Directed Acyclic Transformer for Non-Autoregressive Machine Translation

Fei Huang, Hao Zhou, Yang Liu, Hang Li, Minlie Huang

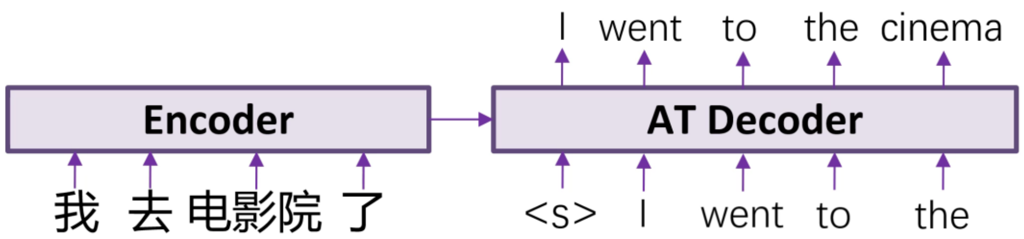
CoAI group, Tsinghua University

ByteDance AI Lab

Background

Autoregressive Translation (AT)

- Generate token by token
- Latency: ~600ms per sample*

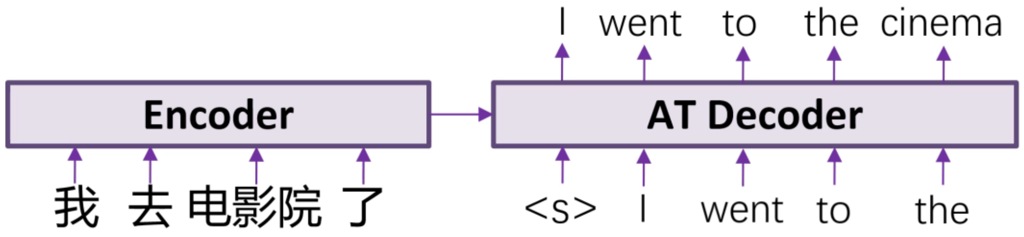


*: Reported by Gu et al. *Non-autoregressive Machine Translation*. ICLR2018.
The latency is evaluated on IWLST16 En-De with batch size=1 on a Nvidia Tesla P100

Background

Autoregressive Translation (AT)

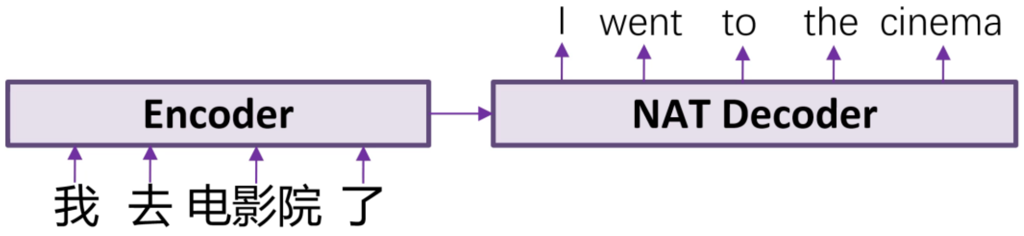
- Generate token by token
- Latency: ~600ms per sample*



Reduce the inference latency

Non-Autoregressive Translation (NAT)

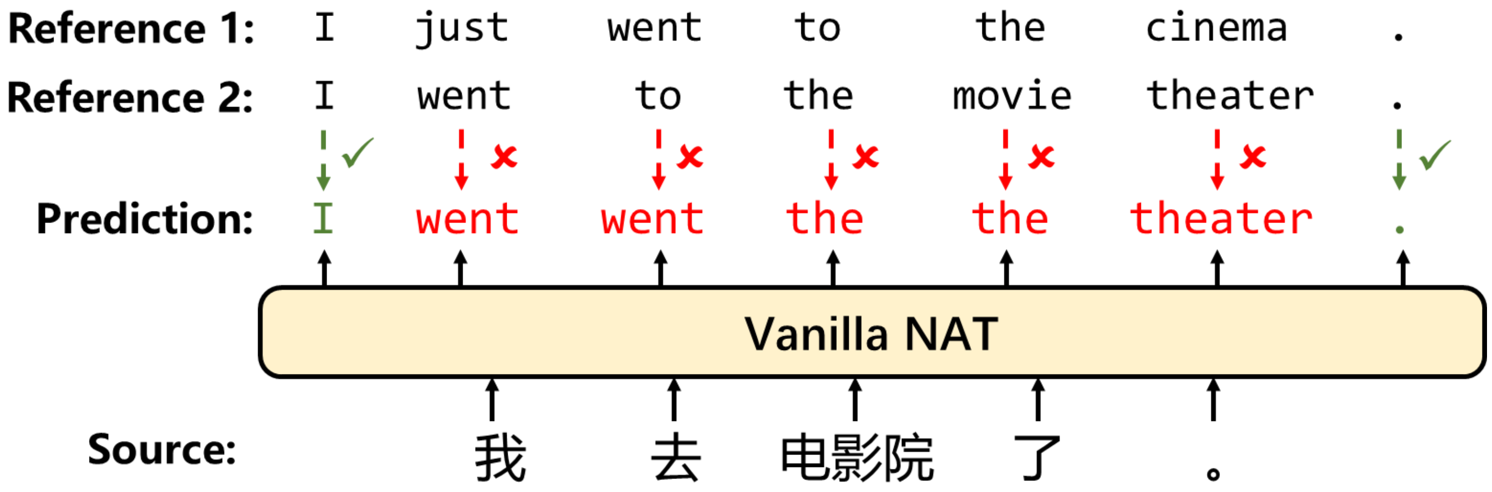
- Generate all tokens in parallel (Gu et al., 2018)
- Latency: ~10x speed up*



*: Reported by Gu et al. *Non-autoregressive Machine Translation*. ICLR2018.
The latency is evaluated on IWLST16 En-De with batch size=1 on a Nvidia Tesla P100

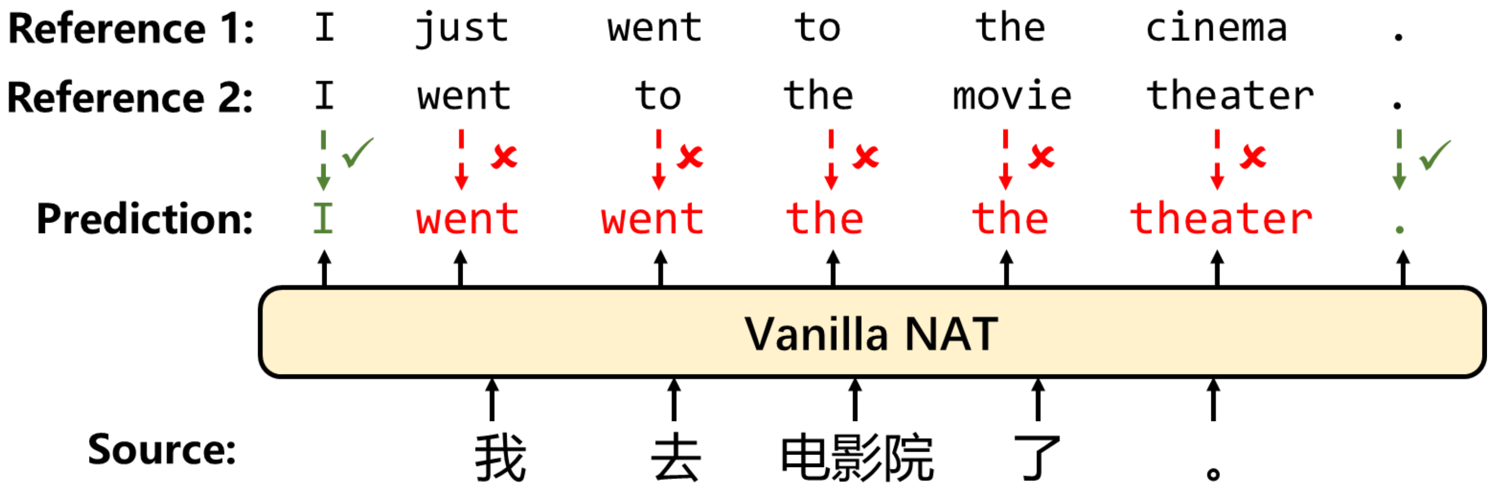
Challenges in NAT

- **Multi-modality Problem:**
  - NATs produce incorrect outputs that mix multiple possible translations



Challenges in NAT

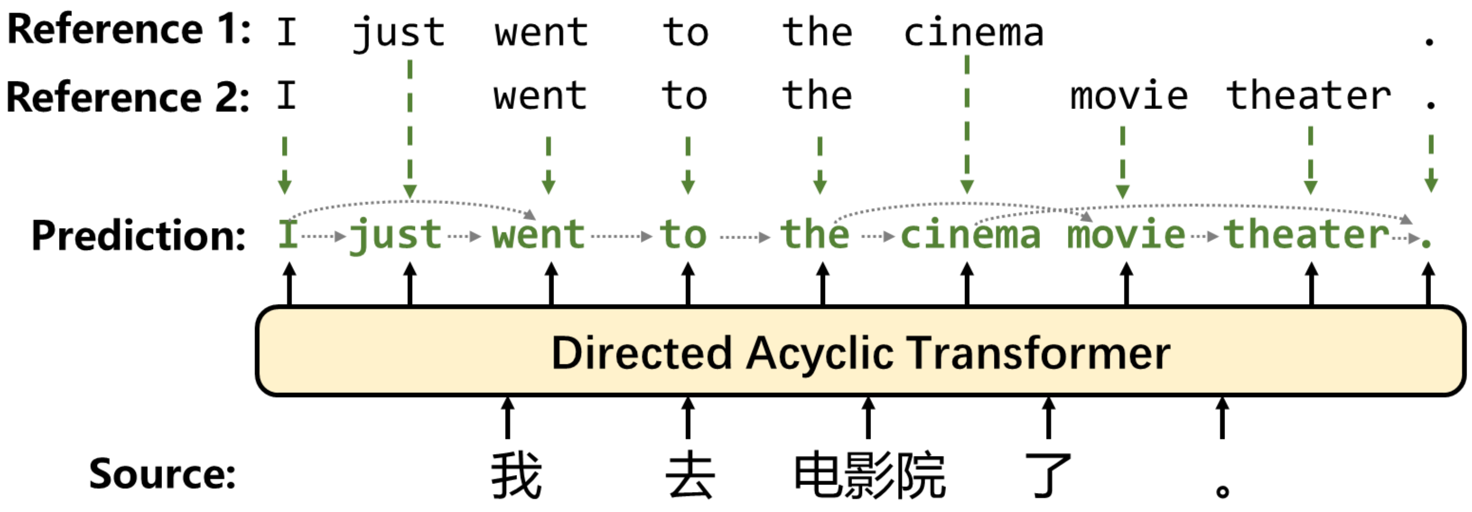
- **Multi-modality Problem:**
  - NATs produce incorrect outputs that mix multiple possible translations



- **Two causes:**
  - **Training**: inconsistent labels in the reference sentences
  - **Inference**: cannot preserve correct lexical dependencies during inference

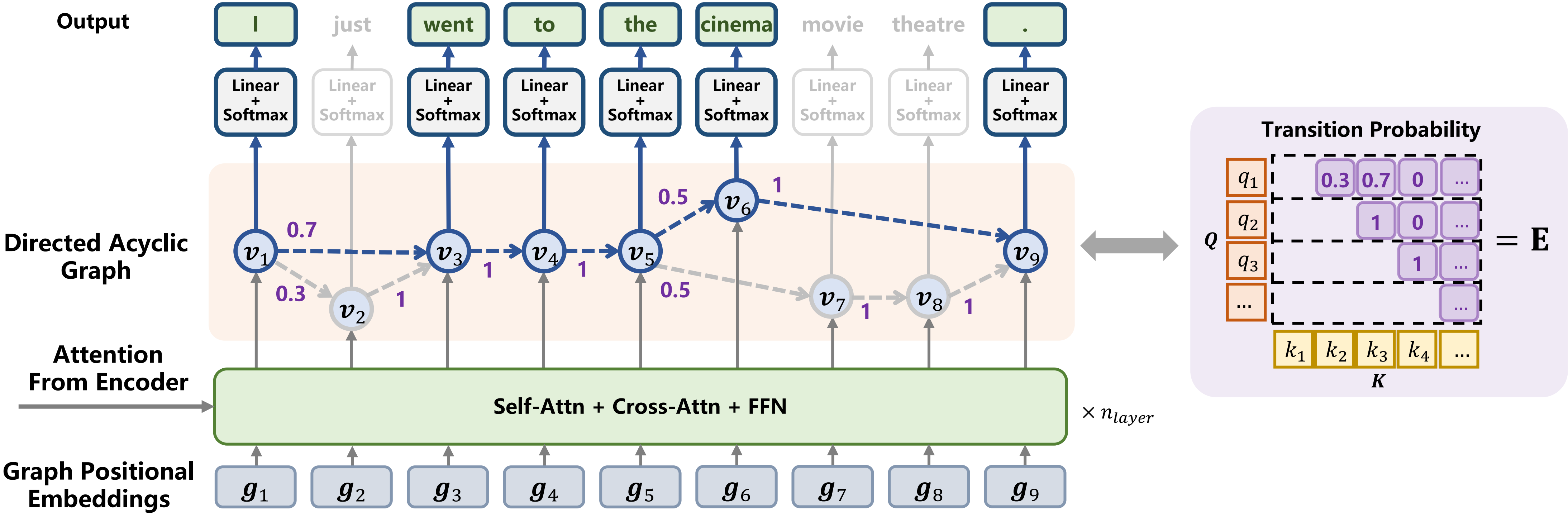
Our Proposed Method

- **Utilize Directed Acyclic Graph (DAG)**
  - to organize the decoding hidden states (and predicted tokens)



  - **In training**: alleviate conflicts by assigning tokens to different vertices
  - **In inference**: recover the translation following predicted transitions

Directed Acyclic Transformer (DA-Transformer)

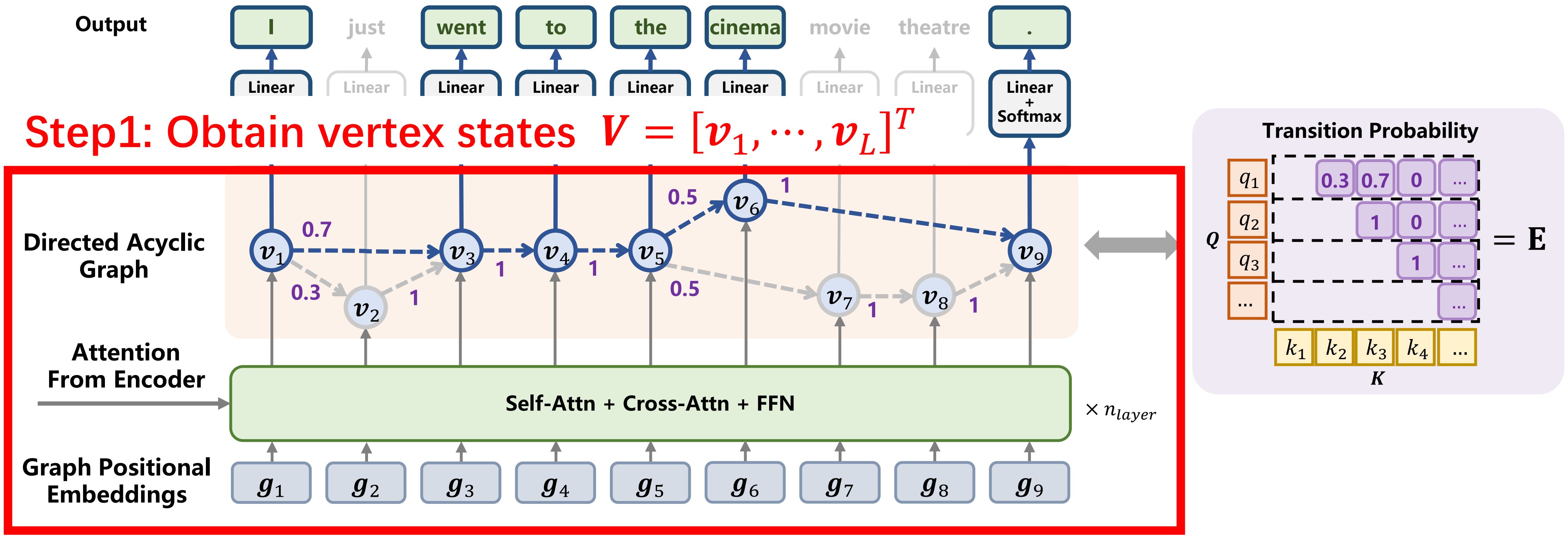


Reference $Y =$ `I went to the cinema .`

Path $A = \{1, 3, 4, 5, 6, 9\}$

Directed Acyclic Transformer (DA-Transformer)

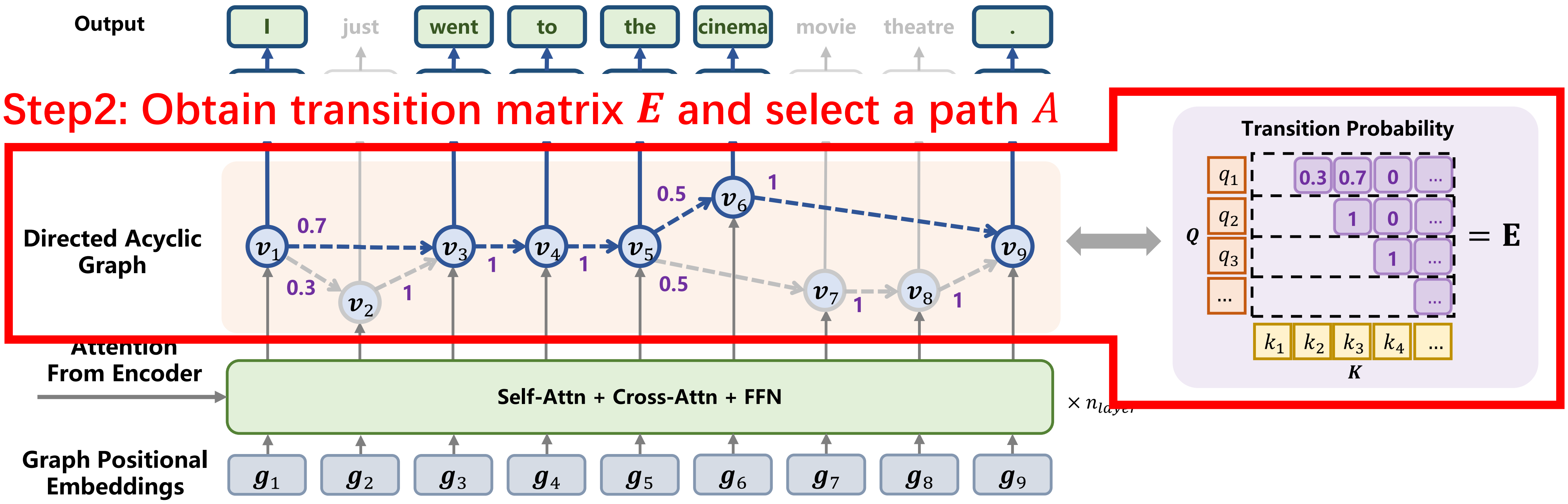
Step1: Obtain vertex states $V = [v_1, \cdots, v_L]^T$



Reference $Y =$ `I went to the cinema .`

Path $A = \{1, 3, 4, 5, 6, 9\}$

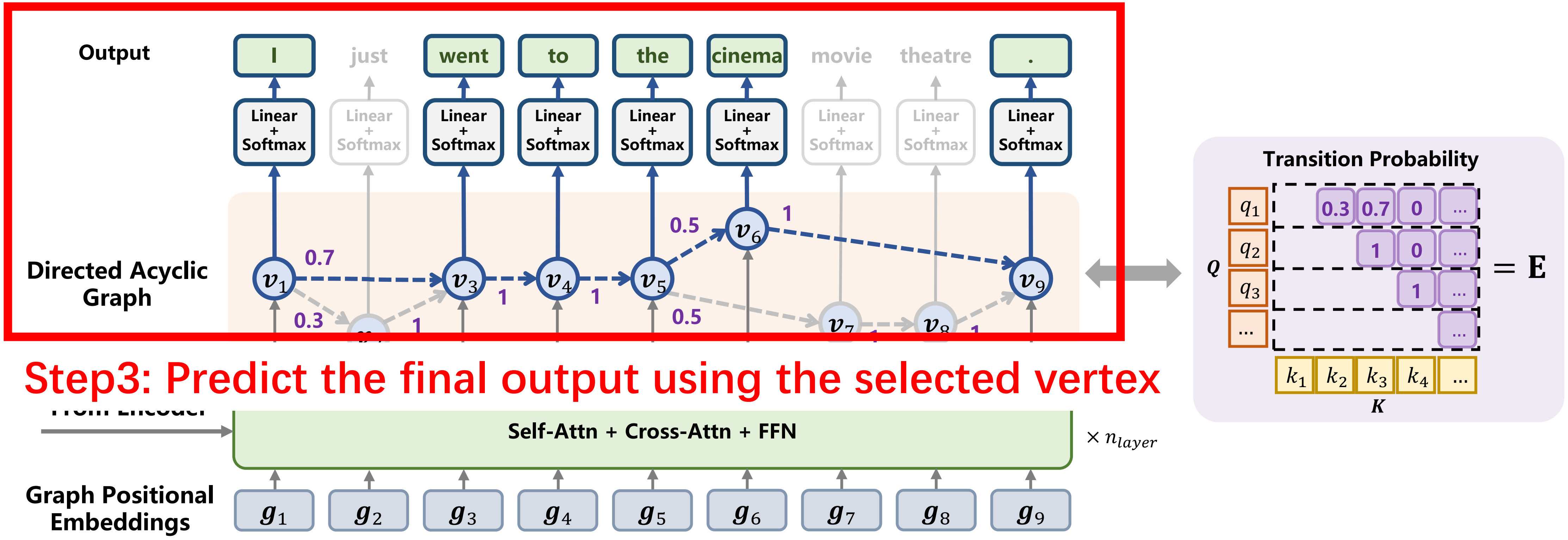
Directed Acyclic Transformer (DA-Transformer)



Reference $Y =$ `I went to the cinema .`

Path $A = \{1, 3, 4, 5, 6, 9\}$

Directed Acyclic Transformer (DA-Transformer)



Reference $Y =$ `I went to the cinema .`

Path $A = \{1,3,4,5,6,9\}$

DA-Transformer – Training & Inference

- **Training with only one reference**

$$\mathcal{L} = -\log P_\theta(Y|X) = -\log \sum_{A\in\Gamma} P_\theta(Y, A|X)$$

  - Use dynamic programming to do the marginalization
  - We find that the objective can avoid inconsistent labels by **assigning a single reference to several paths sparsely**
  - The whole DAG can be **learned across different training instances**

- **Inference with various decoding strategies on the DAG**
  - Greedy / Lookahead Decoding / Beam Search

Main Results

Model	Iter #	Avg Gap ↓ Raw	KD	Speedup
Transformer (Vaswani et al., 2017)	M	0.45	0.49	1.0x
Transformer (Ours)	M	0	0	1.0x
CMLM (Ghazvininejad et al., 2019)	10	3.00	1.37	2.2x
SMART (Ghazvininejad et al., 2020b)	10	2.67	0.67	2.2x
DisCo (Kasai et al., 2020)	≈4	2.43	0.59	3.5x
Imputer (Saharia et al., 2020)	8	3.07	0.04	2.7x
CMLMC (Anonymous, 2021a)	10	1.35	0.15	1.7x
Vanilla NAT (Gu et al., 2018)	1	15.78	8.26	15.3x
AXE† (Ghazvininejad et al., 2020a)	1	7.36	4.34	14.2x
CTC (Libovický & Helcl, 2018)	1	9.41	3.47	14.6x
GLAT (Qian et al., 2021a)	1	6.05	2.59	15.3x
OaXE† (Du et al., 2021)	1	5.4	2.0	14.2x
CTC + GLAT (Qian et al., 2021a)	1	3.52	1.98	14.6x
CTC + DSLP (Huang et al., 2021)	1	3.44	0.73	14.0x
DA-Transformer + Greedy (Ours)	1	1.47	0.75	14.0x
+ Lookahead	1	1.20	0.58	13.9x
+ BeamSearch	1	0.61	0.18	7.1x
+ BeamSearch + 5-gram LM	1	**0.30**	**0.05**	7.0x

Part of Table1

Avg Gap = BLEU gap against the best AT averaged on WMT14 En↔De and WMT17 Zh↔En

1. DA-Transformer outperforms existing **non-iterative NATs by 2~3 BLEU** with competitive latency speedup when Knowledge Distillation is not applied.

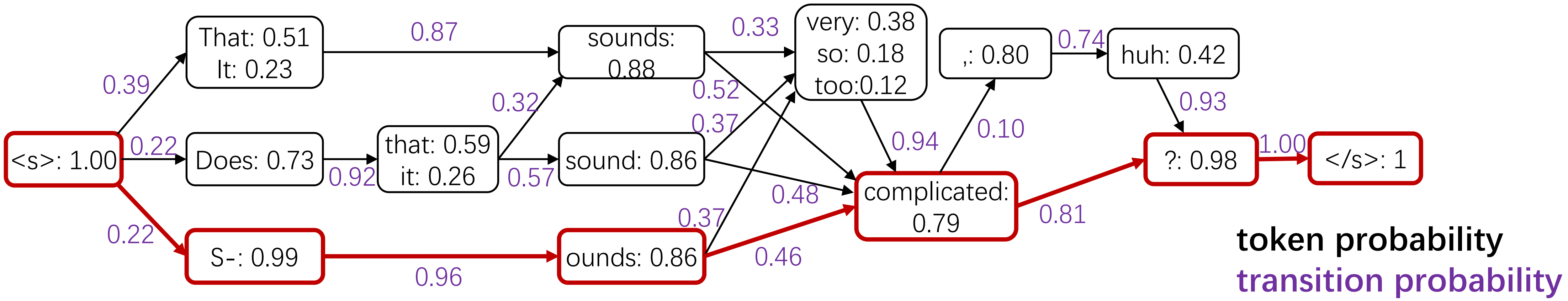
2. DA-Transformer **reduces the average gap against the AT to <0.30 BLEU**, while achieving **7x~14x speedups**.

Case Study

Source: 听 起来 很 复杂 ?　　**Reference:** S- ounds tricky ?

Vanilla NAT: It ounds sounds sounds complicated ?

DA-Transformer:



Rank	Hypotheses of BeamSearch	Score
1	S- ounds complicated ?	-0.55
2	S- ounds very complicated ?	-0.66
3	Does that sound very complicated ?	-0.79
4	S- ounds very complicated , huh ?	-0.94

Other Results & Analysis

- DA-Transformer effectively improves the token prediction accuracy

- DA-Transformer facilitates diverse generation

- DA-Transformer provides flexible quality-speed tradeoff by tuning graph size, decoding method

Thanks for Your Attention

GitHub (code): https://github.com/thu-coai/DA-Transformer

If you are interested, welcome to see our other paper at ICML2022!

On the Learning of Non-Autoregressive Transformer

CoAI group, Tsinghua University

ByteDance AI Lab