NeurIPS 2026

SNACK

A Sequential Notation Framework for Probabilistic Graph Generation

Hohyun Kim* Hyesung Kim* Min-hwan Oh† Seunggeun Lee†

Seoul National University · *equal contribution †corresponding

github.com/leelabsg/snack

We want to sample graphs in proportion to reward

$p(G) \propto R(G)$

Many good, diverse candidates instead of a single mode — that is what GFlowNets do.
It requires the backward policy **q(x | G)**: a distribution over *every* string that decodes to G.

Two paradigms, each missing something

GRAPH-BASED

q is computable. The cost is not.

- Valid actions by construction
- Needs a specialized architecture
- q is computable. The cost is not — One forward pass per step

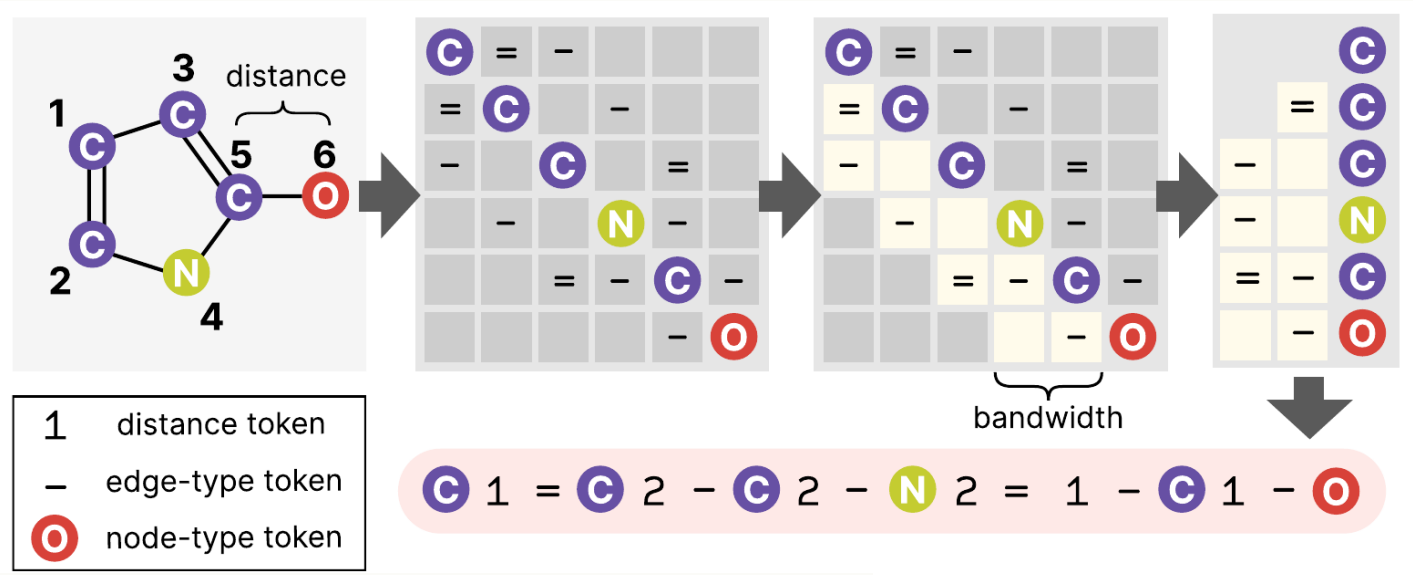
STRING-BASED

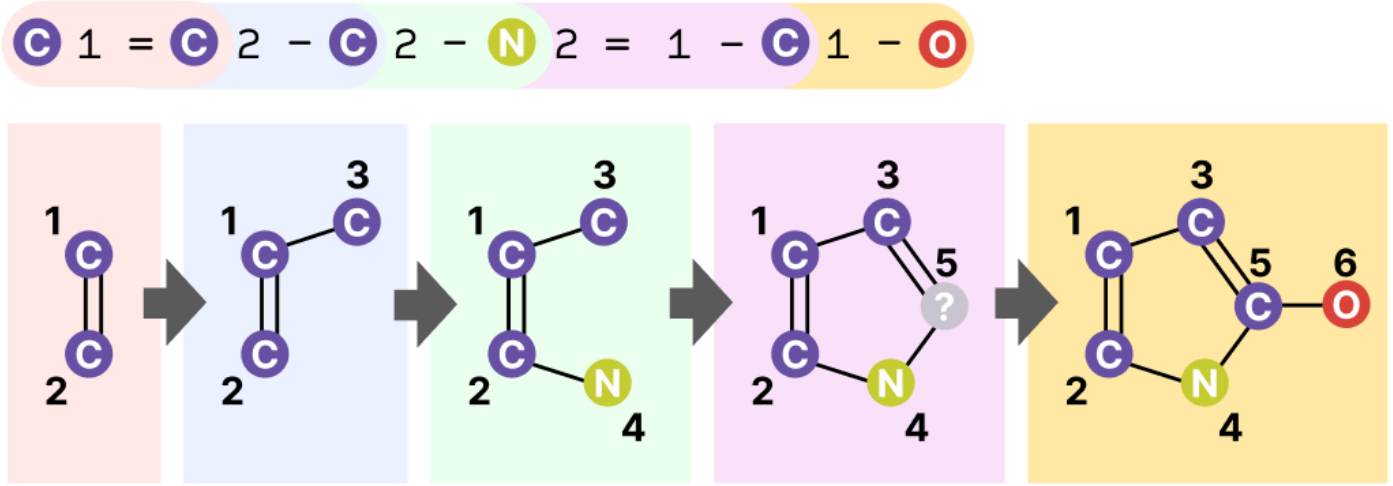
Fast. But q is unknown.

- Standard, efficient sequence models
- Many strings map to one graph, with no way to count them
- Fast. But q is unknown — q is set to 1, and reward leaks to graphs with more representations

Each side is missing what the other has, which is why GFlowNets have stayed graph-based — and paid the scaling cost.

SNACK: read the adjacency matrix, row by row





THE KEY PROPERTY

A SNACK string and an ordered adjacency matrix are the same object

SELFIES, SMILES

```
[C][N]
[Ring1][C][N]
```

Different strings, same matrix — q is not computable.

SNACK — bijective

One string ⇄ one ordered matrix, so the only ambiguity left is the node ordering itself:

$$q(x|G) \;=\; |\mathbf{Aut}(\mathbf{A})| \cdot q^{ord}(\pi|G)$$

The state space is a DAG, not a tree. Distinct strings merge at the final string → graph step. To our knowledge that converging transition had not been addressed before, and handling it is what makes the correction exact.

Every prefix is a partial graph — so we can mask

Decode the prefix, read off which operations are admissible, and zero out the rest of the logits at generation time. Prior work used BFS ordering only to build training sequences, then discarded it.

Grammar

Always a well-formed SNACK string, and termination is enforced.

Valence

Chemically valid molecules by construction.

BFS

Generation obeys the same ordering as training — no exposure bias.

100% valid on every benchmark — and BFS masking aligns the supports of p_θ and q, so every sampled sequence has strictly positive backward flow. This is why V+B beats V throughout.

String-based GFlowNets, done properly

$$L_{TB} = [\log Z + \log p_\theta(x) - \log q(x|G) - \log R(G)]^2$$

q distributes the reward across every SNACK string of the same graph — the term nobody could compute before.

One forward pass, not N + M

Trajectory log-probabilities come from a single pass over the sequence; graph policies re-encode every intermediate state.

Cubic → quadratic in N

In both compute and memory, under standard dense attention.

Conditioned TB too

Start from a scaffold's string and keep appending — substructure preserved by construction.

Same symmetry, applied at the entry

$$p_0(x_0|G_0) = |\mathrm{Aut}(A)| \cdot p_0^{ord}(\pi|G_0)$$

A fragment maps to many strings. Fix one as the prompt and you have optimized p(G | x_0), not p(G | G_0) — so we correct the entry step too, and exact proportionality is restored for conditional generation.

Competitive on distribution learning — and far ahead when conditioned

MOSES

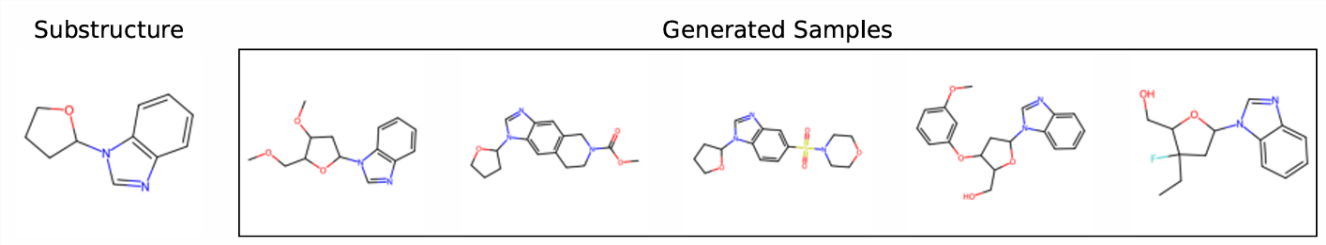
Method	Valid ↑	FCD ↓	Scaf ↑
SMILES	0.993	0.100	0.898
SELFIES	1.000	0.119	0.896
GEEL	0.997	0.104	0.894
SNACK (V+B)	1.000	0.103	0.895

Same architecture for every representation, no method-specific tricks.

SUBSTRUCTURE-CONDITIONED, NO FINE-TUNING

Method	Success ↑
SELFIES	0.181
SMILES	0.152
GEEL	0.000
SNACK	0.390

SNACK can attach to any earlier atom in the band; SMILES only extends from the last one.



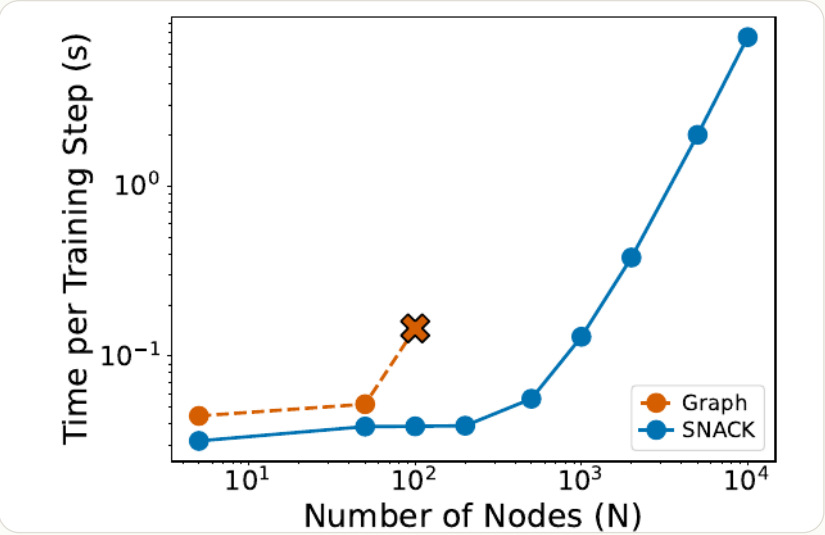
Goal-directed: the gap widens as graphs grow

GFLOWNET, TOP-100 REWARD

Task	Graph	SNACK (V+B)
HOMO–LUMO gap	1.118	1.118
Aripiprazole sim.	0.204	0.469
Albuterol sim.	0.171	0.736
Unique fraction	0.998	1.000

Parity on the small benchmark (N ≤ 9); on the two larger tasks (max N = 16), 2–4× the graph policy's top-100 reward.
GFlowNets trained on raw SMILES never got off the ground — early samples are invalid, zero-reward strings.

TRAINING TIME PER STEP



The graph baseline runs out of GPU memory around 10^2 nodes; SNACK keeps going to 10^4.

Graph methods had exactness, string methods had scale — SNACK has both

All of it follows from one bijection with the ordered adjacency matrix.

01

First exact backward policy

q(x|G) in closed form via automorphisms, and prefixes that turn constraints into a logit mask.

02

Fragment-conditioned TB

Exact proportionality for conditional generation, not just unconditional.

03

$O(N^3) \rightarrow O(N^2)$

Trajectory evaluation in a single forward pass, and graphs two orders of magnitude larger.

Limitations and next steps: the ordering policy is fixed and hand-designed — we would like to learn it jointly — plus stereochemistry, chirality, and fragment-level generation.

github.com/leelabsg/snack Thank you — come find us at the poster.