

# **Diffusion Causal Models for Counterfactual Estimation**

Thomas Melistas

# Overview

- What are counterfactuals?
- Introduction to Structural Causal Models
- Unifying Diffusion and SCMs
- Related Work

# What are counterfactuals?

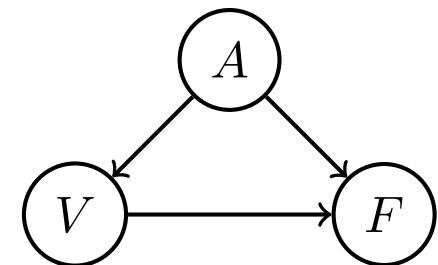
- Hypothetical scenarios
- Example: I got the vaccine and did not catch the flu, what is the probability I wouldn't catch the flu if I didn't get vaccinated?
- It cannot be computed straight away (I already got vaccinated)

Let's formalize this

# Graphs as Joint Distribution Factorizations

- Represent Bayesian Networks as DAGs
- Encode conditional independence (Markovian)

$$p(a, v, f) = p(a)p(v|a)p(f|a, v)$$

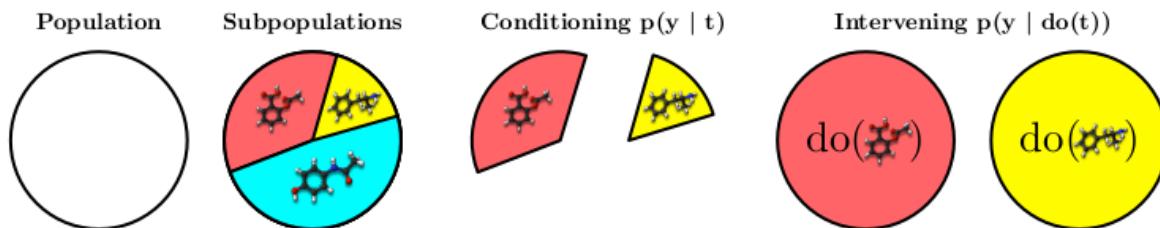


$A$ : Age,  $V$ : Vaccination,  $F$ : Flu

# Interventions

- Applying an action to a population → getting the vaccine
- We use the do-operator  $do(V = v)$  or  $do(v)$
- Different than conditioning:  $p(f|do(v)) \neq p(f|v)$

$$p(a, f|do(v)) = p(a) \cancel{p(v|a)}^1 p(f|a, v) \neq p(a, f|v)$$



# Counterfactuals

What is the probability that I (27 years old) wouldn't get the flu if I didn't get the vaccine?

$$p(f|do(v'), f, a) ?$$

- **Conflict:** first  $f$  is the hypothetical scenario and second  $f$  is the observed one
- We denote the first by  $f_{v'}$ :  
**Counterfactual flu probability under intervention of no vaccination**

$p(f|do(v'), a)$  : How probable is flu **if** we don't vaccinate people

$p(f_{v'}|v, f, a)$ : How probable the flu **would be** if I hadn't vaccinated

# Pearl's Causal Hierarchy

Layer	Activity	Semantics	Example
(1) Associational $p(y   x)$	Seeing 	How would seeing $x$ change my belief in $Y$ ?	What does a symptom tell us about the disease?
(2) Interventional $p(y   \text{do}(x), z)$	Doing 	What happens to $Y$ if I do $x$ ?	What if I take aspirin, will my headache be cured?
(3) Counterfactual $p(y_{x'}   x, y)$	Imagining 	Was it $x$ that caused $Y$ ?	Was it the aspirin that stopped my headache?

# **Structural Causal Models**

- Extension of Bayesian Networks
- Causal relationships → Deterministic, functional equations
- Stochasticity → Some variables remain unobserved

# Structural Causal Models

An SCM  $\mathcal{M} := (\mathbf{S}, p(\epsilon))$  consists of:

- (i) structural assignments  $\mathbf{S} = \{f_i\}_{i=1}^N$ , s.t.  $x_i := f_i(\epsilon_i, \mathbf{pa}_i)$ ,
- (ii) a joint distribution  $p(\epsilon) = \prod_{i=1}^N p(\epsilon_i)$  over mutually independent noise variables

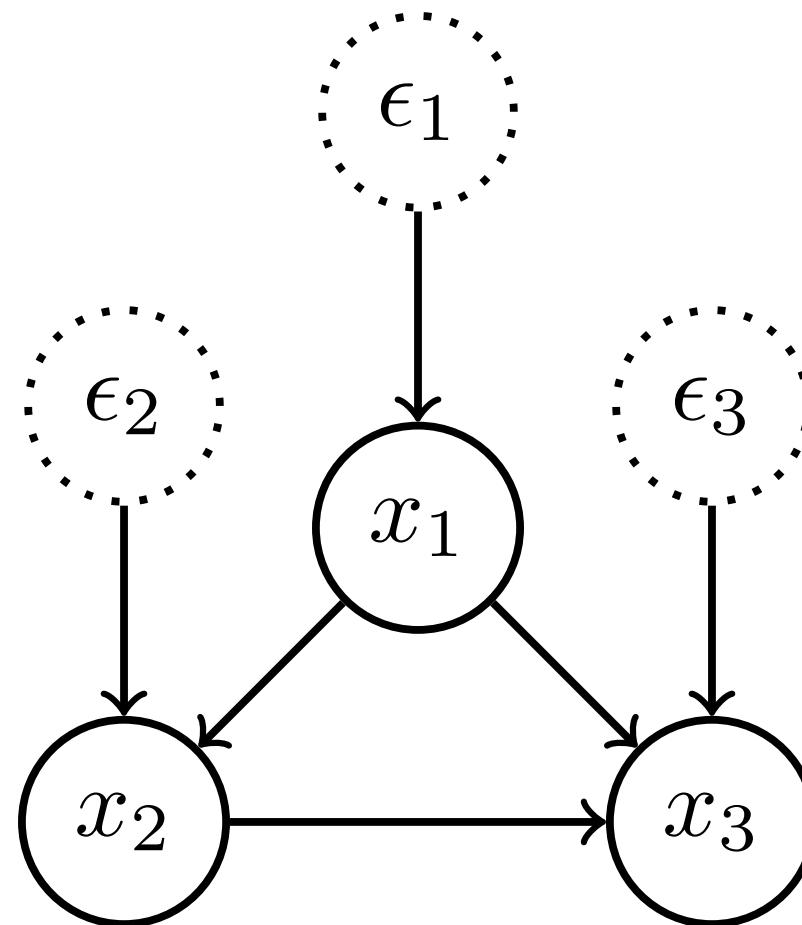
$x_i$ : an **endogenous** variable (observed)

$\mathbf{pa}_i$ : the parents of  $x_i$  (its direct causes, endogenous)

$\epsilon_i$ : an **exogenous** variable (unobserved)

## Structural Causal Models

- In any SCM, each variable  $x_i$  is caused by parent variables and unobserved exogenous **noise** variables  $\epsilon_i$



# Counterfactual Inference with SCMs

In order to compute counterfactuals we need to:

1. Estimate the noise  $\epsilon$  given the observed datum  $\mathbf{x}$  (factual)  
We define  $\mathcal{M}_x := (\mathbf{S}, p(\epsilon|\mathbf{x}))$
2. To intervene on its structural assignments  $\mathbf{S}$  with intervention  $do(x_i = \tilde{x}_i)$

We define the modified, counterfactual SCM

$$\widetilde{\mathcal{M}} := \mathcal{M}_{\mathbf{x}; do(\tilde{x}_i)} = (\widetilde{\mathbf{S}}, p(\epsilon|\mathbf{x}))$$

# Counterfactual Inference with SCMs

Three-step procedure:

1. **Abduction:** Infer  $p(\epsilon|x)$ , the state of the world (exogenous noise) that is compatible with the observations  $x$ .
2. **Action:** Replace the structural equations ( $do(\tilde{x}_i)$ ) corresponding to the intervention, resulting in a modified SCM  $\widetilde{\mathcal{M}} := \mathcal{M}_{x;do(\tilde{x}_i)} = (\tilde{\mathbf{S}}, p(\epsilon|x))$
3. **Prediction:** Use the modified model to compute  $p_{\widetilde{\mathcal{M}}}(\mathbf{x})$

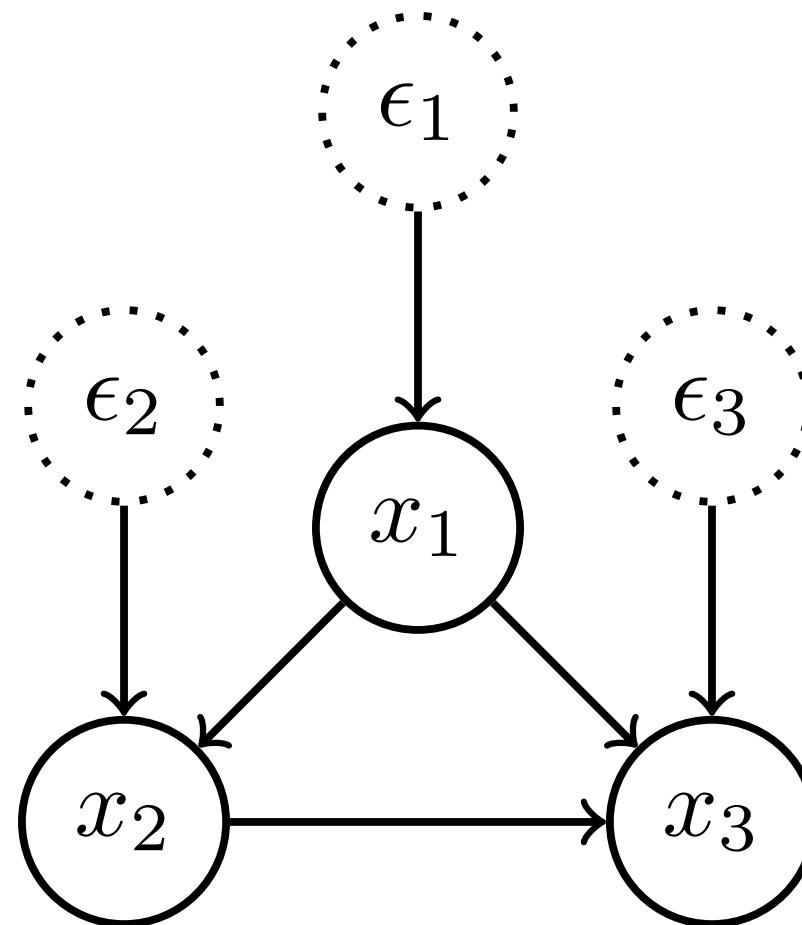
# Principle of Independent Mechanisms

$$p(\mathbf{x}) = \prod_{i=1}^N p(x_i | \mathbf{pa}_i)$$

If we intervene on a subset  $S$ , then  
for all  $i$ :

1. If  $i \notin S$ ,  $p(x_i | \mathbf{pa}_i)$  remains unchanged
2. If  $i \in S$ ,  $p(x_i | \mathbf{pa}_i) = 1$ , if  $x_i$  the value set by the intervention,  
otherwise 0

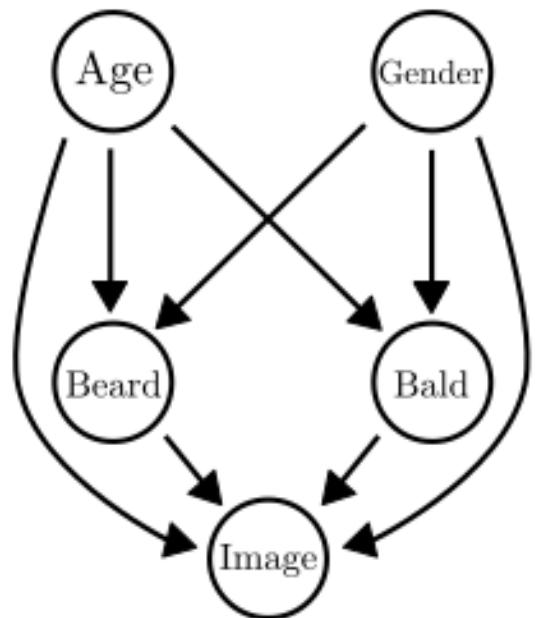
e.g. when intervening on  $x_2$  the  
edges towards  $x_2$  are removed



# What about images?

We can think of the image as a variable and its attributes as parents

**Example** of a possible graph and counterfactual estimation



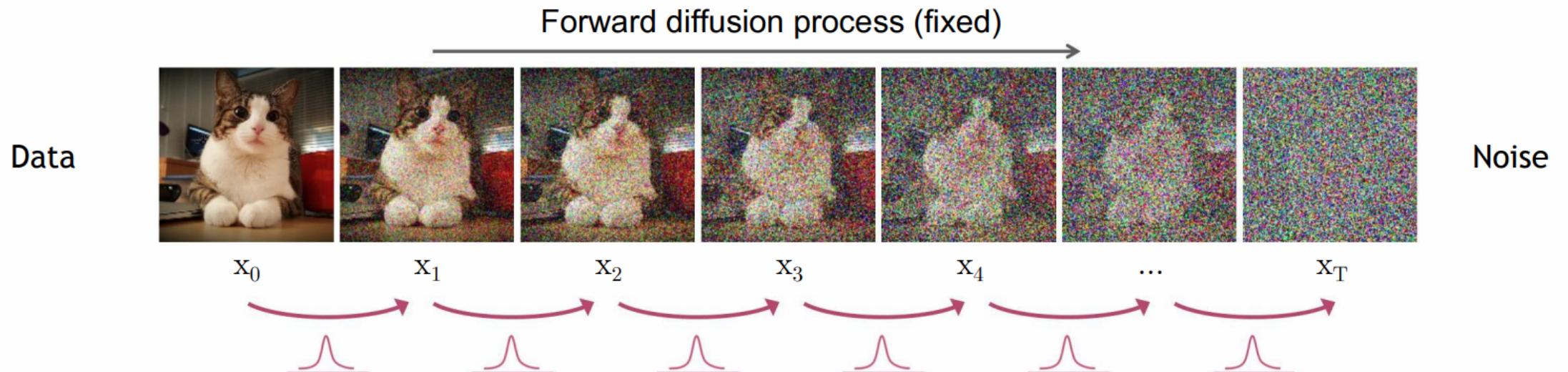
age : old  
gender : female  
beard : absent  
bald : not



age : old  
gender : male  
beard\* : present  
bald : true

# Recap on DDPM

- $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1 - \alpha_t)I)$  (noising step)
- $p_\theta(x_T) = \mathcal{N}(x_T; 0, I)$ .
- $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma(x_t, t)^2 I)$  (denoising step)



$$\operatorname{argmin}_\theta = \underbrace{\frac{(1-a_t)^2}{2\sigma_q^2(t)(1-\bar{a}_t)a_t}}_{\lambda_t} [||e_0 - e_\theta(x_t, t)||_2^2]$$

## Recap on Score-based models

$$\operatorname{argmin}_{\theta} = \frac{(1-a_t)^2}{2\sigma_q^2(t)a_t} [||s_{\theta}(x_t, t) - \nabla_{x_t} \log p(x_t)||_2^2]$$

and SDEs

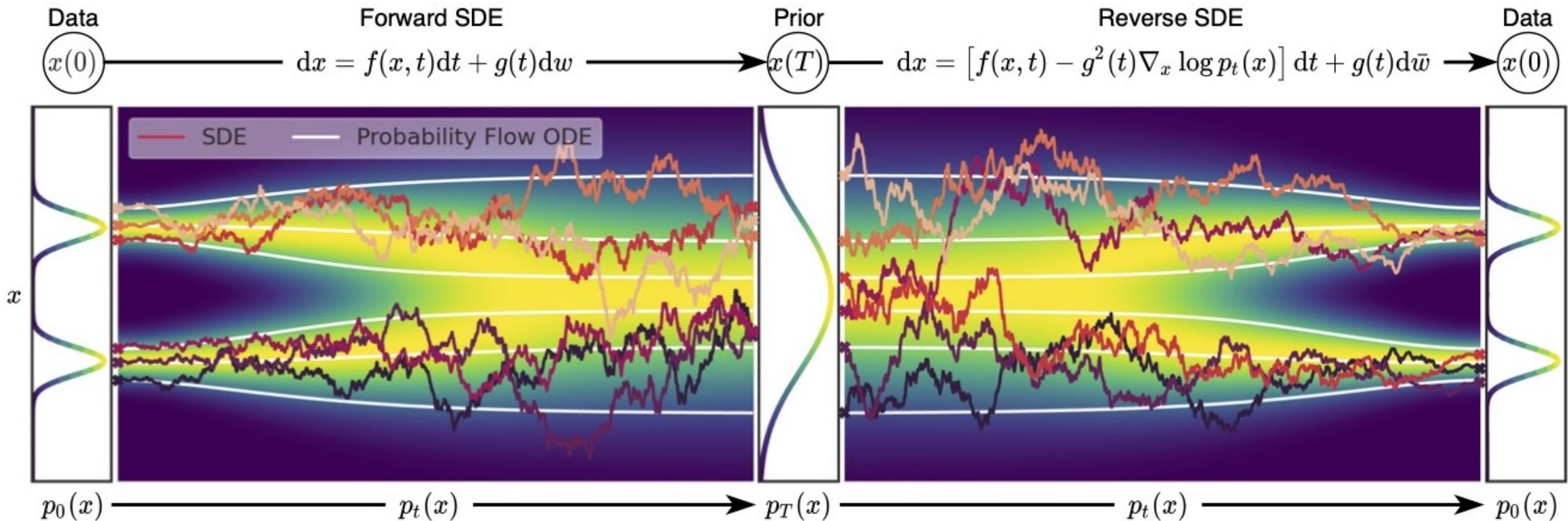
$$dx = f(x, t)dt + g(t)dw$$

To sample from  $x(T) \sim p_T$  and get new data from  $p_{data}$ , we can reverse the SDE:

$$dx = [f(x, t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)d\tilde{w}.$$

With **(i)** the terminal distribution  $p(T) \approx \pi(x)$  and **(ii)** the score  $\nabla_x \log p_t(x)$ , we train a *time dependent score model*, with objective:

$$\mathbb{E}_{t \in U(0, T)} \mathbb{E}_{p_t(x)} [\lambda(t) \|\nabla_x \log p_t(x) - s_{\theta}(x, t)\|_2^2]$$



# Teaser on DDIM and Classifier-Guidance

---

**Algorithm 1** Classifier guided diffusion sampling, given a diffusion model  $(\mu_\theta(x_t), \Sigma_\theta(x_t))$ , classifier  $p_\phi(y|x_t)$ , and gradient scale  $s$ .

---

```
Input: class label  $y$ , gradient scale  $s$ 
 $x_T \leftarrow$  sample from  $\mathcal{N}(0, \mathbf{I})$ 
for all  $t$  from  $T$  to 1 do
     $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$ 
     $x_{t-1} \leftarrow$  sample from  $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$ 
end for
return  $x_0$ 
```

---

---

**Algorithm 2** Classifier guided DDIM sampling, given a diffusion model  $\epsilon_\theta(x_t)$ , classifier  $p_\phi(y|x_t)$ , and gradient scale  $s$ .

---

```
Input: class label  $y$ , gradient scale  $s$ 
 $x_T \leftarrow$  sample from  $\mathcal{N}(0, \mathbf{I})$ 
for all  $t$  from  $T$  to 1 do
     $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t) - \sqrt{1 - \bar{\alpha}_t} \nabla_{x_t} \log p_\phi(y|x_t)$ 
     $x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left( \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}} \hat{\epsilon}$ 
end for
return  $x_0$ 
```

---

## Diff-SCM: Motivation

- SCMs have been previously used to provide a causal interpretation of SDEs for modeling time-dependent problems
- SDEs have been used to formalize diffusion process in a continuous manner
- Diff-SCM models the dynamics of causal variables as an Ito process  $\mathbf{x}_t^{(k)}, \forall t \in [0, T]$ , going from an observed endogenous variable  $\mathbf{x}_0^{(k)} = \mathbf{x}^{(k)}$  to its respective exogenous noise  $\mathbf{x}_T^{(k)} = \mathbf{u}^{(k)}$  and back

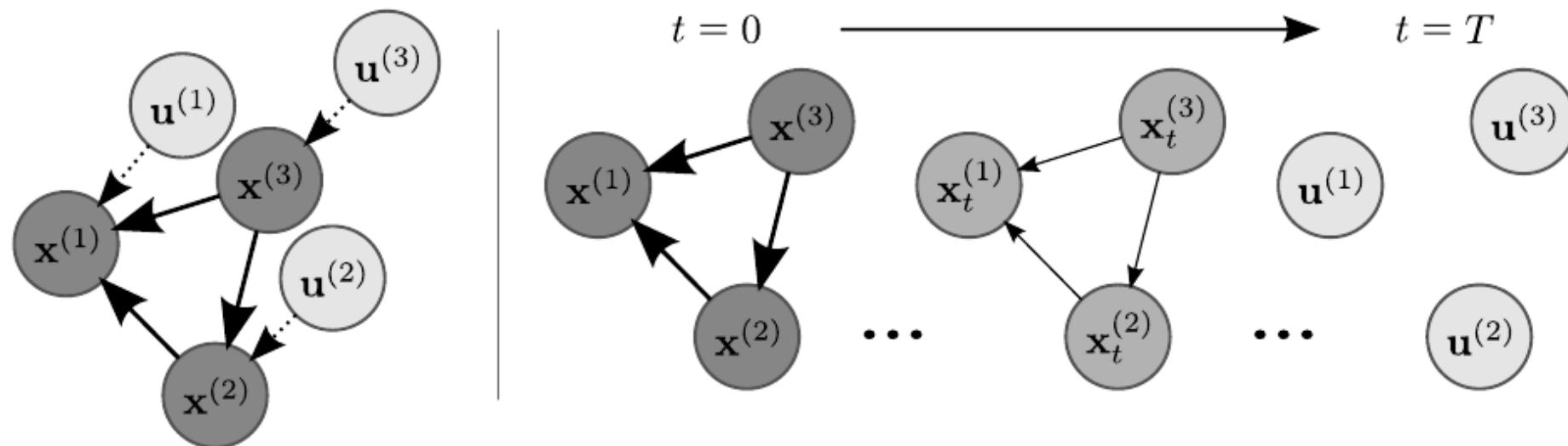
Sokol, A. & Hansen N. (2014). Causal interpretation of stochastic differential equations

Bongers, S., & Mooij, J.M. (2018). From Random Differential Equations to Structural Causal Models: the stochastic case

Song, Y. et al. (2020). Score-Based Generative Modeling through Stochastic Differential Equations

# Diff-SCM: Unifying Diffusion Processes and Causal Models

- Forward diffusion  $\rightarrow$  gradual weakening of the causal relations between variables of a SCM



- The original joint distribution  $p_{\mathcal{G}}$  diffuses to independent Gaussians  $p(U)$

# Diff-SCM: Unifying Diffusion Processes and Causal Models

- We can define a Deep SCM as a set of SDEs (1 for each node  $k$ ):

$$d\mathbf{x}^{(k)} = -\frac{1}{2}\beta_t \mathbf{x}^{(k)} dt + \sqrt{\beta_t} d\mathbf{w}, \forall k \in [1, K],$$

where  $p(\mathbf{x}_0^{(k)}) = \prod_{j=k}^K p(\mathbf{x}^{(j)} | \mathbf{pa}^{(j)})$  and  $p(\mathbf{x}_T^{(k)}) = p(\mathbf{u}^{(k)})$

- The reverse-time SDE of the above is given by:

$$d\mathbf{x}^{(k)} = \left[ -\frac{1}{2}\beta_t + \beta_t \nabla_{\mathbf{x}_t^{(k)}} \log p(\mathbf{x}_t^{(k)}) \right] + \sqrt{(\beta_t)} \bar{\mathbf{w}}$$

To solve: Iteratively update  $\mathbf{x}_T^{(k)} = \mathbf{u}^{(k)}$  with the gradient w.r.t. the input variable  $\nabla_{\mathbf{x}_t^{(k)}} \log p(\mathbf{x}_t^{(k)})$  until it becomes  $\mathbf{x}_0^{(k)} = \mathbf{x}^{(k)}$

# Apply Interventions with Anti-Causal Predictors

- We train an anti-causal classifier for each edge and a diffusion model for each node
- We use the gradients of the classifiers and diffusion models to propagate the intervention in the causal direction over the nodes

**Proposition 1 (Interventions as anti-causal gradient updates)** *We consider the SCM  $\mathfrak{G}$  and a variable  $\mathbf{x}^{(j)} \in \text{an}^{(k)}$ . The effect observed on  $\mathbf{x}^{(k)}$  caused by an intervention on  $\mathbf{x}^{(j)}$ ,  $p_{\mathfrak{G}}(\mathbf{x}^{(k)} | \text{do}(\mathbf{x}^{(j)} = x^{(j)}))$ , is equivalent to solving a reverse-diffusion process for  $\mathbf{x}_t^{(k)}$ . Since the sampling process involves taking into account the distribution entailed by  $\mathfrak{G}$ , it is guided by the gradient of an **anti-causal predictor** w.r.t. the effect when the cause is assigned a specific value:*

$$\nabla_{x_t^{(k)}} p_{\mathfrak{G}^-}(\mathbf{x}^{(j)} = x^{(j)} | x_t^{(k)})$$

# Counterfactual Estimation with Diff-SCM

Three step procedure:

1. **Abduction:** Infer  $p(\mathbf{u}|\mathbf{x})) \rightarrow \text{Forward diffusion*}$
2. **Action:** Modify SCM  $\widetilde{\mathcal{M}} := \mathcal{M}_{\mathbf{x}; do(\tilde{x}_i)} = (\widetilde{\mathbf{S}}, p(\mathbf{u}|\mathbf{x})) \rightarrow \text{Remove the edges between the intervened variable and its parents}$
3. **Prediction:** Compute  $p_{\widetilde{\mathcal{M}}}(\mathbf{x}) \rightarrow \text{Reverse diffusion* controlled by the gradients of an anti-causal classifier}$

\*It actually follows the DDIM algorithm:

Song, J. et al. (2020). Denoising Diffusion Implicit Models

## Counterfactual Estimation with Diff-SCM

**Formally:** Estimate counterfactual  $x_{CF}^{(k)}$  based on factual (observed)  $x_F^{(k)}$  after  
 $do(\mathbf{x}^{(j)} = x_{CF}^{(j)}), \mathbf{x}^{(j)} \in \mathbf{an}^{(k)}$

They do experiments for the simple case, two variables:  $\mathbf{x}^{(j)} \rightarrow \mathbf{x}^{(k)}$

## Counterfactual Estimation with Diff-SCM

**Abduction of Exogenous Noise** By performing forward diffusion with an invertible process (e.g. DDIM, Neural ODEs) we get a latent  $u^{(k)}$  that is invertible

**Prediction under Intervention** Counterfactual estimation: apply an intervention in the reverse diffusion process with the gradients of an anti-causal predictor  
(see *classifier guidance of DDIM*)

---

**Algorithm 1** Inference of **counterfactual** for a variable  $\mathbf{x}^{(k)}$  from an intervention on  $\mathbf{x}^{(j)} \in \mathbf{an}^{(k)}$ 

---

**Models:** trained diffusion model  $\epsilon_\theta$  and anti-causal predictor  $p_\phi(x^{(j)} | x_t^{(k)})$

**Input :** factual variable  $x_{0,F}^{(k)}$ , target intervention  $x_{0,CF}^{(j)}$ , scale  $s$

**Output:** counterfactual  $x_{0,CF}^{(k)}$

### Abduction of Exogenous Noise – Recovering $u^{(k)}$ from $x_{0,F}^{(k)}$

**for**  $t \leftarrow 0$  **to**  $T$  **do**

$$x_{t+1,F}^{(k)} \leftarrow \sqrt{\alpha_{t+1}} \left( \frac{x_{t,F}^{(k)} - \sqrt{1-\alpha_t} \epsilon_\theta(x_{t,F}^{(k)}, t)}{\sqrt{\alpha_t}} \right) + \sqrt{\alpha_{t+1}} \epsilon_\theta(x_{t,F}^{(k)}, t)$$

**end**

$$u^{(k)} = x_{T,F}^{(k)} = x_T^{(k)}$$

### Generation under Intervention

**for**  $t \leftarrow T$  **to**  $0$  **do**

$$\epsilon \leftarrow \epsilon_\theta(x_t^{(k)}, t) - s\sqrt{1-\alpha_t} \nabla_{x_t^{(k)}} \log p_\phi(x_{0,CF}^{(j)} | x_t^{(k)})$$

$$x_{t-1}^{(k)} \leftarrow \sqrt{\alpha_{t-1}} \left( \frac{x_t^{(k)} - \sqrt{1-\alpha_t} \epsilon}{\sqrt{\alpha_t}} \right) + \sqrt{\alpha_{t-1}} \epsilon$$

**end**

$$x_{0,CF}^{(k)} = x_0^{(k)}$$

# Controlling the Intervention

Three factors enable counterfactual estimation:

- (i) The inferred  $u^{(k)}$  keeps information of the factual observation
- (ii)  $\nabla_{\mathbf{x}_t^{(k)}} \log p_\phi(x_{CF}^{(j)} | x_t^{(k)})$  guides the intervention towards the desired counterfactual class
- (iii)  $\epsilon_\theta(x_t^{(k)}, t)$  forces the estimation to belong to the data distribution

A hyperparameter  $s$  controls the scale of  $\nabla_{\mathbf{x}_t^{(k)}} \log p_\phi(x_{CF}^{(j)} | x_t^{(k)})$

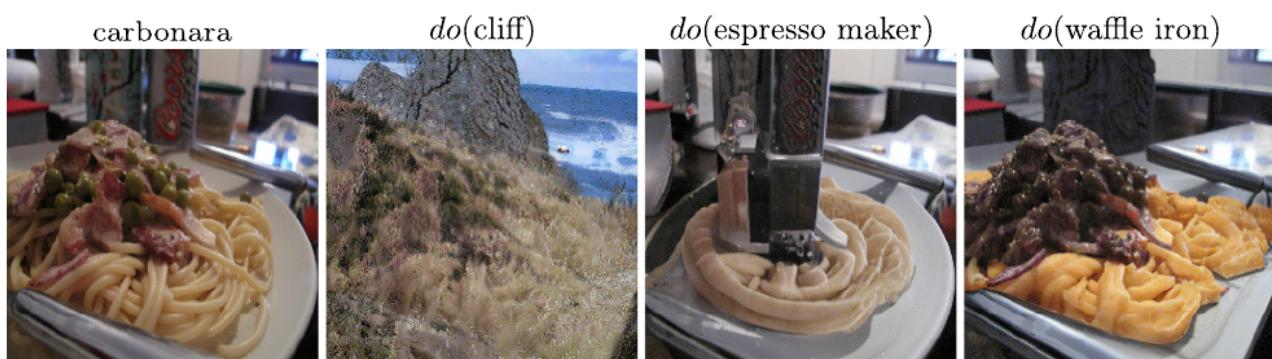
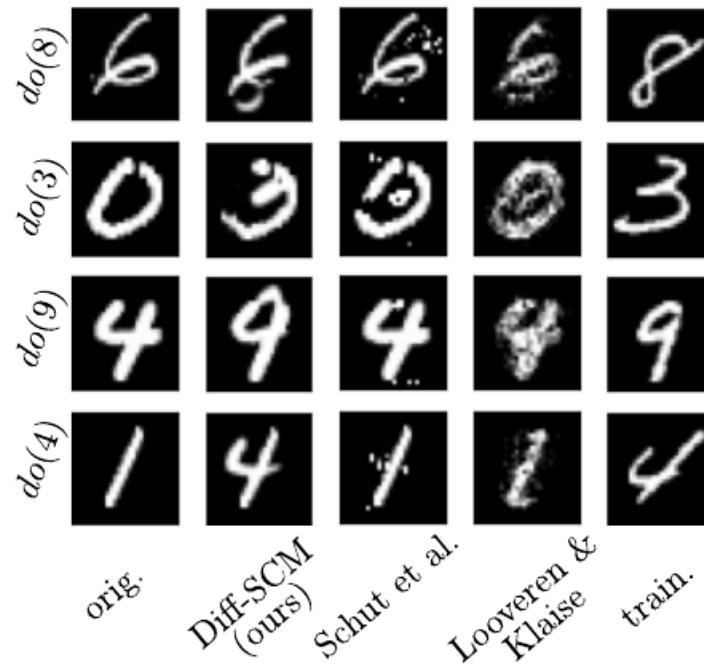
## Experimental Setup

Given an SCM  $\mathcal{G}_{image}$  with two variables  $\mathbf{x}^{(1)} \leftarrow \mathbf{x}^{(2)}$ ,  $\mathbf{x}^{(1)}$ : image,  $\mathbf{x}^{(2)}$ : class

- A diffusion model  $\epsilon_\theta$  (U-Net) learns the score of the marginal distribution of  $\mathbf{x}^{(1)}$
- An anticausal classifier  $p_\phi(\mathbf{x}^{(2)} | \mathbf{x}^{(1)})$  (encoder of  $\epsilon_\theta$  + pooling + linear layer)

Both conditioned on  $t$  and trained separately

Trained and tested on MNIST and ImageNet



Schut, L. et al. (2021) Generating Interpretable Counterfactual Explanations By Implicit Minimisation of Epistemic and Aleatoric Uncertainties

Van Looveren, A et al. (2021). Interpretable Counterfactual Explanations Guided by Prototypes

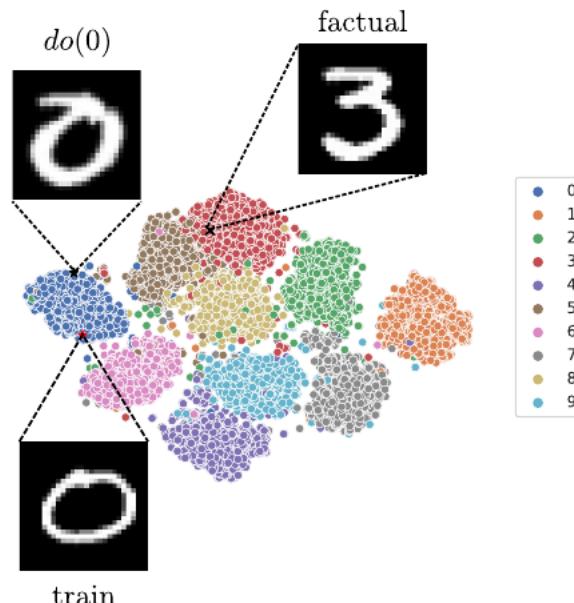
# Counterfactual Latent Divergence (CLD)

- Used to evaluate minimality of intervention in latent space
- They train a VAE and measure KL divergence

$$div = D(x_{\text{CF}}^{(1)}, x_{\text{F}}^{(1)}), \quad \text{with } D(x_i^{(1)}, x_j^{(1)}) = D_{\text{KL}}(\mathcal{N}(\mu_i, \sigma_i), \mathcal{N}(\mu_j, \sigma_j))$$

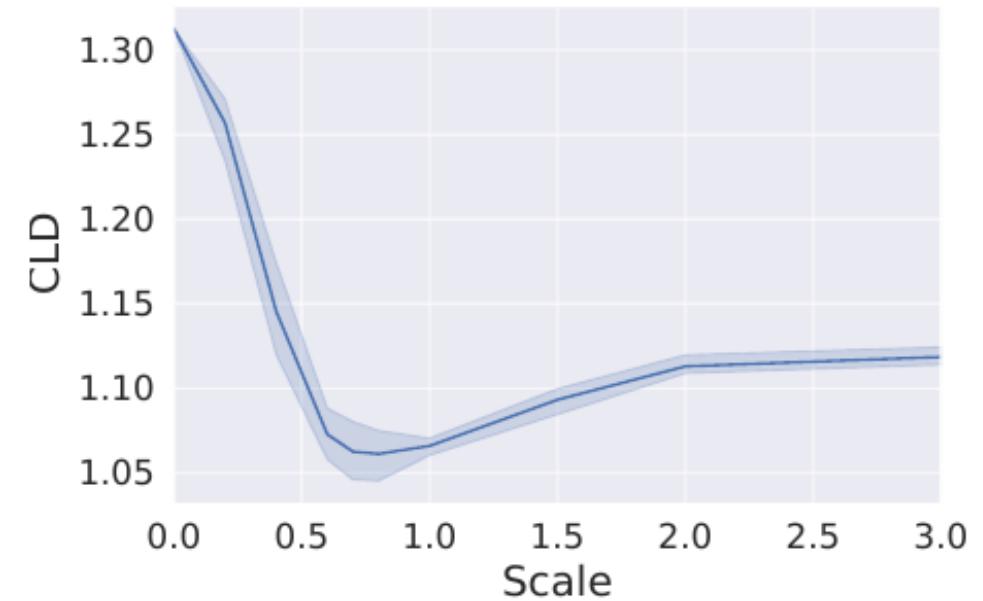
$$\mathcal{S}_{\text{class}} = \{D(x^{(1)}, x_{\text{F}}^{(1)}) \mid (x^{(1)}, x^{(2)}) \in \mathcal{D} \wedge x^{(2)} = x_{\text{class}}^{(2)}\}$$

$$\text{CLD} = \log (\exp (P (\mathcal{S}_{\text{CF}} \leq div)) + \exp (P (\mathcal{S}_{\text{F}} \geq div)))$$



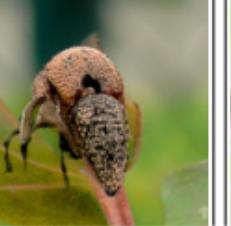
## Use CLD to tune $s$

orig.	7 3 / 2 9 7 9 6 0
rec.	7 3 / 2 9 7 9 6 0
$s = 0.1$	7 3 / 7 9 7 5 6 0
$s = 0.7$	5 5 5 5 5 5 5 5 5 5
$s = 2.0$	5 5 5 5 5 5 5 5 5 5



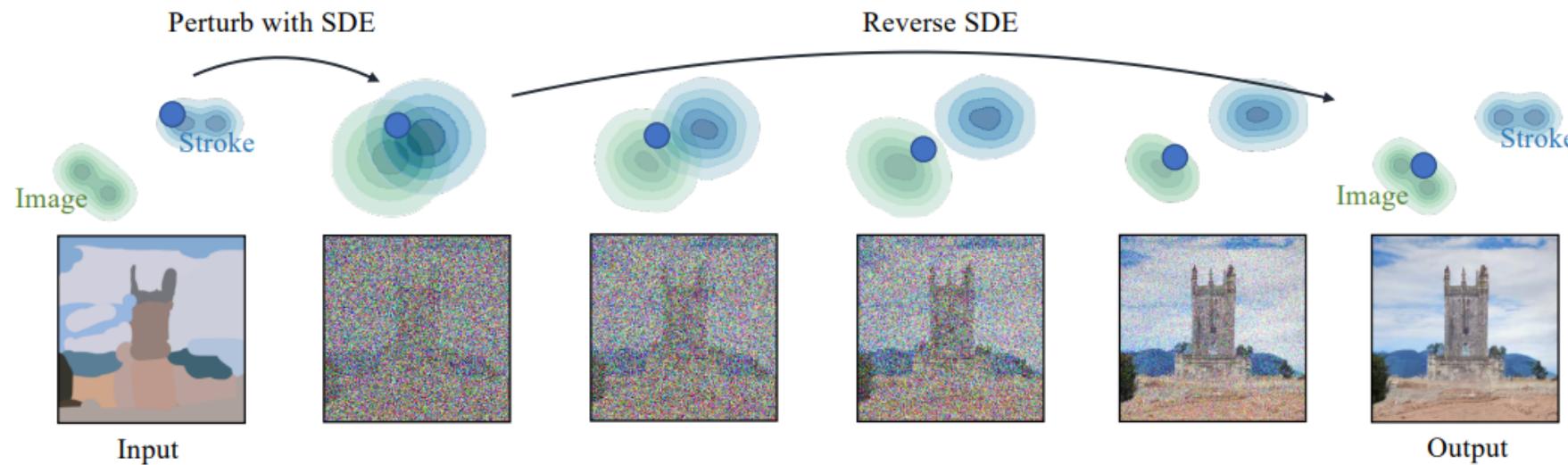
# Related work

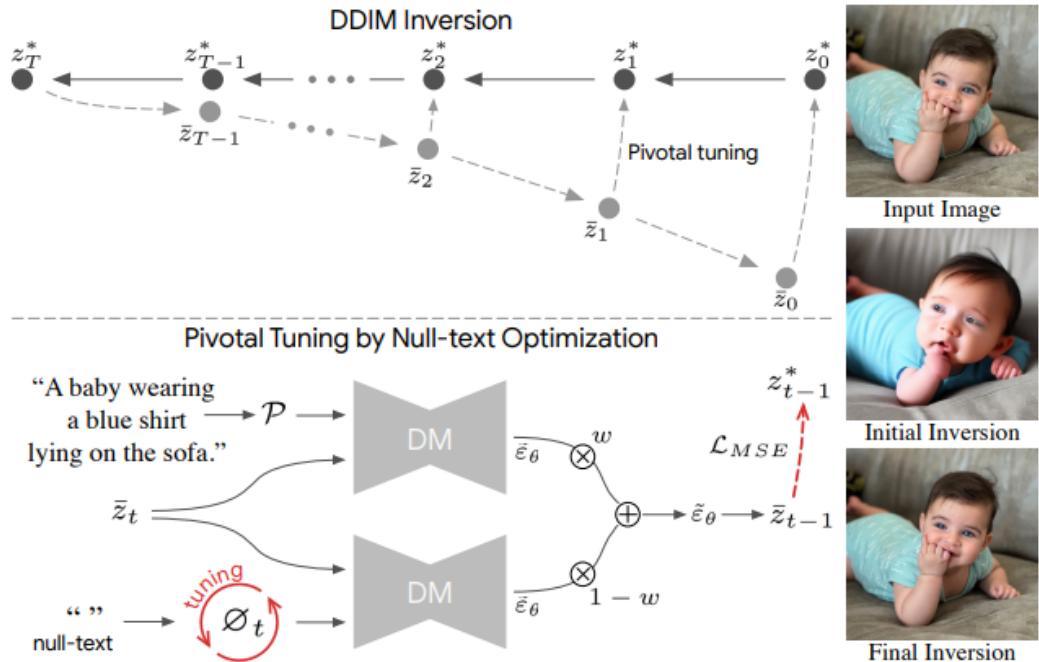
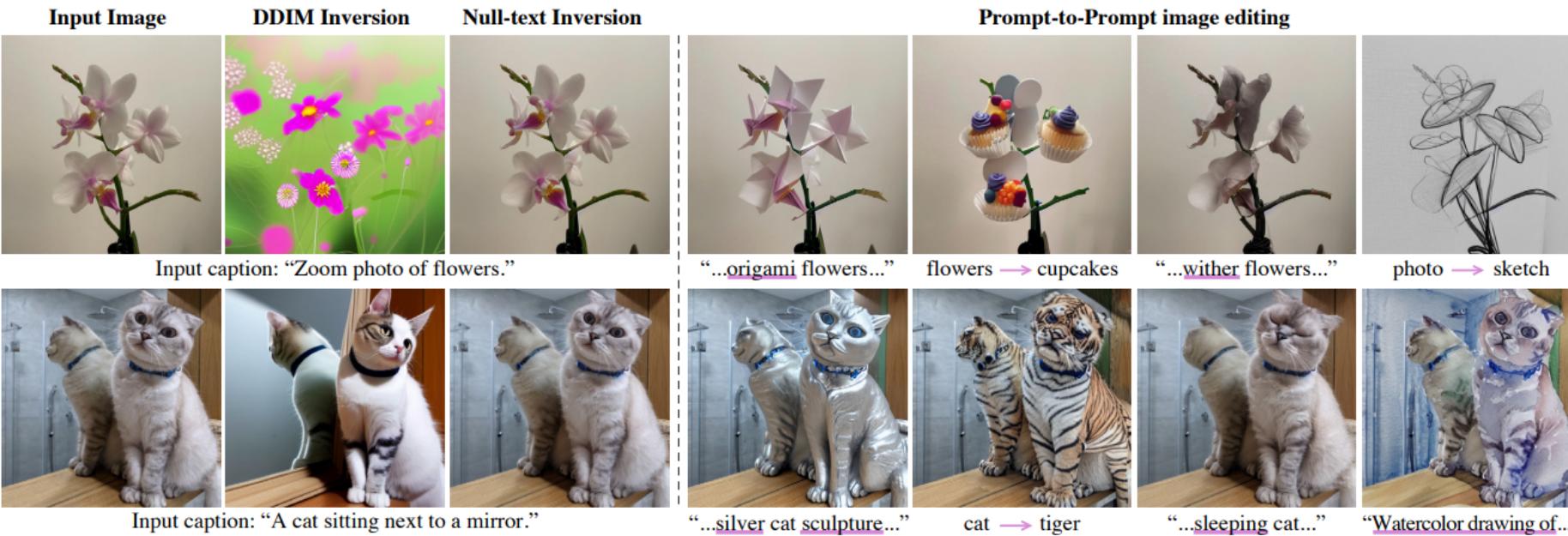
- Counterfactual Explanations is a technique to test the robustness and bias of models by making minimal changes in order to change a classifier's prediction

Original	Non-robust	Robust	Cone Proj.	Original	Non-robust	Robust	Cone Proj.
ladybug	weevil: 0.99	weevil: 1.00	weevil: 0.99	ringlet	monarch: 0.47	monarch: 1.00	monarch: 0.98
							

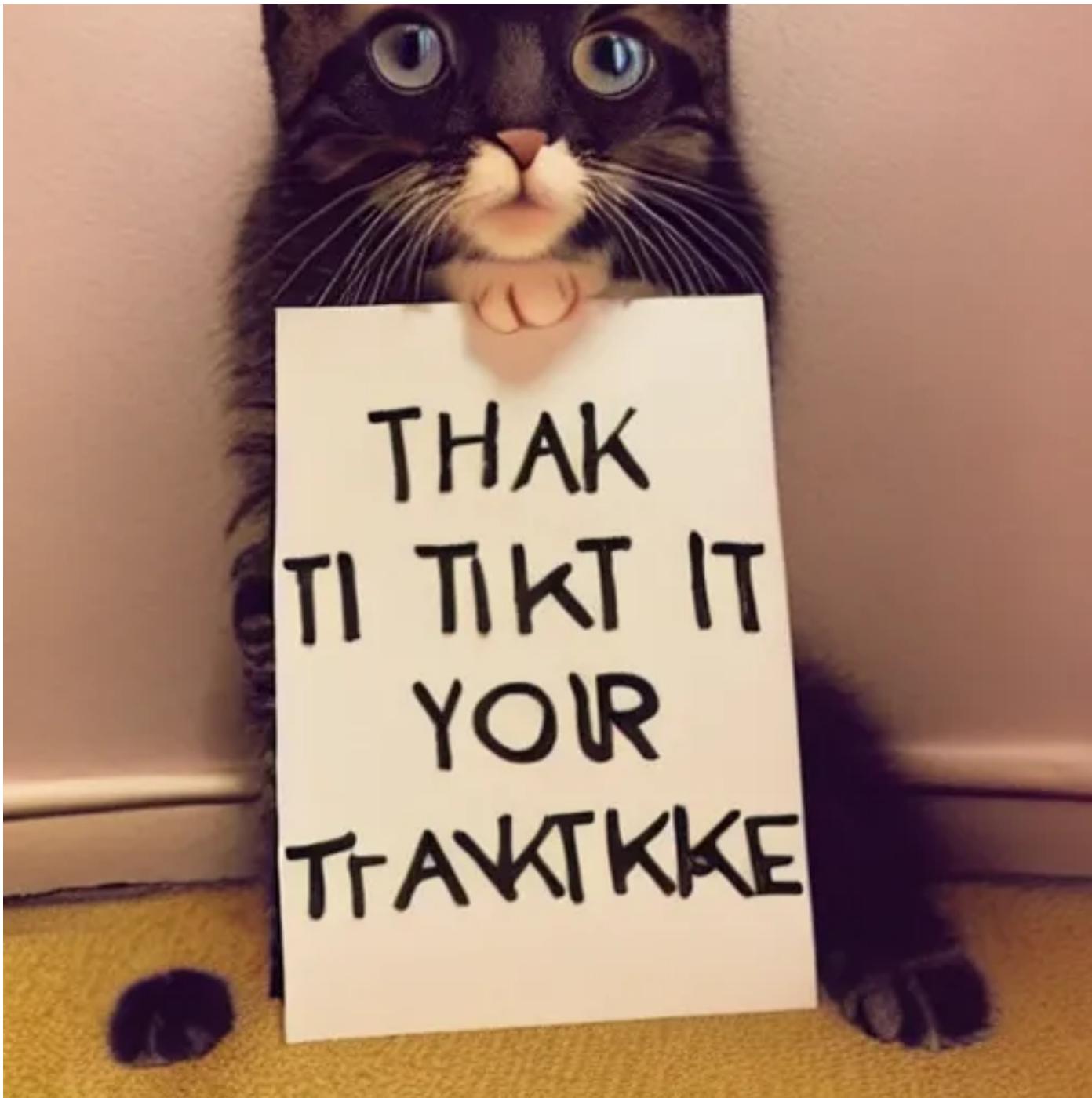
# Related work

- Image-to-Image translation can be thought of as a case of counterfactual estimation
- Existing methods such as SDEdit or Null-text-Inversion deploy a very similar technique





Thank you  
for your time!



# Questions

- What are the difficulties in extending Diff-SCM to the general case?
- How can counterfactual properties such as minimality be evaluated?
- What can this framework offer, more than the classic image editing framework?