

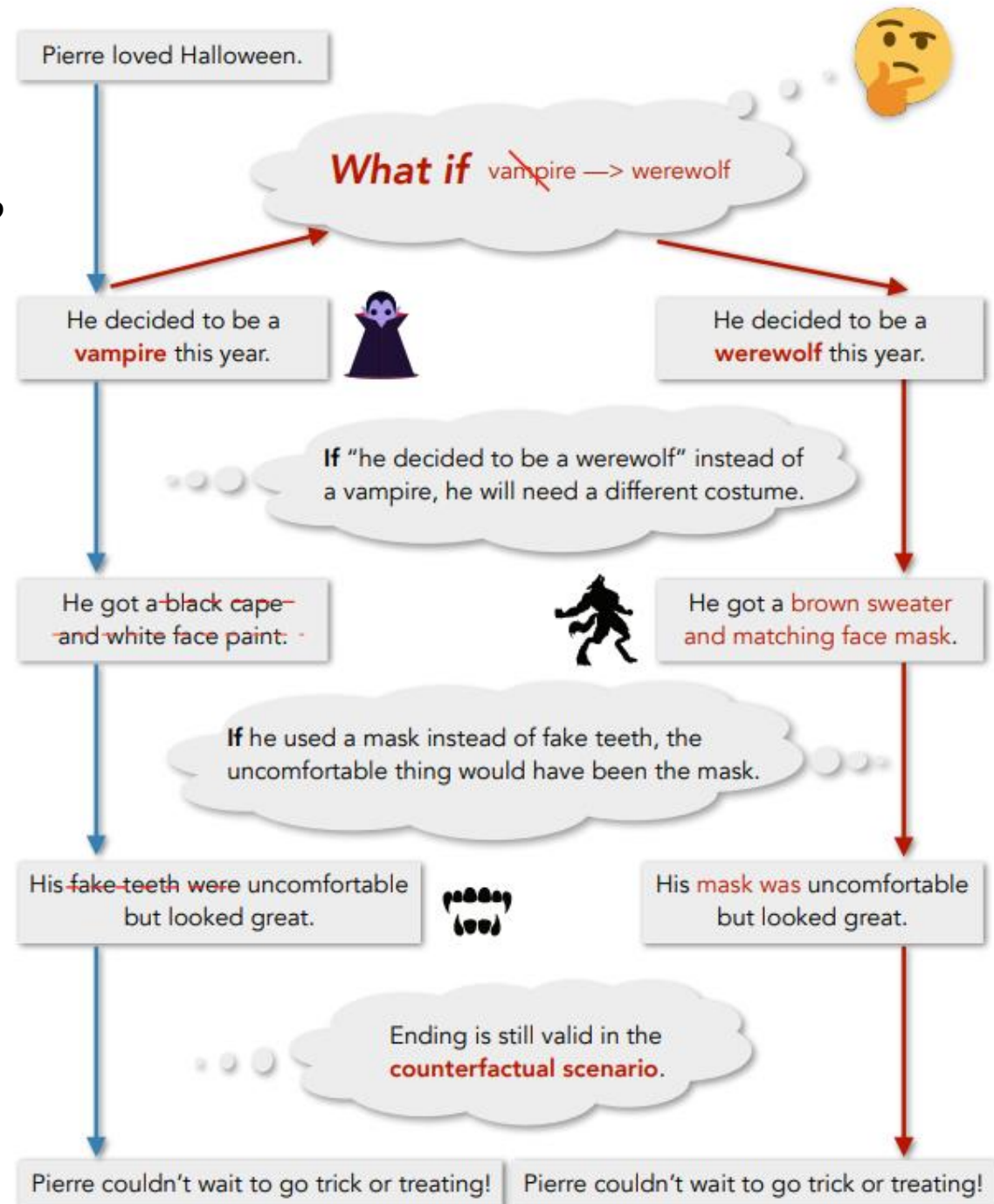
Some papers about
counterfactual

Outlines

- 1.A start: counterfactual for NLI
- 2.How to use: a VQA case
- 3.Counterfactual for Explanation
- 4.Counterfactual for Text: a dialogue generation case
- 5.~~Generation (Inference) problems~~

Counterfactual

What would have happened if

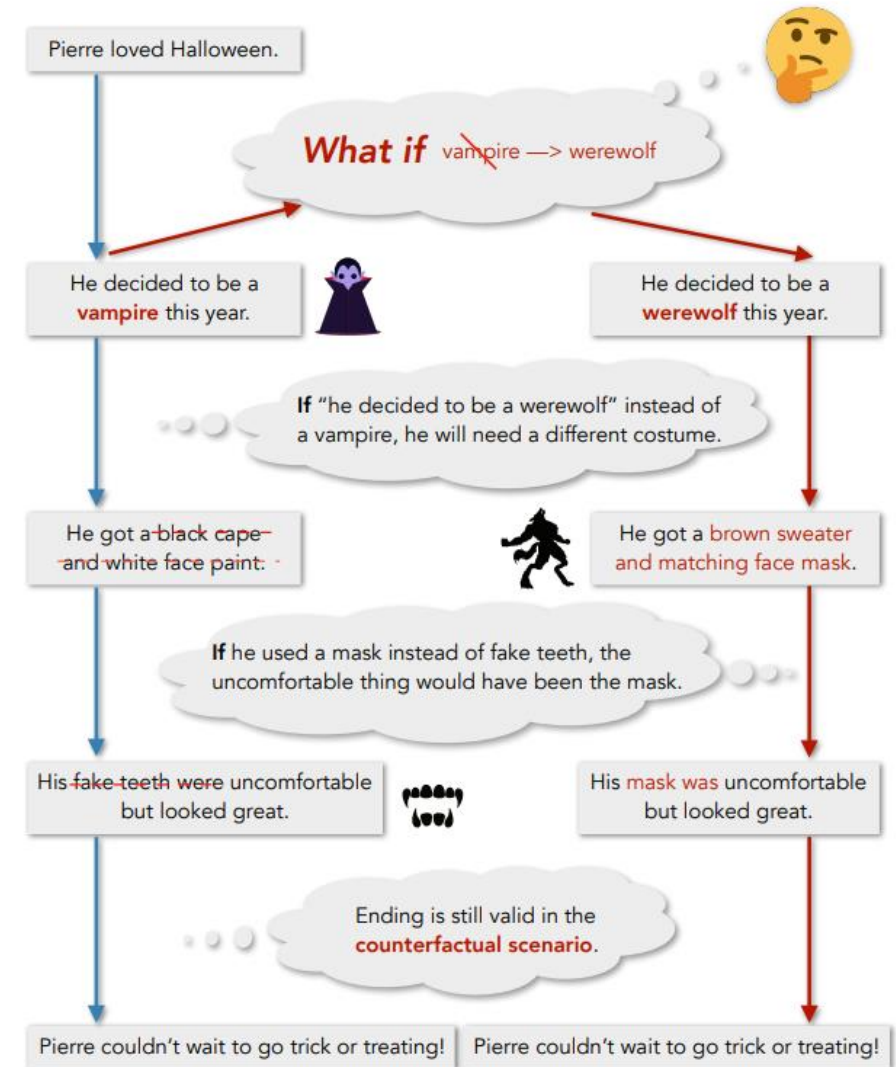


Counterfactual for NLI

- **Counterfactual Story Reasoning and Generation** Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark and Yejin Choi EMNLP, 2019
- **Back to the Future: Unsupervised Backprop-based Decoding for Counterfactual and Abductive Commonsense Reasoning (DeLorean)** Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D. Hwang, Ronan Le Bras, Antoine Bosselut and Yejin Choi EMNLP, 2020

Counterfactual Story Reasoning and Generation

- Counterfactual reasoning is meaningful for intelligence
- Introducing *counterfactual story rewriting* Task
- No dataset before——**TIMETRavel**
(from ROCStories)



TIMETRAVEL

Premise	Alec’s daughter wanted more blocks to play with.
Initial	Alec figured that blocks would develop her scientific mind.
Original Ending	Alec bought blocks with letters on them. Alec’s daughter made words with them rather than structures. Alec was happy to see his daughter developing her verbal ability.
Counterfactual	Alec couldn’t afford to buy new blocks for his daughter.
Edited Ending	Alec decided to make blocks with letters on them instead. Alec’s daughter made words with the blocks. Alec was happy to see his daughter developing her verbal ability.
Premise	Ana had just had a baby girl.
Initial	She wanted her girl to have pierced ears.
Original Ending	She took her baby to the studio and had her ears pierced. Then she fastened tiny diamond studs into the piercings. Ana loved the earrings.
Counterfactual	She didn’t like the idea of having her ears pierced.
Edited Ending	She decided not to take her baby to the studio to get her ears pierced. So she took tiny diamond stickers and stuck them to her ear. Ana loved the fake earrings.

Table 1: Examples from TIMETRAVEL

	Train	Valid	Test
<i>ROCStories data:</i>			
# Stories	98,159	1,871	1,871
TIMETRAVEL:			
# Counterfactual Context	96,867	5,613	7,484
# Edited Ending	16,752	5,613	7,484
#For Unsupervised	80000+		

Table 2: Dataset statistics

Baseline Method

1. **Basic Model:** GPT、GPT-S、GPT-M

2. Baseline Method:

a) **Zero-shot:** *“At test time, the model receives the premise and the counterfactual context (s_1, s_2) as input and generates the tokens that constitute the rewritten counterfactual outcome.”*

b) **Fine-tuning :** fine-tune on ROCStories by $\mathcal{L}^{ft}(\theta) = \log p_{\theta}(S)$

c) **Fine-tuning+Counterfactual:**

$$\mathcal{L}^{cf}(\theta) = \log p_{\theta}(s'_2 | s_1). \quad \mathcal{L}^{ft+cf}(\theta) = \mathcal{L}^{ft} + \mathcal{L}^{cf},$$

d) **Reconstruction + Counterfactual (RC + CF)**

$$\mathcal{L}^{rc}(\theta) = \log p_{\theta}(s_{3:5} | S, [s], s_1, [mask]).$$

e) **Supervised learning**

$$\mathcal{L}^s(\theta) = \log p_{\theta}(s'_{3:5} | S, [s], s_1, s'_2).$$

Evaluation

- **Human**

- Does the rewritten ending keep in mind details of the original premise sentence?
- Is the plot of the rewritten ending relevant to the plot of the original ending?
- Does the rewritten ending respect the changes induced by the counterfactual sentence?

(0~5)

- **Automatic**

- Overlap Metrics
 - BLEU
 - ROUGE-L
- Model-based Metrics
 - Word Mover's Similarity 、 Sentence + Word Mover's Similarity
 - BERTScore、 BERT-FT

Case study

Premise	Rob loved to play soccer and was on his middle school team.
Initial	They had just started their season and were playing their first game.
Original Ending	Rob accidentally tripped over another player during the game. The other player was mad, but Rob apologized and kept his composure. Rob ended up playing the best game of his life!
Counterfactual	When he graduated from middle school, he started learning to play the guitar.
Model	Generated Ending
GPT-M + ZS	He played in bands for about a year, then got into touring with a band called The Muffs. He wrote the music for a few of The Muffs' songs. He was a very talented singer-songwriter and guitarist.
GPT-M + FT + CF	He took classes on the guitar for years to get better. He took a solo lesson and had a great time. Rob now is a famous guitarist.
GPT-M + Supervised	He accidentally tripped over the guitar during the song. The other person was mad, but Rob apologized and kept his composure. Rob ended up playing the best song of his life.
Premise	I called Carly to ask her on a date.
Initial	I got her voicemail.
Original Ending	I texted her that we should talk soon. She texted me back, asking why we should talk soon. I told her I liked her.
Counterfactual	She said that we should just stay friends which made me really sad.
Model	Generated Ending
GPT-M + ZS	She said that she went out and did something really stupid and that was all I could think about. I did my best to be nice to Carly by telling her how much I loved her and that I would be her best friend no matter what. There were nights when I felt like crying because I couldn't get out of bed.
GPT-M + FT + CF	I decided to watch a movie instead. After that I decided I should see a movie I had never seen before. I went to the theater and bought a ticket.
GPT-M + Sup	I texted her that we should just stay friends soon. She texted me back, asking why we should just stay friends soon. I told her I liked her.

Table 3: Model Outputs

Back to the Future: Unsupervised Backprop-based Decoding for Counterfactual and Abductive Commonsense Reasoning (DeLorean)

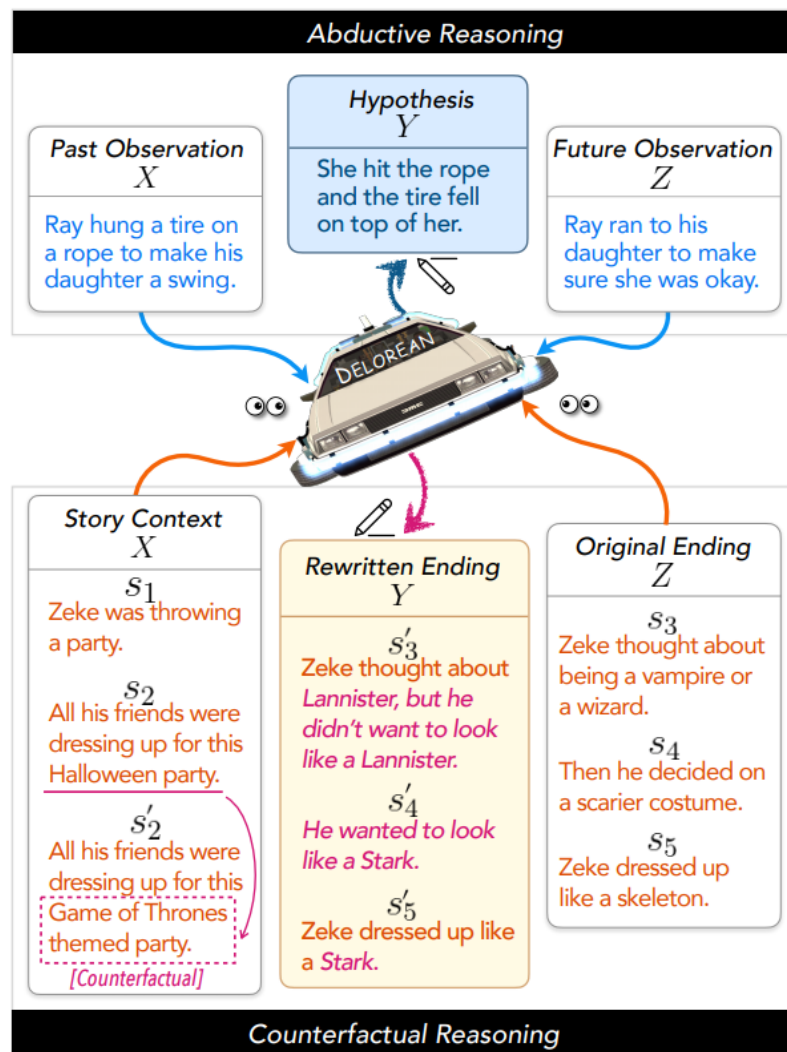
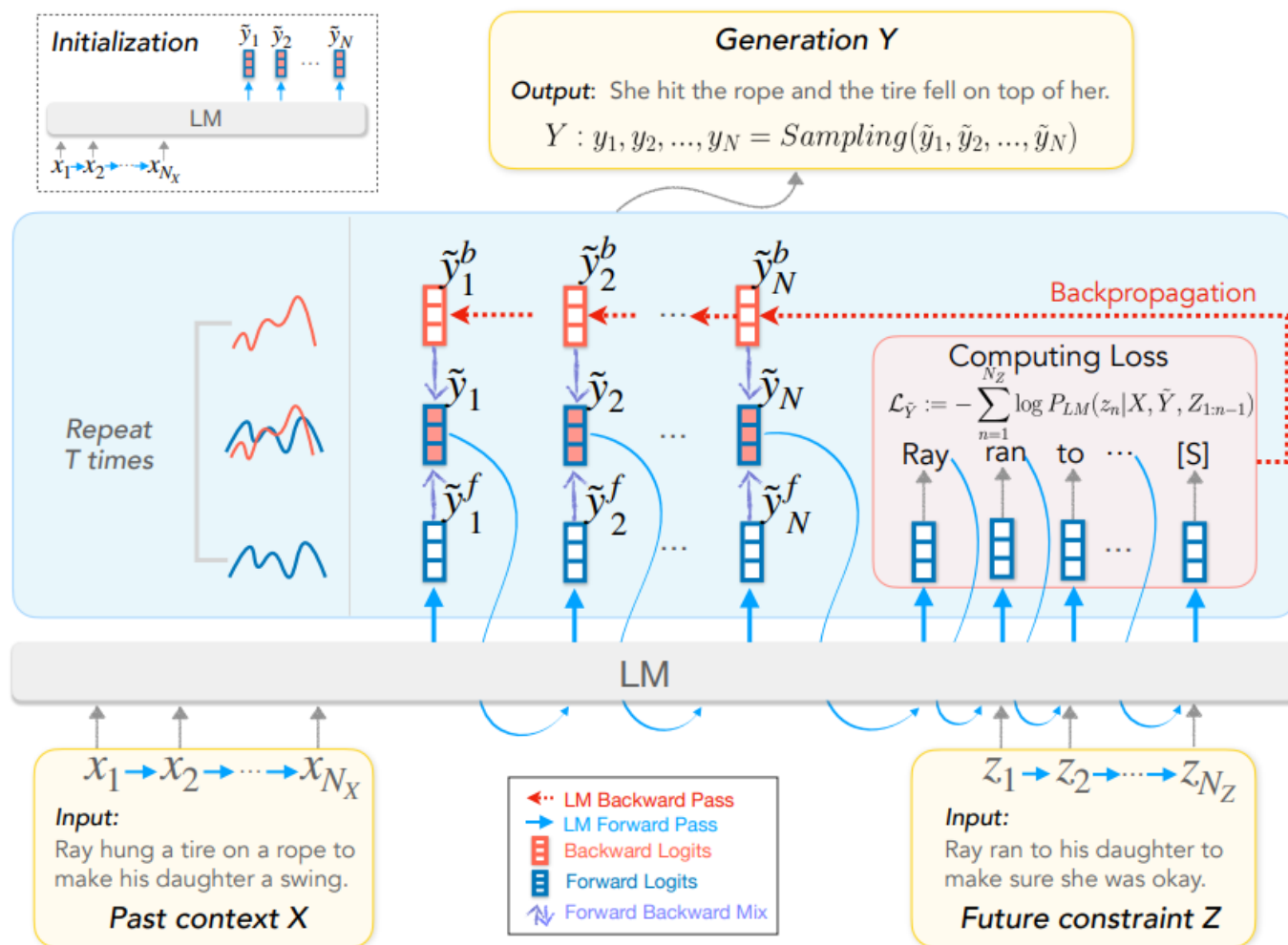


Figure 1: DELOREAN, our proposed method, with generated reasoning results. **Top:** the goal in abductive reasoning is to generate a hypothesis (Y) of what happened between the observed past (X) and future (Z) contexts. **Bottom:** In counterfactual reasoning, given a story context altered by a counterfactual condition, X , and the original ending Z , the goal is to generate a new ending Y which is coherent with X while remaining similar to Z . The story from TIMETRAVEL (Qin et al., 2019a) consists of five sentences. Our approach alternates forward (left-to-right) and backward (right-to-left) passes that iteratively refine the generated texts w.r.t context from each side.

Method



Forward:

$$\tilde{y}_n^{(t),f} = \text{LM}(X, \tilde{Y}_{1:n-1}^{(t)}).$$

Backward

$$\tilde{y}_n^{(t),b} = \tilde{y}_n^{(t-1)} - \lambda \cdot \nabla_{\tilde{y}_n} \mathcal{L}(X, \tilde{Y}^{(t-1)}, Z),$$

Combine loss:

$$\tilde{y}_n^{(t)} = \gamma \cdot \tilde{y}_n^{(t),f} + (1 - \gamma) \cdot \tilde{y}_n^{(t),b},$$

$$\tilde{y}_n^{(t)} = \tilde{y}_n^{(t),f} \text{ for } n > N.$$

To generate discrete output :

$$y_n \sim \text{softmax}(\tilde{y}_n / \tau),$$

Ranking :

$$c(A, B) = \text{BERT_NSP}(A, B),$$

Method

Algorithm 1: DELOREAN Decoding

Input: Pre-trained language model (LM)

Context X

Future constraint Z

1: Initialize logits $\tilde{Y}^{(0)}$

2: Initialize Y_s , list of candidate generations

3: **for** $t \leftarrow 1$ to T **do**

4: // Backward pass

5: **for** $n \leftarrow N$ to 1 **do**

6: Compute backward logits \tilde{y}_n^b , Eq.(1)

7: **end for**

8: // Forward pass

9: **for** $n \leftarrow 1$ to N **do**

10: Compute forward logits \tilde{y}_n^f , Eq.(2)

11: Mix forward and backward logits, Eq.(3)

12: **end for**

13: Sample candidate Y from logits \tilde{Y} and add to Y_s

14: **end for**

15: Rank Y_s by coherence

Output: The most coherent generated text Y from Y_s

Constraints:

$$\mathcal{L}(X, \tilde{Y}, Z) := \text{KL} \left(Z \parallel \text{softmax}(\tilde{Y}/\tau) \right)$$

Ranking

$$\text{ranking_score}(Y) = c(X, Y) + \sum_{s=1}^{S-1} c(Y[s], Y[s+1]).$$

Result

	BLEU	ROUGE	BERT
<i>Supervised + Discriminative</i>			
<i>Sup+Disc</i>	75.71	72.72	62.39
<i>Unsupervised+ Discriminative</i>			
<i>Recon+CF</i>	75.92	70.93	62.49
<i>Unsupervised</i>			
<i>FT</i>	4.06	24.09	62.55
<i>FT+CF</i>	4.02	24.35	62.63
<i>Pretrained-only</i>			
Zero-Shot _{$s_1 s'_2$}	1.74	21.41	59.31
Zero-Shot _{$s_1 s'_2$} -Ranked	2.26	25.81	60.07
DELOREAN	21.35	40.73	63.36
Human	64.93	67.64	61.87

Table 4: Automatic evaluation results of counterfactual story rewriting, on the test set of TIMETRAVEL.

Coherence - Human Judges Preferred				
	Our model	Neutral	Comparator	
DELOREAN	25%	58%	17%	Sup+Disc
DELOREAN	23%	70%	7%	Recon+CF
DELOREAN	22%	48%	30%	FT
DELOREAN	18%	60%	22%	Zero-Shot _{$s_1 s'_2$}
DELOREAN	27%	42%	31%	Zero-Shot _{$s_1 s'_2$} -Ranked
DELOREAN	10%	29%	61%	Human
Min-Edits - Human Judges Preferred				
	Our model	Neutral	Comparator	
DELOREAN	4%	17%	79%	Sup+Disc
DELOREAN	1%	14%	85%	Recon+CF
DELOREAN	21%	76%	3%	FT
DELOREAN	28%	71%	1%	Zero-Shot _{$s_1 s'_2$}
DELOREAN	37%	56%	7%	Zero-Shot _{$s_1 s'_2$} -Ranked
M+Sup	8%	22%	70%	Human

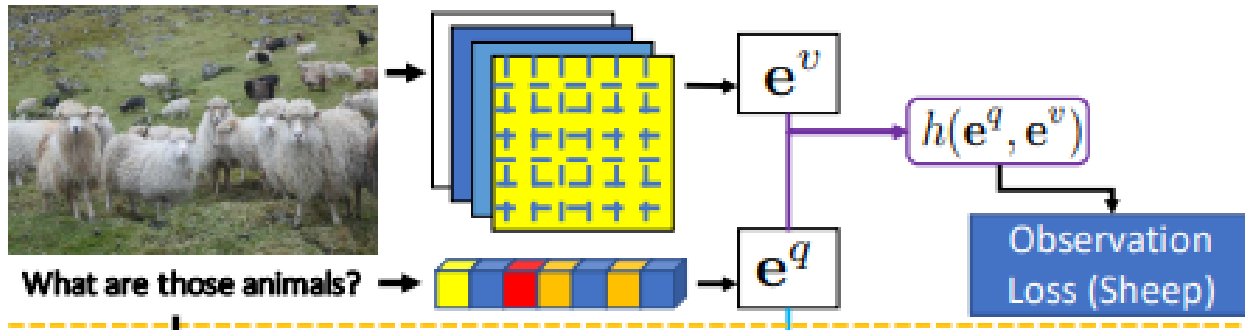
Table 5: Human pairwise comparison results on the counterfactual task, between our best model and each baseline with respect to coherence and min-edits.

“mining worlds that could have been”

--Judea Pearl

- **Counterfactual Vision and Language Learning** Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, Anton van den Hengel CVPR, 2020

Visual Question Answer



Dataset: $\mathcal{D} = \{\langle q_i, v_i, a_i \rangle\}_{i=1}^n$.

Input: image: $e^v = f_v(v)$, question: $e^q = f_q(q)$.

Model: joint space: $z = h(e^q, e^v)$

Output: answer: $p(a|q, v, \theta)$

Structural Causal Model

Definition:

Structural causal model: \mathcal{M}

Independent random variable(noisy, latent): $\mathbf{u} = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ with distribution $P(\mathbf{u})$

A set of variable(observed): $\mathbf{X} = \{X_1, \dots, X_n\}$

A set of functions(deterministic, transition, Mechanism $\mathbf{F} = \{f_1, \dots, f_n\}$

We have

$$X_i = f_i(\mathbf{PA}_i, \mathbf{u}_i), \forall i, \text{ where } \mathbf{PA}_i \subseteq \mathbf{X} \setminus X_i$$

And $P^{\mathcal{M}}$ is determined by prior distribution $P(\mathbf{u})$ and function.

Intervention

For an SCM \mathcal{M} an intervention $I = \text{do}(X_i := \tilde{f}_i(\tilde{\mathbf{PA}}_i, \mathbf{u}_i))$ replace the structural mechanism $f_i(\mathbf{PA}_i, \mathbf{u}_i)$ with $\tilde{f}_i(\tilde{\mathbf{PA}}_i, \mathbf{u}_i)$

The result SCM is \mathcal{M}^I and distribution $P^{\mathcal{M};I}$

Counterfactual queries

Abduction: Predict the 'state of the world' (the exogenous noise, u) that is compatible with the observations, x , i.e. infer $P_M(u|x)$.

Action: Perform an intervention.

Prediction: Compute the quantity of interest based on the distribution entailed by the counterfactual SCM.

Counterfactual VQA

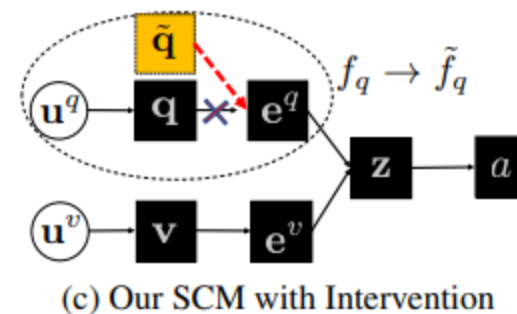
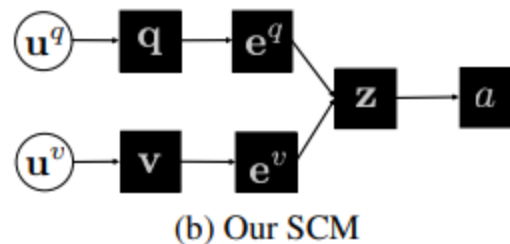
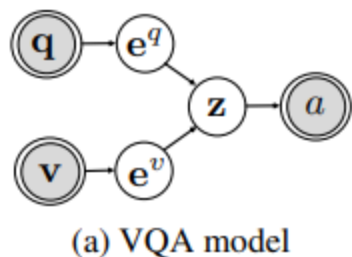


Figure 2: The difference between a typical VQA graphical model (in Fig. 2a), our corresponding causal model (in Fig. 2b) and an example of intervention in the question representation of this model (in Fig. 2c). In our model two exogenous variables u^q and u^v are incorporated to learn and reason about the intervention caused by these variables.

Motivation

*“these methods exploit statistical regularities and **biases** in the data to achieve this performance”*

*“**particular signals in the input trigger specific answers**; for instance, when the image contains a banana, the answer is most likely to be yellow, irrespective of the remainder of the image, or the question.”*

*“Training a model to both learn to answer, **and “reason” about the intervention** in the questions and images allows better generalization”*

Method

Counterfactual distribution

Counterfactual distribution is the **posterior** of the exogenous obtained from the observation:

$$p(\mathbf{u}|\mathcal{D}) \propto p(\mathbf{u}) \prod_{i=1}^n p(a_i|\mathbf{q}_i, \mathbf{v}_i)p(\mathbf{v}_i|\mathbf{u}^v)p(\mathbf{q}_i|\mathbf{u}^q).$$

Prior:

$$\mathbf{u}^v \sim \text{Beta}(\alpha_0, \beta_0)$$

Likelihood:

$$\begin{aligned} \mathbf{q} \sim p(\mathbf{q}|\mathbf{u}^q) &= \begin{cases} \mathbf{q} & \mathbf{u}^q \geq 1 - \epsilon \\ \mathbf{u}^q \mathbf{q} \oplus (1 - \mathbf{u}^q) \mathbf{q}', & \text{otherwise} \end{cases}, \quad \text{and} \\ \mathbf{v} \sim p(\mathbf{v}_i|\mathbf{u}^v) &= \begin{cases} \mathbf{v} & \mathbf{u}^v \geq 1 - \epsilon \\ \mathbf{u}^v \mathbf{v} \oplus (1 - \mathbf{u}^v) \mathbf{v}', & \text{otherwise} \end{cases} \end{aligned} \quad (2)$$

\mathbf{q}' and \mathbf{v}' are uniformly sampled at random from the dataset and \oplus denotes an **interpolation**.

Method

Generating Counterfactual

By conjugate prior, the posterior :

$$p(\mathbf{u}|\mathcal{D}) \sim \text{Beta}(\alpha, \beta)$$

Where

$$\alpha = \alpha_0 + \sum \mathbb{I}[a_i = \arg \max p(a_i|\mathbf{q}_i, \mathbf{v}_i)]$$

$$\beta = \beta_0 + \sum \mathbb{I}[a_i \neq \arg \max p(a_i|\mathbf{q}_i, \mathbf{v}_i)]$$

Method

Generating Counterfactual

Perform the intervention: generating the counterfactuals(sample from the posterior) and replace the \mathbf{v} (or \mathbf{q}) with its alternative $\tilde{\mathbf{v}}$ (or $\tilde{\mathbf{q}}$).

Minimum intervention: changing the answer for a given question-answer pair (\mathbf{q}, \mathbf{v}) to $(\tilde{\mathbf{q}}, \tilde{\mathbf{v}})$ when using the generating process.

$$\begin{aligned} \max_{\mathbf{u}} \quad & \log(p^{\text{do}(I)}|\mathbf{q}, \mathbf{v}(\tilde{\mathbf{q}}, \tilde{\mathbf{v}}|\mathbf{u})) \\ \text{s.t.} \quad & \tilde{a} = \operatorname{argmax}_{a'} p^{\text{do}(I)}|\mathbf{q}, \mathbf{v}(a'|\tilde{\mathbf{q}}, \tilde{\mathbf{v}}), \forall \tilde{a} \neq a \\ & 0 \leq \mathbf{u} < 1 \end{aligned}$$

To compute feasible:

$$\max_{\mathbf{u}} \quad \|\mathbf{u}\|^2 - \lambda \log \left(p^{\text{do}(I)}|\mathbf{q}, \mathbf{v}(a|\tilde{\mathbf{q}}, \tilde{\mathbf{v}}) \right)$$

Method

Counterfactual loss

Common practical empirical risk minimization:

$$\mathbb{E}_{\mathbf{q}, \mathbf{v}} \mathbb{E}_{p(a|\mathbf{q}, \mathbf{v})} [\ell(f_{\boldsymbol{\theta}}(\mathbf{q}, \mathbf{v}), a)]$$

Add counterfactuals

$$\begin{aligned} R(\boldsymbol{\theta}) &= \mathbb{E}_{\mathbf{q}, \mathbf{v}} \mathbb{E}_{p(a|\mathbf{q}, \mathbf{v})} [\ell(f_{\boldsymbol{\theta}}(\mathbf{q}, \mathbf{v}), a)] \\ &= \mathbb{E}_{\mathbf{q}, \mathbf{v}} \mathbb{E}_{p^{\text{do}(I)}|\mathbf{q}, \mathbf{v}}(a|\tilde{\mathbf{q}}, \tilde{\mathbf{v}}) \left[\ell(f_{\boldsymbol{\theta}}(\mathbf{q}, \mathbf{v}), a) \frac{p(a|\mathbf{q}, \mathbf{v}, \boldsymbol{\theta})}{p^{\text{do}(I)}|\mathbf{q}, \mathbf{v}}(a|\tilde{\mathbf{q}}, \tilde{\mathbf{v}}, \boldsymbol{\theta}) \right] \end{aligned}$$

Note that $p^{\text{do}(I)}|\mathbf{q}, \mathbf{v}}(a|\tilde{\mathbf{q}}, \tilde{\mathbf{v}}, \boldsymbol{\theta})$ has part of SCM altered. Intuitively, the counterfactuals that have smaller scores are more penalized and conversely the over-confident ones are discouraged. This subsequently adjusts the decision boundary to be discriminative for both observations and counterfactuals. Furthermore, since this risk can have a very high variance we can clip this value similar to [12],

Method

Counterfactual loss

since this risk can have a very high variance we can clip this value:

$$R^M(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{q}, \mathbf{v}} \mathbb{E}_{\tilde{p}^u(a|\mathbf{q}, \mathbf{v})} \left[\ell(f_{\boldsymbol{\theta}}(\mathbf{q}, \mathbf{v}), a) \right. \\ \left. \times \min \left\{ M, \frac{p(a|\mathbf{q}, \mathbf{v}, \boldsymbol{\theta})}{p^{\text{do}(I)}(\mathbf{q}, \mathbf{v})(a|\tilde{\mathbf{q}}, \tilde{\mathbf{v}}, \boldsymbol{\theta})} \right\} \right]$$

$$\hat{R}^M(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \ell(f_{\boldsymbol{\theta}}(\mathbf{q}_i, \mathbf{v}_i), a_i) \times \omega_i(\boldsymbol{\theta}) \quad (4)$$

$$\text{where } \omega_i(\boldsymbol{\theta}) = \min \left\{ M, \frac{p(a_i|\mathbf{q}_i, \mathbf{v}_i, \boldsymbol{\theta})}{p^{\text{do}(I)}(\mathbf{q}, \mathbf{v})(a|\tilde{\mathbf{q}}, \tilde{\mathbf{v}}, \boldsymbol{\theta})} \right\}.$$

$$\hat{R}^{M*} = \arg \min_{\boldsymbol{\theta}} \hat{R}^M(\boldsymbol{\theta})$$

When training. Alternating between the conventional ERM and the counterfactual risk.

Experiment

Unimodal problem

Stanford Sentiment Treebank (SST) (11855 instances with vocabulary size of 17836 and 5 classes.)

	LSTM	T	LSTM+P	T+P	LSTM+C	T+C
Random	84.4	82.0	84.53	85.21	85.61	85.56
GloVe	84.9	86.4	85.77	87.1	87.24	88.4

Table 1: Accuracy (%) obtained by the testing methods using LSTM (with randomly initialized, trainable embeddings). Best results highlighted in Bold. T abbreviates TreeLSTM [40]; +P and +C indicate posterior and Counterfactuals respectively.

CIFAR-10 and CIFAR-100

Dataset	Model	Baseline	Ours+P	Ours+C
CIFAR-10	VGG-19	95.04	95.92	96.73
	ResNet-18	93.02	94.2	94.91
	ResNet-101	93.75	94.1	95.34
	DenseNet-121	95.04	95.92	96.73
CIFAR-100	VGG-19	72.23	73.45	74.8
	ResNet-18	75.61	76.5	77.75
	ResNet-101	77.78	78.9	80.0
	DenseNet-121	77.01	79.67	79.67

Table 2: Test errors for the CIFAR experiments.

Experiment

Result

VQA-CP and VQA v2

Model	Overall	Yes/No	Number	Other
Question-Only [6]	15.95	35.09	11.63	7.11
RAMEN [34]	39.21	-	-	-
BAN [26]	39.31	-	-	-
MuRel [11]	39.54	42.85	13.17	45.04
UpDn [8]	39.74	42.27	11.93	46.05
UpDn+Q-Adv+DoE [33]	41.17	65.49	15.48	35.48
UpDn+C Images	41.01	44.61	12.38	46.11
UpDn+C Questions	40.62	42.33	14.17	48.32
UpDn+C (Q+I)	42.12	45.72	12.45	48.34

Table 3: State-of-the-art results on VQA-CP test. **UpDn+C** indicates our approach based on UpDn baseline. **(Q+I)** denotes both question and images are intervened.

Model	Overall
Question-Only [6]	25.98
BAN [26]	69.08
MuRel [11]	65.14
UpDn [8]	63.48
UpDn+Q-Adv+DoE [33]	62.75
Pythia [37]	68.49
Pythia+C	68.77

Table 4: Performance of our approach on VQA v2 validation. **Pythia+C** is our counterfactual implementation of [37].

Experiment

Smaller set

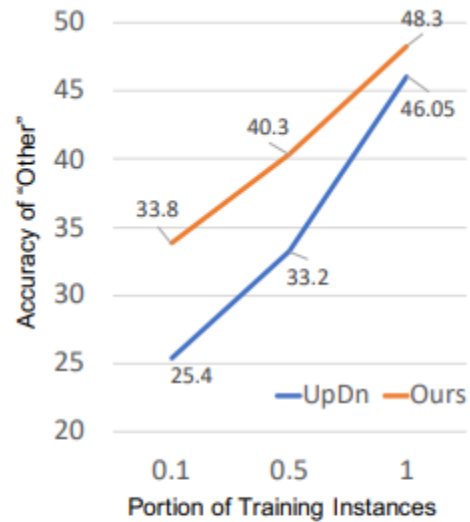


Figure 6: The performance of our approach vs. the baseline using fraction of the training data.

Experiment

Case study













Question Image		Counterfactual Questions	Counterfactual Images
	Is this in Australia?	<ol style="list-style-type: none"> 1. Is the grass green? 2. Is there grass on the ground? 3. Are they standing on a green grass field? 4. Is the stop light green? 	  
	What color is the person's helmet?	<ol style="list-style-type: none"> 1. What color jacket is the girl wearing? 2. What color jacket is the person wearing? 3. What color is the jacket? 4. What color is the woman's jacket? 	  
	Where did the shadow on the car come from?	<ol style="list-style-type: none"> 1. What kind of dog is this? 2. What type of dog is this? 3. What kind of dog is shown? 4. What is the breed of dog? 	  

Figure 5: Given the image-question pair in the first column, the closest instances of the questions (in second column) and images (in the third column) are found from the VQA v2 test dataset corresponding to the generated counterfactuals (using the exogenous variables).

Counterfactual Explanations for Machine Learning: A Review

Sahil Verma

University of Washington

Arthur AI

vsahil@cs.washington.edu

John Dickerson

Arthur AI

University of Maryland

john@arthur.ai

Keegan Hines

Arthur AI

keegan@arthur.ai

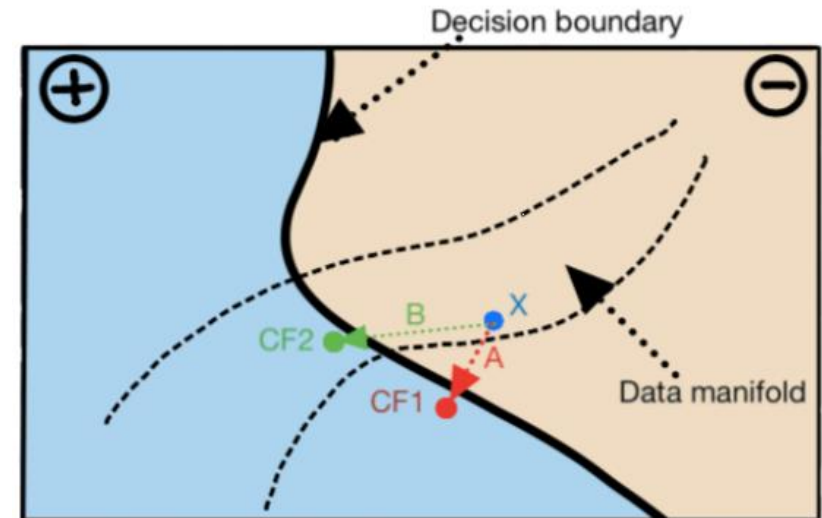
Desiderata and Major Themes of Research

1.Validity: minimize the distance between the counterfactual(x') and the original datapoint(x) subject to the constraint that the output of the classifier on the counterfactual is the desired label ($y' \in \mathcal{Y}$)

$$\arg \min_{x'} d(x, x') \text{ subject to } f(x') = y'$$

To differentiable:

$$\arg \min_{x'} \max_{\lambda} \lambda (f(x') - y')^2 + d(x, x')$$



Desiderata and Major Themes of Research

2.Actionability: “A recommended counterfactual should never change the immutable features.”
Call the set of actionable feature \mathcal{A} ,

$$\arg \min_{x' \in \mathcal{A}} \max_{\lambda} \lambda(f(x') - y')^2 + d(x, x')$$

3.Sparsity: A counterfactual ideally should change smaller number of features in order to be most effective.

$$\arg \min_{x' \in \mathcal{A}} \max_{\lambda} \lambda(f(x') - y')^2 + d(x, x') + g(x' - x)$$

4. Data Manifold closeness: “it is desirable that a generated counterfactual is realistic in the sense that it is near the training data and adheres to observed correlations among the features.”

$$\arg \min_{x' \in \mathcal{A}} \max_{\lambda} \lambda(f(x') - y')^2 + d(x, x') + g(x' - x) + l(x'; \mathcal{X})$$

Desiderata and Major Themes of Research

- 5. **Causality:** a counterfactual should maintain any known causal relations between features.
- 6. **Amortized inference:** : Generating a counterfactual is expensive……
- 7. **Alternative methods:** linear programming, mixed-integer programming, SMT solvers.

Relationship to other related terms

1.Recourse:...the difference with recourse **has blurred**.

2.Inverse classification: Inverse classification aims to perturb an input in a meaningful way in order to classify it into its desired class. Inverse classification has the **same goals** as counterfactual explanations.

3.Contrastive explanation:- Contrastive explanations generate explanations of the form “an input x is classified as y because features f_1, f_2, \dots, f_k are present and f_n, \dots, f_r are absent”. Contrastive explanations are **related** to counterfactual explanations.

4. Adversarial learning: Adversarial learning is a closely-related field, but the terms are **not interchangeable**. Adversarial learning aims to generate the least amount of change in a given input in order to classify it differently, often with the goal of **far-exceeding the decision boundary** and resulting in a highly-confident misclassification.

Research Challenge

1. Unify counterfactual explanations with traditional “explainable AI.”

... (CF explanation) do not tell which feature(s) was the principal reason for the original decision, and why. It would be nice if, along with giving actionable feedback, counterfactual explanations also gave the reason for the original decision, which can help applicants understand the model's logic...

2. Provide counterfactual explanations as discrete and sequential steps of actions.

...in the real world, actions are discrete and often sequential. Therefore the counterfactual generation process must take the discreteness of actions into account and provide a series of actions that would take the individual from the current state to the modified state, which has the desired class label.

3. Counterfactual explanations as an interactive service to the applicants

Counterfactual explanations should be provided as an interactive interface,...

.....

Counterfactual Visual Explanations

Yash Goyal¹ Ziyan Wu² Jan Ernst² Dhruv Batra¹ Devi Parikh¹ Stefan Lee¹

Counterfactual Vision Explanation

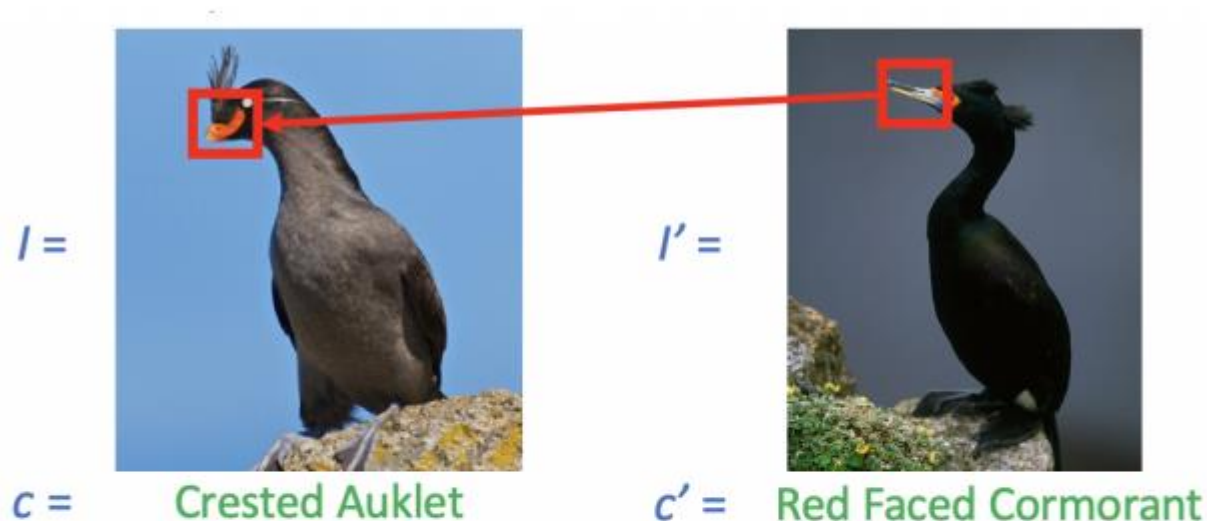


Figure 1. Our approach generates counterfactual visual explanations for a query image I (left) – explaining why the example image was classified as class c (*Crested Auklet*) rather than class c' (*Red Faced Cormorant*) by finding a region in a distractor image I' (right) and a region in the query I (highlighted in red boxes) such that if the highlighted region in the left image looked like the highlighted region in the right image, the resulting image I^* would be classified more confidently as c' .

Method

Minimum-Edit Counterfactual Problem

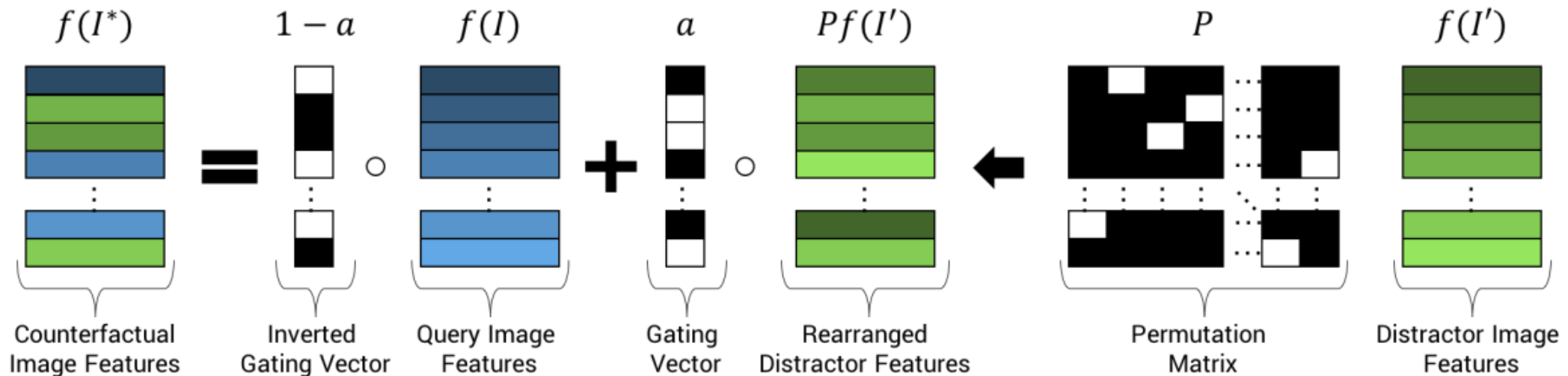


Figure 2. To parameterize our counterfactual explanations, we define a transformation that replaces regions in the query image I with those from a distractor I' . Distractor image features $f(I')$ are first rearranged with a permutation matrix P and then selectively replace entries in $f(I)$ according to a binary gating vector a . This allows arbitrary spatial cells in $f(I')$ to replace arbitrary cells in $f(I)$.

$$a \in \mathbb{R}^{hw}$$

$$P \in \mathbb{R}^{hw \times hw}$$

$$f(I^*) = (\mathbf{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ Pf(I')$$

Method

Minimum-Edit Counterfactual Problem

$$\begin{aligned} & \underset{P, \mathbf{a}}{\text{minimize}} \quad ||\mathbf{a}||_1 \\ & \text{s.t.} \quad c' = \operatorname{argmax} \quad g((\mathbb{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ Pf(I')) \\ & \quad \quad a_i \in \{0, 1\} \quad \forall i \quad \text{and} \quad P \in \mathcal{P} \end{aligned}$$

Where \mathcal{P} is the set of all $hw \times hw$ permutation metrics.

Method

Greedy Sequential Exhaustive Search

$$\underset{P, \mathbf{a}}{\text{maximize}} \quad g_{c'}((\mathbf{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ Pf(I'))$$

$$\text{s.t.} \quad \|\mathbf{a}\|_1 = 1, \quad a_i \in \{0, 1\} \quad \forall i$$

$$P \in \mathcal{P}$$

constrain \mathbf{a} to be one-hot – indicating the edit in I which maximizes the model log-probability $g_{c'} 0$.

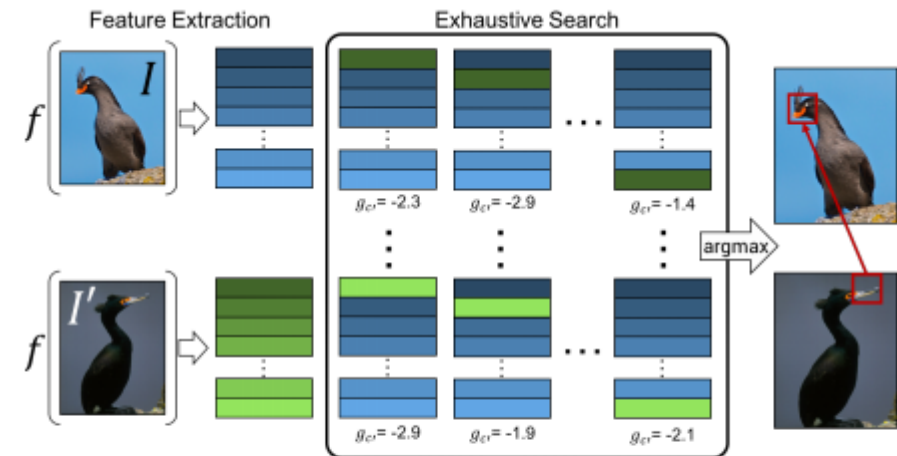


Figure 4. In our exhaustive best-edit search, we check all pairs of query-distractor spatial locations and select whichever pair maximizes the log probability of the distractor class c' .

Method

Continuous Relaxation

$$\begin{aligned} & \underset{P, \mathbf{a}}{\text{maximize}} && g_{c'}((\mathbf{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ Pf(I')) \\ & \text{s.t.} && \|\mathbf{a}\|_1 = 1, \quad a_i \geq 0 \quad \forall i \\ & && \|\mathbf{p}_i\|_1 = 1 \quad \forall i, \quad P_{i,j} \geq 0 \quad \forall i, j \end{aligned}$$

Specifically, we define $\mathbf{a} = \sigma(\alpha)$ and $\mathbf{p}_i^T = \sigma(\mathbf{m}_i^T)$

Experiment

MNIST & Omniglot

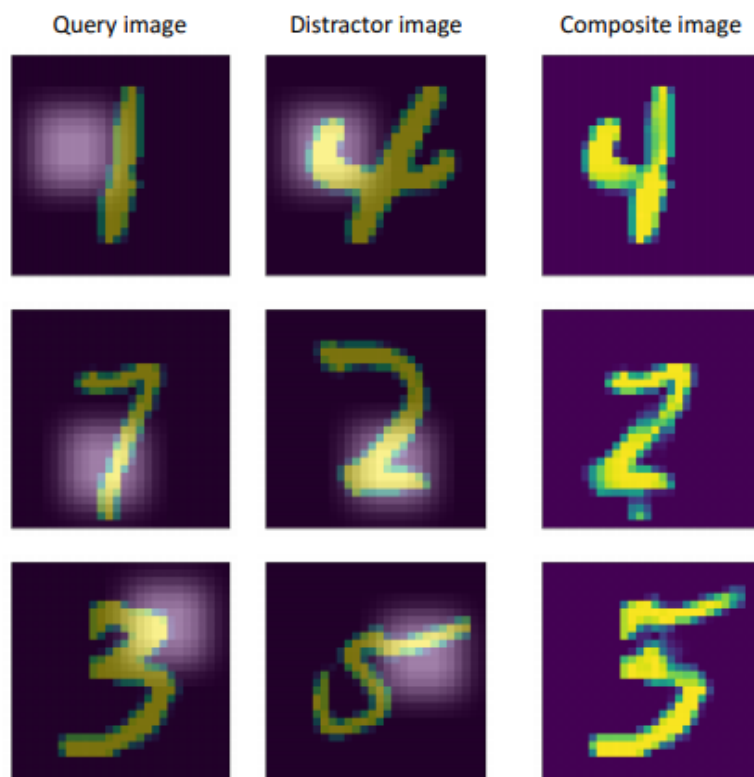


Figure 5. Results on MNIST (LeCun et al., 1998) dataset. The first two columns show the query and distractor images, each with their identified discriminative region highlighted. The third column shows composite images created by making the corresponding replacement in pixel space.

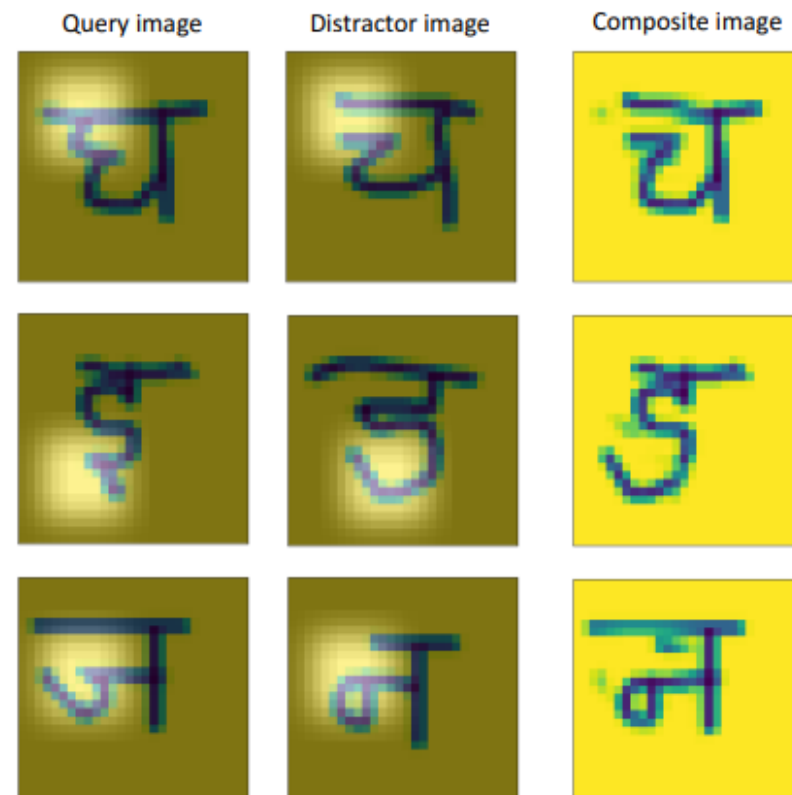


Figure 7. Qualitative results on the Omniglot dataset.

simple pen strokes; however, most humans are not going to a priori know the difference between characters. Hence, Omniglot is an ideal “mid-way point” between our MNIST and CUB experiments.

Experiment

Caltech-UCSD Birds

Query image



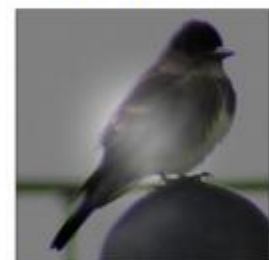
Eared Grebe

Distractor image



Horned Grebe

Composite image



Olive sided Flycatcher



Myrtle Warbler



Blue Grosbeak



Indigo Bunting



Northern Fulmar



Glaucous winged Gull



Anna Hummingbird



Ruby throated
Hummingbird



Experiment

Common Experimental Settings In all our experiments, we operate in output space of the last convolutional layer in the CNN but our approach is equally applicable to the output of any convolutional layer. Further, all qualitative results shown are with the exhaustive search approach presented in Section 2.2 as we are operating on relatively small images.

In our experiments on CUB (Sec. 4.3), we find the continuous relaxation presented in Sec. 2.3 achieves identical solutions to exhaustive search for 79.98% of instances and on average achieves distractor class probability that is 92% of the optimal found via exhaustive search – suggesting its usefulness for larger feature spaces.



Figure 10. Our machine teaching interface. During training phase (shown in (a)), if the participants choose an incorrect class, they are shown a feedback (shown in (b)) highlighting the fine-grained differences between the two classes. At test time (shown in (c)), they must classify the birds from memory.

Woulda, Coulda, Shoulda: Counterfactually-Guided Policy Search

Lars Buesing, Theophane Weber, Yori Zwols, Nicolas Heess, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau ICLR2019

Counterfactual Off-Policy Evaluation with Gumbel-Max Structural Causal Models

Michael Oberst, David Sontag ICML2019

Counterfactual Off-Policy Training for Neural Dialogue Generation

Qingfu Zhu , Weinan Zhang , Ting Liu, William Yang Wang EMNLP2020

Motivation

1. data **insufficiency** problem
 2. real users chatting usually **time-consuming** and **labor-intensive** in practice
 3. **humans could** independently reason potential responses based on past experiences from the true environment
- propose a **counterfactual off-policy training (COPT)**
approach to explore potential responses(**synthesized** responses)

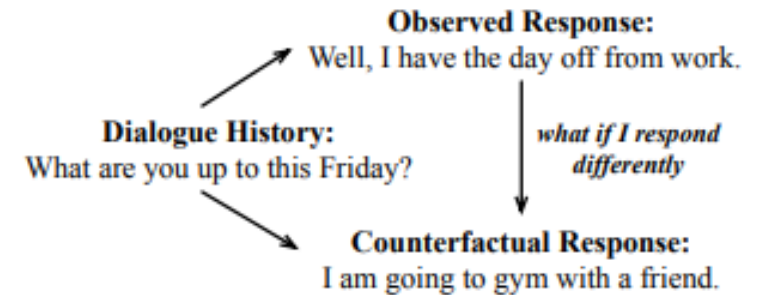


Figure 1: An example of a counterfactual response, which is a potential response inferred in hindsight from given observed response.

Method Notation

SCM

A structural causal model over random variables $V = \{V_1, \dots, V_N\}$ consists of independent noise random variables $U = \{U_1, \dots, U_N\}$ with distribution P_U and deterministic functions $F = \{f_1, \dots, f_N\}$ such that $V_i = f_i(\mathbf{PA}_i, U_i)$, where $\mathbf{PA}_i \subset V$ are the parents of V_i in a given DAG (Buesing et al., 2019). U is called *scenarios*, and F is called *causal mechanisms*. Figure 2 (Left) shows an example of an SCM. Each random variable V_i is determined by its parents in V , U_i , and f_i , e.g., $V_2 = f_2(V_1, U_2)$.

Dialogue Generation

Dialogue history \mathbf{X}

Response \mathbf{Y}

Condition Distribution $P(\mathbf{Y}|\mathbf{X})$

→ Deterministic function $\mathbf{Y} = f_\pi(\mathbf{X}, U)$

Method Notation

Intervention in SCM

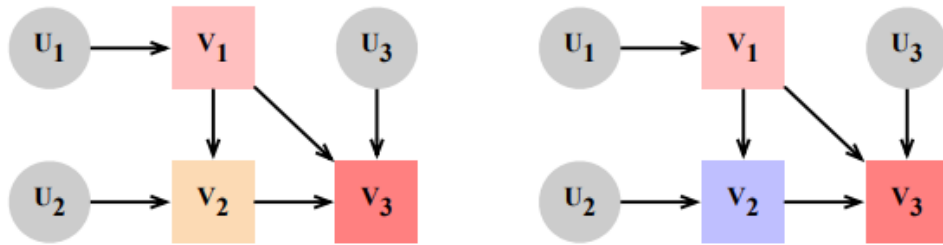


Figure 2: An example of an SCM and an intervention. **Left:** An SCM with random variables \mathbf{V} , scenarios \mathbf{U} , and causal mechanisms \mathbf{F} represented by colored squares. **Right:** A new SCM after taking an intervention on the left SCM. The original causal mechanism $f_2(\mathbf{V}_1, \mathbf{U}_2)$ (denoted by the orange square) is replaced by $f_2^T(\mathbf{V}_1, \mathbf{U}_2)$ (denoted by the purple square).

Counterfactual Reasoning in SCM

Given an SCM and observed a variable, $\mathbf{V}_i = \mathbf{v}_i$ counterfactual answers the question: “*What the variable \mathbf{V}_i would have been if I take an intervention \mathbf{T} while remaining everything else unchanged*” In this way, generating a counterfactual response can be seen as querying: “*Having observed a response $\mathbf{Y} = \mathbf{y}$, what the response \mathbf{Y} would have been if I take an intervention by following the **target policy π** , rather than the **behavior policy μ** that generates the observed responses*”

Method

3-steps

- Observed $Y = y$ when $X = x$, infer the scenario u in hindsight from $Y = f_\mu(X, U)$.
- Take an intervention by replacing the causal mechanism $f_\mu(X, U)$ with $f_\pi(X, U)$.
- Reason a counterfactual response $\hat{y} = f_\pi(x, u)$ by the resulting new SCM.

Method

*Two policies: the target policy that we aim to learn and the behavior policy used for the reasoning of scenarios. The behavior policy is pre-trained and then **froze during adversarial learning** because it aims to maximize the likelihood of a fixed set of observed responses*

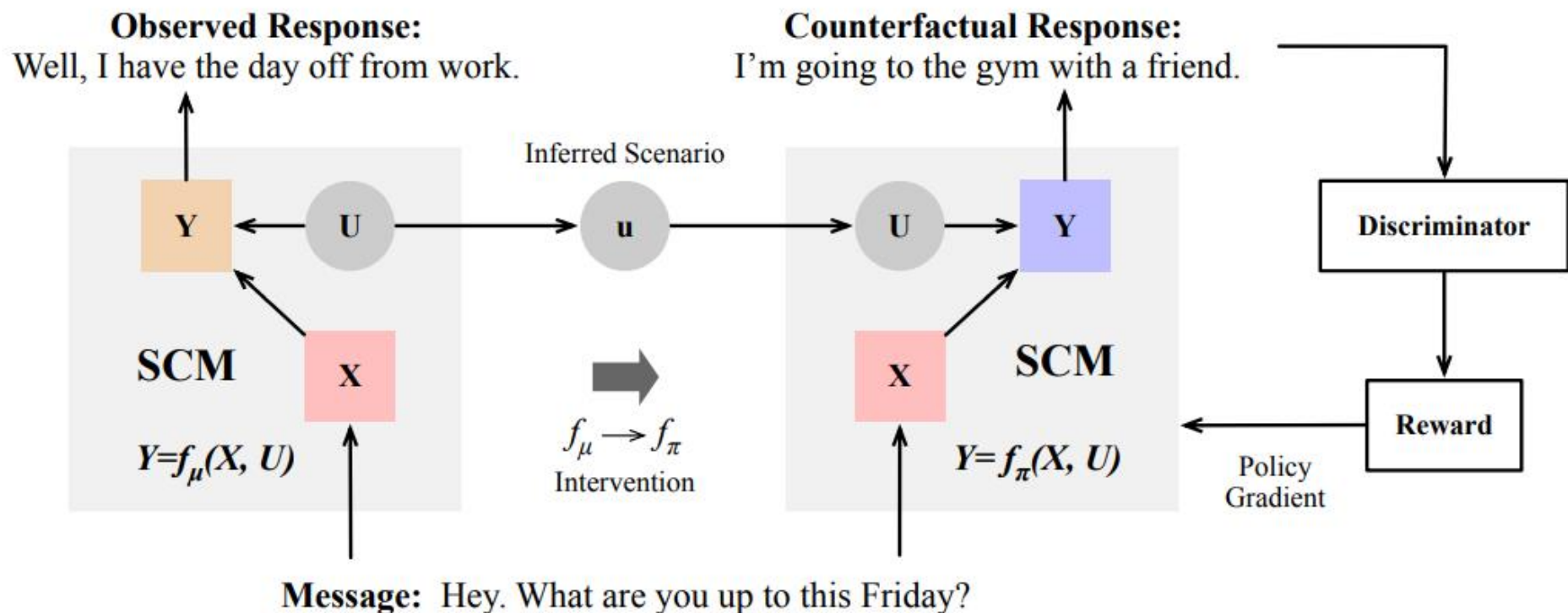


Figure 3: The architecture of our COPT approach. π is the target policy that we aim to learn. μ is the behavior policy that generates observed responses. First, we infer the scenario u where the observed response occurs. Then we update the policy from μ to π , which can be seen as an intervention on the left SCM and results in the right SCM. Then, the counterfactual response is reasoned in the inferred scenario u by the causal mechanism $Y = f_\pi(X, U)$.

Method

Generator

$$\mathbf{H}_i = \text{LSTM}(\mathbf{X}_i, \mathbf{H}_{i-1}),$$

$$\mathbf{S}_j = \text{LSTM}([\mathbf{e}(\hat{\mathbf{Y}}_{j-1}), \mathbf{C}_j], \mathbf{S}_{j-1}),$$

$$\mathbf{P}_j^\pi(\hat{\mathbf{Y}}_j | \mathbf{X}, \hat{\mathbf{Y}}_{1:j-1}) = \text{softmax}(\mathbf{S}_j \cdot \mathbf{O}),$$

Adversarial learning-based dialogue generation model is optimized according to the reward of responses sampled from

$$\mathbf{P}_j^\pi(\hat{\mathbf{Y}}_j | \mathbf{X}, \hat{\mathbf{Y}}_{1:j-1}) \in \mathbb{R}^{|V|} \quad (\text{abbreviated as } \mathbf{P}_j^\pi)$$

Using the Gumbel-Max Trick

$$\hat{\mathbf{Y}}_j = \arg \max_{\hat{\mathbf{Y}}_j} (\log \mathbf{P}_j^\pi + \mathbf{U}_j)$$

\mathbf{U}_j follows the standard Gumbel distribution.

Method

Infer the scenario

For the policy $\mu: y_j = \arg \max_{y_j} (\log p_j^\mu + u_j)$

Two ways to infer the μ

1) Reject sampling : sampling u_j from the standard Gumbel distribution and rejects those where $y_j \neq \arg \max_{y_j} (\log p_j^\mu + u_j)$.

2) Using properties of gumble-max:

$$g = \log p_j^\mu + u_j$$

the maximum of g follows the standard Gumbel distribution and is independent with the argmax of g .

g can be obtained by first sampling a maximum and then sampling the remaining elements truncated at the maximum

Method

Discriminator & training

The output reward $D(\tilde{Y}_j|\mathbf{X}, \tilde{\mathbf{Y}}_{1:j-1})$ is the probability that \tilde{Y}_j is human-generated

Generator

$$J_G(\boldsymbol{\theta}) = -\mathbb{E}_{\hat{\mathbf{Y}}_{1:j} \sim G} D(\hat{Y}_j | \mathbf{X}, \hat{\mathbf{Y}}_{1:j-1})$$

$$\begin{aligned} \nabla J_G(\boldsymbol{\theta}) = & -\mathbb{E}_{\hat{\mathbf{Y}}_{1:j} \sim G} D(\hat{Y}_j | \mathbf{X}, \hat{\mathbf{Y}}_{1:j-1}) \\ & \cdot \nabla \log G_\pi(\hat{Y}_j | \mathbf{X}, \hat{\mathbf{Y}}_{1:j-1}), \end{aligned}$$

Discriminator

$$\begin{aligned} J_D(\phi) = & -\mathbb{E}_{\mathbf{Y}_{1:j} \sim \mathcal{S}} \log D(Y_j | \mathbf{X}, \mathbf{Y}_{1:j-1}) \\ & - \mathbb{E}_{\hat{\mathbf{Y}}_{1:j} \sim G} \log(1 - D(\hat{Y}_j | \mathbf{X}, \hat{\mathbf{Y}}_{1:j-1})), \end{aligned} \quad (6)$$

Method

Algorithm 1 Counterfactual Off-Policy Training

- 1: Pre-train π and μ with MLE loss;
 - 2: Pre-train D on positive instances sampled from observed responses, and negative instances generated by pre-trained π ;
 - 3: **for** epoch in number of epochs **do**
 - 4: **for** g in g-steps **do**
 - 5: Infer u from an observed response;
 - 6: Generate a counterfactual response in u ;
 - 7: Optimize θ according to Equation 5;
 - 8: **end for**
 - 9: **for** d in d-steps **do**
 - 10: Sample positive instances from observed responses;
 - 11: Sample negative instances from responses generated by π ;
 - 12: Update ϕ according to Equation 6;
 - 13: **end for**
 - 14: **end for**
-

Experiment

Data: Daily Dialog

Training Dialogues	11,118
Validation Dialogues	1,000
Test Dialogues	1,000
Average Tokens Per Dialogue	114.7
Average Tokens Per Utterance	14.6

Table 1: Statistics of the DailyDialog dataset.

Experiment Result

Model	Dist-1	Dist-2	BLEU-1	BLEU-2	BLEU-3	BLEU-4
HRED (Serban et al., 2016)	0.011	0.045	33.01	4.54	1.05	0.29
DPGAN (Xu et al., 2018)	0.002	0.008	31.61	3.74	0.43	0.06
REGS (Li et al., 2017a)	0.021	0.097	38.39	6.85	2.00	0.68
REGS+COPT	0.026	0.116	39.80	7.67	2.34	0.77
StepGAN (Tuan and Lee, 2019)	0.018	0.097	34.76	6.51	1.97	0.60
StepGAN+COPT	0.026	0.117	43.83	8.43	2.66	0.95

Table 3: Automatic evaluation results of distinct-1 (Dist-1), distinct-2 (Dist-2), and BLEU scores.

Experiment

Case study

Dialogue History	A: This is our counter-offer. B: We are sincere to make business with you, but the difference between your counter-offer and our price is too great.
HRED	A: We can do that, but you can take a look at this one.
DPGAN	A: Think that the will be, but he said that they had a think we have a good job in a. Me, I am sure that I can be able to have a look at the first course.
REGS	A: If you are willing to withdraw money, we can offer you a private vehicle.
REGS+COPT	A: If you are willing to pay a commission fee, we can offer you \$5 % discount.
StepGAN	A: What do you mean by that?
StepGAN+COPT	A: In that case, we can offer you a discount.

Table 4: An example of generated responses given dialogue history between person A and B.

Conclusion

1. The counterfactual routine
2. A more directional(specific) adverse learning(data augmentation)
3. To design an inference(posterior) method

How to inference the posterior?

Deep Structural Causal Models for Tractable Counterfactual Inference

Nick Pawlowski, Daniel Coelho de Castro, Ben Glocker *NeurIPS 2020*

Notation

SCM $\mathfrak{G} := (\mathbf{S}, P(\epsilon))$

Collection of structural assignment

$$\mathbf{S} = (f_1, \dots, f_K)$$

$$x_k := f_k(\epsilon_k; \mathbf{pa}_k) \text{ (called *mechanisms*),}$$

Joint distribution $P(\epsilon) = \prod_{k=1}^K P(\epsilon_k)$ (independent exogenous noise)

1. **Abduction:** Predict the ‘state of the world’ (the exogenous noise, ϵ) that is compatible with the observations, \mathbf{x} , i.e. infer $P_{\mathfrak{G}}(\epsilon | \mathbf{x})$.
2. **Action:** Perform an intervention (e.g. $\text{do}(x_k := \tilde{x}_k)$) corresponding to the desired manipulation, resulting in a modified SCM $\tilde{\mathfrak{G}} = \mathfrak{G}_{\mathbf{x}; \text{do}(\tilde{x}_k)} = (\tilde{\mathbf{S}}, P_{\mathfrak{G}}(\epsilon | \mathbf{x}))$ [1, Sec. 6.4].
3. **Prediction:** Compute the quantity of interest based on the distribution entailed by the counterfactual SCM, $P_{\tilde{\mathfrak{G}}}(\mathbf{x})$.

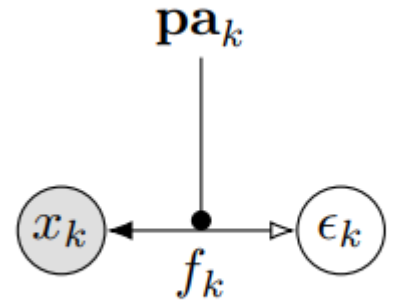
Deep Mechanisms

Invertible, explicit

Normalizing flow:

$$\begin{aligned} \epsilon &\sim P(\epsilon) & x &= f(\epsilon) & \implies & p(x) = p(\epsilon) |\det \nabla f(\epsilon)|^{-1} \\ \text{s.t} & & \epsilon &= f^{-1}(x) \end{aligned}$$

For DSCM (condition situation)



Invertible explicit likelihood

$$x_i := f_i(\epsilon_i; \mathbf{pa}_i), \quad p(x_i | \mathbf{pa}_i) = p(\epsilon_i) \cdot |\det \nabla_{\epsilon_i} f_i(\epsilon_i; \mathbf{pa}_i)|^{-1} \Big|_{\epsilon_i = f_i^{-1}(x_i; \mathbf{pa}_i)}.$$

Problem: Heavy computational for high-dimensional data

Deep Mechanisms

Amortised, explicit

“Separate the f_k into a “low-level”, **invertible component** h_k and a “high-level”, **non-invertible component** g_k with corresponding noise decomposition: $\epsilon_k = (u_k, z_k)$.

$$x_k := f_k(\epsilon_k; \mathbf{pa}_k) = h_k(u_k; g_k(z_k; \mathbf{pa}_k), \mathbf{pa}_k)$$

$$P(\epsilon_k) = P(u_k)P(z_k)$$

Amortised variational inference

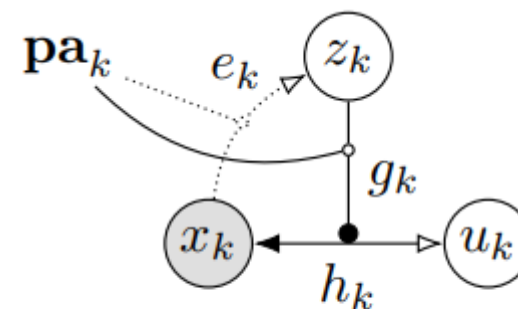
$$\log p(x_k | \mathbf{pa}_k) \geq \mathbb{E}_{Q(z_k | x_k, \mathbf{pa}_k)} [\log p(x_k | z_k, \mathbf{pa}_k)] - D_{\text{KL}}[Q(z_k | x_k, \mathbf{pa}_k) \| P(z_k)] .$$

Where

$$p(x_k | z_k, \mathbf{pa}_k) = p(u_k) \cdot |\det \nabla_{u_k} h_k(u_k; g_k(z_k, \mathbf{pa}_k), \mathbf{pa}_k)|^{-1} \Big|_{u_k = h_k^{-1}(x_k; g_k(z_k, \mathbf{pa}_k), \mathbf{pa}_k)} .$$

And a encoder

$$Q(z_k | x_k, \mathbf{pa}_k)$$

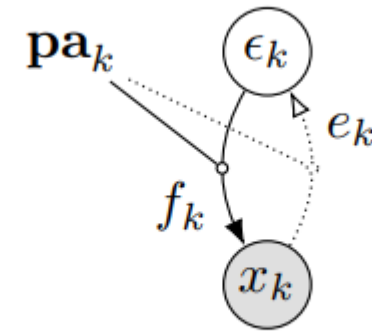


Amortised explicit likelihood

Deep Mechanisms

Amortised, implicit

VAE, GAN...



(c) Amortised implicit likelihood

Deep counterfactual inference

Abduction

For invertible mechanisms: $\epsilon_i = f_i^{-1}(x_i; \mathbf{pa}_i)$

For implicit-likelihood mechanisms: $\epsilon_j \approx e_j(x_j; \mathbf{pa}_j)$

For amortised, explicit-likelihood mechanisms:

$$\begin{aligned} P_{\mathfrak{G}}(\epsilon_k | x_k, \mathbf{pa}_k) &= P_{\mathfrak{G}}(z_k | x_k, \mathbf{pa}_k) P_{\mathfrak{G}}(u_k | z_k, x_k, \mathbf{pa}_k) \\ &\approx Q(z_k | e_k(x_k; \mathbf{pa}_k)) \delta_{h_k^{-1}(x_k; g_k(z_k; \mathbf{pa}_k), \mathbf{pa}_k)}(u_k). \end{aligned}$$

Deep counterfactual inference

Action

$$\text{SCM } \tilde{\mathfrak{G}} = (\tilde{\mathbf{S}}, P_{\tilde{\mathfrak{G}}}(\epsilon | \mathbf{x}))$$

Deep counterfactual inference

Prediction

Sample from $\tilde{\mathcal{G}}$

$$\begin{aligned}z_k^{(s)} &\sim Q(z_k | e_k(x_k; \mathbf{pa}_k)) \\u_k^{(s)} &= h_k^{-1}(x_k; g_k(z_k^{(s)}; \mathbf{pa}_k), \mathbf{pa}_k) \\\tilde{x}_k^{(s)} &= \tilde{h}_k(u_k^{(s)}; \tilde{g}_k(z_k^{(s)}; \tilde{\mathbf{pa}}_k), \tilde{\mathbf{pa}}_k) .\end{aligned}$$