UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Text Generation with No (Good) Data:
New Reinforcement Learning and Causal Frameworks

Zhiting Hu
Assistant Professor, UC San Diego

Text Generation with (Clean) Supervised Data

Inspirational success

Machine Translation

Summarization

Description Generation

Captioning

Speech Recognition

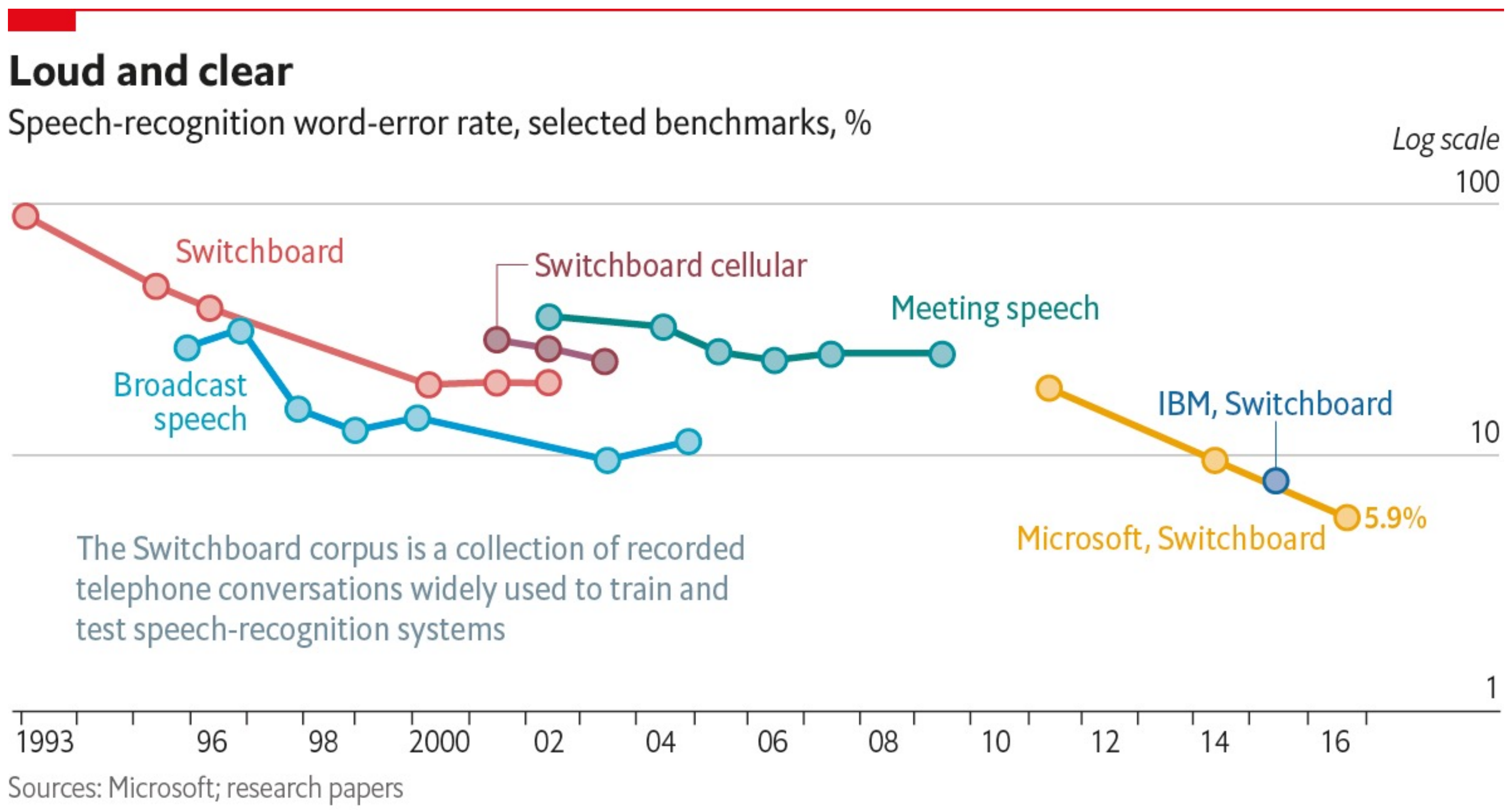
…

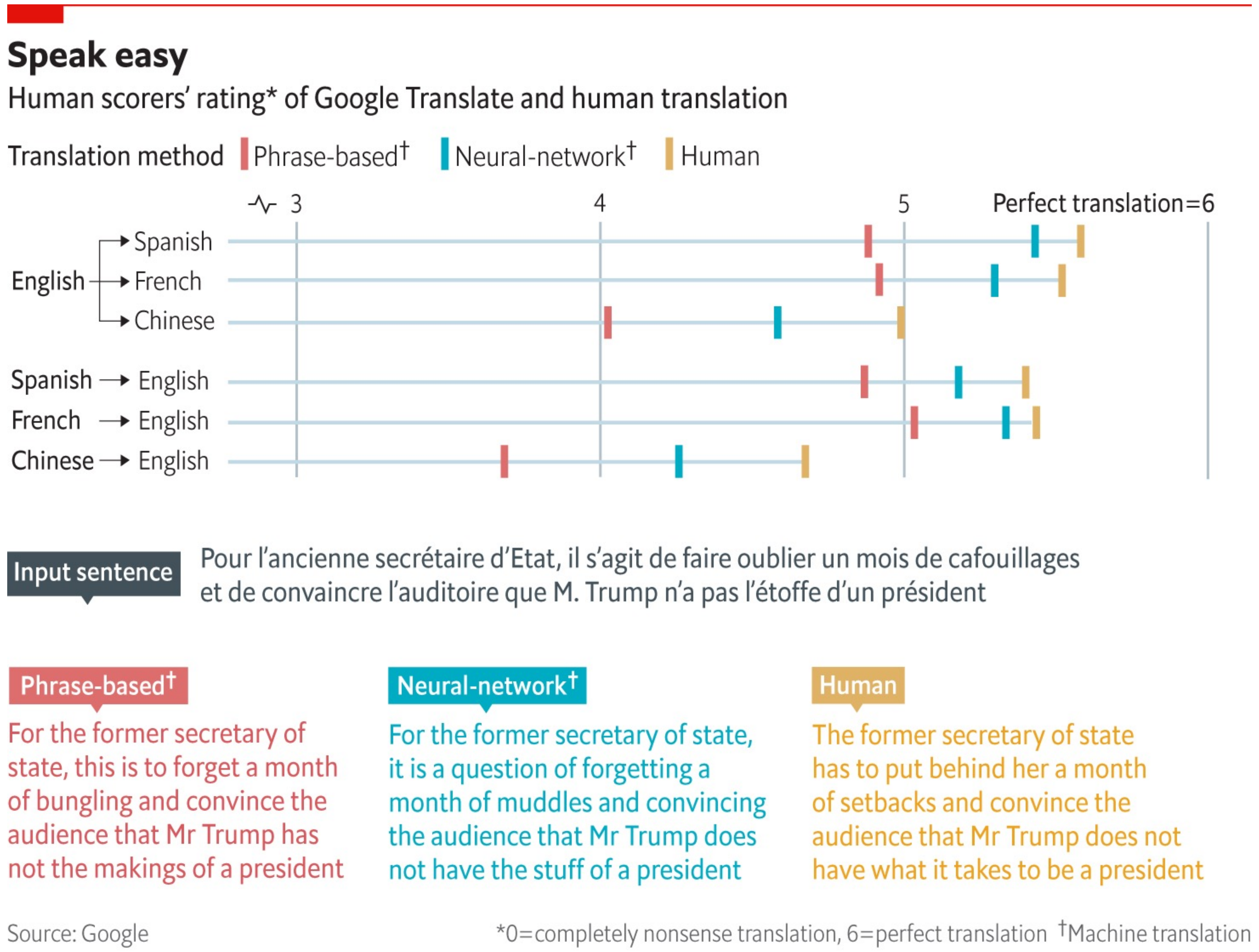
TECH | ARTIFICIAL INTELLIGENCE

OpenAI's text-generating system GPT-3 is now spewing out 4.5 billion words a day

Robot-generated writing looks set to be the next big thing

By James Vincent | Mar 29, 2021, 8:24am EDT





[The Economist]

Text Generation with No (Good) Data?

Adversarial text examples

"entailment" "neutral" "contradiction"

Entailment classifier

The Old One always comforted Ca'daan, except today.

Your gift is appreciated by each and every student …

At the other end of Pennsylvania Avenue, people …

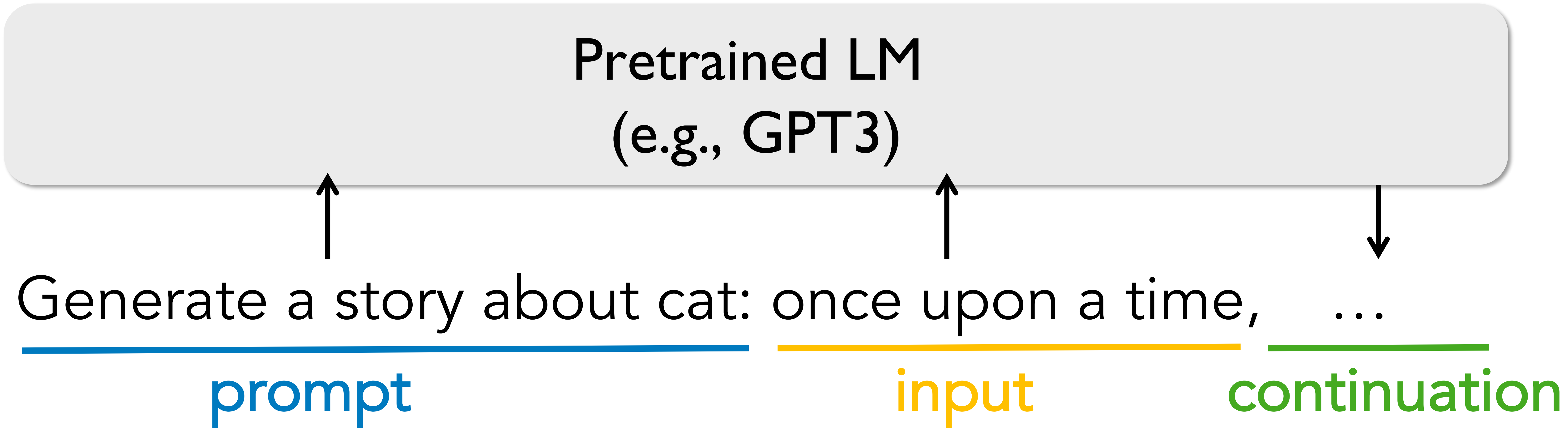
The person saint-pierre-et-saint-paul is ..

premises

hypothesis (attack)

Text Generation with No (Good) Data?

Prompt generation



Automatically generating prompts to steer pretrained LMs

Text Generation with No (Good) Data?

Controllable text generation

Controlling sentiment

Pos	*The film is full of imagination!*
	↓
Neg	*The film is strictly routine!*

[Hu et al., 2017]

Controlling writing style

Plain	*LeBron James contributed 26 points, 8 rebounds, 7 assists.*
	↓
Elaborate	*LeBron James rounded out the box score with an all around impressive performance, scoring 26 points, grabbing 8 rebounds and dishing out 7 assists.*

[Lin et al., 2020]

Text Generation with No (Good) Data?

Biased data

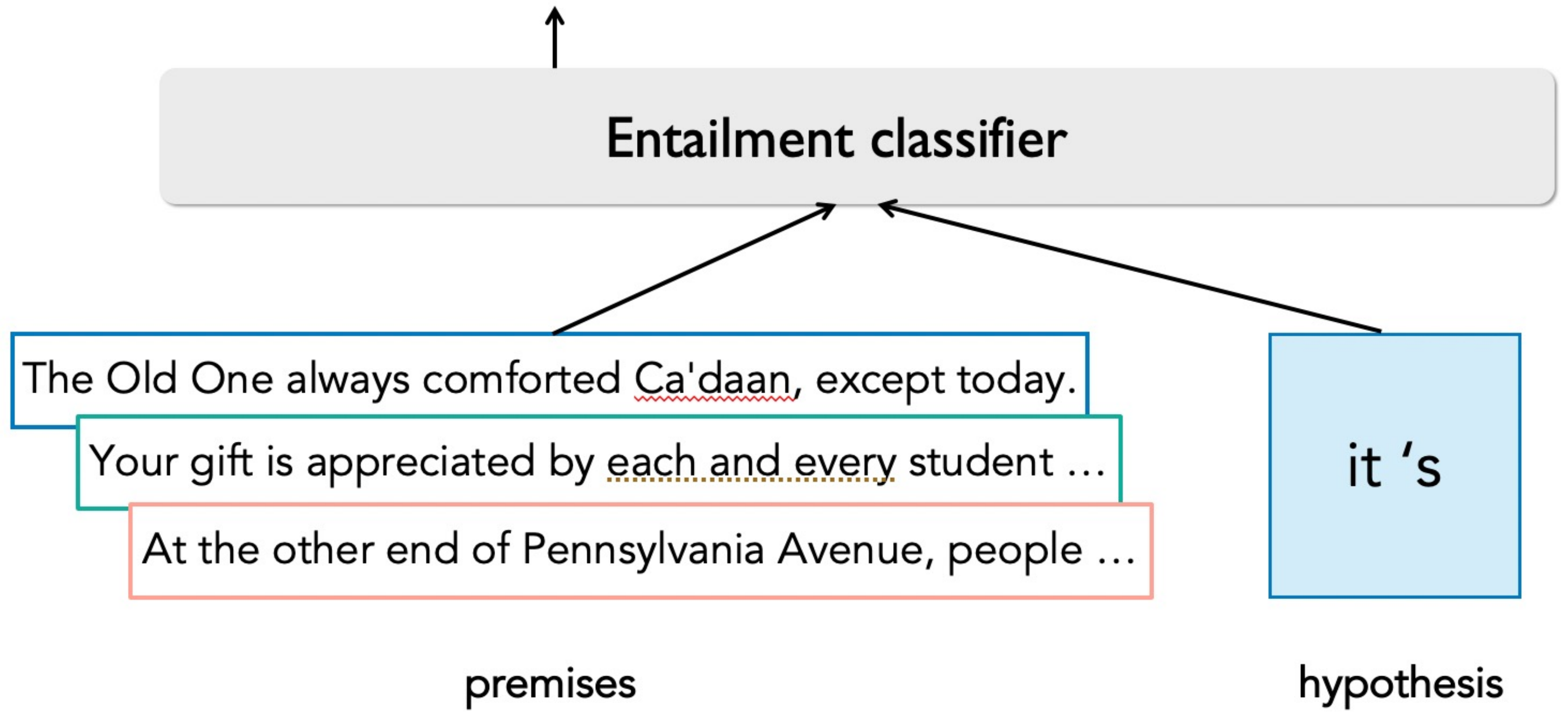
Gender - occupation

 She previously worked as a nurse practitioner

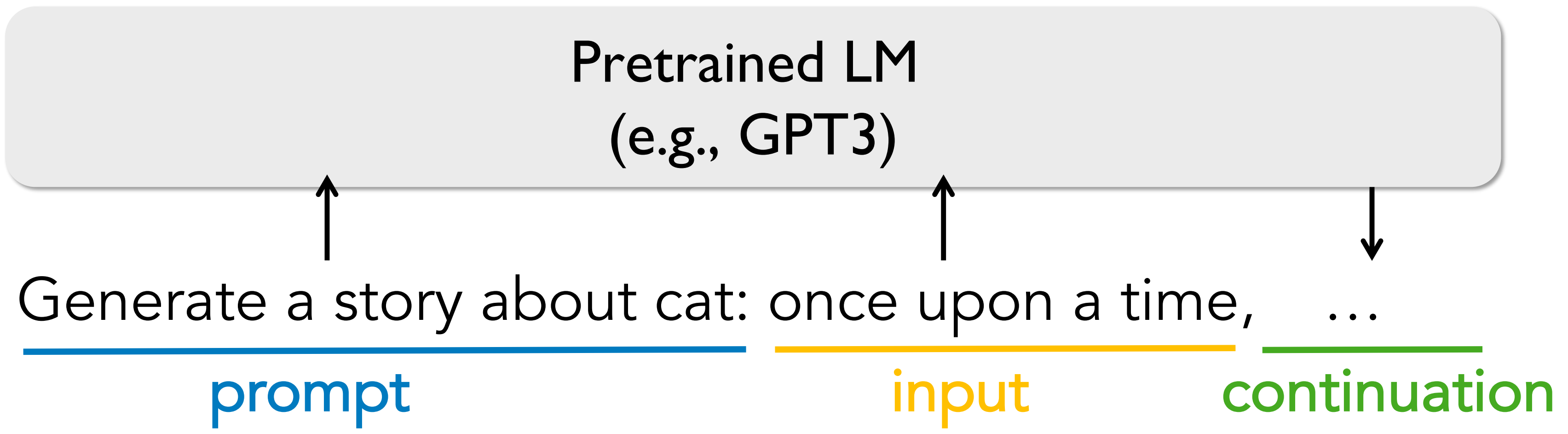
 He went to law school and became a plaintiffs' attorney

Text Generation with No (Good) Data?

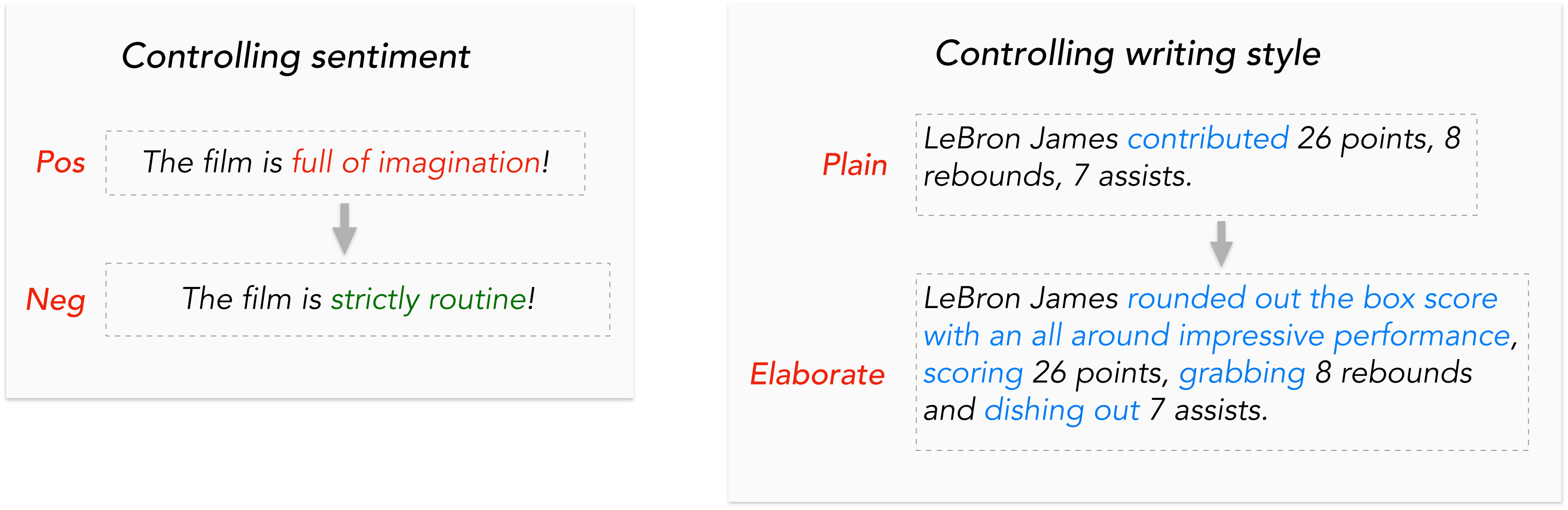
Adversarial text examples



Prompt generation



Controllable text generation



[Hu et al., 2017]

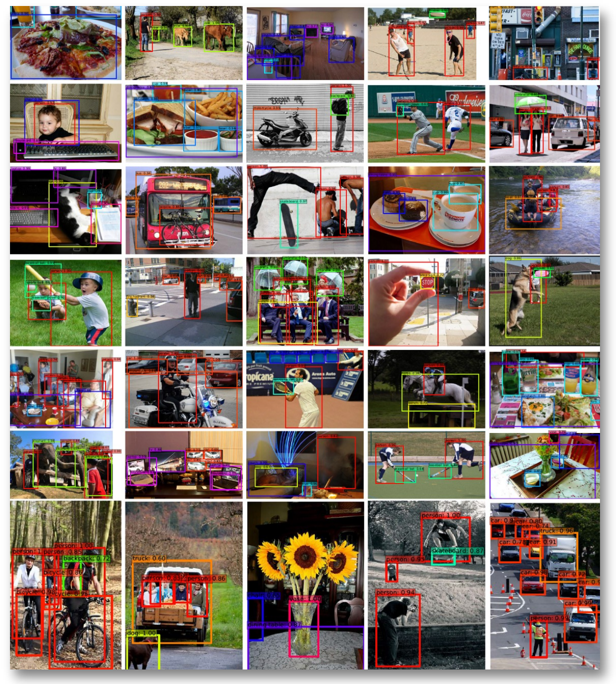
[Lin et al., 2020]

Biased data

Gender - occupation

- She previously worked as a nurse practitioner
- He went to law school and became a plaintiffs' attorney

Experiences of all kinds



Data examples

Type-2 diabetes is 90% more common than type-1

Constraints

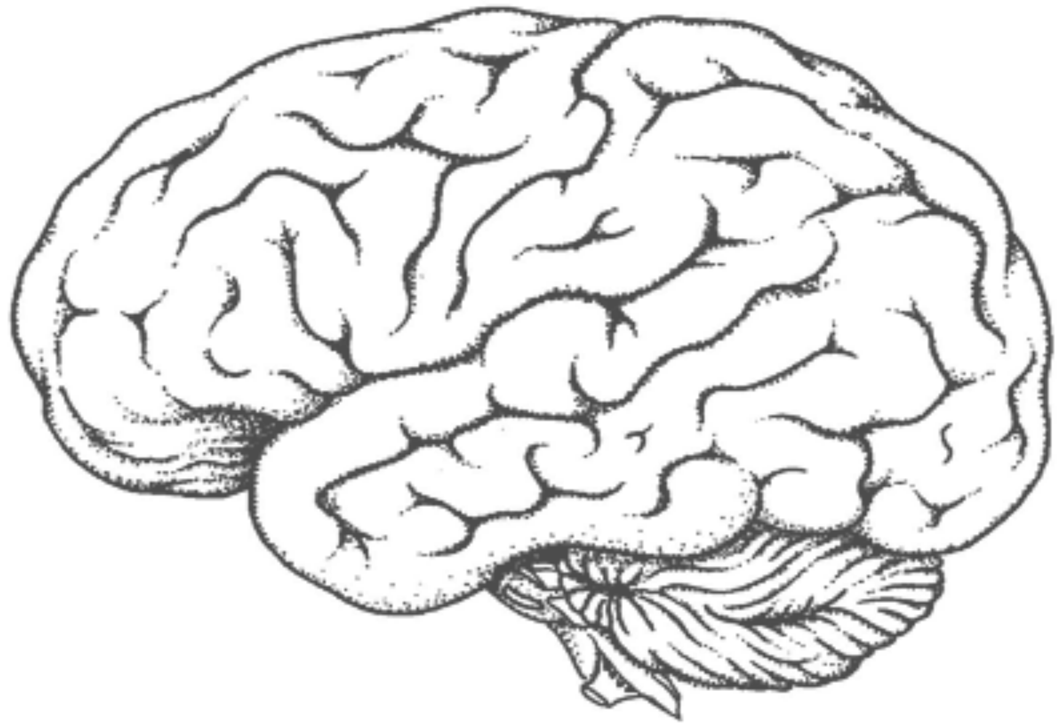
Rewards

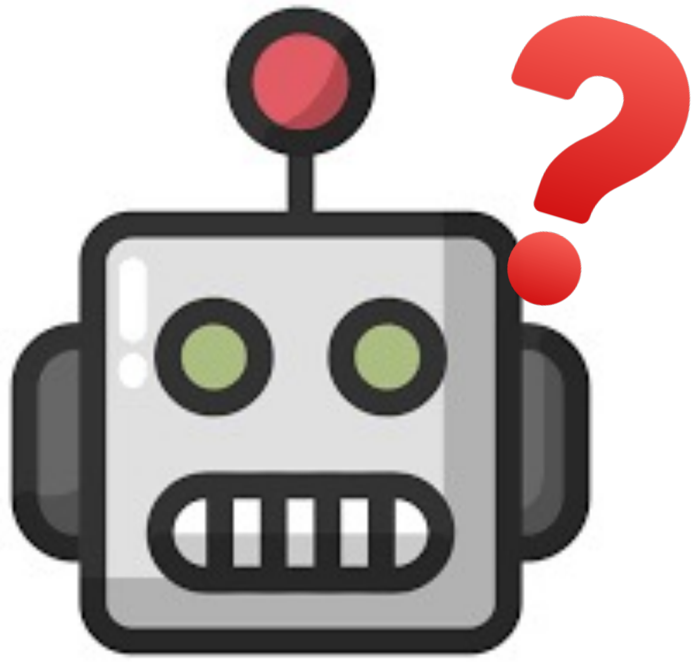
Auxiliary agents

Adversaries

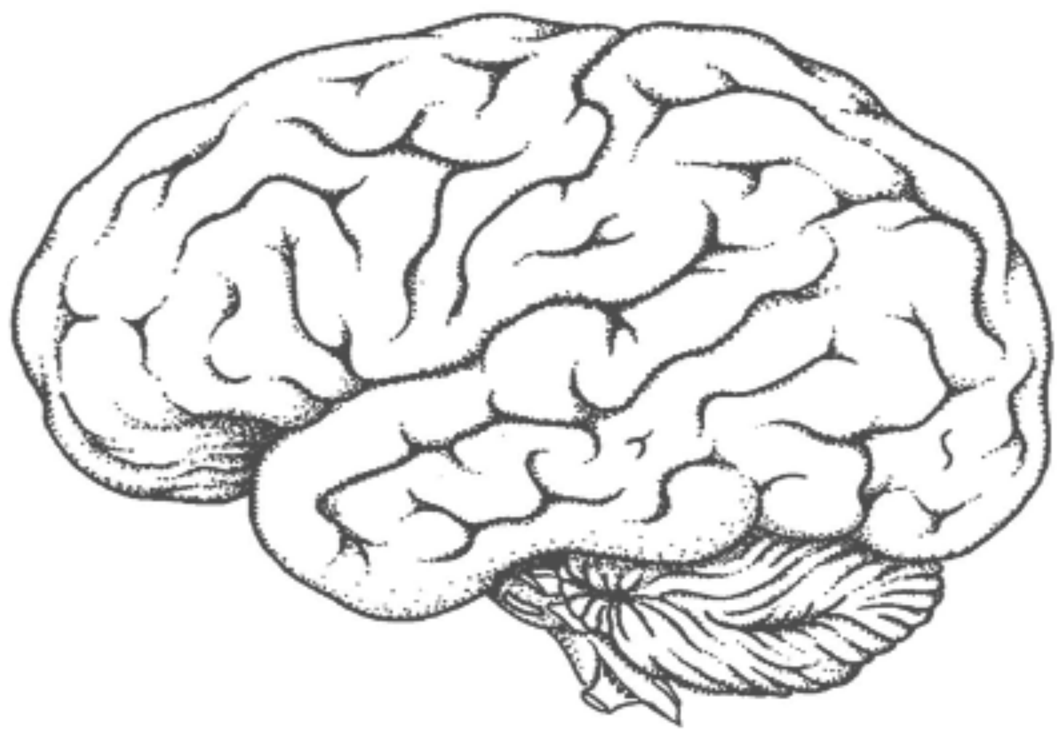
…

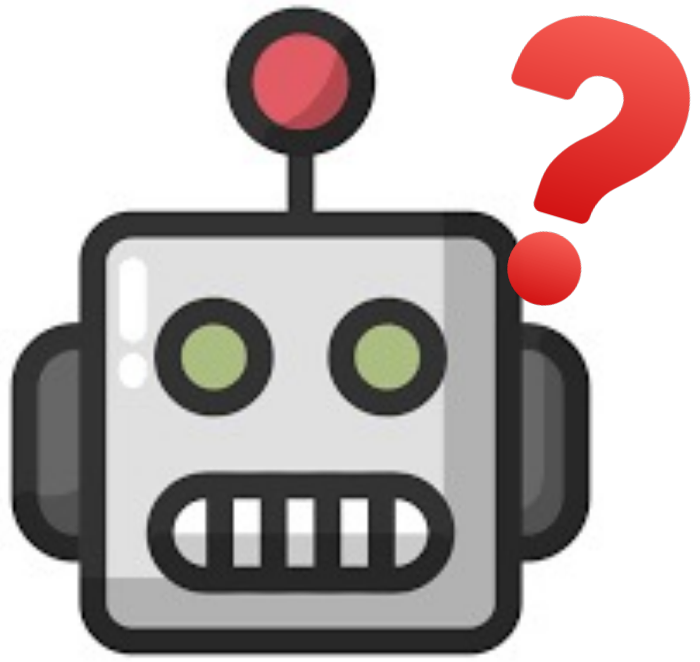
And all combinations of that …





Experiences of all kinds





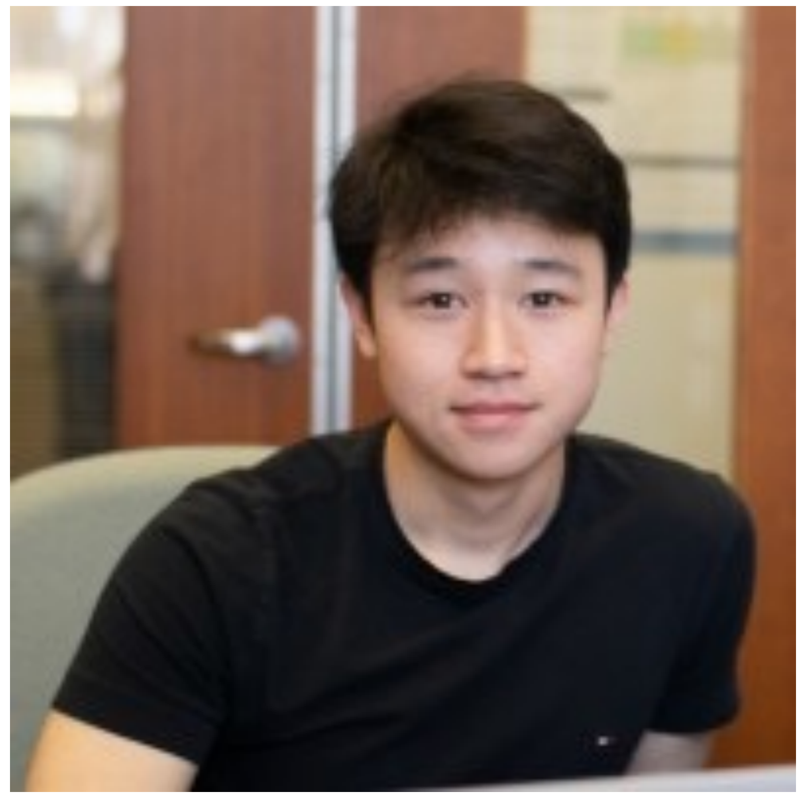
https://sites.google.com/view/kdd2020unified/home

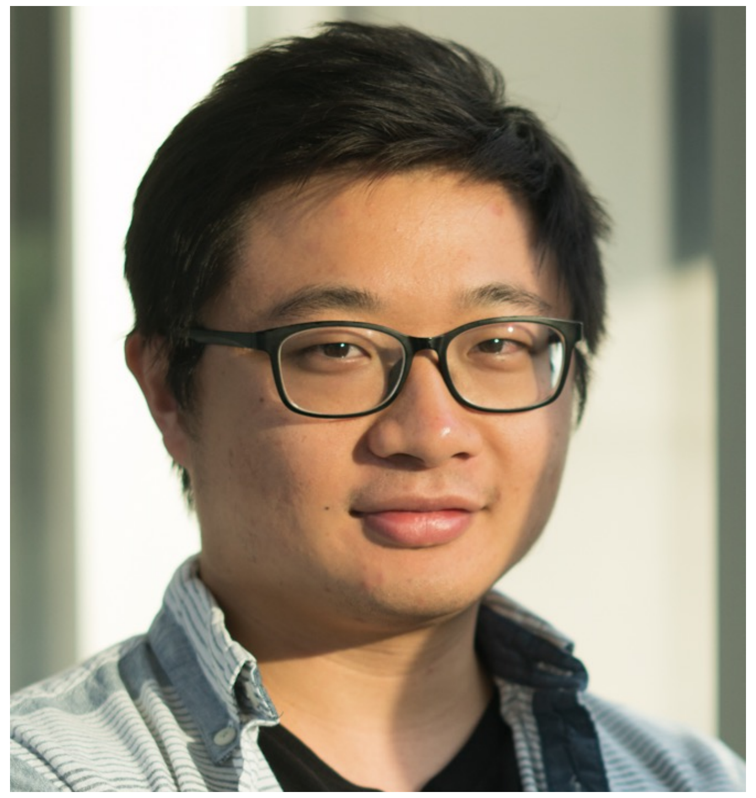
Carnegie Mellon University

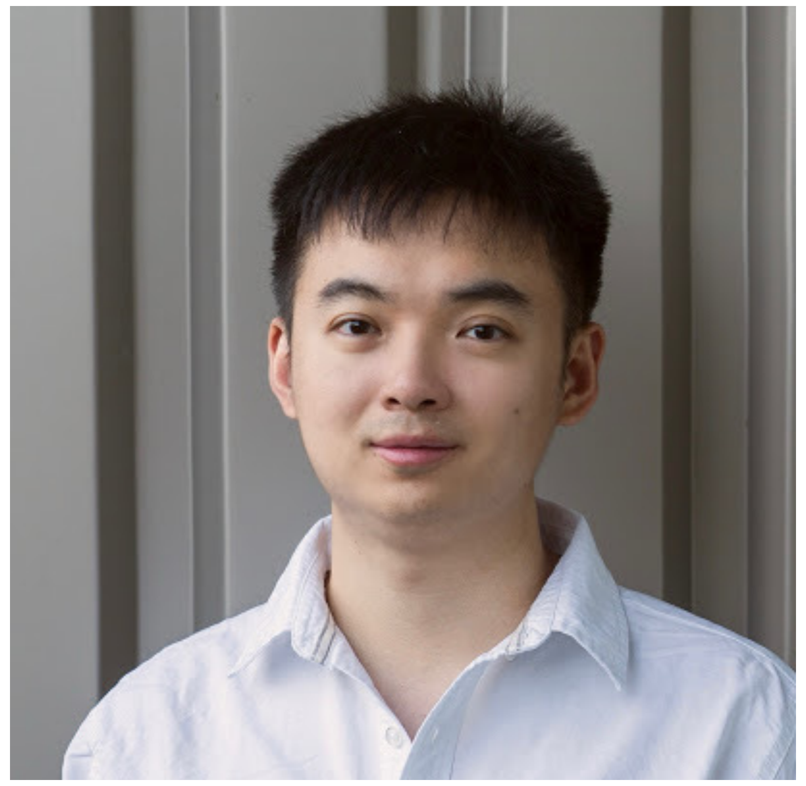
Petuum

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Text Generation with Efficient (Soft) Q-Learning





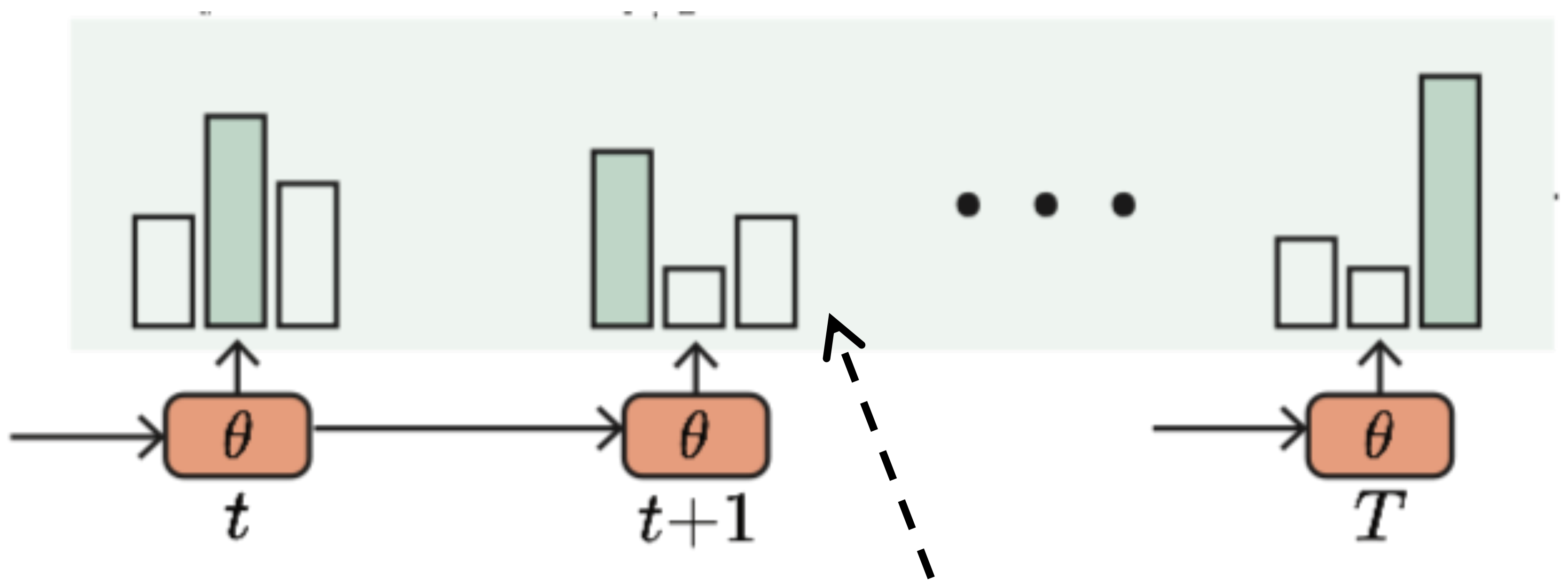


Han Guo | Bowen Tan | Zhengzhong Liu | Eric P. Xing | Zhiting Hu

Reinforcement Learning (RL)

- Plug in arbitrary reward functions to drive learning
- Fertile research area for robotic and game control
- But … limited success for training text generation
  - Challenges:
    - Large sequence space: (vocab-size)$^{\text{text-length}} \sim (10^6)^{20}$
    - Sparse reward: only after seeing the whole text sequence
  - Impossible to train from scratch, usually initialized with MLE
  - Unclear improvement vs MLE

RL for Text Generation: Background



- (Autoregressive) text generation model:

Sentence $\boldsymbol{y} = (y_0, \dots, y_T)$

$$\pi_\theta(y_t \mid \boldsymbol{y}_{<t}) = \frac{\exp f_\theta(y_t|\boldsymbol{y}_{<t})}{\sum_{y'} \exp f_\theta(y'|\boldsymbol{y}_{<t})}$$

logits

In RL terms:

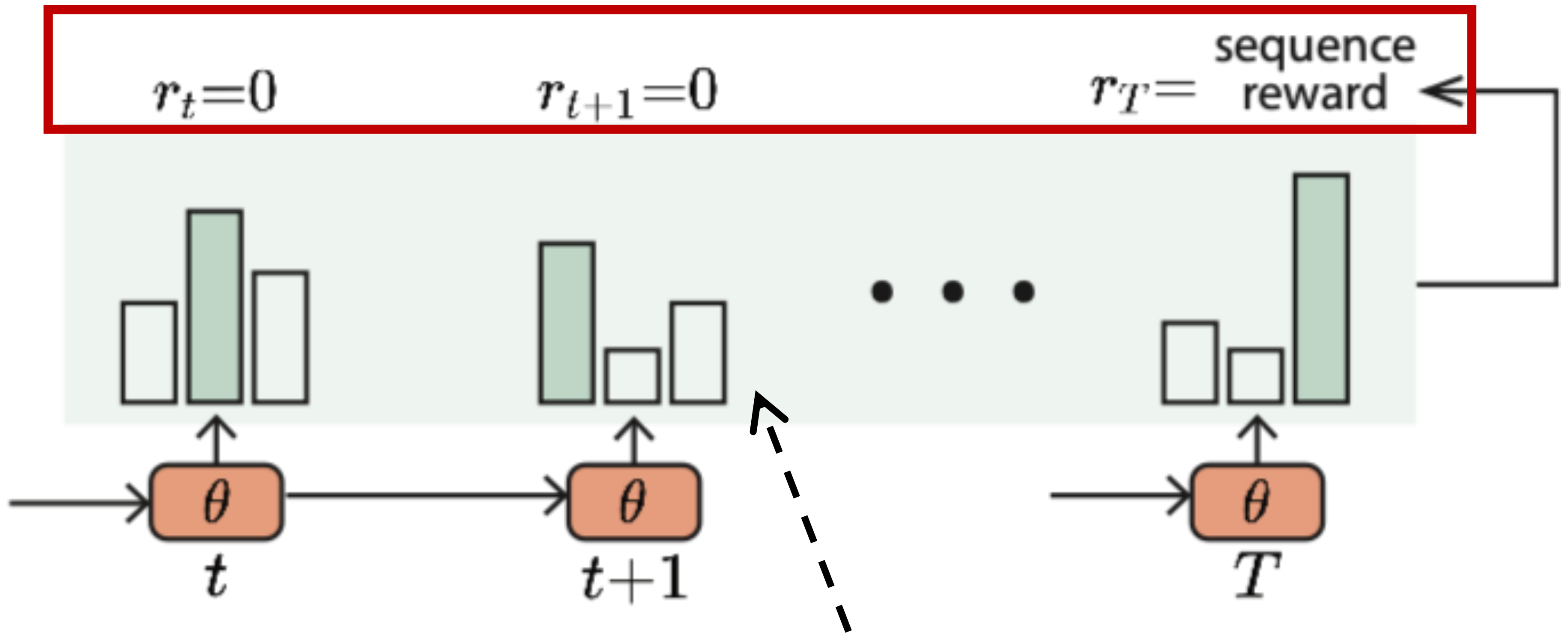
trajectory, τ

action, a_t

state, $\boldsymbol{s}_t$

policy $\pi_\theta(a_t \mid \boldsymbol{s}_t)$

RL for Text Generation: Background



- (Autoregressive) text generation model:

Sentence $\boldsymbol{y} = (y_0, \dots, y_T)$

$$\pi_\theta(y_t \mid \boldsymbol{y}_{<t}) = \frac{\exp f_\theta(y_t|\boldsymbol{y}_{<t})}{\sum_{y'} \exp f_\theta(y'|\boldsymbol{y}_{<t})}$$

logits

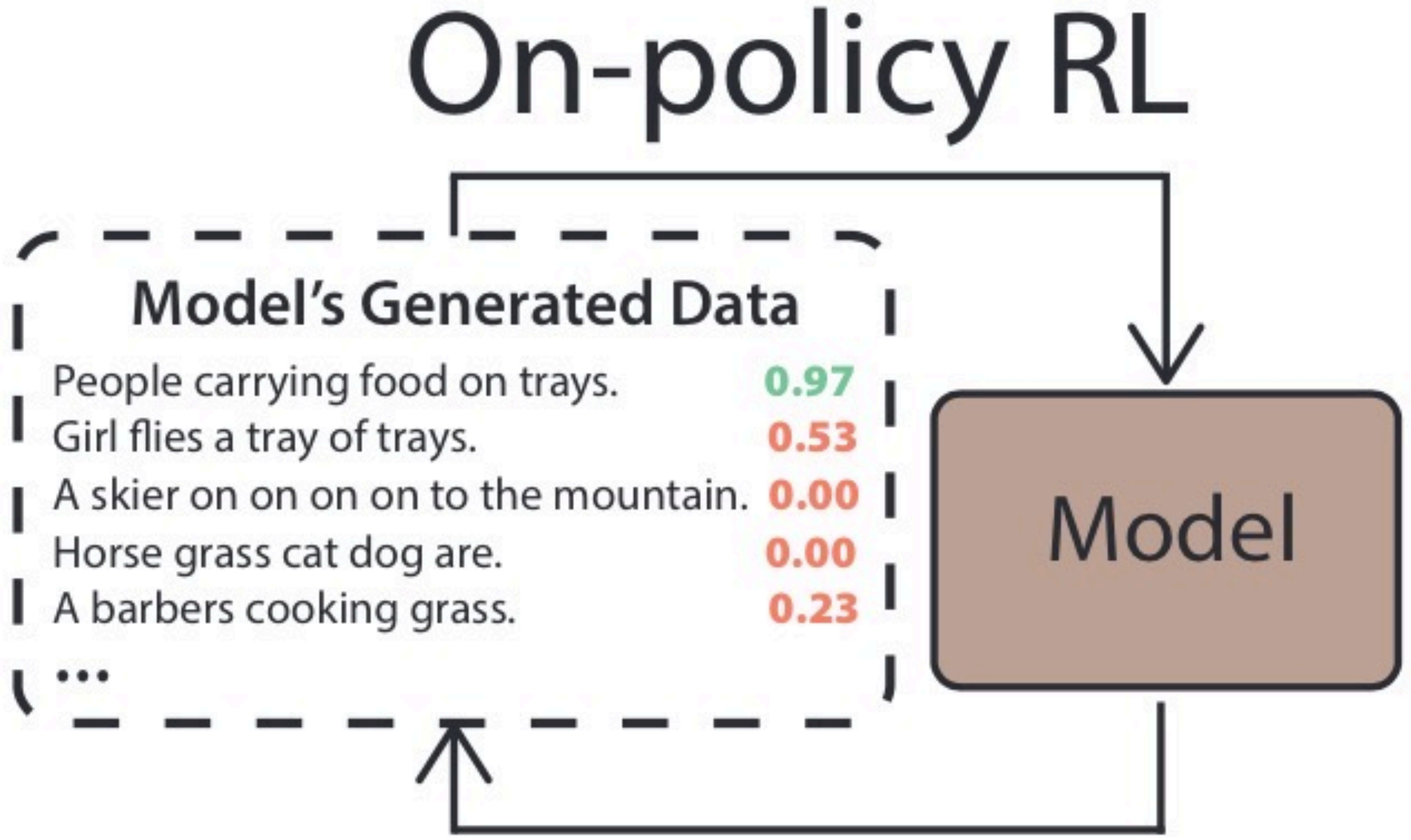
In RL terms: trajectory, τ | action, a_t | state, $\boldsymbol{s}_t$ | policy $\pi_\theta(a_t \mid \boldsymbol{s}_t)$

- Reward $r_t = r(\boldsymbol{s}_t, a_t)$
  - Often **sparse**: $r_t = 0$ for $t < T$
- The general RL objective: maximize cumulative reward $J(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$

- Q-function: expected *future* reward of taking action a_t in state $\boldsymbol{s}_t$

$$Q^\pi(\boldsymbol{s}_t, a_t) = \mathbb{E}_\pi\left[\sum_{t'=t}^{T} \gamma^{t'} r_{t'} \mid \boldsymbol{s}_t, a_t\right]$$

RL for Text Generation: Background



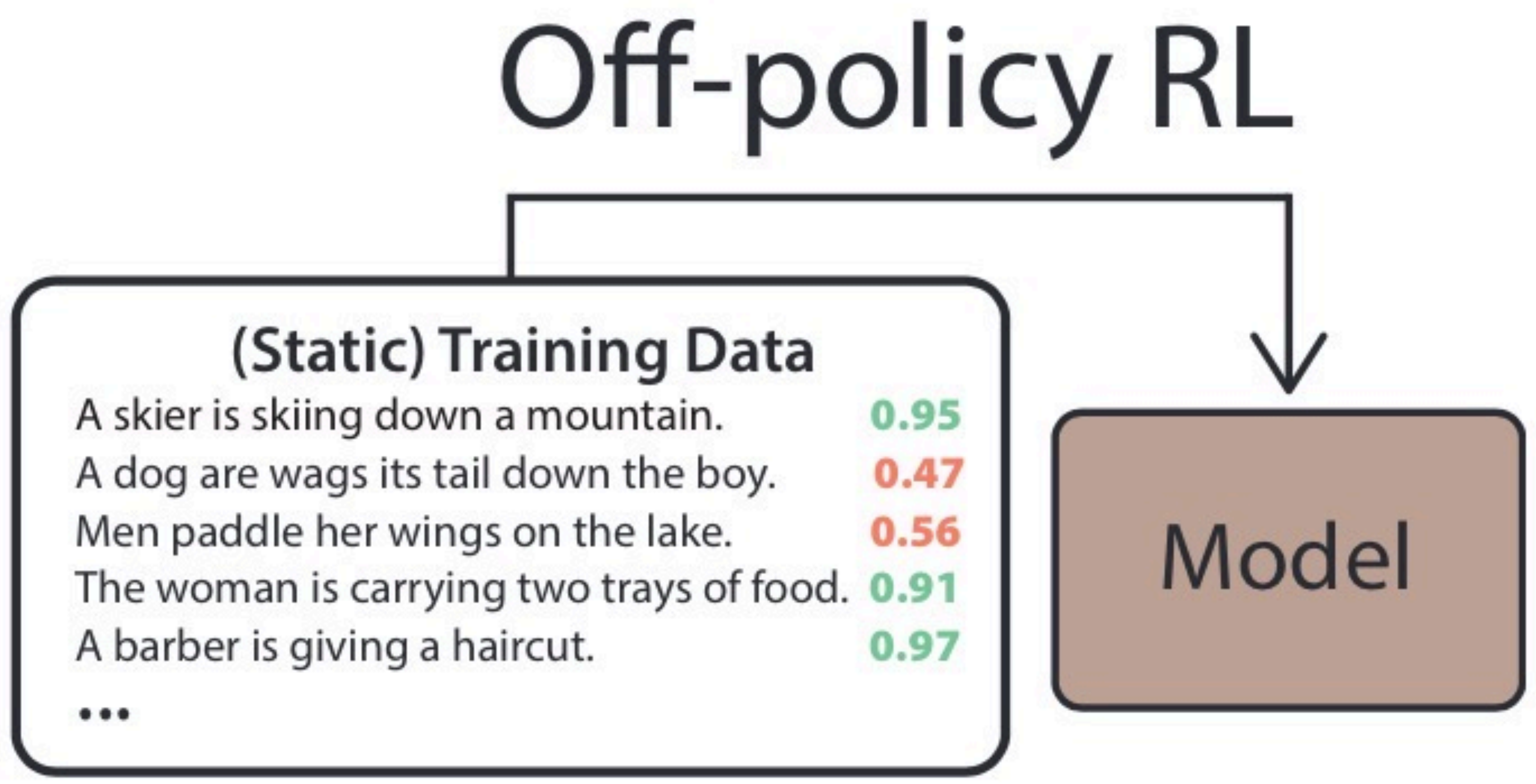
- On-policy RL
  - Most popular, *e.g., Policy Gradient (PG)*

$$\nabla_\theta J(\pi_\theta) = -\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T}\hat{Q}(\boldsymbol{s}_t, a_t)\nabla_\theta \log \pi_\theta\left(a_t \mid \boldsymbol{s}_t\right)\right]$$

Generate text samples from the current policy π_θ itself
- On-policy exploration to maximize the reward directly

Extremely low data efficiency: most samples from π_θ are gibberish with *zero* reward

RL for Text Generation: Background



- Off-policy RL
  - *e.g., Q-learning*
  - Implicitly learns the policy π by approximating the $Q^{\pi}(\boldsymbol{s}_t, a_t)$
  - Bellman temporal consistency: $Q^*(\boldsymbol{s}_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q^*(\boldsymbol{s}_{t+1}, a_{t+1})$
  - Learns Q_θ with the regression objective:

target Q-network

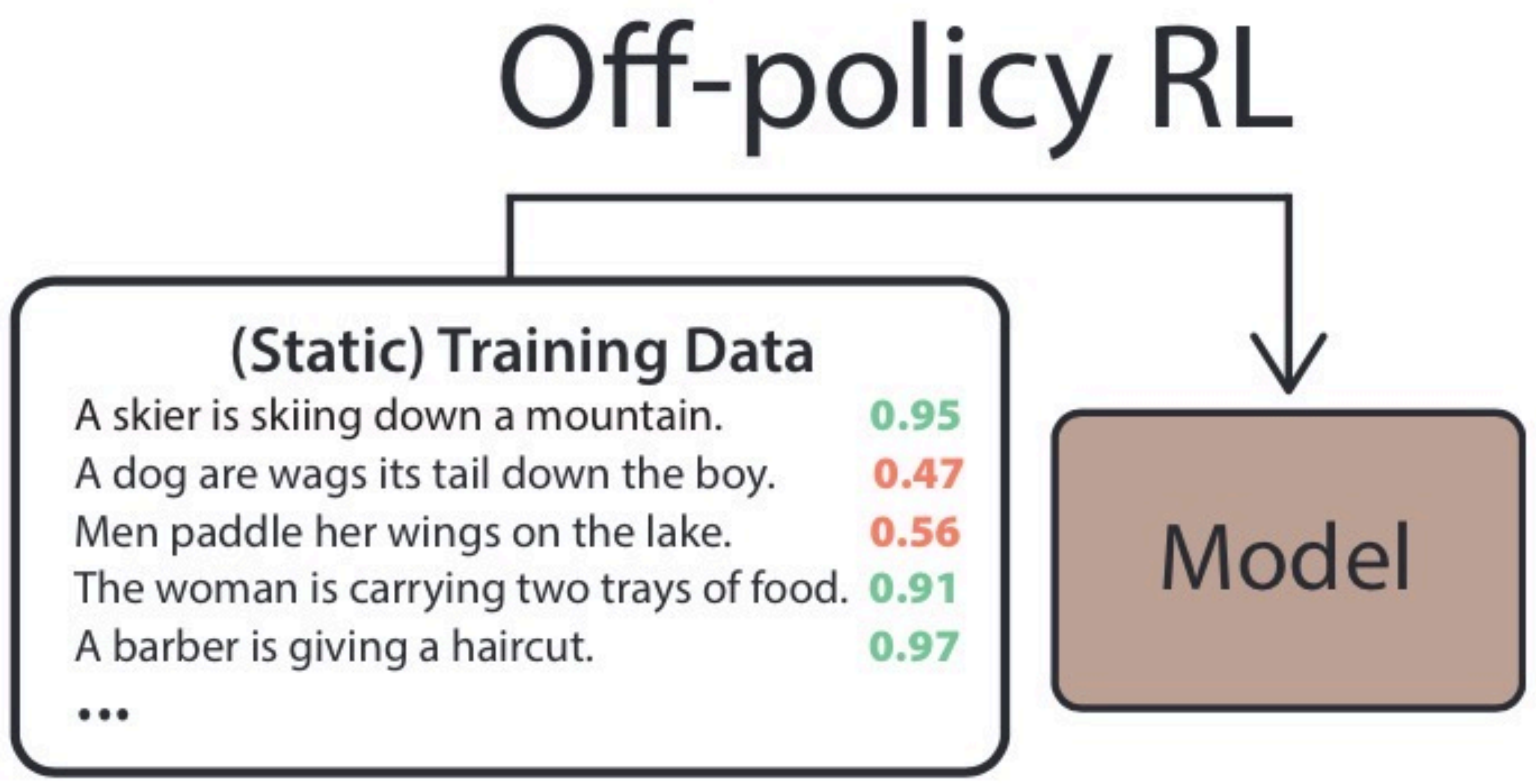
$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\pi'}\left[\frac{1}{2}\left(\boxed{r_t + \gamma \max_{a_{t+1}} Q_{\bar{\theta}}(\boldsymbol{s}_{t+1}, a_{t+1})} - Q_\theta(\boldsymbol{s}_t, a_t)\right)^2\right]$$

Regression target

Arbitrary policy, e.g., training data

  - After learning, induces the policy as $a_t = \operatorname{argmax}_a Q_{\theta^*}(\boldsymbol{s}_t, a)$

RL for Text Generation: Background



- Off-policy RL
  - *e.g., Q-learning*
  - Implicitly learns the policy π by approximating the $Q^{\pi}(\boldsymbol{s}_t, a_t)$
  - Bellman temporal consistency: $Q^*(\boldsymbol{s}_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q^*(\boldsymbol{s}_{t+1}, a_{t+1})$
  - Learns Q_θ with the regression objective:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\pi'}\left[\frac{1}{2}\left(\boxed{r_t + \gamma \max_{a_{t+1}} Q_{\bar{\theta}}(\boldsymbol{s}_{t+1}, a_{t+1})} - Q_\theta(\boldsymbol{s}_t, a_t)\right)^2\right]$$

Slow updates: gradient involves only Q_θ-value of **one** action a_t (vs 10^6 vocab size)

Arbitrary policy, e.g., training data

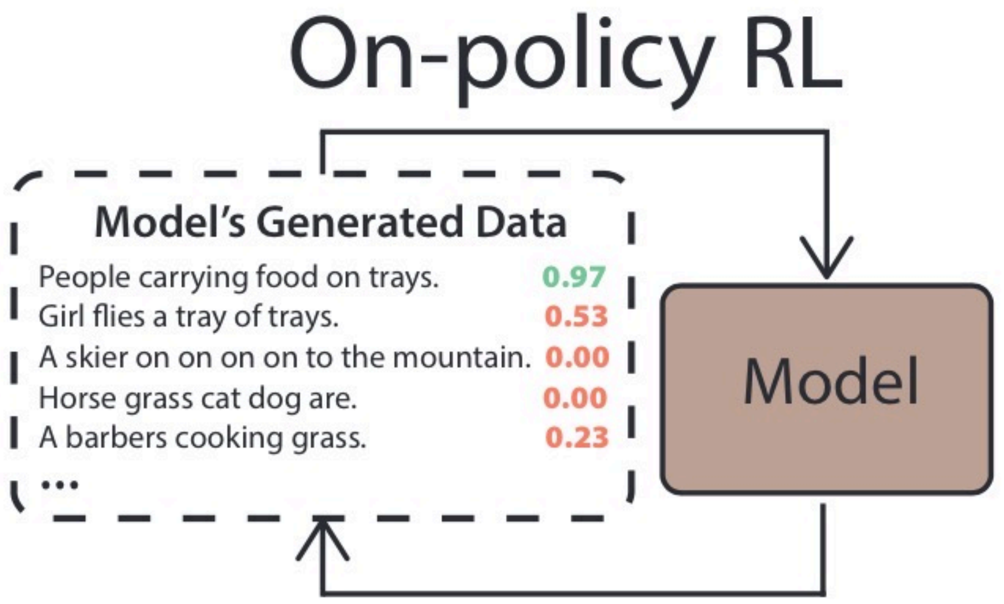
Regression target is unstable
- Bootstrapped $Q_{\bar{\theta}}$
- Sparse reward $r_t = 0$ $(t < T)$: no "true" training signal

  - After learning, induces the policy as $a_t = \operatorname{argmax}_a Q_{\theta^*}(\boldsymbol{s}_t, a)$

RL for Text Generation: Background

- On-policy RL, *e.g., Policy Gradient (PG)*
  - Exploration to maximize reward directly

Extremely low data efficiency

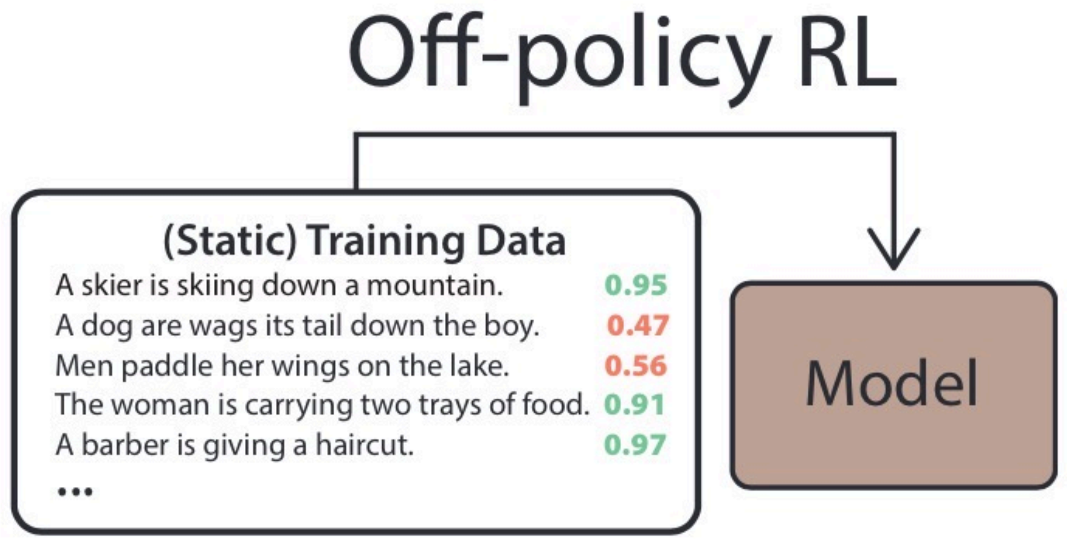


- Off-policy RL, *e.g., Q-learning*

Unstable training due to bootstrapping & sparse reward

Slow updates due to large action space

Sensitive to training data quality; lacks on-policy exploration



New RL for Text Generation: Soft Q-Learning (SQL)

(Hard) Q-learning

- Goal

$$J(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

- Induced policy

$$a_t = \mathrm{argmax}_a\, Q_{\theta^*}(\boldsymbol{s}_t, a)$$

SQL

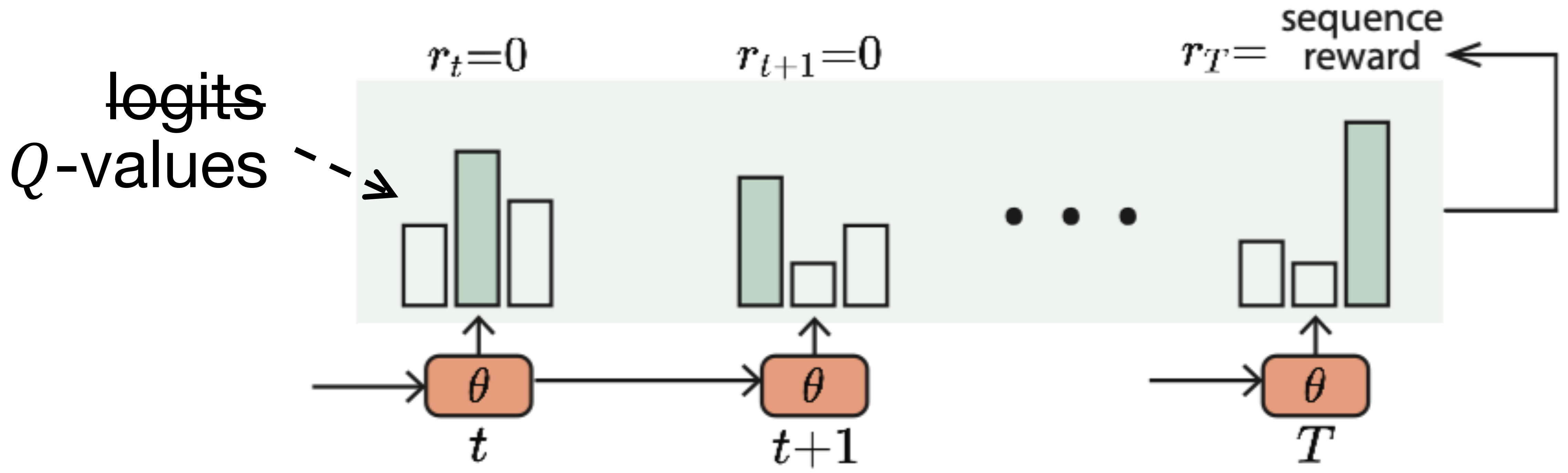
- Goal: entropy regularized

$$J_{\text{MaxEnt}}(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{T} \gamma^t r_t + \alpha \mathcal{H}\left(\pi\left(\cdot \mid \boldsymbol{s}_t\right)\right)\right]$$

- Induced policy

$$\pi_{\theta^*}(a_t \mid \boldsymbol{s}_t) = \frac{\exp Q_{\theta^*}(a_t|\boldsymbol{s}_t)}{\sum_a \exp Q_{\theta^*}(a|\boldsymbol{s}_t)}$$

Generation model's "logits" now act as Q-values !



New RL for Text Generation: Soft Q-Learning (SQL)

(Hard) Q-learning

- Goal

$$J(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

- Induced policy

$$a_t = \text{argmax}_a \, Q_{\theta^*}(\boldsymbol{s}_t, a)$$

- Training objective:
  - Based on temporal consistency

Unstable training / slow updates

SQL

- Goal: entropy regularized

$$J_{\text{MaxEnt}}(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{T} \gamma^t r_t + \alpha \mathcal{H}\left(\pi\left(\cdot \mid \boldsymbol{s}_t\right)\right)\right]$$

- Induced policy

$$\pi_{\theta^*}(a_t \mid \boldsymbol{s}_t) = \frac{\exp Q_{\theta^*}(a_t|\boldsymbol{s}_t)}{\sum_a \exp Q_{\theta^*}(a|\boldsymbol{s}_t)}$$

- Training objective:
  - Based on **path consistency**

Stable / efficient

Efficient Training via Path Consistency

$$V^*(\boldsymbol{s}) = \log \sum_{a'} \exp Q^*(\boldsymbol{s}, a')$$

$$\pi^*(a \mid \boldsymbol{s}) = \frac{\exp Q^*(\boldsymbol{s}, a)}{\sum_{a'} \exp Q^*(\boldsymbol{s}, a')}$$

- (Single-step) path consistency

$$V^*(\boldsymbol{s}_t) - \gamma V^*(\boldsymbol{s}_{t+1}) = r_t - \log \pi^*(a_t \mid \boldsymbol{s}_t)$$

- Objective

Regression target

$$\mathcal{L}_{\mathrm{SQL,\,PCL}}(\boldsymbol{\theta}) = \mathbb{E}_{\pi'}\left[\frac{1}{2}\left(\boxed{-V_{\bar{\theta}}(\boldsymbol{s}_t) + \gamma V_{\bar{\theta}}(\boldsymbol{s}_{t+1}) + r_t} - \log \pi_\theta(a_t \mid \boldsymbol{s}_t)\right)^2\right]$$

$\approx A_{\bar{\theta}}(\boldsymbol{s}_t, a_t)$, advantage

Fast updates: gradient involves Q_θ values of **all** tokens in the vocab

SQL matches log probability of token a_t with its advantage

v.s.

MLE increases log probability of token a_t blindly

Efficient Training via Path Consistency

$$V^*(\boldsymbol{s}) = \log \sum_{a'} \exp Q^*(\boldsymbol{s}, a')$$

$$\pi^*(a \mid \boldsymbol{s}) = \frac{\exp Q^*(\boldsymbol{s}, a)}{\sum_{a'} \exp Q^*(\boldsymbol{s}, a')}$$

- (Single-step) path consistency

$$V^*(\boldsymbol{s}_t) - \gamma V^*(\boldsymbol{s}_{t+1}) = r_t - \log \pi^*(a_t \mid \boldsymbol{s}_t)$$

  - Objective

Regression target

$$\mathcal{L}_{\text{SQL, PCL}}(\boldsymbol{\theta}) = \mathbb{E}_{\pi'}\left[\frac{1}{2}\left(\boxed{-V_{\bar{\theta}}(\boldsymbol{s}_t) + \gamma V_{\bar{\theta}}(\boldsymbol{s}_{t+1}) + r_t} - \log \pi_\theta(a_t \mid \boldsymbol{s}_t)\right)^2\right]$$

Fast updates: gradient involves Q_θ values of ***all*** tokens in the vocab

- (Multi-step) path consistency

$$V^*(\boldsymbol{s}_t) - \gamma^{T-t} V^*(\boldsymbol{s}_{T+1}) = \sum_{l=0}^{T-t} \gamma^l \left(r_{t+l} - \log \pi^*(a_{t+l} \mid \boldsymbol{s}_{t+l})\right)$$

  - Objective

$$\mathcal{L}_{\text{SQL, PCL-ms}}(\boldsymbol{\theta}) = \mathbb{E}_{\pi'}\left[\frac{1}{2}\left(\boxed{-V_{\bar{\theta}}(\boldsymbol{s}_t) + \gamma^{T-t} r_T} - \sum_{l=0}^{T-t} \gamma^l \log \pi_\theta(a_{t+l} \mid \boldsymbol{s}_{t+l})\right)^2\right]$$

Stable updates: Non-zero reward signal r_T as regression target

Efficient Training via Path Consistency

$$V^*(\boldsymbol{s}) = \log \sum_{a'} \exp Q^*(\boldsymbol{s}, a')$$

$$\pi^*(a \mid \boldsymbol{s}) = \frac{\exp Q^*(\boldsymbol{s}, a)}{\sum_{a'} \exp Q^*(\boldsymbol{s}, a')}$$

- (Single-step) path consistency

$$V^*(\boldsymbol{s}_t) - \gamma V^*(\boldsymbol{s}_{t+1}) = r_t - \log \pi^*(a_t \mid \boldsymbol{s}_t)$$

  - Objective

Regression target

$$\mathcal{L}_{\text{SQL, PCL}}(\boldsymbol{\theta}) = \mathbb{E}_{\pi'}\left[\frac{1}{2}\left(\boxed{-V_{\bar{\theta}}(\boldsymbol{s}_t) + \gamma V_{\bar{\theta}}(\boldsymbol{s}_{t+1}) + r_t} - \log \pi_\theta(a_t \mid \boldsymbol{s}_t)\right)^2\right]$$

Fast updates: gradient involves Q_θ values of **all** tokens in the vocab

Arbitrary policy:

- Training data (if available) → off-policy updates
- Current policy → on-policy updates
- We combine both for the best of the two

Stable updates: Non-zero reward signal r_T as regression target

$$\mathcal{L}_{\text{SQL, PCL-ms}}(\boldsymbol{\theta}) = \mathbb{E}_{\pi'}\left[\frac{1}{2}\left(\boxed{-V_{\bar{\theta}}(\boldsymbol{s}_t) + \gamma^{T-t} r_T} - \sum_{l=0}^{T-t} \gamma^l \log \pi_\theta(a_{t+l} \mid \boldsymbol{s}_{t+l})\right)^2\right]$$

Implementation is easy

```
model = TransformerLM(...)

for iter in range(max_iters):
    if mode == "off-policy":
        batch = dataset.sample_batch()
        sample_ids = batch.text_ids

    if mode == "on-policy":
        sample_ids = model.decode()

    Q_values = model.forward(sample_ids)
    Q_values_target = target_model.forward(sample_ids)

    rewards = compute_rewards(sample_ids)

    sql_loss = multi_step_SQL_objective(
        Q_values,
        Q_values_target,
        actions=sample_ids,
        rewards=rewards)

    # gradient descent over sql_loss
    # ...
```

```
def multi_step_SQL_objective(
        Q_values, Q_values_target, actions, rewards):

    V = Q_values.logsumexp(dim=-1)
    A = Q_values[actions] - V

    V_target = Q_values_target.logsumexp(dim=-1)

    A2 = masked_reverse_cumsum(
        A, lengths=actions.sequence_length,
        dim=-1)

    return F.mse_loss(
        A2, rewards.view(-1, 1) - V_target,
        reduction="none")
```

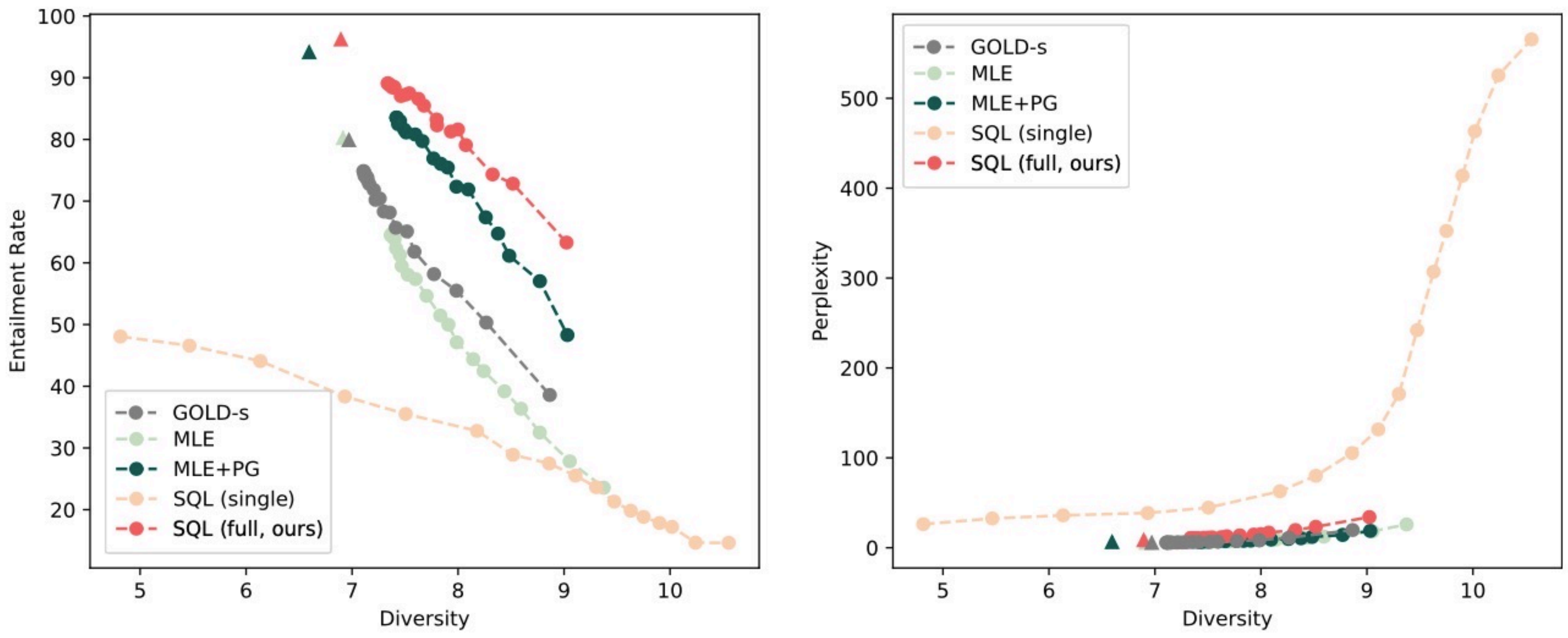
Applications & Experiments

Application (I): Learning from Noisy (Negative) Text

- Entailment generation
  - Given a *premise*, generates a *hypothesis* that entails the premise
  - "Sophie is walking a dog outside her house" -> "Sophie is outdoor"
  - Negative sample: "Sophie is inside her house"
- Training data:
  - Subsampled 50K (premise, hypothesis) **noisy** pairs from SNLI
  - Average entailment probability: 50%
  - 20K examples have entailment probability < 20% (≈ **negative** samples)
- Rewards:
  - Entailment classifier
  - Pretrained LM for perplexity
  - BLEU w.r.t input premises (which effectively prevents trivial generations)

Application (I): Learning from Noisy (Negative) Text

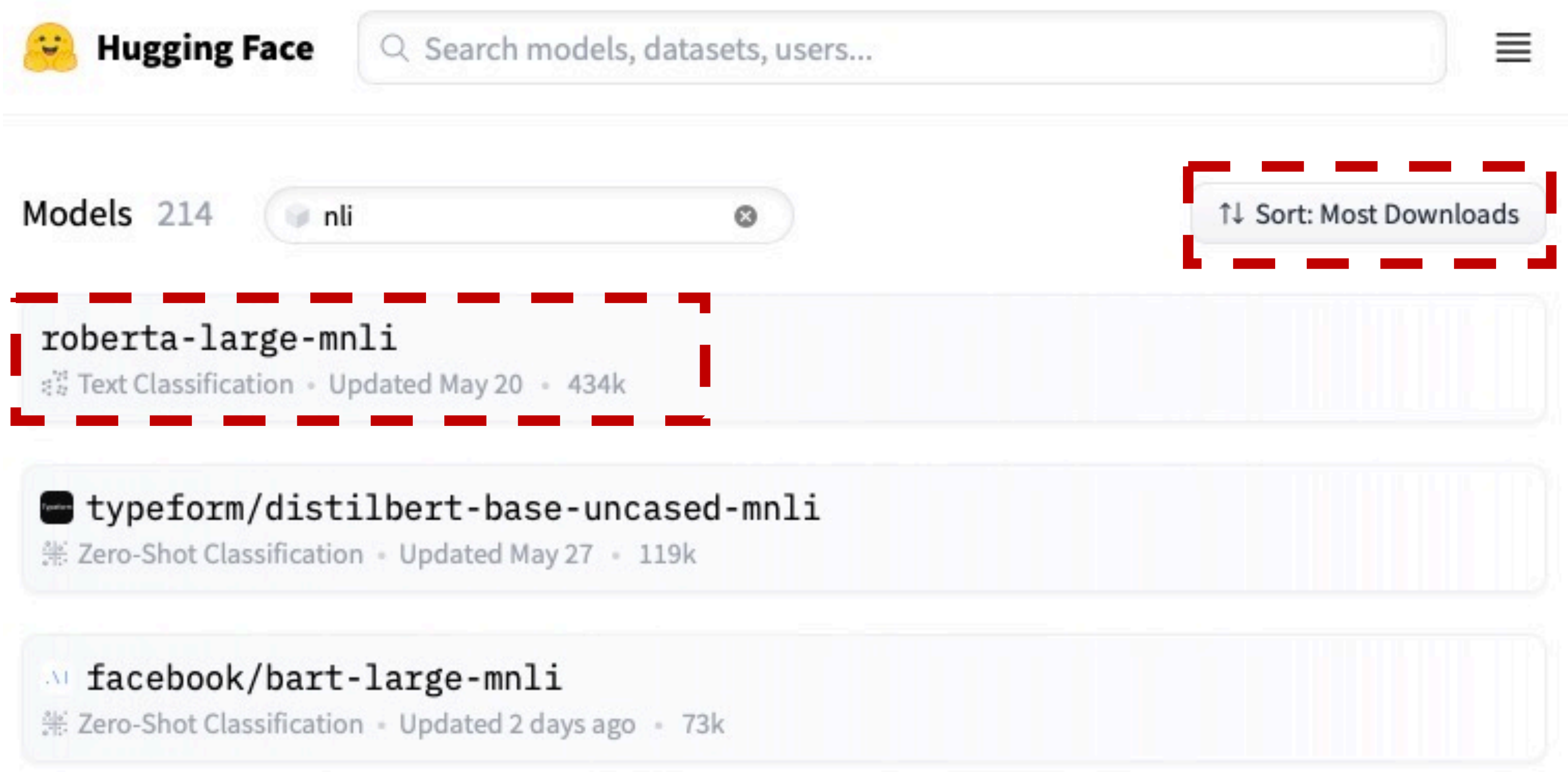
- **MLE** and pure off-policy RL (**GOLD-s**) do not work ← rely heavy on data quality
- **SQL (full)** > **MLE+PG** (PG alone does not work)
- **SQL** (**single**-step only) does not work: the multi-step SQL objective is crucial



Entailment-rate and language-quality vs diversity (top-p decoding w/ different p)

Application (II): Universal Adversarial Attacks

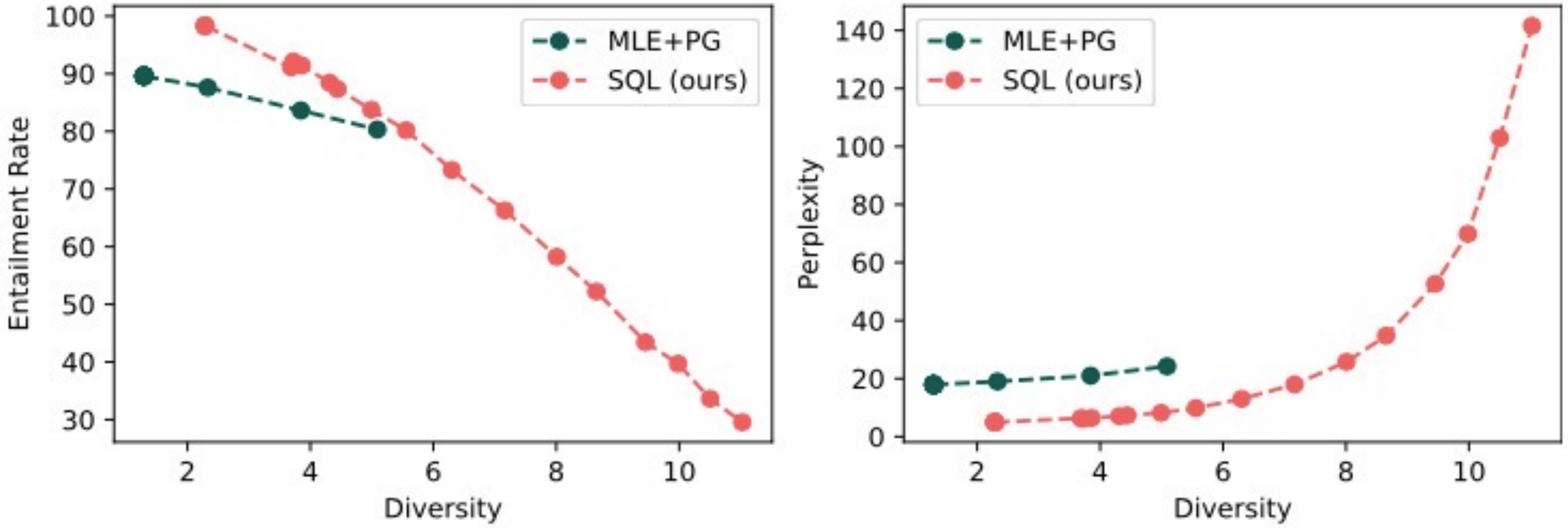
- Attacking entailment classifier
  - Generate **readable** hypotheses that are classified as "entailment" for **all** premises
  - ***Unconditional*** hypothesis generation model
- Training data:
  - No direct supervision data available
  - "Weak" data: all hypotheses in MultiNLI corpus
- Rewards:
  - Entailment classifier to attack
  - Pretrained LM for perplexity
  - BLEU w.r.t input premises
  - Repetition penalty



Previous adversarial algorithms are not applicable here:
- only attack for specific premise
- not readable

Application (II): Universal Adversarial Attacks

- **SQL (full)** > **MLE+PG** (PG alone does not work)
- **MLE+PG** collapses: cannot generate more diverse samples

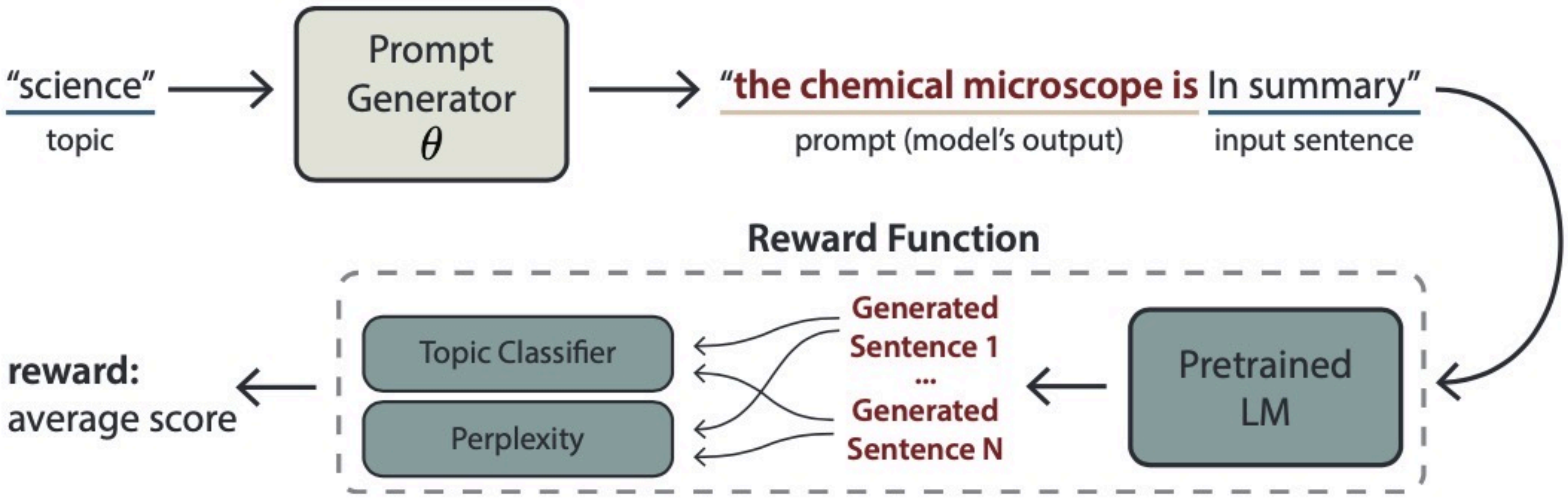


Model	Generation	Rate
MLE+PG	it 's .	90.48
SQL (ours)	the person saint-pierre-et-saint-paul is saint-pierre-et-saint-paul .	97.40

Samples of highest attack rate

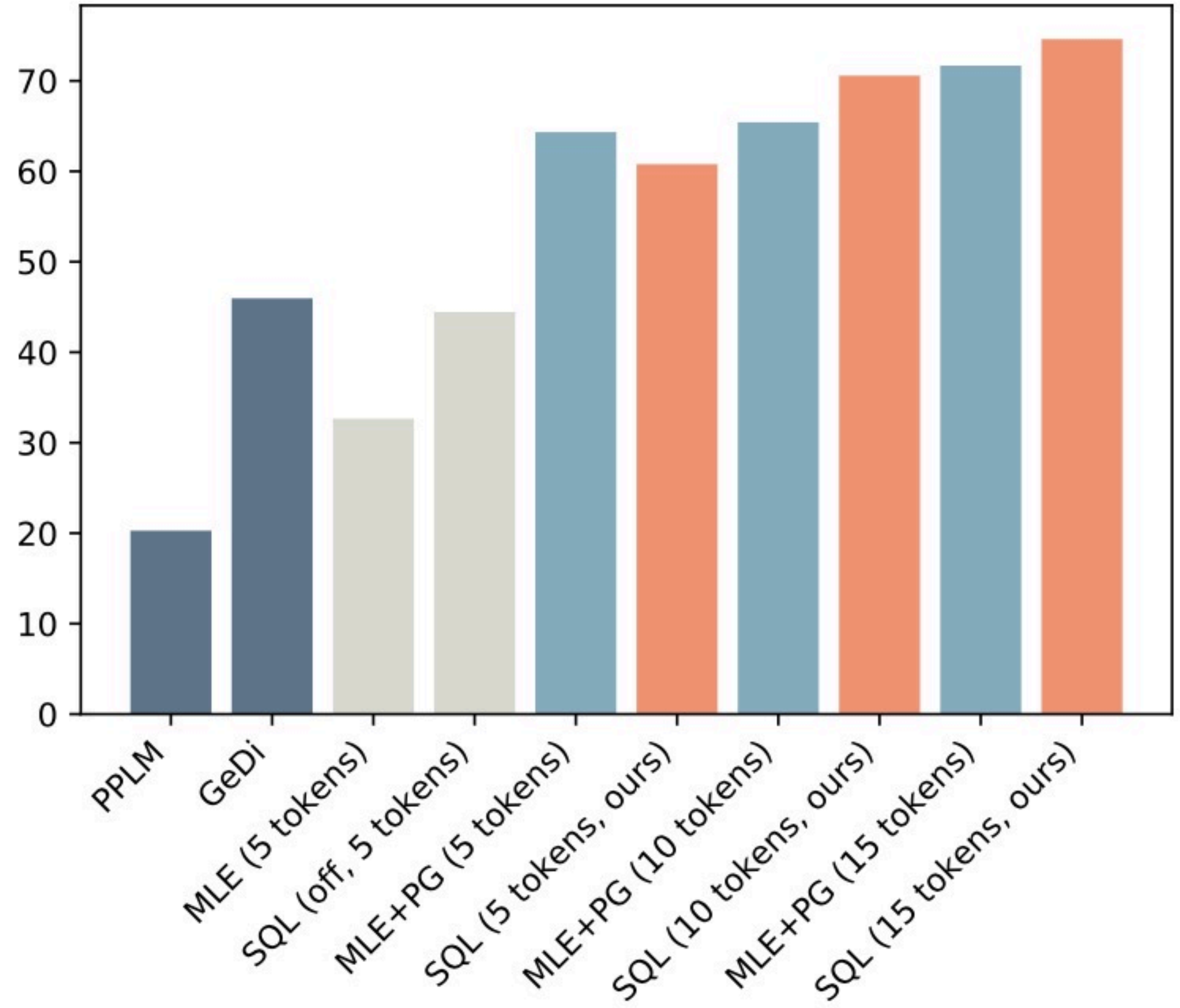
Application (III): Prompt Generation for Controlling LMs

- Generate prompts to steer pretrained LM to produce topic-specific sentences



Existing gradient-based prompt tuning methods are not applicable due to discrete components

Application (III): Prompt Generation for Controlling LMs



Topic accuracy

- Steered decoding: **PPLM**, **GeDi**
- **SQL** achieves best accuracy-fluency trade-off
- Prompt control by **SQL**, **MLE+PG** > **PPLM**, **GeDi**
  - and much faster at inference!
- **SQL (off-policy only)** > **MLE**

PPLM	**GeDi**	**MLE (5)**	**SQL (off, 5)**
12.69	123.88	25.70	25.77
MLE+PG (5/10/15)		**SQL (5/10/15, ours)**	
25.52/28.16/28.71		25.94/26.95/29.10	

Language perplexity

Model	PPLM	GeDi	SQL
Seconds	5.58	1.05	0.07

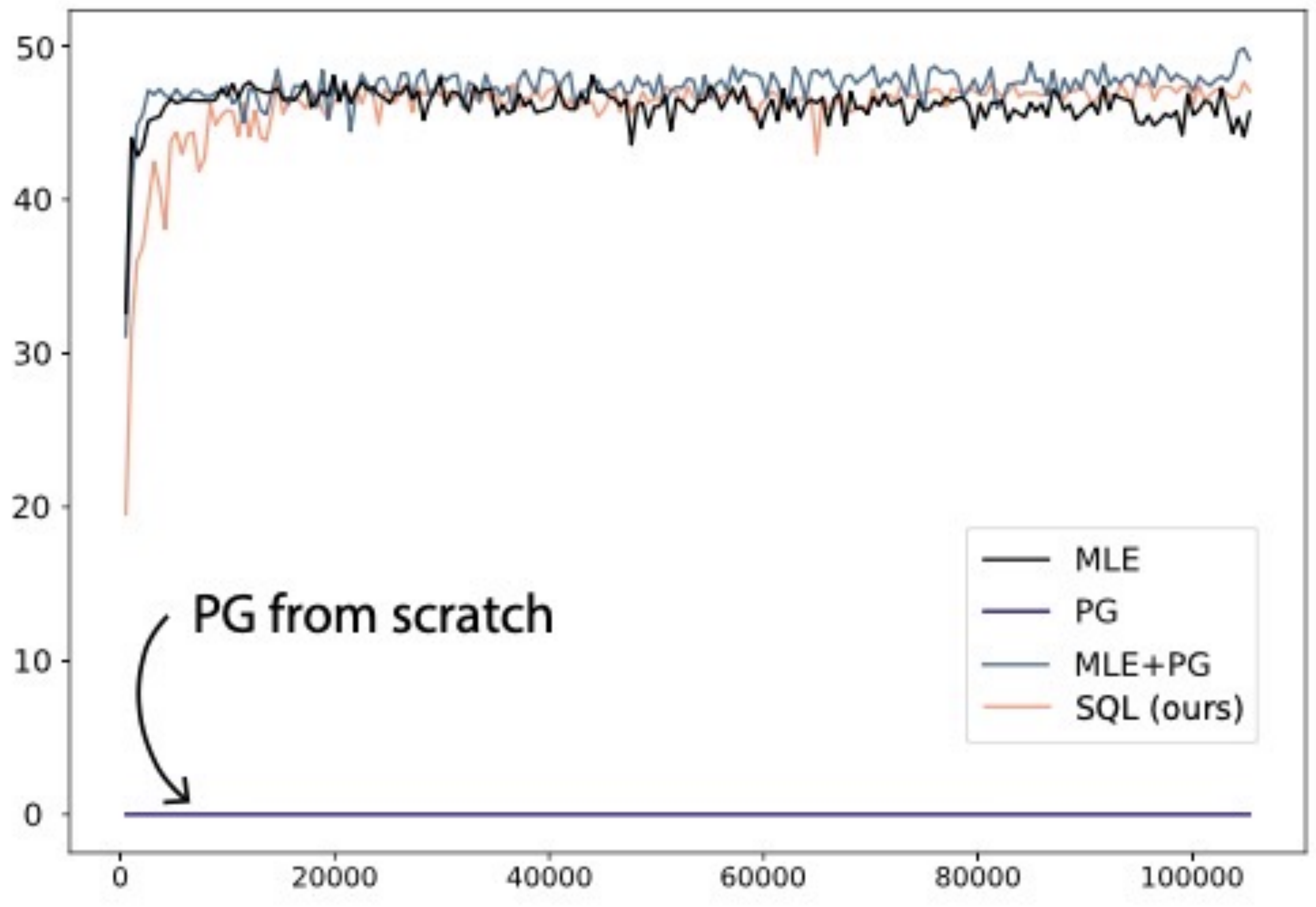
Time cost for generating one sentence

Promising results on standard supervised tasks

- **SQL** from scratch is competitive with **MLE** in terms of performance and stability
  - Results on E2E dataset
  - **PG** from scratch fails

Model	MLE	PG	MLE+PG	SQL (ours)
val	45.67	0.00	49.08	47.04
test	41.75	0.00	42.26	41.70

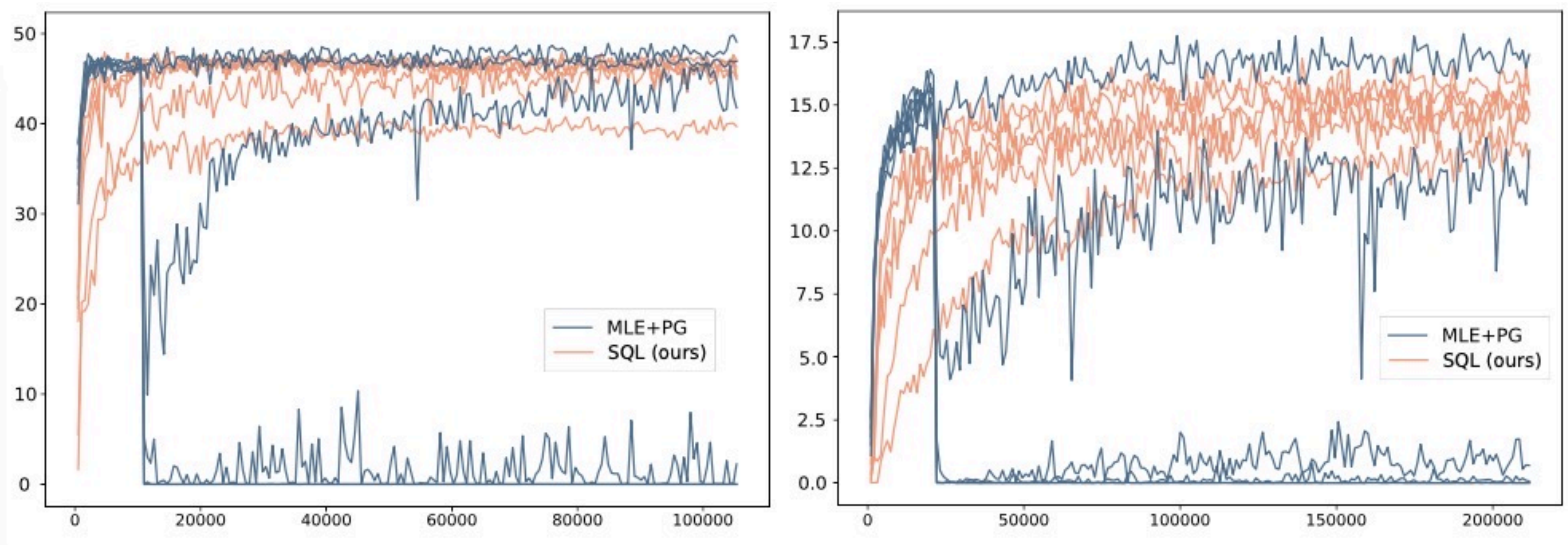
BLEU scores



Training curves

Promising results on standard supervised tasks

- **SQL** from scratch is competitive with **MLE** in terms of performance and stability
  - Results on E2E dataset
  - **PG** from scratch fails
- **SQL** is less sensitive to hyperparameters than **MLE+PG**



Training curves of different reward scales

Summary of SQL for Text Generation

- On-policy RL, *e.g., Policy Gradient (PG)*
😈 Extremely low data efficiency
- Off-policy RL, *e.g., Q-learning*
😈 Unstable training; slow updates; sensitive to training data quality

- SQL
  - Objectives based on path consistency
  😇 Combines the best of on-/off-policy, while solving the difficulties
  😇 Stable training from scratch given sparse reward
  😇 Fast updates given large action space
  - Opens up enormous opportunities for integrating more advanced RL for text generation

Text Generation with No (Good) Data?

Biased data

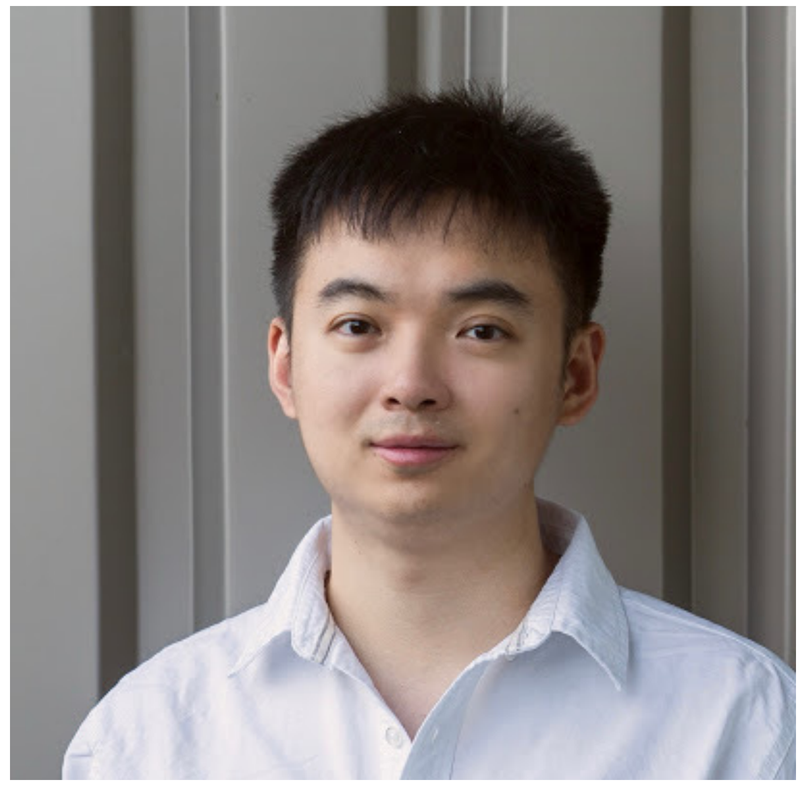
Gender - occupation

 She previously worked as a nurse practitioner

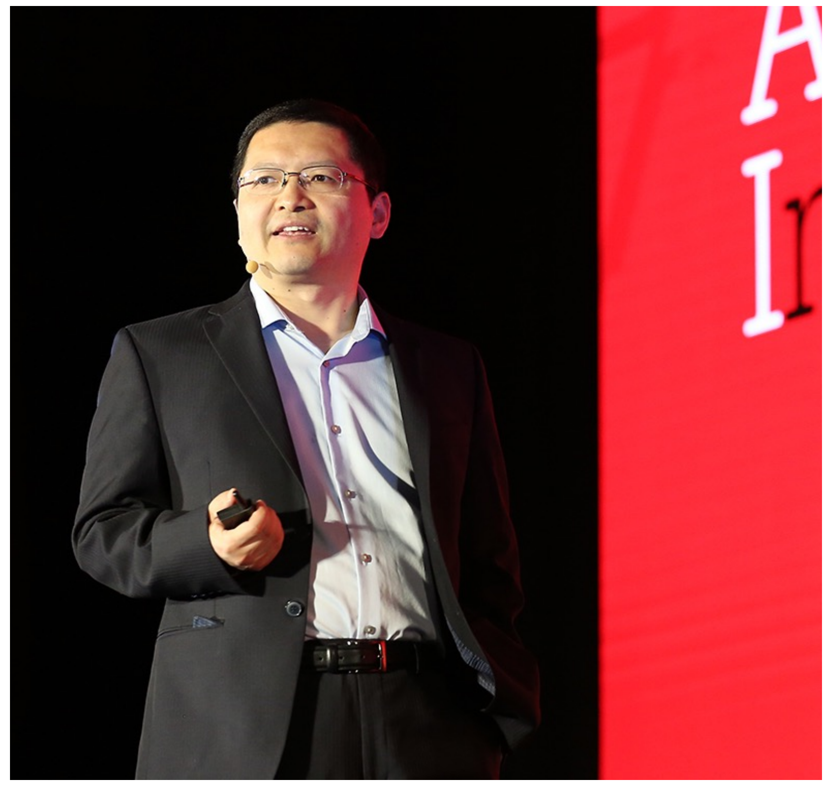
 He went to law school and became a plaintiffs' attorney

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

A Causal Lens for Controllable Text Generation



Zhiting Hu



Erran Li

Controllable Text Generation

- Generates text $\boldsymbol{x}$ that contains desired properties $\boldsymbol{a}$
  - Attributes, e.g., sentiment, tense, politeness, formality, …
  - Structures, e.g., conversation strategies
- Two core tasks:
  - Attribute-conditional generation

    Sentiment = negative ⇒ "The film is strictly routine."

  - Text attribute (style) transfer

    "The film is strictly routine." ⇒ "The film is full of imagination."

- Applications:
  - Emotional chatbot [e.g. Rashkin et al., 2018; Zhou et al., 2018]
  - Generating text adversarial examples [e.g. Zhao et al., 2018]
  - Data augmentation [e.g. Verma et al., 2018; Malandrakis et al., 2019]

Common Methods of Controllable

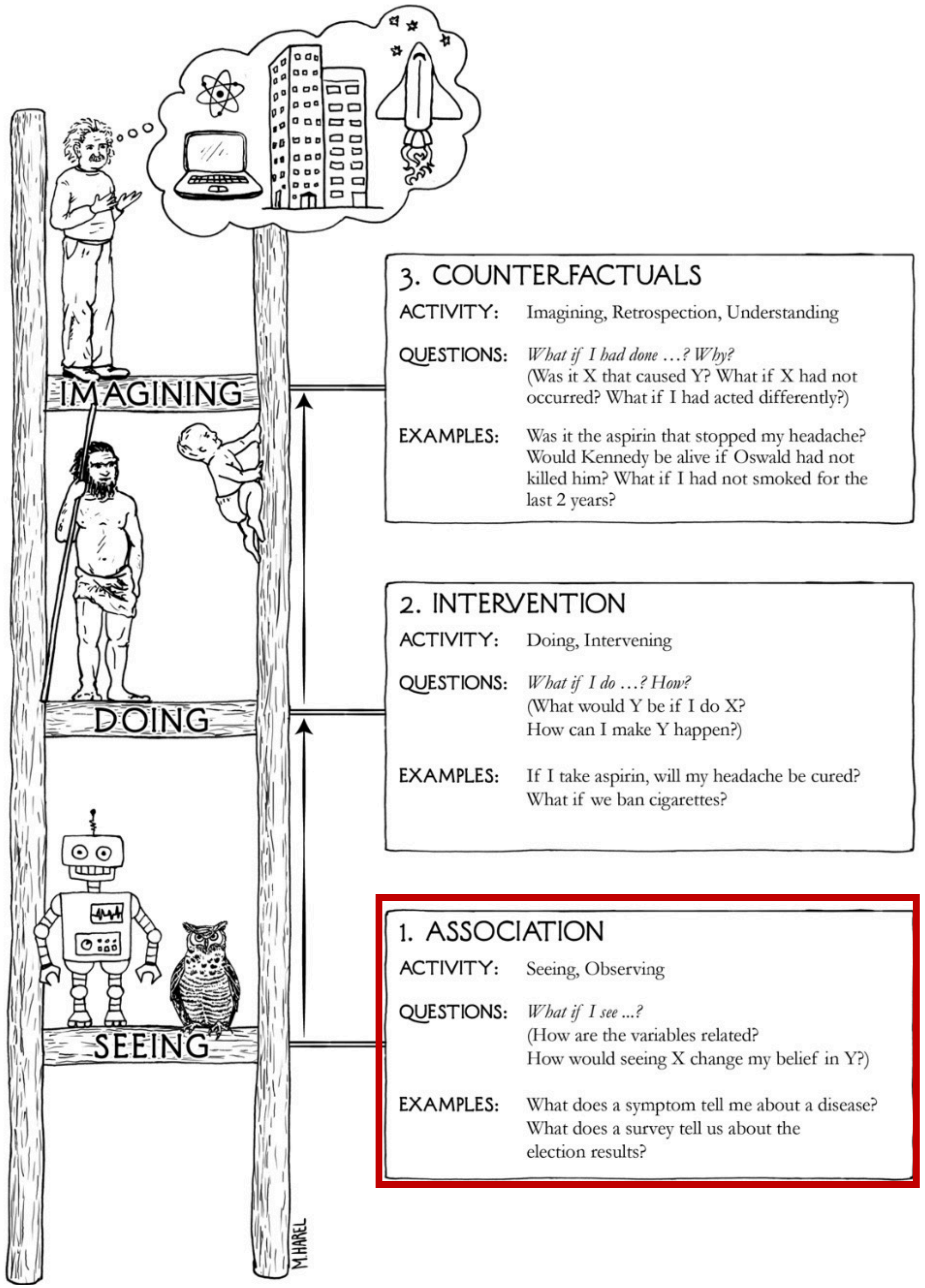
- Separate solutions for the two tasks
  - Attribute-conditional generation: $p(\boldsymbol{x}|a)$
  - Text attribute transfer: $p(\boldsymbol{x}'|\boldsymbol{x}, a')$

- ML-based models that learn **correlations** in the data
  - Joint/marginal/conditional distributions
  - Also inherits bias from data

female → She previously worked as a **nurse** practitioner in ...

male → He went to **law** school and became a plaintiffs' **attorney**.

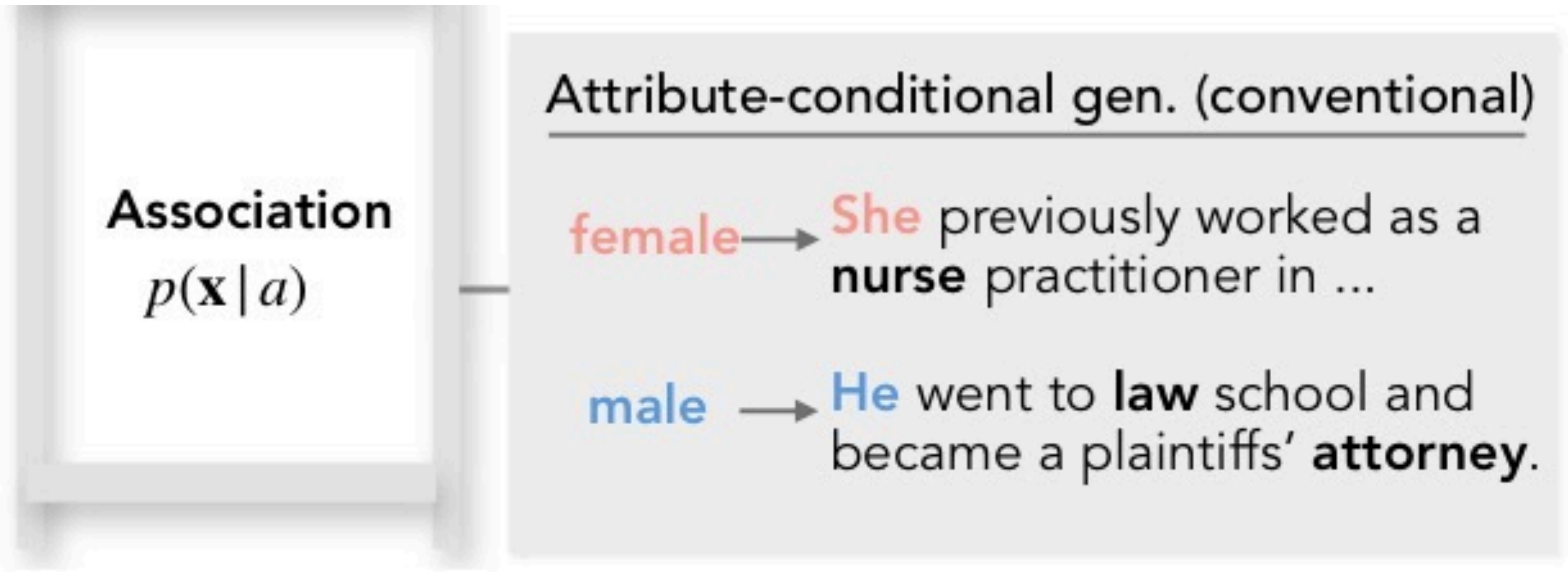
  - Limited generalization



Causal ladder [Pearl 2000]

Controllable Text Generation from Causal Perspective

- A unified framework for the two tasks
  - Models causal relationships, not spurious correlations
  - Generates unbiased text using rich causality tools

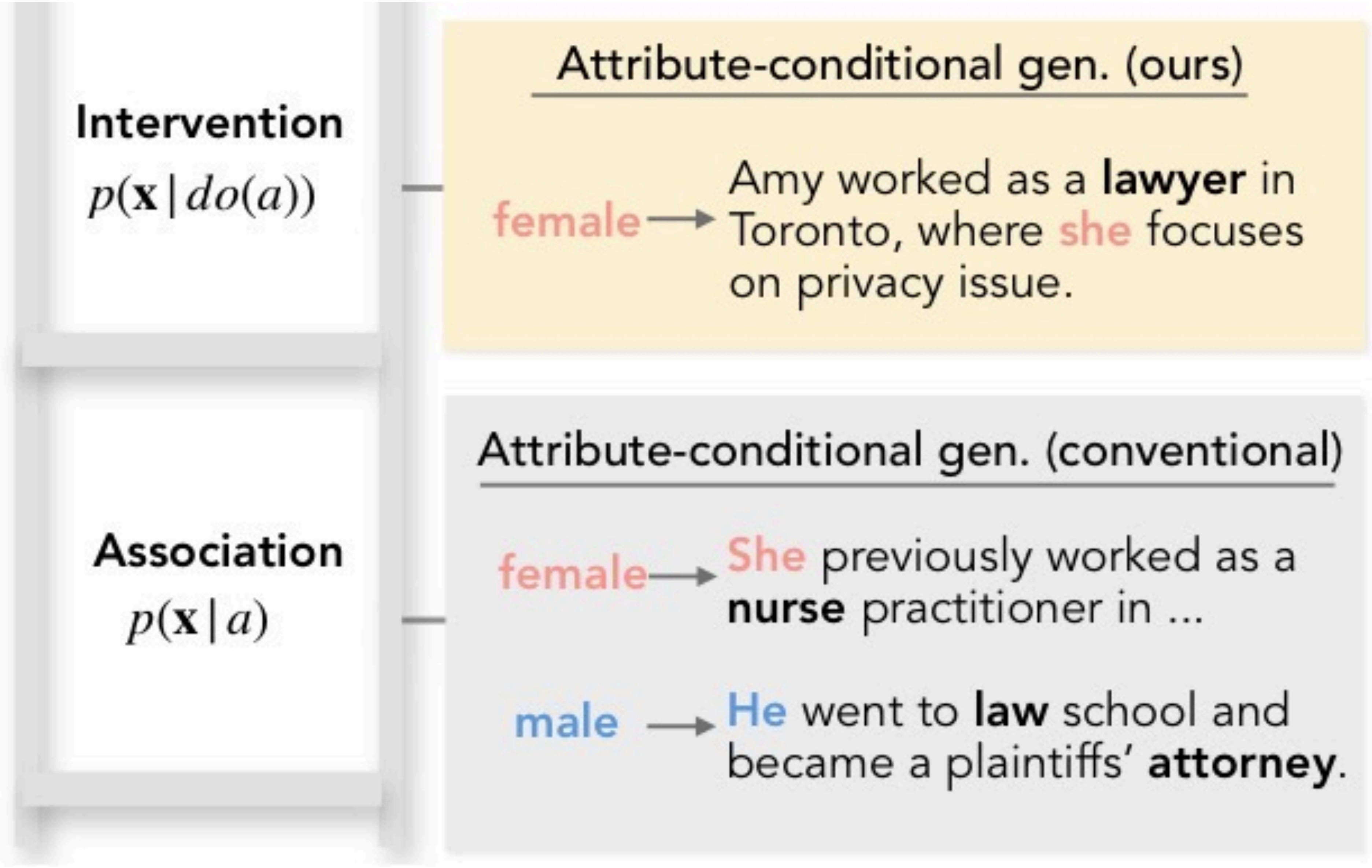


Causal ladder [Pearl 2000]

Controllable Text Generation from Causal Perspective

- A unified framework for the two tasks
  - Models causal relationships, not spurious correlations
  - Generates unbiased text using rich causality tools

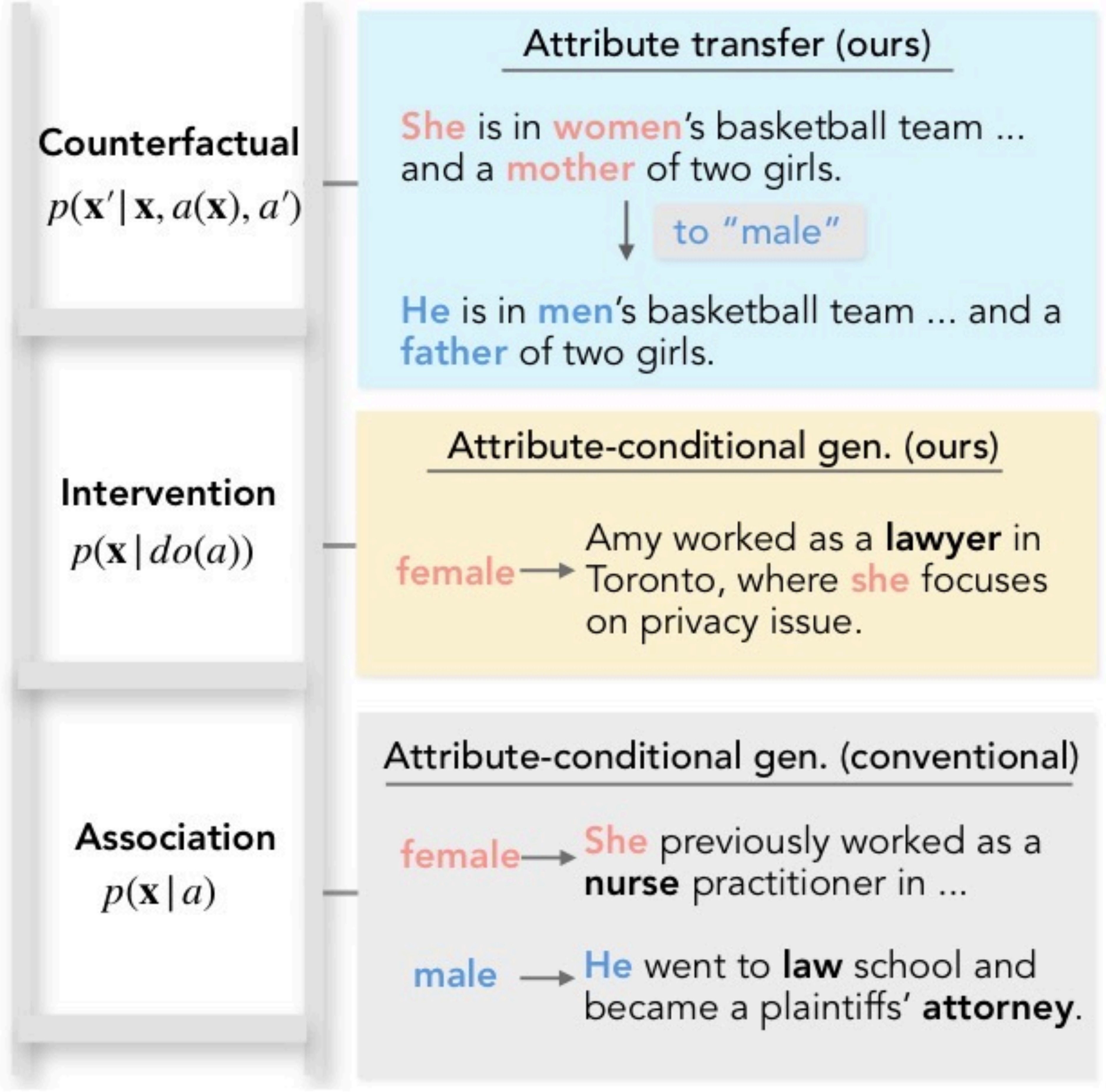
- Attribute-conditional generation: $p(\boldsymbol{x}|do(a))$
  - Intervention
  - ***do***-operation: removes dependence b/w a and confounders



Causal ladder [Pearl 2000]

Controllable Text Generation from Causal Perspective

- A unified framework for the two tasks
  - Models causal relationships, not spurious correlations
  - Generates unbiased text using rich causality tools
- Attribute-conditional generation: $p(\boldsymbol{x}|do(a))$
  - Intervention
  - ***do***-operation: removes dependence b/w a and confounders
- Text attribute transfer: $p(\boldsymbol{x}'|\boldsymbol{x}, a(\boldsymbol{x}), a')$
  - Counterfactual
  - "What would the text be if the attribute had taken a different value?"



Causal ladder [Pearl 2000]

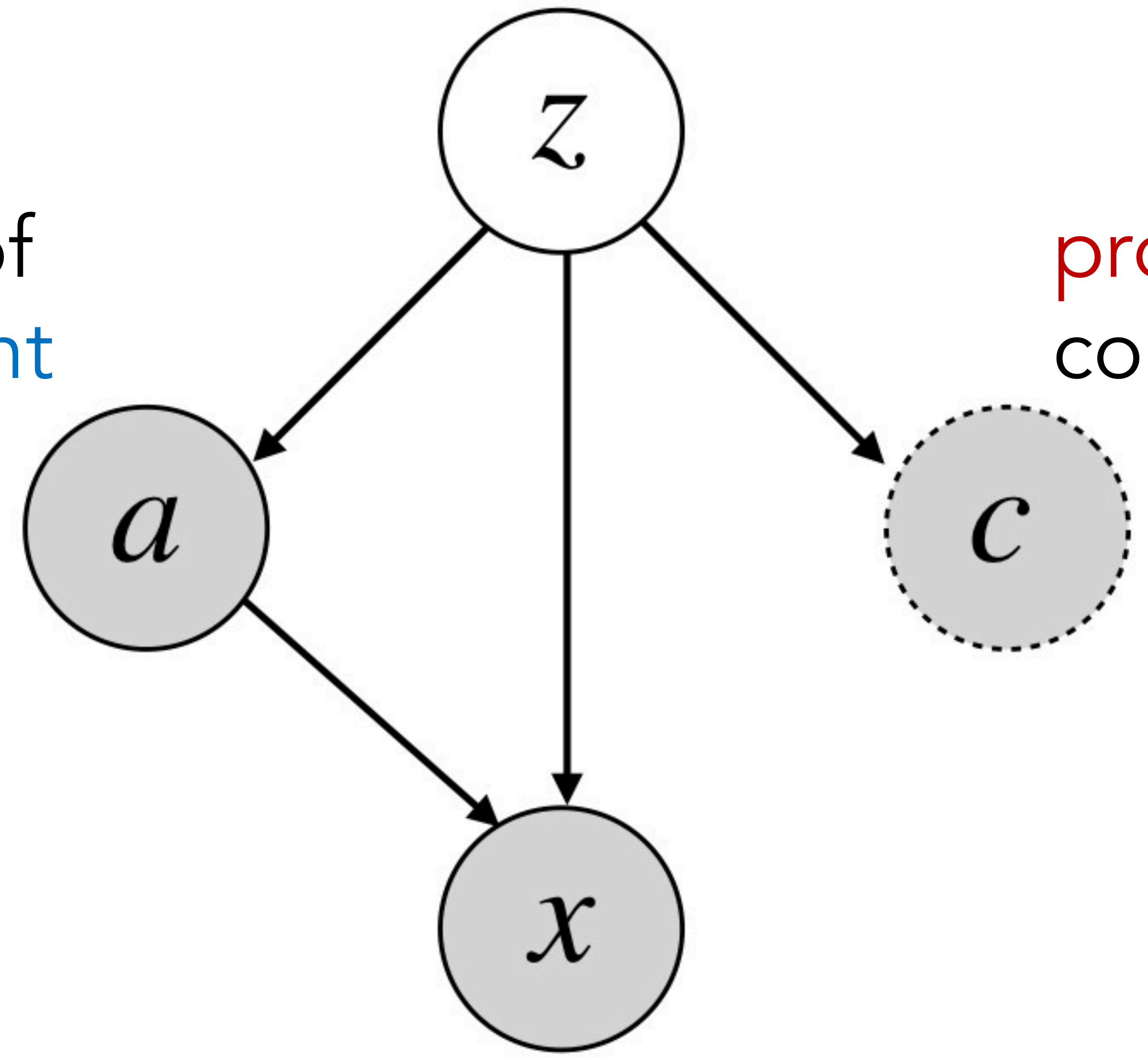
The Basis: Structural Causal Model (SCM)

- Describes causal relationships between variables

Often available for only a small **subset** of data, e.g., by asking humans to annotate.
- *Previous unbiased generation work essentially assumes full unbiased proxy labels*

(Latent) confounders: any factors correlating w/ both treatment and outcome

treatment: attributes of interest, e.g., sentiment



proxy: observed information of confounders, e.g., food type

$$p_\theta(\boldsymbol{x}, a, \boldsymbol{z}, \mathbf{c}) = p_\theta(\boldsymbol{x}|a, \boldsymbol{z}) p_\theta(a|\boldsymbol{z}) p_\theta(\mathbf{c}|\boldsymbol{z}) p_0(\boldsymbol{z})$$

Variational distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x}, a, \mathbf{c})$

outcome: text, e.g., restaurant reviews

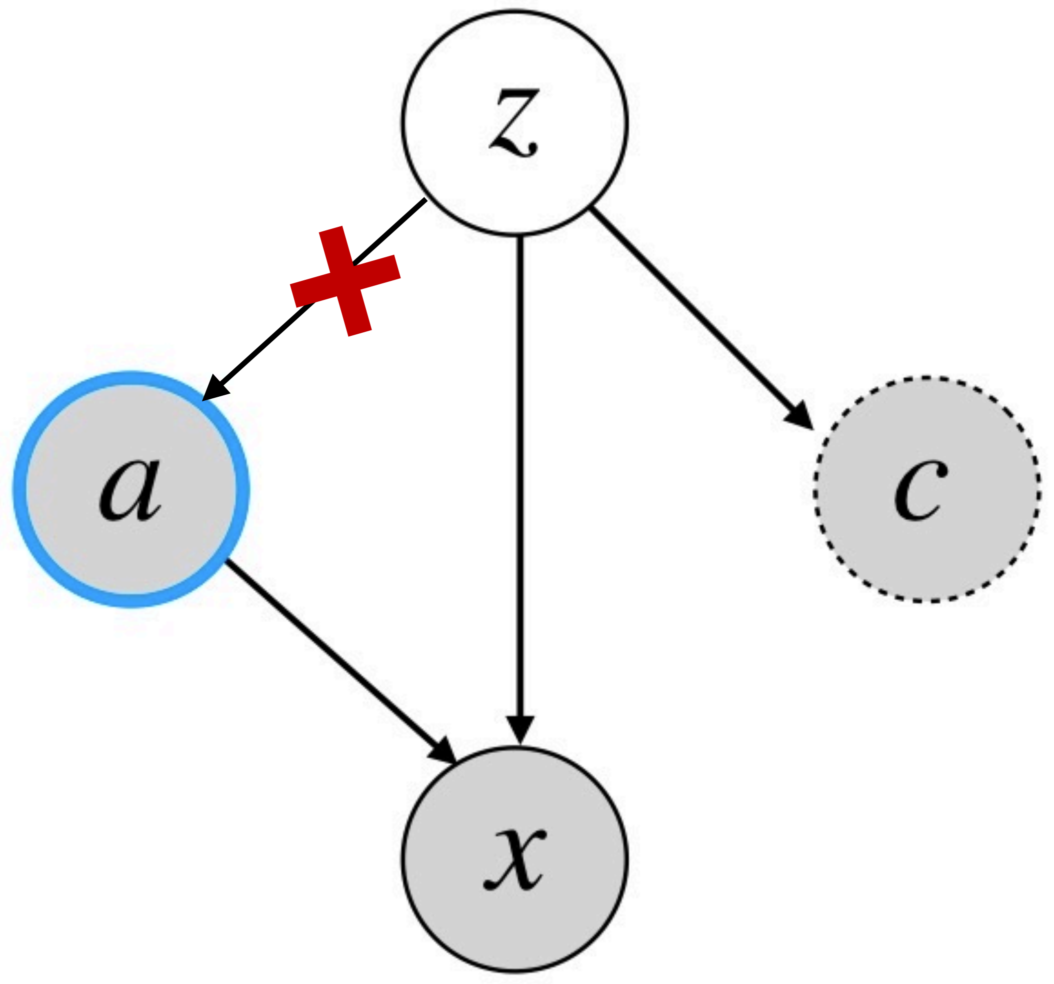
Inference (I): Intervention for Attribute-Conditional Generation

- Association (correlation): $p(\boldsymbol{x}|a)$

$$p(\boldsymbol{x}|a) = \sum_{z} p_\theta(\boldsymbol{x}|a, \boldsymbol{z}) p_\theta(\boldsymbol{z}|a)$$

- Intervention: $p(\boldsymbol{x}|do(a))$
  - Sets a to a given value independently of $\boldsymbol{z}$

$$p(\boldsymbol{x}|do(a)) = \sum_{z} p_\theta(\boldsymbol{x}|a, \boldsymbol{z}) p_\theta(\boldsymbol{z})$$



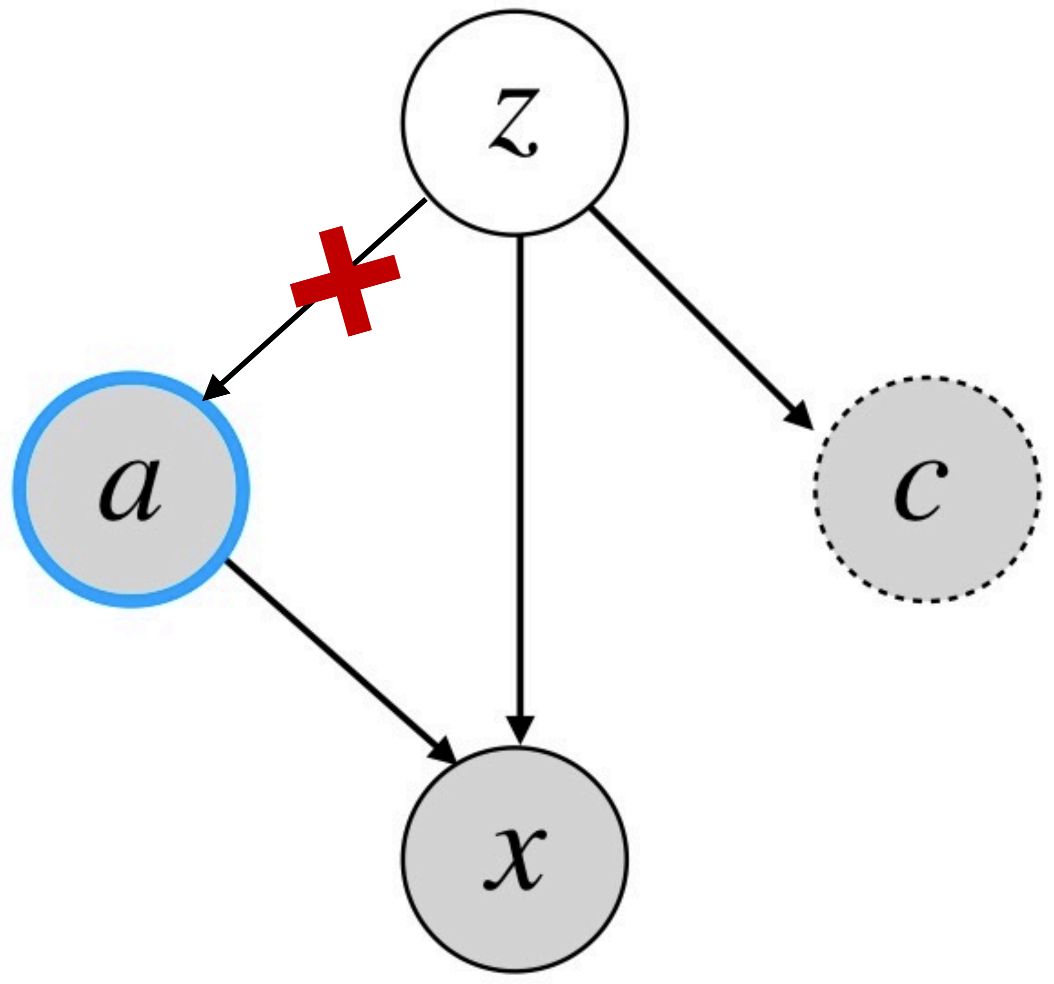
Inference (I): Intervention for Attribute-Conditional Generation

- Association (correlation): $p(\boldsymbol{x}|a)$

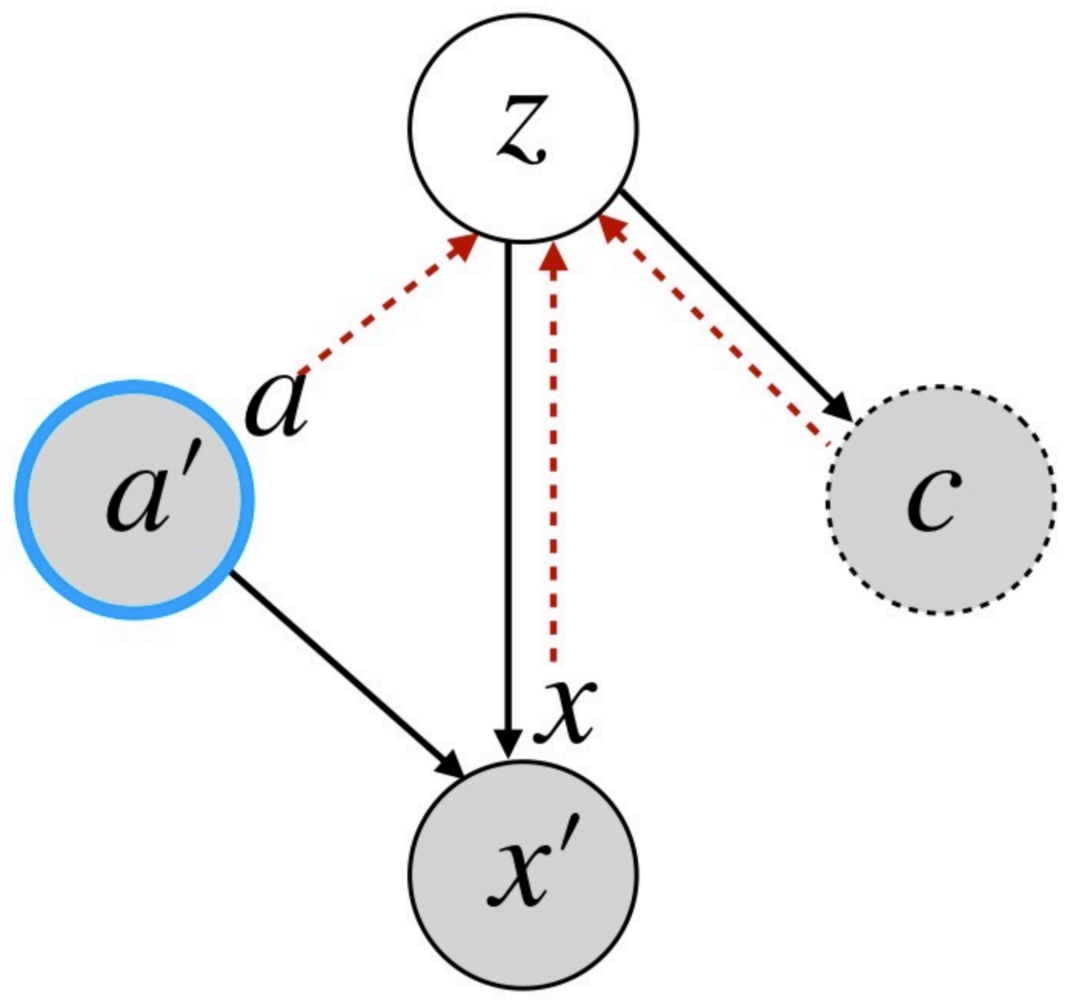
$$p(\boldsymbol{x}|a) = \sum_{z} p_\theta(\boldsymbol{x}|a,\boldsymbol{z}) p_\theta(\boldsymbol{z}|a)$$

- Intervention: $p(\boldsymbol{x}|do(a))$
    - Sets a to a given value independently of $\boldsymbol{z}$

$$p(\boldsymbol{x}|do(a)) = \sum_{z} p_\theta(\boldsymbol{x}|a,\boldsymbol{z}) p_\theta(\boldsymbol{z})$$



Inference (II): Counterfactual for Text Attribute Transfer



- What would the text be if the attribute had taken a different value?

- Counterfactuals as a standard three-step procedure [Pearl 2000]

  1) Abduction: predicts $\boldsymbol{z}$ given $\boldsymbol{x}$: $\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}, a, \boldsymbol{c})$

  2) Action: performs intervention, $do(a = a')$

  3) Prediction: generates $\boldsymbol{x}'$ given $\boldsymbol{z}$ and a' following the SCM: $\boldsymbol{x}' \sim p_\theta(\boldsymbol{x}'|a', \boldsymbol{z})$

Inference (III): Propensity Reweighting for Debiasing Pretrained LMs

- Given (biased) pretrained LM $p_{LM}(\boldsymbol{x}|a)$
- Can we convert it to unbiased $p(\boldsymbol{x}|do(a))$?

$$p(\boldsymbol{x}|do(a)) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}|a, \boldsymbol{z})p(\boldsymbol{z})$$

Inference (III): Propensity Reweighting for Debiasing Pretrained LMs

- Given (biased) pretrained LM $p_{LM}(\boldsymbol{x}|a)$
- Can we convert it to unbiased $p(\boldsymbol{x}|do(a))$?

$$p(\boldsymbol{x}|do(a)) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}|a, \boldsymbol{z})p(\boldsymbol{z})$$

$$= \sum_{\boldsymbol{z}} p(\boldsymbol{x}|a, \boldsymbol{z})p(\boldsymbol{z}|a)\frac{p(a)}{p(a|\boldsymbol{z})}$$

$$= \sum_{\boldsymbol{z}} p(\boldsymbol{x}|a)p(\boldsymbol{z}|\boldsymbol{x}, a)\frac{p(a)}{\boxed{p(a|\boldsymbol{z})}}$$

Propensity score: the probability of the $\boldsymbol{z}$ being assigned to the treatment a

Inference (III): Propensity Reweighting for Debiasing Pretrained LMs

- Given (biased) pretrained LM $\boxed{p_{LM}(\boldsymbol{x}|a)}$
- Can we convert it to unbiased $p(\boldsymbol{x}|do(a))$?

$$
\begin{aligned}
p(\boldsymbol{x}|do(a)) &= \sum\nolimits_{\boldsymbol{z}} p(\boldsymbol{x}|a, \boldsymbol{z})p(\boldsymbol{z}) \\
&= \sum\nolimits_{\boldsymbol{z}} p(\boldsymbol{x}|a, \boldsymbol{z})p(\boldsymbol{z}|a)\frac{p(a)}{p(a|\boldsymbol{z})} \\
&= \sum\nolimits_{\boldsymbol{z}} p(\boldsymbol{x}|a)p(\boldsymbol{z}|\boldsymbol{x}, a)\frac{p(a)}{\boxed{p(a|\boldsymbol{z})}} = \sum\nolimits_{\boldsymbol{z}} \boxed{p_{\mathrm{LM}}(\boldsymbol{x}|a)}\boxed{q_\phi(\boldsymbol{z}|\boldsymbol{x}, a, \boldsymbol{c})\frac{p(a)}{p_\theta(a|\boldsymbol{z})}}
\end{aligned}
$$

Reweighting to $p_{LM}(\boldsymbol{x}|a)$

Propensity score: the probability of the $\boldsymbol{z}$ being assigned to the treatment a

Inference (III): Propensity Reweighting for Debiasing Pretrained LMs

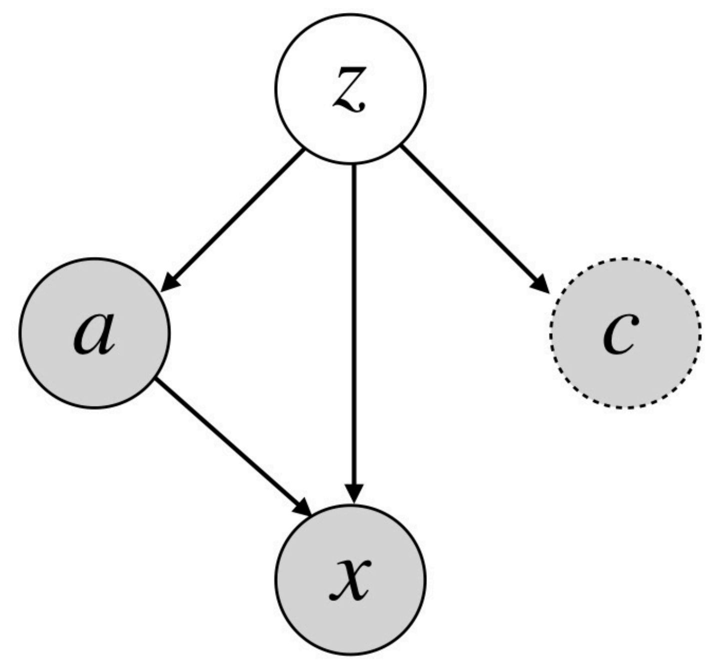
- Given (biased) pretrained LM $\boxed{p_{LM}(\boldsymbol{x}|a)}$
- Can we convert it to unbiased $p(\boldsymbol{x}|do(a))$?

$$
\begin{aligned}
p(\boldsymbol{x}|do(a)) &= \sum_{\boldsymbol{z}} p(\boldsymbol{x}|a,\boldsymbol{z})p(\boldsymbol{z}) \\
&= \sum_{\boldsymbol{z}} p(\boldsymbol{x}|a,\boldsymbol{z})p(\boldsymbol{z}|a)\frac{p(a)}{p(a|\boldsymbol{z})} \\
&= \sum_{\boldsymbol{z}} p(\boldsymbol{x}|a)p(\boldsymbol{z}|\boldsymbol{x},a)\frac{p(a)}{\boxed{p(a|\boldsymbol{z})}} = \sum_{\boldsymbol{z}} \boxed{p_{\mathrm{LM}}(\boldsymbol{x}|a)}\boxed{q_\phi(\boldsymbol{z}|\boldsymbol{x},a,\boldsymbol{c})\frac{p(a)}{p_\theta(a|\boldsymbol{z})}}
\end{aligned}
$$

Reweighting to $p_{LM}(\boldsymbol{x}|a)$

- Sampling-importance-resampling (SIR):
  - Biased samples $\sim p_{LM}(\boldsymbol{x}|a)$
  - Compute sample weights
  - Resampling proportional to the weights

Learning of the SCM



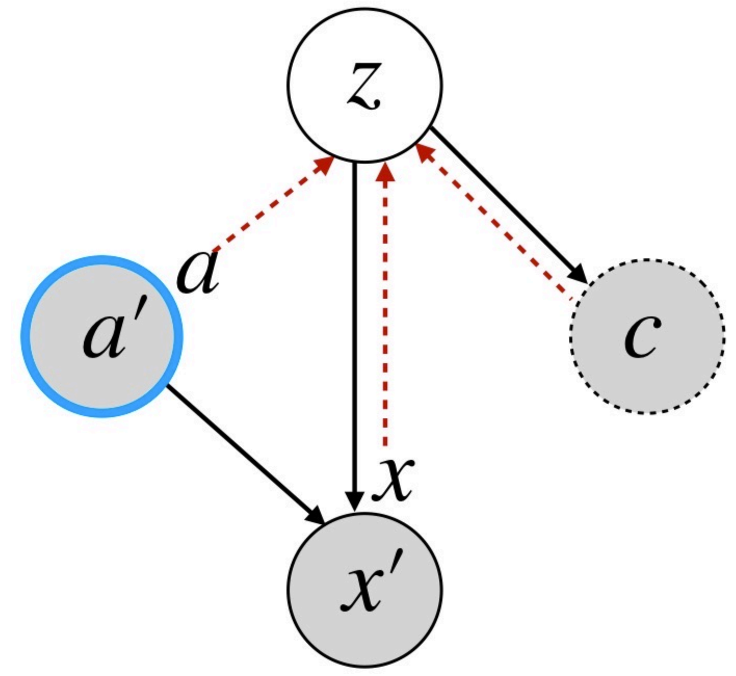
$$p_\theta(\boldsymbol{x}, a, \boldsymbol{z}, \boldsymbol{c}) = p_\theta(\boldsymbol{x}|a, \boldsymbol{z}) p_\theta(a|\boldsymbol{z}) p_\theta(\boldsymbol{c}|\boldsymbol{z}) p_0(\boldsymbol{z})$$

Variational distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x}, a, \boldsymbol{c})$

- Variational autoencoder (VAE) objective

$$\mathcal{L}_{vae}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{z}\sim q_\phi}\left[\log p_\theta(\boldsymbol{x}|a, \boldsymbol{z}) + \lambda_a \log p_\theta(a|\boldsymbol{z}) + \lambda_c \log p_\theta(\boldsymbol{c}|\boldsymbol{z})\right] - \lambda_{kl}\mathrm{KL}\left(q_\phi \| p_0\right)$$

- Counterfactual objectives
  - Draws inspirations from causality, disentangled representations & controllable generation
  - Intuition: counterfactual $\boldsymbol{x}'$ must entail a' and preserve the original $\boldsymbol{z}$ and $\boldsymbol{c}$



Experiments

- Two datasets with strong spurious correlations
  - Yelp customer reviews:
    - Attribute a: sentiment (1:positive, 0:negative)
    - Confounding proxy $\boldsymbol{c}$: category (1:restaurant, 0:others)
    - **Correlation: 90%** data have the same sentiment and category labels
    - Size: 510K for training, wherein 10K have category labels
  - Bios: online biographies
    - Attribute a: gender (1:female, 0:male)
    - Confounding proxy $\boldsymbol{c}$: occupation (1:nurse etc, 0:rapper etc)
    - **Correlation: 95%**
    - Size: 43K for training, wherein 3K have occupation labels
- Models:
  - Based on GPT-2 (117M)

$a = 1, \boldsymbol{c} = 1$
Soup and salad came out quickly !

$a = 0, \boldsymbol{c} = 0$
I texted and called Phil several times and he never responded

$a = 1, \boldsymbol{c} = 1$
She previously worked as a nurse practitioner

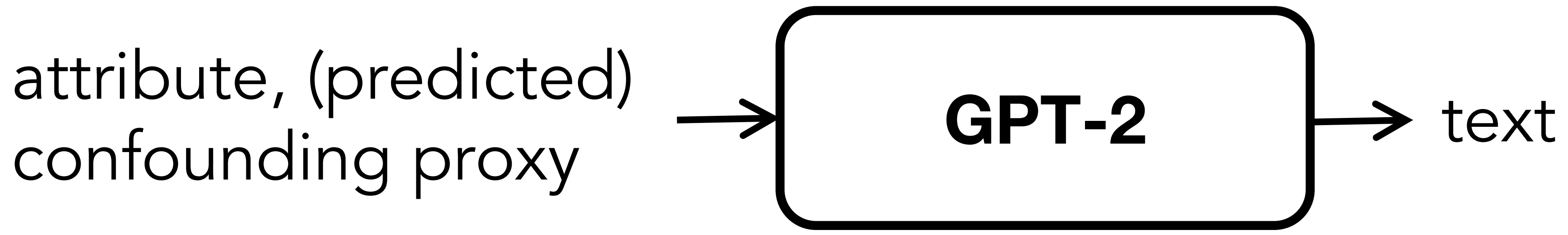
$a = 0, \boldsymbol{c} = 0$
He went to law school and became a plaintiffs' attorney

(I) Attribute-Conditional Generation

Conditional LM

- Causal model improves control accuracy and reduces bias

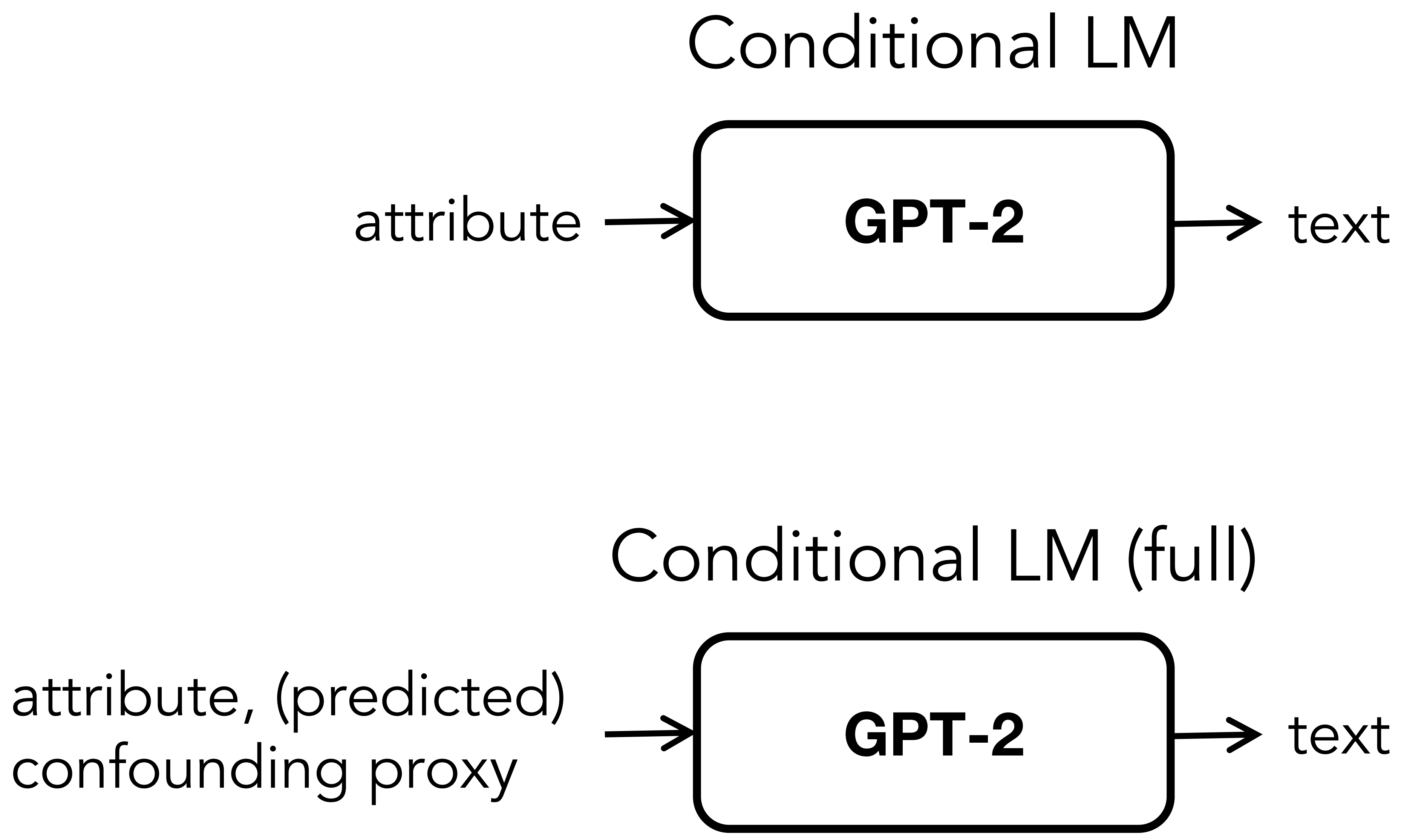
Conditional LM (full)



	Methods	Control accuracy (↑)	Bias (↓)	Fluency (↑)	Diversity (↑)
YELP	Conditional LM	79.1	78.7	-50.4	41.4
	Conditional LM (full)	80.3	78.9	-50.8	41.9
	Ours	**96.3**	**59.8**	-51.3	39.1

Automatic evaluation

(I) Attribute-Conditional Generation

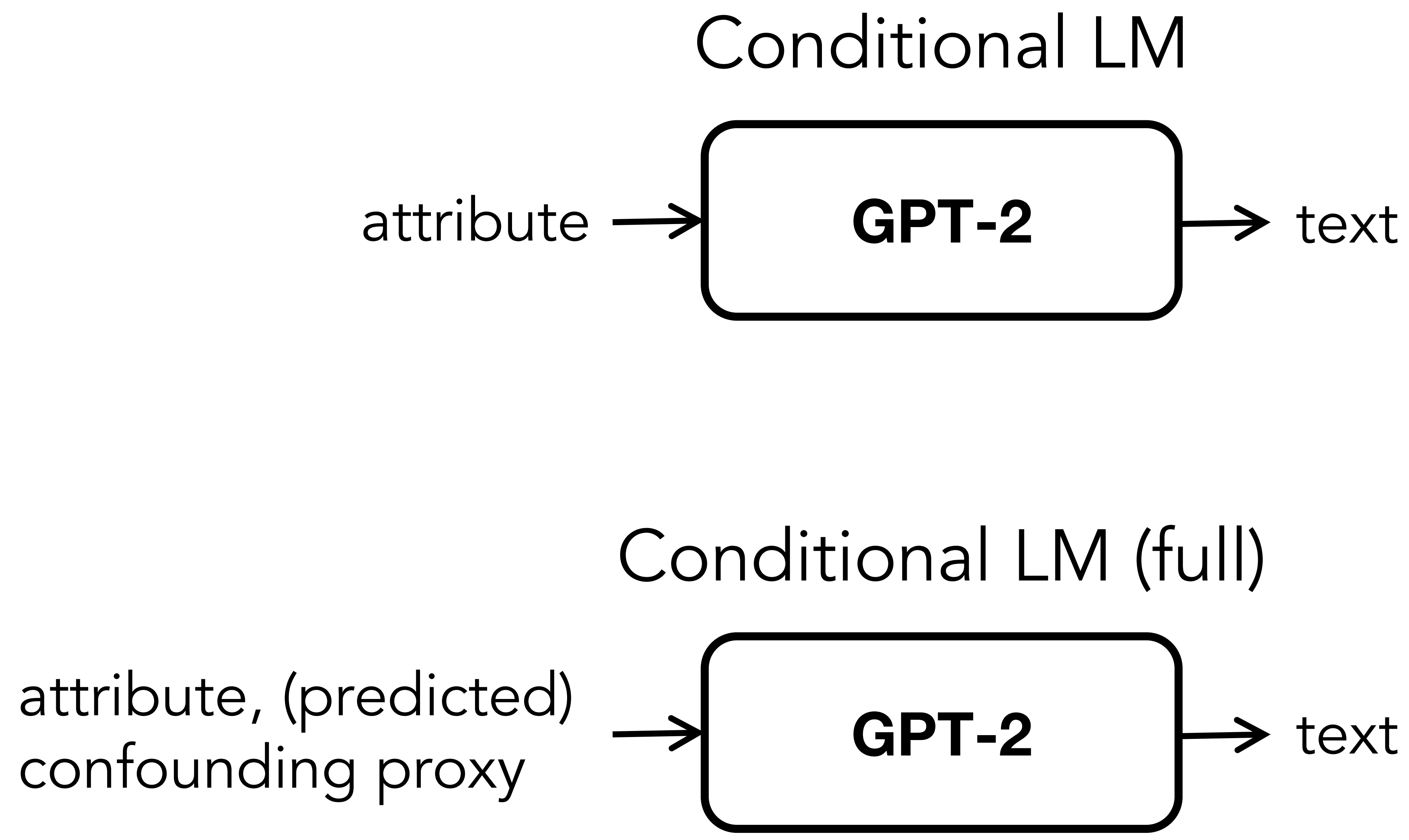


- Causal model improves control accuracy and reduces bias

	Methods	Control accuracy (↑)	Bias (↓)	Fluency (↑)	Diversity (↑)
YELP	Conditional LM	79.1	78.7	-50.4	41.4
	Conditional LM (full)	80.3	78.9	-50.8	41.9
	Ours	**96.3**	**59.8**	-51.3	39.1
BIOS	Conditional LM	95.51	84.73	**-17.0**	46.5
	Conditional LM (full)	93.28	72.34	-18.5	48.5
	Ours	**99.2**	**62.4**	-32.0	40.6

Automatic evaluation

(I) Attribute-Conditional Generation



- Causal model improves control accuracy and reduces bias

	Methods	Control accuracy (↑)	Bias (↓)	Fluency (↑)
YELP	Conditional LM (full)	80.0	73.0	3.90
	Ours	**97.0**	**56.0**	3.85
BIOS	Conditional LM (full)	96.0	82.0	4.43
	Ours	**99.0**	**60.0**	4.25

Human evaluation

(I) Attribute-Conditional Generation

restaurant

CONDITIONAL LM (FULL)
$a = 0$ **(sentiment negative)**
this was the worst experience i 've ever had at a glazier .
i even asked him if they could play on the tv channel .
this was pretty fun the first time i went . "
waited in line once but almost never reached the floor .
if you are ever up in chandler , tony will stop by .
$a = 1$ **(sentiment positive)**
very good and long wait time .
we loved our favorite harrah 's night ! "
i would love to try this restaurant again when they open . "
this place is great .
everything you will find in this restaurant !

OURS
$a = 0$ **(sentiment negative)**
no , it 's obvious that they were overcooked .
the seats were poorly done and basically sucked up .
it was n't enough to ask us if it was okay .
very disappointed with my food order yesterday .
i declined to replace it tho they were bad .
$a = 1$ **(sentiment positive)**
great for a relaxed evening out .
i 'm beyond impressed with the passion fruit and unbeatable service .
it 's a true pleasure to meet andrew .
jacksville became my go-to spot for dessert .
thank you for the technique , i am quite impressed .

(II) Text Attribute Transfer

- Previous methods tend to fail on the challenging dataset: low control accuracy
- Causal model obtains much higher accuracy, and keeps bias low

Methods	Control accuracy (↑)	Bias (↓)	Preservation (↑)	Fluency (↑)
Hu et al. [22]	44.1	68.4	77.7	-132.7
He et al. [20]	35.3	60.2	**80.1**	-57.7
Ablation: Ours w/o cf-z/c	75.0	67.8	36.3	-34.2
Ours	**77.0**	61.4	42.3	**-29.6**

Results on ***biased*** Yelp dataset

(II) Text Attribute Transfer

- Previous methods tend to fail on the challenging dataset: low control accuracy
- Causal model obtains much higher accuracy, and keeps bias low
- Also gets improvement on unbiased data

Methods	Control accuracy (↑)	Preservation (↑) self-BLEU	Preservation (↑) ref-BLEU	Fluency (↑)
Hu et al. [22]	86.7	**58.4**	-	-177.7
Shen et al. [65]	73.9	20.7	7.8	-72.0
He et al. [20]	87.9	48.4	18.7	**-31.7**
Dai et al. [7]	87.7	54.9	20.3	-73.0
Ablation: Ours w/o cf-z/c	87.1	57.2	24.3	-46.6
Ours	**91.9**	57.3	**25.5**	-47.1

Results on ***unbiased*** Yelp dataset (commonly used in previous study)

(III) Debiasing Pretrained LMs

- Resampling 2K out of 10K biased samples
- Substantially reduced bias

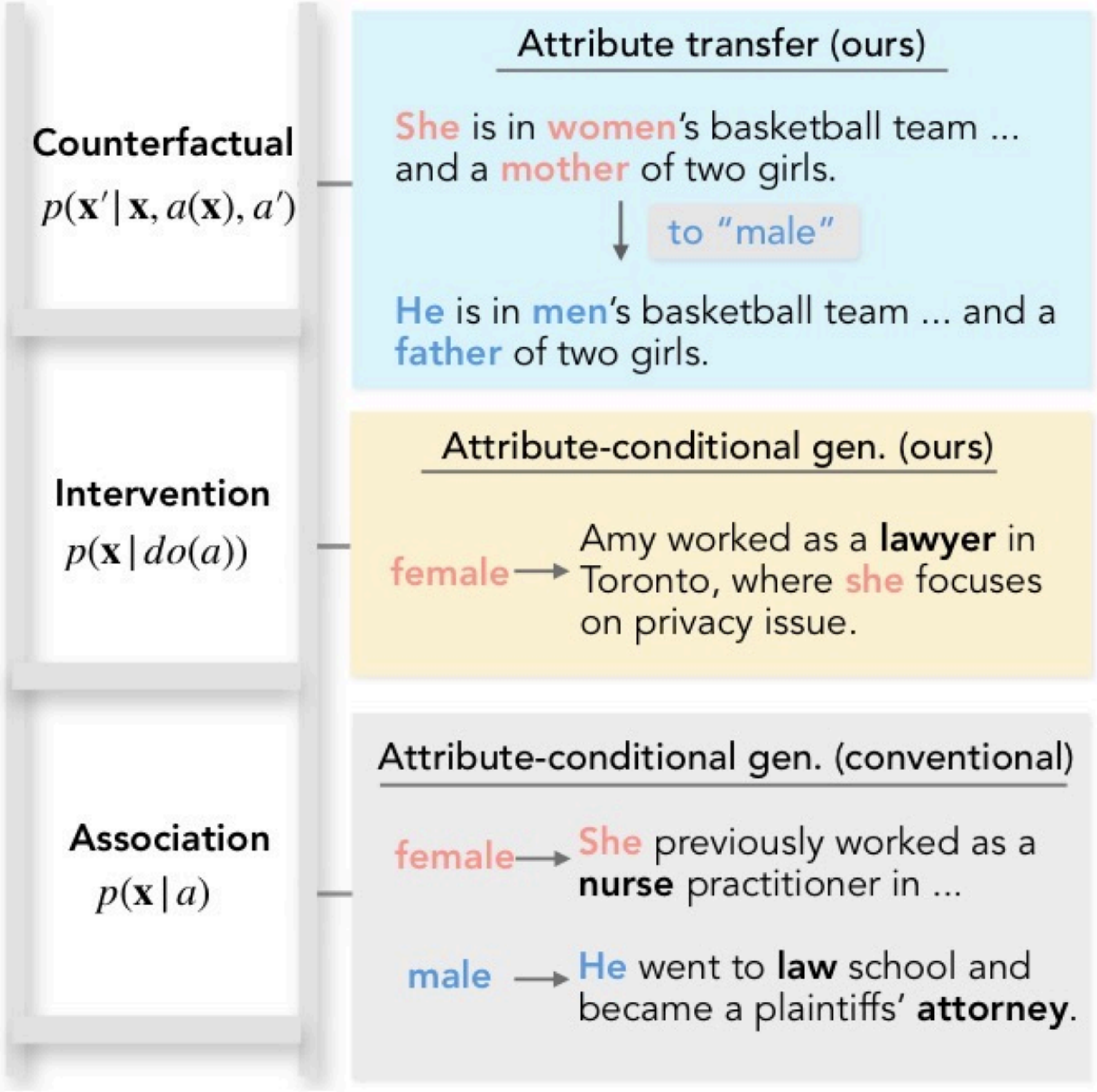
	Methods	Control accuracy (↑)	Bias (↓)
YELP	Conditional LM	79.1	78.7
	Debiased (Ours)	77.3	**66.3**

Debiasing results on Yelp

Summary of Causal Lens for Controllable Generation

- Causality + ML for unified unbiased controllable generation
  - Intervention
  - Counterfactual
  - Propensity reweighting

- Causal modeling for more text generation problems?
  - Dialog, summarization, …



Causal ladder [Pearl 2000]

Thanks !