Ctf-Editing using a Diffusion NCM

Atharv Sardesai
Department of Computer Science
Columbia University
New York, NY 10027
ahs2204@columbia.edu

1 Problem Statement

Diffusion based models (1; 2) have become the state of the art for image generation. However, these models often suffer from confounding bias. These biases exhibit themselves during image editing tasks when we want to modify certain parts of the image while keeping other parts constant. Spurious correlation between two latent generative factors might cause them to interfere with each other during the editing task.

We propose an NCM (Neural Causal Model) (3; 4), to address some of these issues in the context of diffusion based image generation and editing. We leverage the A-NCM and estimation of i-ctf distribution through a more relaxed set of assumptions and causal graphs (5) in order to estimate our counterfactual queries. Furthermore, we also tackle the challenge of editing real images in the diffusion setting, i.e. when we do not have access to the initial noise distribution.

2 Introduction

Structural Causal Models are better able to capture causal relationships present in the data. An SCM is defined by the tuple $< U, V, FP(U) >$ where U are the exogenous variables, V are the known variables, F is the functional relationship between these variables and P(U) is the probability function over U. This SCM induces a Causal Graph G. An NCM is a subset of SCM's where the functional relationships are calculated using a neural network.

Image editing tasks often require answers to "What If" questions, which can be defined in causal terms as a counterfactual query. If we have access to the true SCM we can perform an intervention on it and sample from this distribution. We regard this as finding the i-ctf distribution. It has been shown (5) that finding the true i-ctf distribution from purely observational data is not possible as this query is non-identifiable. However, there have been attempts at numerically estimating this quantity by constraining it to a Causal Diagram G. Namely, we try to find a surrogate SCM $\hat{M}$ which has the same Causal diagram G and same $P(U)$ as our original SCM. Then we sample from this surrogate SCM and perform interventions on it.

3 Ctf-image editing in Diffusion models

Latent Diffusion Models have become widely used for high quality image generation. These models transform data from an initial distribution to a target distribution in multiple steps. However, these models also suffer from the same challenges that all non-causal models suffer from i.e. confounding bias in the dataset. In the context of image editing tasks, this means modifying certain features may lead to unwanted changes in other generated features.

In order to solve this challenge in the context of diffusion based models we can adopt the ASCM proposed here (5). We replace the image generator from a VAE to a diffusion based model.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

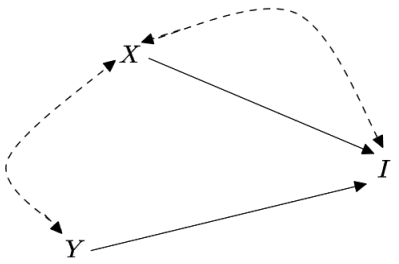
An additional challenge to using diffusion-based models is that although we can edit images generated by the diffusion model itself as the initial distribution is known, doing so with real images is difficult as we do not know the distribution U which can generate this image.

Let us see why this is the case, by taking a look at a simple ASCM example.

Let us consider an ASCM $< U, \{V, I\}, F, P(U) >$ such that $V = \{X, Y\}$, $U = \{u_x, u_y, u_{xy}, u_I\}$, $P(u_x = 0) = 0.4, P(u_y = 0) = 0.6, P(u_{xy} = 0) = 0.5, P(u_I = 0) = 0.5$ All variables except for I are binary variables for simplicity.

We define an $\mathscr{F}$ such that
$X \longleftarrow u_x \oplus u_{xy}$
$Y \longleftarrow u_y \oplus u_{xy}$
$I \longleftarrow \{I_1, I_2, I_3, I_4\}$
where each pixel is calculated using the following functional relationship $\mathscr{F}_I$
$I_1 = u_x \wedge X$
$I_2 = Y$
$I_3 = X \oplus u_I$
$I_4 = X \oplus Y \oplus u_i$

We can illustrate the causal diagram for this SCM as follows:



Now, let us see how the SCM evolves over the domain of U.

u_I	u_x	u_y	u_{xy}	X	Y	I_1	I_2	I_3	I_4	I	P(I)
0	0	0	0	0	0	0	0	0	0	0	0.06
0	0	0	1	1	1	0	1	1	0	6	0.06
0	0	1	0	0	1	0	1	0	1	5	0.04
0	0	1	1	1	0	0	0	1	1	3	0.04
0	1	0	0	1	0	1	0	1	1	11	0.09
0	1	0	1	0	1	0	1	0	1	5	0.09
0	1	1	0	1	1	1	1	1	0	14	0.06
0	1	1	1	0	0	0	0	0	0	0	0.06
1	0	0	0	0	0	0	0	1	1	3	0.06
1	0	0	1	1	1	0	1	0	1	5	0.06
1	0	1	0	0	1	0	1	1	0	6	0.04
1	0	1	1	1	0	0	0	0	0	0	0.04
1	1	0	0	1	0	1	0	0	0	8	0.09
1	1	0	1	0	1	0	1	1	0	6	0.09
1	1	1	0	1	1	1	1	0	1	13	0.06
1	1	1	1	0	0	0	0	1	1	3	0.06

Now let us try to consider a situation where we are provided an Image where I=5. Can we find the P(U|I)? We can see that there are three corresponding rows which generate the same Image

I. However, we do not know exactly which set of initial conditions yielded the image we want to perform the intervention on.

Furthermore, when we try to perform an intervention such that X=1. Then we can observe that the first set of exogenous variables produces $I' = \{0, 1, 1, 0\}$. However the second highlighted set of exogenous variables produces $I' = \{1, 1, 1, 0\}$. Hence, we know that even the counterfactual images generated will be different depending on the initial U.

We tackle this problem in our SCM by trying instead to predict $P(V|U, I)$. This means that our SCM has access to the observational distribution for I during the training process.

4 Architecture

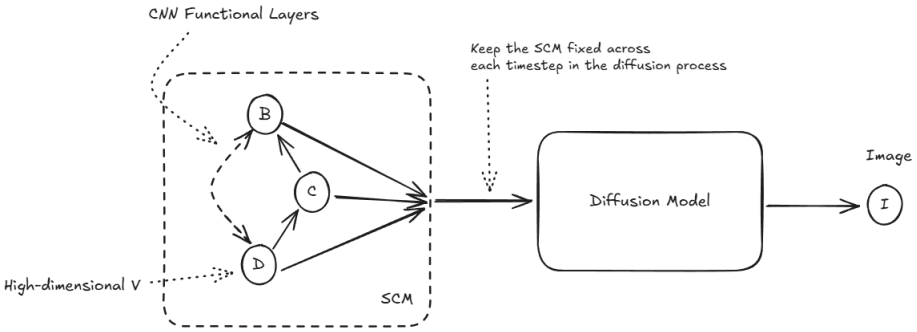


Figure 1: Overall Architecture of our Causal Image Generation Model

We propose a method for effectively learning disentangled representations from observational data. Our architecture is shown 1. We have two parts of the model. A generative NCM which generates high dimension generative factors. And a diffusion model which takes these generative factors as input to predict noise to be removed from the final image.

For generating the counterfactual image, we perform an intervention on our desired variable while keeping the exogenous variables fixed. Then we pass this new V to the diffusion model to generate an intervened image. We can observe this in the diagram 2

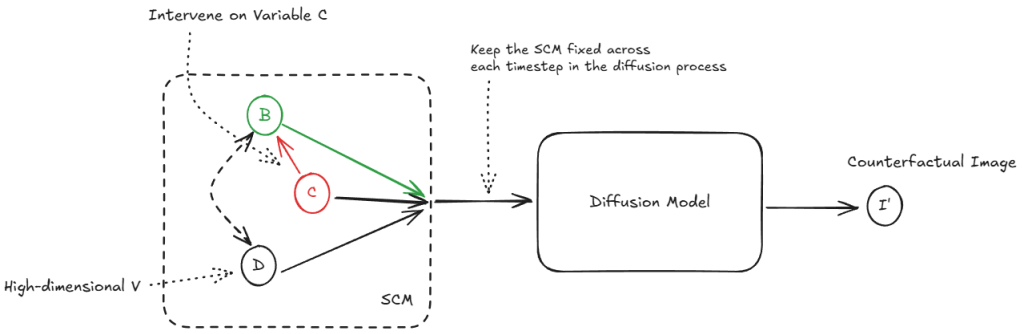


Figure 2: Generating Counterfactual Image through intervention on SCM

4.1 Generative NCM

The generative NCM uses CNN's for representing functional relationships between each variables. The variables are high-dimensional with the intent of preserving as much information as possible to guide the diffusion process adequately.

4.2 Diffusion Model

The diffusion model we use is a standard U-net model that takes conditional labels along with noisy input. The total size of the model is around 100M parameters.

4.3 Performing Interventions in High-dimensional Space

Since each endogenous V in our SCM is in a high dimensional space, we try to learn a bi-directional mapping V' between the actual discrete feature space and the high dimensional space. For this we use an autoencoder-like architecture for every V as shown 3. This is important during interventions, since we cannot directly intervene on the high dimensional V.

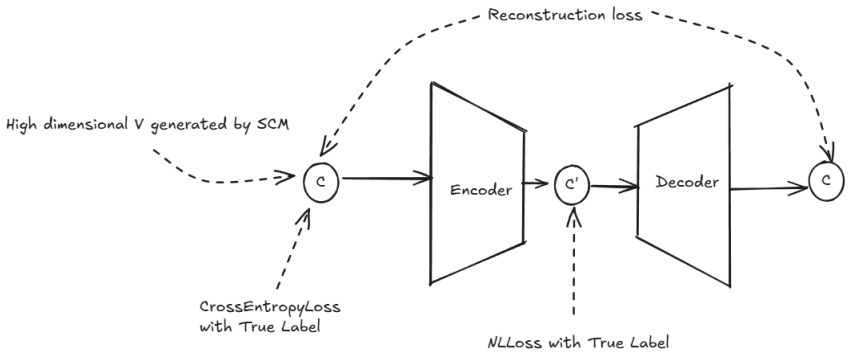


Figure 3: AutoEncoder for mapping V from higher to lower dimension

The implementation of this architecture can be found in the `causal_image_model.py` file here (6).

5 Training

We have trained our model on the MNist Bar dataset for the front-door case. This dataset contains digits(0-9), bar(presence or absence), digit-color(green or red). The entire model including the SCM, V-autoencoders and Diffusion model are trained end to end on each iteration. We train the model on a single A100 GPU for 200 epochs.

On each iteration, we first resolve the values for the SCM and then feed these to the Diffusion model to generate a denoised image. Each node in the SCM takes the true image as input as an exogenous variable U_i . Essentially we are trying to find $P(V|I, U_i)$.

We express our loss as follows:

$$\mathcal{L}_{\text{total}} = \mathbb{E}_{I_0,\epsilon,t,V}\left[||\epsilon - \epsilon_\theta(I_t, t, V)||^2\right] + \frac{1}{N_v}\sum_{v\in V} MSE(v, v_c) + CE(v, y_v) + NLL(v_l, y_v) \quad (1)$$

Our Diffusion loss tries to minimize the MSE between the predicted noise and actual noise in the image at various time steps t(0-1000). Unlike our SCM, it does not gain access to the original image I, but instead obtains a noisy version of it as input. The loss here is same as the noise loss in the DDPM paper (1).

Our SCM tries to minimize the cross-entropy between the true labels and the high-dimensional V generated by our SCM. Furthermore we can see in 3 how, we also have a reconstruction loss and negative log likelihood loss to align the values in our autoencoder for each variable V. You can find the entire training process here (6)

6 Results and Analysis

We have only tested our model for the Mnist task, however, our architecture is not specific to the Mnist example.

6.1 Intervention on color, front-door MNIST example

Let us consider the Bar Mnist example with the following causal diagram 4.

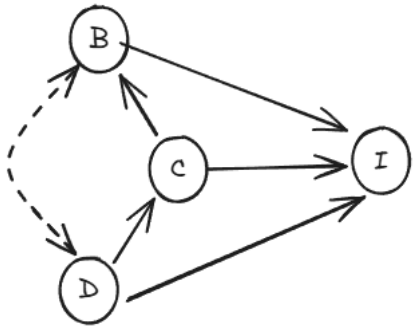


Figure 4: Front-door case for Bar Mnist

We first test the reconstruction quality of a sample for our model. Our model essentially tries to learn and encode information about the image it has seen, similarly to a human does. We can see the real images in the dataset 5

Figure 5: Original Image from the dataset

As we can observe 6 our model is reconstructing the image quite accurately, indicating that it has learned the internal representation well.

Figure 6: Our model's reconstruction of the image

Figure 7: After intervening on the color do(C=1)

Now, in the causal diagram given above 4 we know that if we perform an intervention on the color, then the digit should remain unaffected. We perform the intervention $do(C = 1)$ on the SCM and generate the counterfactual image I'. As we can see from 7, we have successfully intervened only on the color, changing it from red to green. However, the digit remains unchanged, and the style remains consistent as well. The code for this example is here (6) under the bar minst notebook.

6.2 Intervention on Digit, front-door MNIST example

Now, let us perform an intervention on the digit. Naturally, we would want just the digit to change and not the bar. However the color is something that might change since it is affected by the digit. We can observe the results for this below.

Figure 8: Original image from the dataset

Figure 9: Our model's reconstruction of the image

As seen from the images, we are able to construct an accurate reconstruction and intervene on just the digit while keeping the bar the same. Surprisingly the color also remains the same.

Figure 10: After performing intervention do(Digit=8)

7 Conclusion

We propose a method of performing counterfactual image editing using Diffusion models in a causal setting. Diffusion models produce higher quality images compared to their VAE counterparts. Furthermore, we explore a new method of learning disentangled representations of endogenous variables in high dimensional setting. This is crucial to preserve crucial information through the SCM. We also show through preliminary experiments on the mnist example that we are able to preserve styles and other latent variables while performing an intervention.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2022. [Online]. Available: https://arxiv.org/abs/2112.10752

[3] K. Xia, K. Lee, E. Bengio, and E. Bareinboim, "The causal-neural connection: Expressiveness, learnability, and inference," in *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[4] K. Xia, Y. Pan, and E. Bareinboim, "Neural causal models for counterfactual identification and estimation," in *The 11th International Conference on Learning Representations*, Feb 2023.

[5] Y. Pan and E. Bareinboim, "Counterfactual image editing," Causal Artificial Intelligence Lab, Columbia University, Tech. Rep. R-103, December 2023.

[6] A. Sardesai, "Causal diffusion model," https://github.com/athmihir/Causal-Diffusion-Model, 2025.