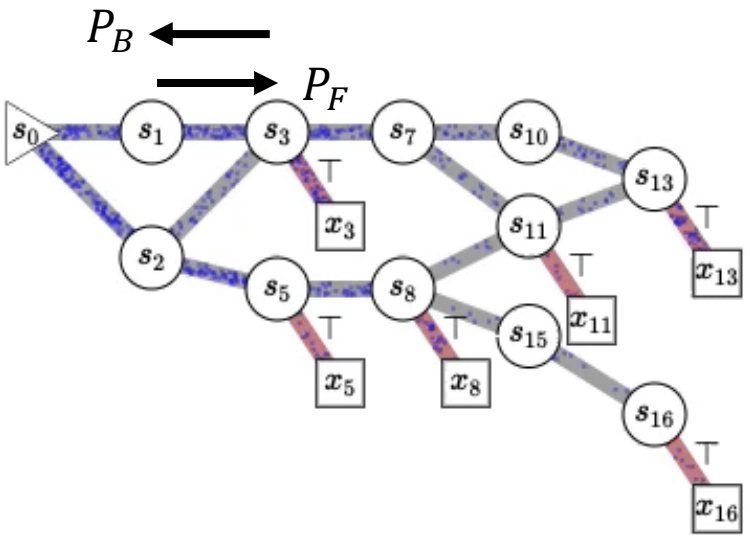


Forward policy and Backward policy

- GFlowNets consider two kind of policies
  - Forward policy $P_F(s_i|s_{i-1};\theta) = \frac{F_\theta(s_{i-1}\to s_i)}{F_\theta(s_{i-1})}$
  - Backward Policy $P_B(s_{i-1}|s_i;\theta) = \frac{F_\theta(s_i\to s_{i-1})}{F_\theta(s_i)}$
- GFlowNets' neural network learn the flows $F_\theta(s_{i-1}\to s_i), F_\theta(s_i\to s_{i-1}), F_\theta(s_i)$, and samples are **generated through the forward policy**.

$$\pi(a_t|s_t) = \pi(s_{t+1} = T(s_t, a_t)|s_t) = P_F(s_t|s_{t-1})$$

"The backward flow is a virtual flow used for training."



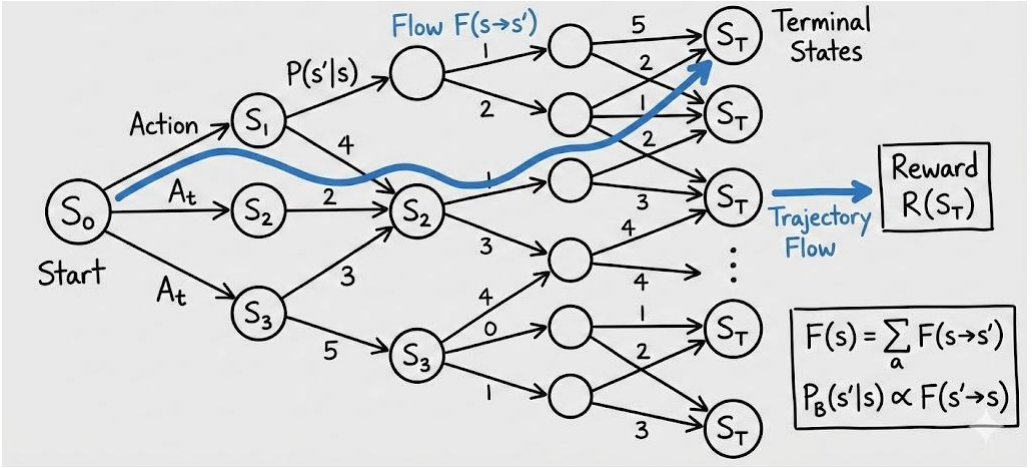
Training objective

- **Detail Balanced Objective** is optimizing the flow matching constraints

$$L_{DB}\left(s_T, R(s_T)\right) = \sum_{i\in[T-1]} \left(\log \frac{F_\theta(s_{i-1})P_F(s_i|s_{i-1};\theta)}{F_\theta(s_i)P_B(s_{i-1}|s_i;\theta)}\right)^2 + \left(\log \frac{F_\theta(s_{T-1})P_F(s_T|s_{T-1};\theta)}{R(s_T)}\right)^2$$

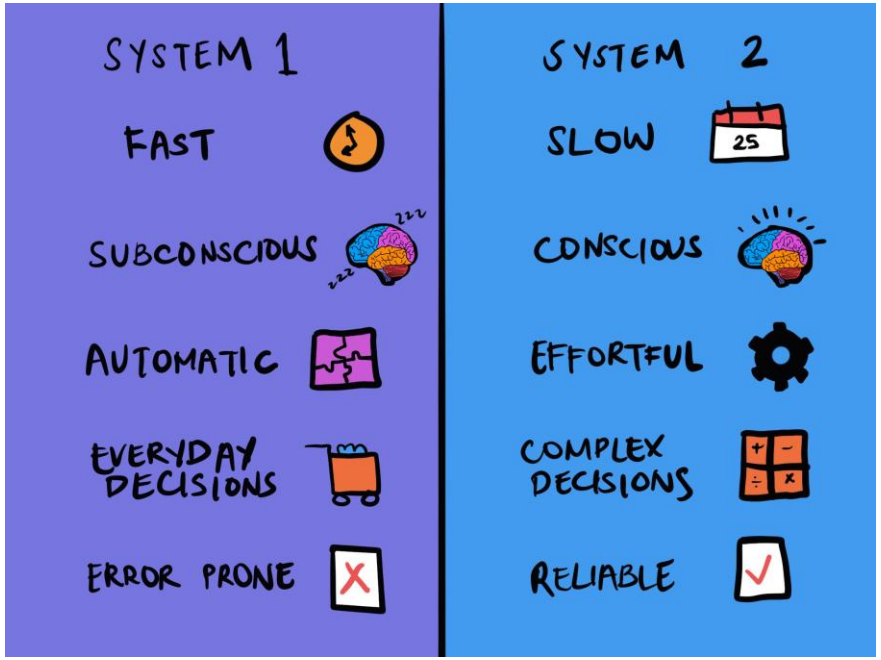
  - First term ensure the conservation of flow between s_{i-1}, s_i,
  - Second term ensure the state flow of final state s_T align to reward $R(s_T)$

- Detailed Balance Objective is mainly used to **convert a global constraint into a local constraint**.



Why GFlowNets? (1)

- **Bengio** argues **current ML is System-1** and must move **toward System-2** for human-like reasoning.
  - **System 1 thinking** : intuitive, fast, unconscious, **implicit knowledge**, …
  - **System 2 thinking** : slow, logical, sequential, conscious, **planning, reasoning**, explicit knowledge, …

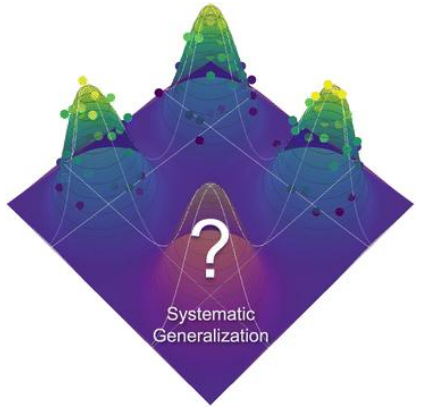


[3] Scaling in the service of reasoning & model-based ML - Yoshua Bengio (2023)
[4] Yoshua Bengio: System 2 GFlowNets: Induction Biases Towards Machines that Understand and Think - YouTube

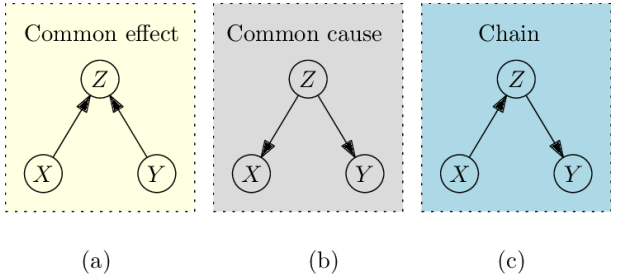
Why GFlowNets? (2)

- **Bengio** outlines the conditions necessary to approach **human-like reasoning**.
  1. **Modularity & Composability :** Reuses knowledge as modular, compositional units and forms compact abstract concepts.
  2. **Causality** : Represents the world causally, enabling intervention and counterfactual reasoning.
  3. **Uncertainty** : Assess uncertainty about a statement, and handle risks more appropriately.

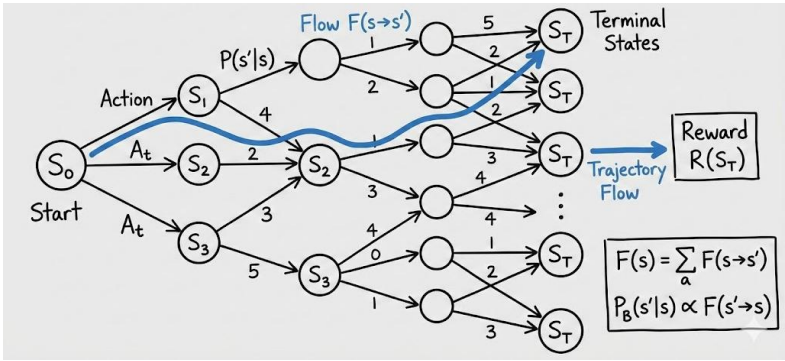
  → Generalization problem!!



Modularity



Causality

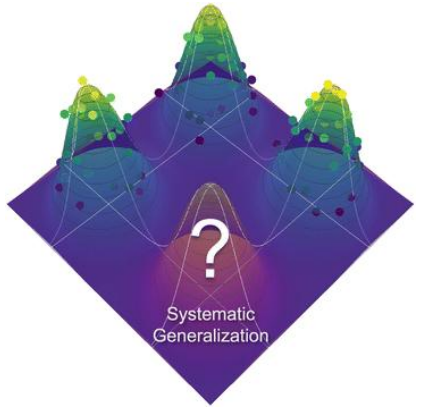


GFlowNets

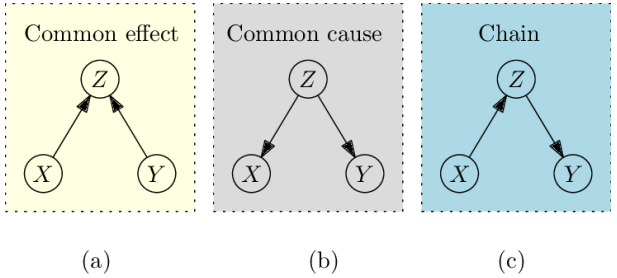
[3] Scaling in the service of reasoning & model-based ML - Yoshua Bengio (2023)
[4] Yoshua Bengio: System 2 GFlowNets: Induction Biases Towards Machines that Understand and Think - YouTube

Why GFlowNets? (3)

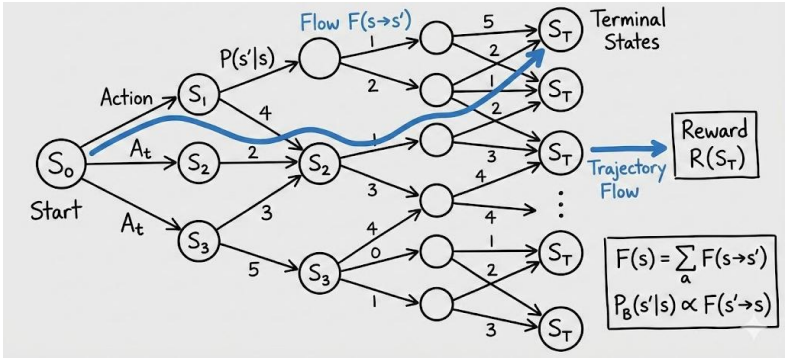
- **GFlowNets** are
  - **Operate on DAG**, so it cover **causality, modularity**.
  - **Learn reward distribution**, so it cover **uncertainty**.
    - **Monte Carlo Markov Chain (MCMC)** also guarantee the sampling from the distribution.
    - But **GFlowNets** has a structure that is **highly favorable for mode seeking**.



Modularity



Causality



GFlowNets



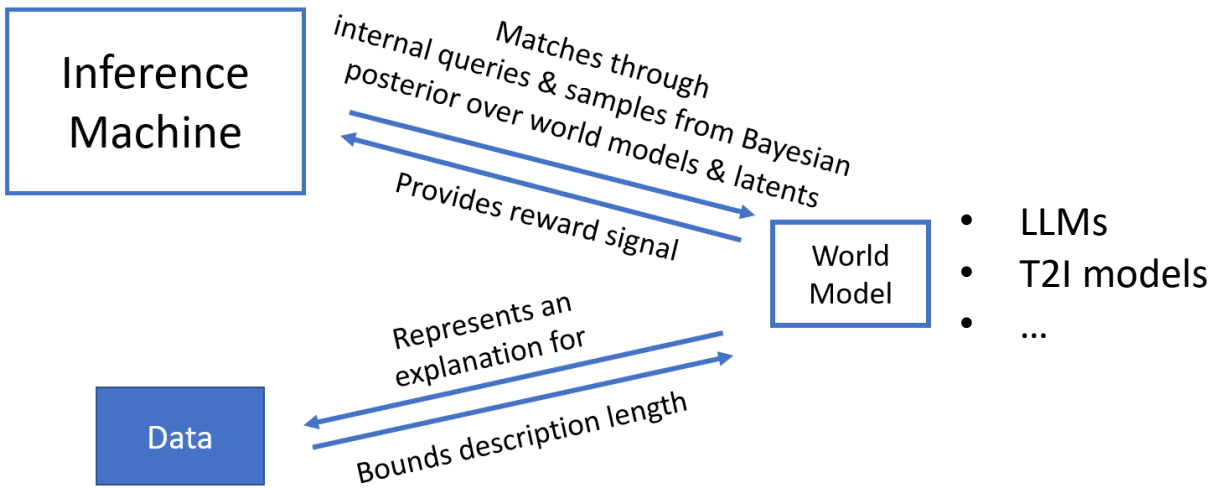
MCMC

Why GFlowNets? (4)

- Also current LLMs is that the **world model and inference machine are one and the same.**

- We should **separate the world model and the inference machine** and learn them simultaneously

- GFlowNets



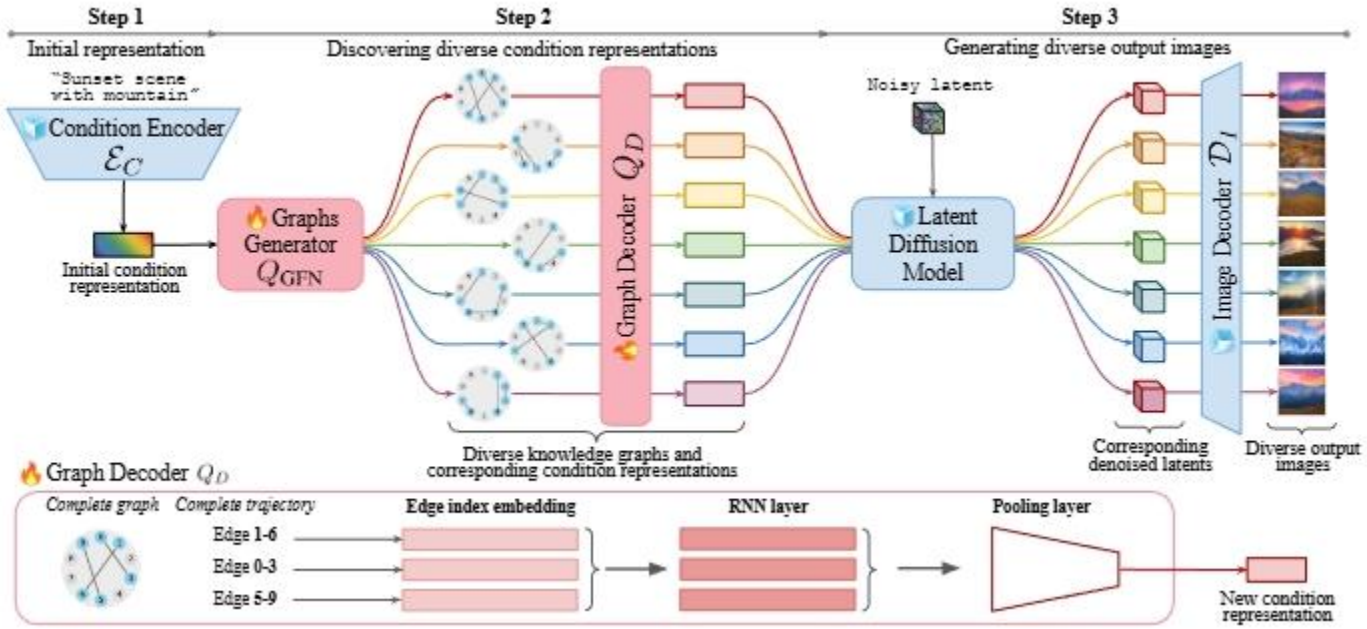
- LLMs
- T2I models
- …

[3] Scaling in the service of reasoning & model-based ML - Yoshua Bengio (2023)

Rainbow

Rainbow

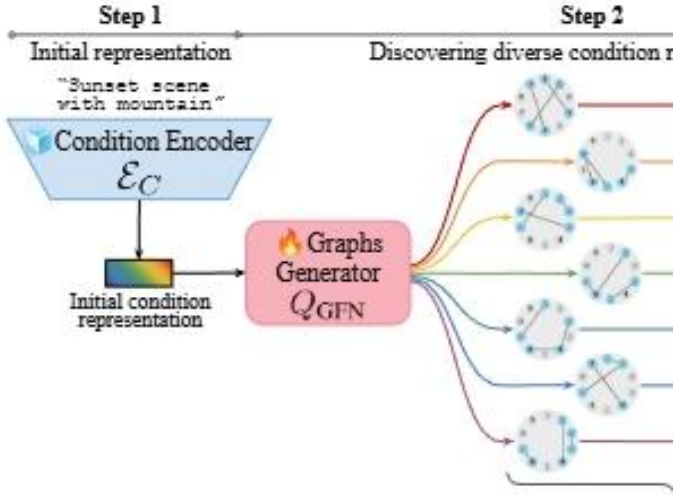
- Rainbow uses GFlowNets to **generate multiple condition** representations from a given condition, enabling an LDM to produce diverse images.
- Explain about
  - The components and detail of train of GFlowNets



[2] Trang, B., Saremi, P., Wang, A. Q., Huang, F., TehraniNasab, Z., Kumar, A., ... & Adeli, E. Discovering Latent Graphs with GFlowNets for Diverse Conditional Image Generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Stage 1: Discovering Diverse Graph Representations

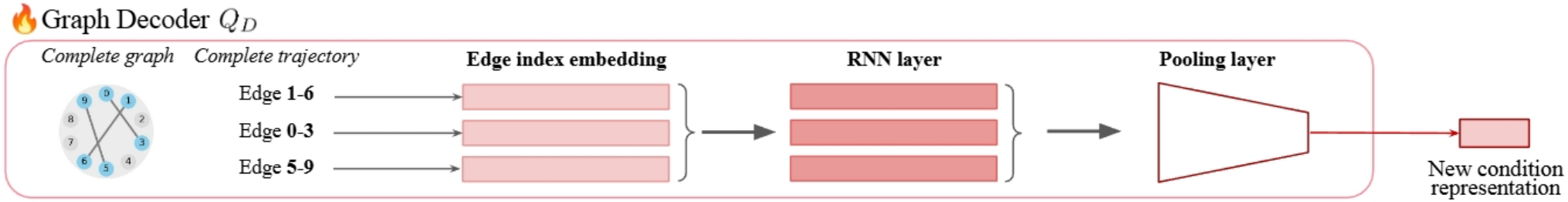
- **Setting**
  - M : number of **trajectories**, N : number of **nodes**
  - ρ : target **sparsity**, $S = (1-\rho)\frac{N(N-1)}{2}$ target number of edge.
  - Initialize M trajectories $\mathrm{T}_{s=0}^{1:M} = \{\tau_{s=0}^{1}, \tau_{s=0}^{2}, \dots, \tau_{s=0}^{M}\}$ with empty graphs.
- Iterate loop S times,
  - Computing flow $\mathrm{F}^{1:M} = Q_{GFN}\left(\mathrm{T}_{s=i-1}^{1:M}, c\right)$ and compute $P_F(s_i|s_{i-1};\theta)$ for each possible state.
  - Sampling new M edges per trajectory from $P_F(s_i|s_{i-1};\theta)$.
  - Accumulate detail balance loss : $\log\frac{F_\theta(s_{i-1})P_F(s_i|s_{i-1};\theta)}{F_\theta(s_i)P_B(s_{i-1}|s_i;\theta)}$
- Outputs : M graphs with S edges and loss.



Stage 2: Decoding Graphs into Condition Representations.

- **Decode M trajectories** $\mathrm{T}_{s=S}^{1:M}$ into **condition representation** $\hat{c}^{1:M} = Q_D\left(\mathrm{T}_{s=S}^{1:M}\right)$
  - **Using RNN** to ensure the **order correlation of edge** (selected order).
  - While training using **blended representation for stability**. $\hat{c}^{1:M} = \gamma Q_D\left(\mathrm{T}_{s=S}^{1:M}\right) + (1-\gamma)\boldsymbol{c}$

M X



Stage 3: Getting reward and computing losses.

- After generate $\hat{c}^{1:M}$, Rainbow train below loss function with **frozen Latent Diffusion Model**.

$$L_{Total} = \alpha L_{GFN} + \beta L_{LDM}$$

  - With random sampled $\epsilon \sim N(0, I)$, time step $\boldsymbol{t}$,
  - L_{LDM} : Loss function of **latent diffusion model**

$$L_{LDM} = E_{z_0 = E_I(X), \epsilon \sim N(0,I), t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c)\|^2\right]$$

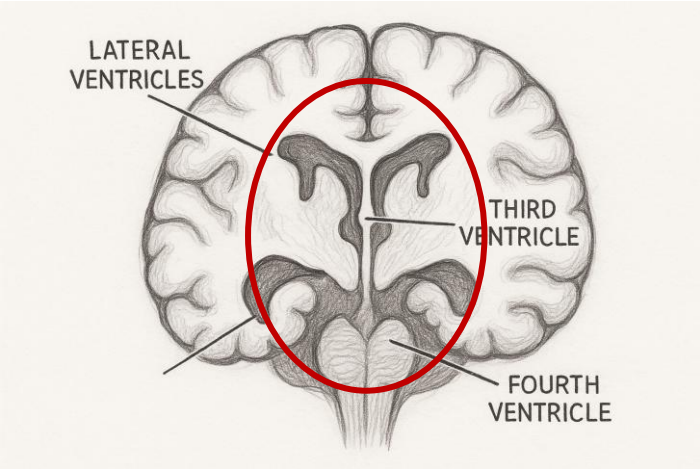
  - Rainbow set the **rewards model as** $R\left(\mathrm{T}_{\mathrm{s=S}}^{1:M}\right) = e^{-MSE(\epsilon, \hat{\epsilon}^{1:M})}$ where $\hat{\epsilon}^{1:M} = \epsilon_\theta(z_t^{1:M}, \boldsymbol{t}, \hat{c}^{1:M})$
    - It means GFlowNets will sampling conditions which **LDM generate well.**
  - L_{GFN} : Loss function of **GFlowNets**

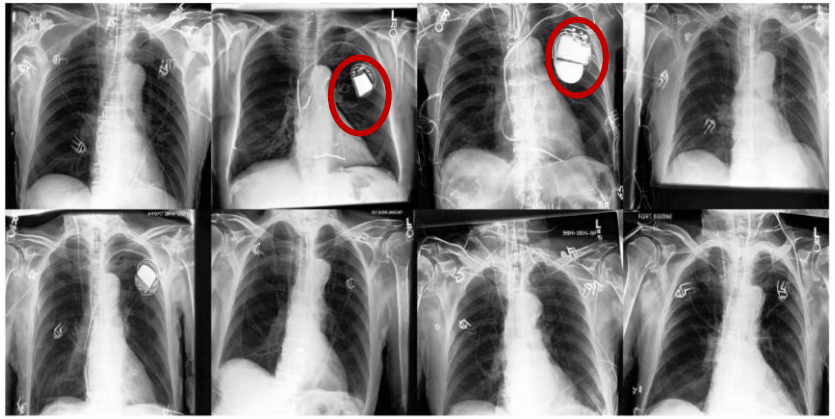
$$L_{GFN} = L_{DB}\left(\hat{\epsilon}^{1:M}, R(\hat{\epsilon}^{1:M})\right)$$

Experiments

Experiment Setup

- **Medical**
  - **3D Brain MRIs :**
    - **Size of Ventricles**: interconnected hollow spaces within the brain.
  - **Chest X-rays :**
    - **Medical devices** (such as chest tubes and wires), Diseases (such as pneumonia and pleural-effusion) and Anatomical details
    - Generate X-ray images through SD1.5 with various prompt ex) *"Chest X-ray showing **Support Devices**"*

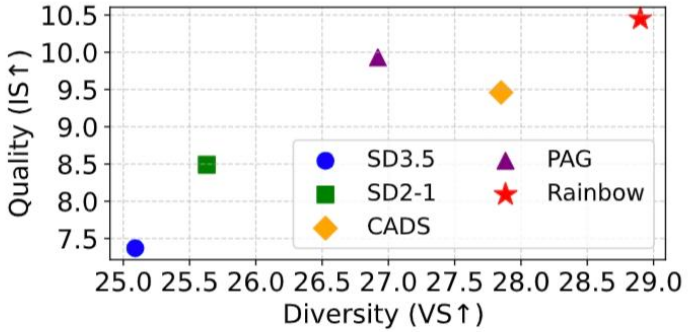




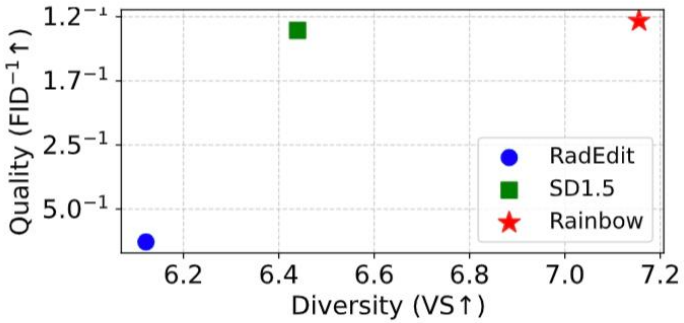
(c) Generations by Rainbow

Experiments : *Capturing Diversity*



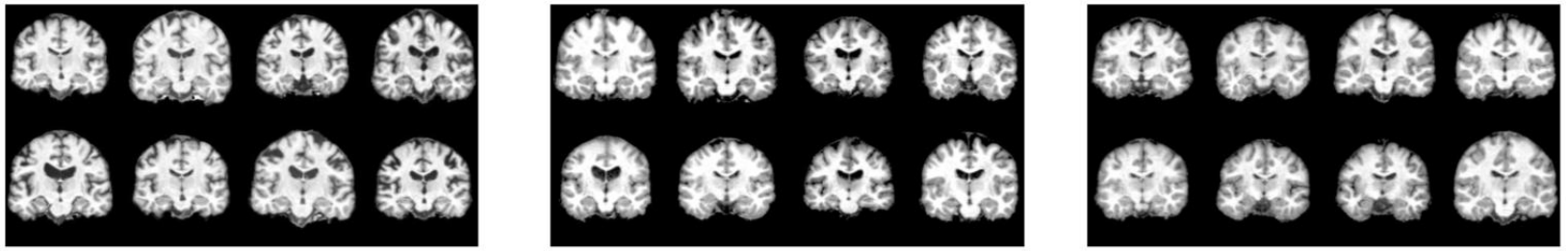


(a) Quantitative comparison on Natural Images



(b) Quantitative comparison on Chest X-rays

Experiments : *Capturing Diversity*



(a) Actual samples from dataset (b) Generations by Rainbow (c) Generations by baseline LDM

Figure 5: **Comparison of MRI image generations for *65-year-old male* individual.** Compared to actual samples from males aged 63-65, Rainbow captures greater diversity in details like ventricle sizes, while the baseline LDM generates images with less variation.



(a) Generations by RadEdit (b) Generations by SD1.5 (c) Generations by Rainbow

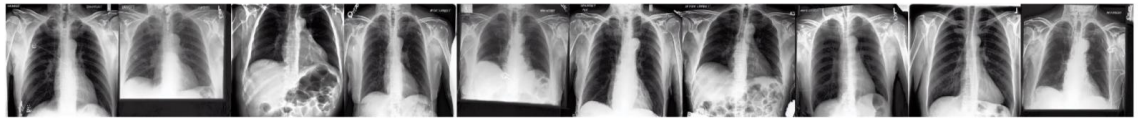
Figure 6: Comparison of chest X-ray image generations for the prompt *Chest X-ray with support devices*. Rainbow is able to provide high-quality images while generating a more diverse set of medical devices compared to the baselines - (a) and (b).

Experiments : Interpretability

- Interpret Process
  1. **Generate** much **samples and cluster** images based seasonal features.
  2. Extract the 10 **most frequently added edges**
  3. **Append that edge** into original trajectories and **generate new image**.



(a) "Sunset scene with mountain" + **Spring** edges



(a) Base prompt

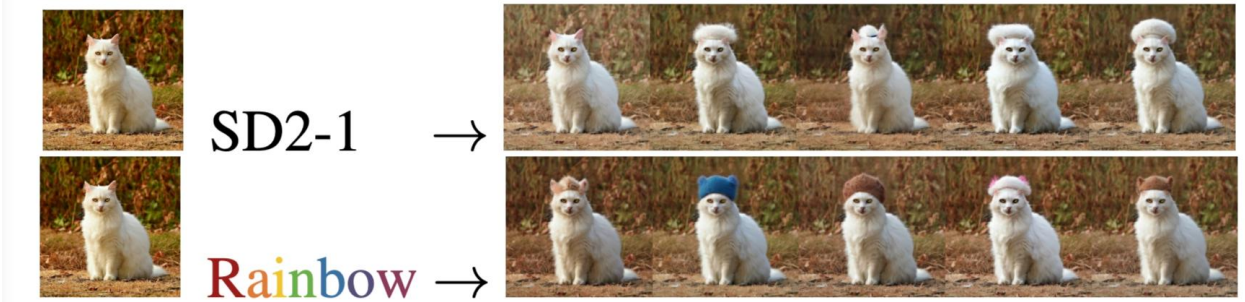


(b) Base prompt + **suport devices** edges

Experiments : Down stream task

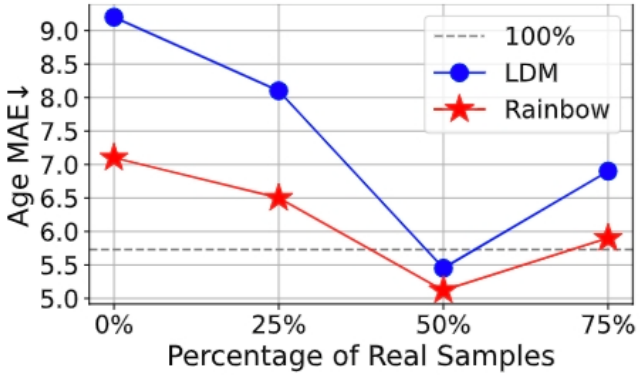
1. Image Editing

A cat wearing a wool cap.



2. Counterfactual Generation

- Generate 3D Brain MRIs include ages(1~100) x gender(Male/Female) then train the age prediction model.
- Both LDM and Rainbows performed best and better than models trained with fully real data at 50%.



Conclusion

Limitation and Future work

- Interpretability
  - Relationship between Edge and Semantic exists, but the **analysis is post-hoc and not automatic**.
  - Integrating **pre-defined concepts or domain knowledge** into graph embeddings.
- Computational cost
  - Training Rainbow requires substantial compute because **M trajectories are updated in parallel**.
- Domain generalization
  - Apply into text generation, recommendation, ….

Conclusion

- A Novel **Combination of GFlowNets and Latent Diffusion Models**

- **Improve both diversity and quality** in natural and medical image generation.

- **Enables meaningful manipulation** within the latent graph

E.O.D

- 정의

$$P_F(s'|s) = \frac{F(s \to s')}{F(s)}, \qquad P_B(s|s') = \frac{F(s' \to s)}{F(s')}$$

- **Detailed Balance(모든 인접 s, s' 쌍에 대해)**

$$F(s)P_F(s'|s) = F(s')P_B(s|s')$$

- 위 식에 정의를 대입하면

$$F(s)\frac{F(s \to s')}{F(s)} = F(s')\frac{F(s' \to s)}{F(s')} \quad \Rightarrow \quad F(s \to s') = F(s' \to s)$$

- 이제 한 state s에 대해 **outflow**

$$\sum_{s \to s''} F(s \to s'') = \sum_{s \to s''} F(s)P_F(s''|s) = F(s)\sum_{s \to s''} P_F(s''|s) = F(s)$$

- **inflow**는 DB를 사용해서

$$\sum_{s' \to s} F(s' \to s) \overset{(\star)}{=} \sum_{s' \to s} F(s \to s') = F(s)\sum_{s' \to s} P_F(s'|s) = F(s)$$

따라서

$$\sum_{s' \to s} F(s' \to s) = \sum_{s \to s''} F(s \to s'')$$

즉, Detailed Balance가 성립하면 inflow = outflow,

곧 flow matching constraint가 자동으로 만족된다.