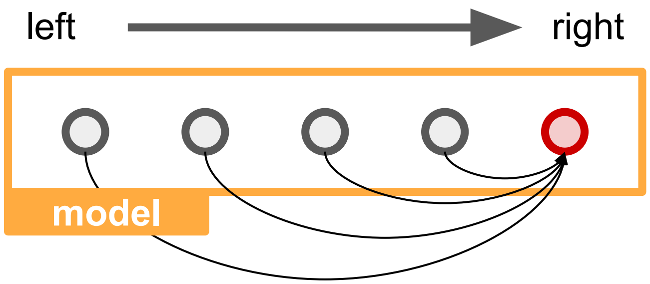
Discovering Non-Monotonic Autoregressive Orderings with Variational Inference

Xuanlin Li[1*], Brandon Trabucco[1*], Dong Huk Park[1],
Michael Luo[1], Sheng Shen[1], Trevor Darrell[1], Yang Gao[2]
1: UC Berkeley 2: Tsinghua University

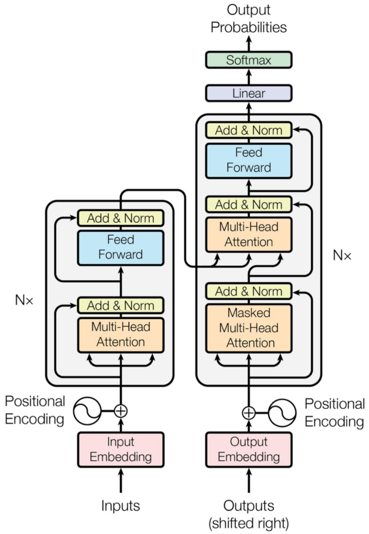
BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

Non-Monotonic Sequence Generation



A pine tree sways in the spring breeze. → Un pin se balance dans la brise printanière.

Machine Translation



many domains have natural structure

```
class StreamingBuffer(object):
    def __init__(self):
        self.vals = []
```

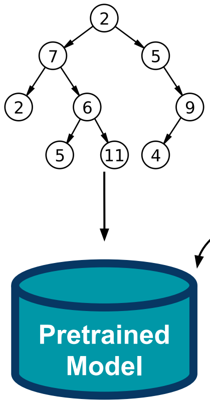
Code Generation

Problem Statement: Given a sequence generation dataset, find the *generation order* that most *naturally* describes the data.

Discovering Generation Orders

choosing the right loss function

Suppose we know in advance the right generation order.



$+ \mathcal{L}_{\text{fixed}}(\theta)$

Suppose we know a generation order that can be fine-tuned.

$+ \mathcal{L}_{\text{adapt}}(\theta)$

often must be domain-specific

Can we find a natural generation order efficiently without domain-specific knowledge?

Yes: Our approach, **Variational Order Inference (VOI)**, does exactly this!

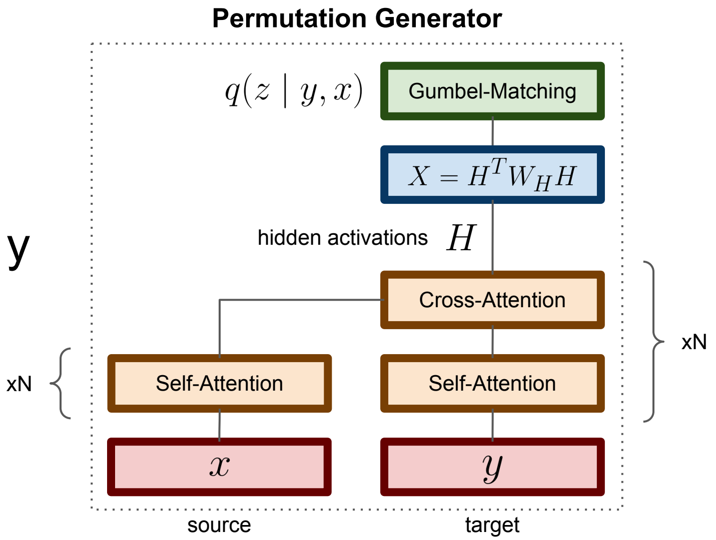
Variational Order Inference

natural ➡ highly probable to co-occur with y

x - input variable (such as an image)

y - target sequence

z - generation order of y



$$q(z \mid y, x) = \frac{\exp\langle X, P = z\rangle_F}{\text{perm}(\exp(X))}$$

Approximate density with Bethe permanent

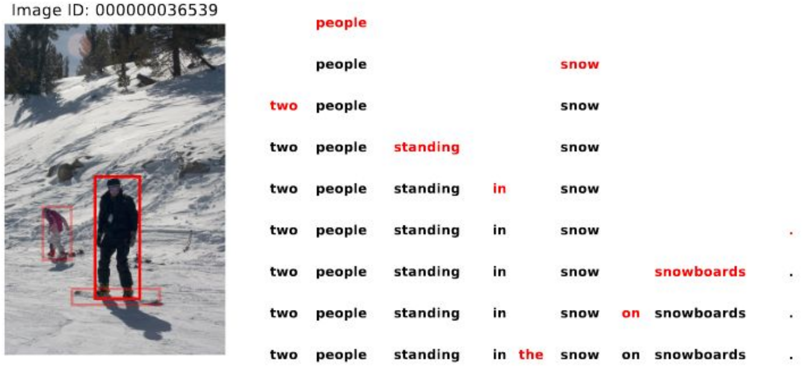
$$\max_{\theta} \mathbb{E}_{y,x}[\log p_\theta(y \mid x)] \geq$$

$$\max_{\theta} \max_{q} \mathbb{E}_{y,x,z\sim q(z \mid y,x)}[\log p_\theta(y, z \mid x)] + \mathcal{H}(q)$$

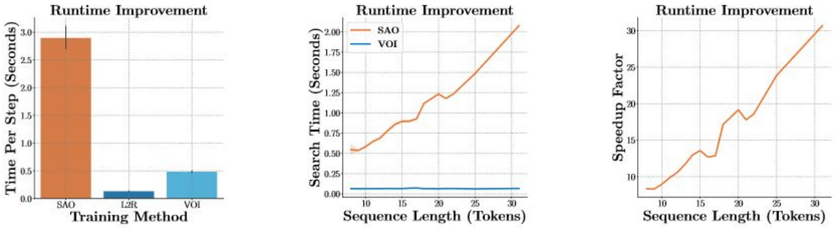
Results

Order	MS-COCO BLEU	MS-COCO Meteor	MS-COCO R-L	MS-COCO CIDEr	Django BLEU	Django Accuracy	Gigaword R-1	Gigaword R-2	Gigaword R-L	WMT16 Ro-En BLEU↑	WMT16 Ro-En Meteor↑	WMT16 Ro-En TER↓
InDIGO - SAO [1]	29.3	24.9	54.5	92.9	42.6	32.9	—	—	—	32.5	53.0	49.0
Ours - Random	28.9	24.2	55.2	92.8	21.6	26.9	30.1	11.6	27.6			
Ours - L2R	30.5	25.3	54.5	95.6	40.5	33.7	35.6	17.2	33.2	32.7	54.4	50.2
Ours - Common	28.0	24.8	55.5	90.3	37.1	29.8	33.9	15.0	31.1	27.4	50.1	53.9
Ours - Rare	28.1	24.5	52.9	91.4	31.1	27.9	34.1	15.2	31.3	26.0	48.5	55.1
Ours - VOI	**31.0**	**25.7**	**56.0**	**100.6**	**44.6**	**34.3**	**36.6**	**17.6**	**34.0**	32.9	54.6	49.3

- **Better performance** than fixed generation orders.
- Training is **highly efficient** and readily parallelizable.
- Prioritizes **descriptive** tokens first, **modifier** tokens last.



Example Generation: COCO 2017



Time Complexity of SAO: $O(Ndl^3)$ Time Complexity of VOI: $O(NKdl^2)$

For more details read our paper and check out our poster.