Diffusion Causal Models for Counterfactual Estimation

Author names withheld

Editor: Under Review for CLeaR 2022

Appendix A. Theory for Training Diffusion Models

We now review with more detailed the formulation of Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020). In DDPM, samples are generated by reversing a diffusion process with a neural network from a Gaussian prior distribution. We begin by defining our data distribution $x_0 \sim p(\mathbf{x}_0)$ and a Markovian noising process which gradually adds noise to the data to produce noised samples $\mathbf{x}_t$ up to $\mathbf{x}_T$. In particular, each step of the noising process adds Gaussian noise according to some variance schedule given by β_t:

$$p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)=\mathcal{N}\left(\mathbf{x}_t ; \sqrt{1-\beta_t}\, \mathbf{x}_{t-1}, \beta_t \mathrm{I}\right) \tag{1}$$

In addition, it's possible to sample $\mathbf{x}_t$ directly from $\mathbf{x}_0$ without repeatedly sample from $\mathbf{x}_t \sim p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$. Instead, $p\left(\mathbf{x}_t \mid \mathbf{x}_0\right)$ can be expressed as a Gaussian distribution by defining a variance of the noise for an arbitrary timestep $\alpha_t := \prod_{j=0}^{t}\left(1-\beta_j\right)$. We, therefore, proceed to define

$$\begin{aligned} p\left(\mathbf{x}_t \mid \mathbf{x}_0\right) &= \mathcal{N}(\mathbf{x}_t ; \sqrt{\alpha}_t \mathbf{x}_0,(1-\alpha_t) \mathrm{I}) && (2)\\ &= \sqrt{\alpha_t} \mathbf{x}_0+\epsilon \sqrt{1-\alpha_t},\ \epsilon \sim \mathcal{N}(0, \mathrm{I}) && (3) \end{aligned}$$

However, we are interested in a generative process which consists in performing a reverse diffusion, going from noise $\mathbf{x}_T$ to data $\mathbf{x}_0$. As such, the model trained with parameters θ should correspond to conditional distribution $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$.

Using Bayes theorem, one finds that the posterior $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is also a Gaussian with mean $\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0)$ and variance $\tilde{\beta}_t$ defined as follows:

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\alpha_{t-1}-\alpha_t}}{1-\alpha_t}\mathbf{x}_0+\frac{\alpha_t(1-\alpha_{t-1})}{\alpha_{t-1}(1-\alpha_t)}\mathbf{x}_t \qquad \tilde{\beta}_t := \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t \tag{4}$$

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathrm{I}) \tag{5}$$

Training $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$ such that $p(\mathbf{x}_0)$ learns the true data distribution, the following variational lower-bound L_{vlb} for $p_\theta(\mathbf{x}_0)$ can be optimized:

$$L_{\text{vlb}} := -\log p_\theta(\mathbf{x}_0|x_1) + \sum_{t=2}^{T} D_{KL}(p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \,||\, p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \tag{6}$$

Ho et al. (2020) considered a variational approximation of the Eq. 5 for training $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$ efficiently. Instead of directly parameterize $\mu_\theta(\mathbf{x}_t, t)$ as a neural network, a model $\epsilon_\theta(\mathbf{x}_t, t)$ is trained to predict ϵ from Equation 3. This simplified objective is defined as follows:

$$L_{\text{simple}} := \mathbb{E}_{t, \mathbf{x}_0 \sim p_{\text{data}}, \epsilon \sim \mathcal{N}(0, \mathrm{I})}\left[(1-\alpha_t)\left\|\boldsymbol{\epsilon}_\theta(\sqrt{\alpha_t}\mathbf{x}_0+\sqrt{1-\alpha_t}\epsilon, t)-\epsilon\right\|_2^2\right] \tag{7}$$

Appendix B. Pearl's Causal Hierarchy

Bareinboim et al. (2020) use *Pearl's Causal Hierarchy* (PCH) nonmenclature after Pearl's seminal work on causality which is well illustrated in Pearl and Mackenzie (2018) as the *Ladder of Causation*. PCH states that structural causal models should be able to sample from a collection of three distributions (Peters et al. (2017), Ch. 6) which are related to cognitive capabilities:

1. The *observational* ("seeing") distribution $p_{\mathfrak{G}}(\mathbf{x}^{(k)})$.

2. The do-calculus (Pearl, 2009) formalizes sampling from the *interventional* ("doing") distribution $p_{\mathfrak{G}}(\mathbf{x}^{(k)} \mid do(\mathbf{x}^{(j)} = x^{(j)}))$. The $do()$ operator means an intervention on a specific variable is propagated only through it's descendants in the SCM $\mathfrak{G}$. The causal structure forces that only the descendants of the variable intervened upon will be modified by a given action.

3. Sampling from a *counterfactual* ("imagining") distribution $p_{\mathfrak{G}}(\mathbf{x}^{(k)} \mid do(\mathbf{x}^{(j)} = x^{(j)}); x^{(k)})$ involves applying an intervention $do(\mathbf{x}^{(j)} = x^{(j)})$ on an given instance $\mathbf{x}^{(k)}$. Contrary to the factual observation, a counterfactual corresponds to a hypothetical scenario.

Appendix C. DDIM sampling procedure

A variation of the DDPM (Ho et al., 2020) sampling procedure is done with Denoising Diffusion Implicit Models (DDIM, Song et al. (2021)). DDIM formulates an alternative non-Markovian noising process that allows a deterministic mapping between latents to images. The deterministic mapping means that the noisy term in Eq. **??** is no longer necessary for sampling. This sampling approach has the same forward marginals as DDPM, therefore, it can be trained in the same manner. This approach was used for sampling throughout the paper as explained in Sec. **??**.

Alg. 1 describes DDIM's sampling procedure from $\mathbf{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$ (exogenous noise distribution) to $\mathbf{x}_0$ (data distribution) deterministic procedure. This formulation has two main advantages: (i) it allows a near-invertible mapping between $\mathbf{x}_T$ and $\mathbf{x}_0$ as shown in Alg. 2; and (ii) it allows efficient sampling with fewer iterations even when trained with the same diffusion discretization. This is done by choosing different undersampling t in the $[0, T]$ interval.

Algorithm 1 Sampling with DDIM - Image Generation

Models: trained diffusion model $\boldsymbol{\epsilon}_\theta$.
Input : $x_T \sim \mathcal{N}(0, \mathrm{I})$
Output: x_0 - Image
for $t \leftarrow T$ **to** 0 **do**
  $x_{t-1} \leftarrow \sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1-\alpha_t}\, \boldsymbol{\epsilon}_\theta(x_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{\alpha_{t-1}}\, \boldsymbol{\epsilon}_\theta(x_t, t)$
end

Appendix D. Implementation Details

For each dataset, we train two models that are trained separately: (i) $\boldsymbol{\epsilon}_\theta$ is implemented as an encoder-decoder architecture with skip-connections, *i.e.* a Unet-like network (Ronneberger et al.,

Algorithm 2 Reverse-Sampling with DDIM - Inferring the Noisy Latent

Models: trained diffusion model $\boldsymbol{\epsilon}_\theta$.
Input : x_0 - Image
Output: x_T - Latent Space
for $t \leftarrow T$ **to** 0 **do**
$x_{t+1} \leftarrow \sqrt{\alpha_{t+1}} \left(\frac{x_t - \sqrt{1-\alpha_t}\, \boldsymbol{\epsilon}_\theta(x_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{\alpha_{t+1}}\, \boldsymbol{\epsilon}_\theta(x_t, t)$
end

2015). (ii) A (Anti-causal) classifier that uses the encoder of $\boldsymbol{\epsilon}_\theta$ with a pooling layer followed by a linear classifier. All models are time conditioned. Time, which is a scalar, is embedded using the transformer's sinusoidal position embedding (Vaswani et al., 2017). The embedding is incorporated into the convolutional models with an Adaptive Group Normalization layer into each residual block (Nichol and Dhariwal, 2021). Our architectures and training procedure follow Dhariwal and Nichol (2021). They performed an extensive ablation study of important components from DDPM (Ho et al., 2020) and improved overall image quality and log-likelihoods on many image benchmarks. We use the same hyperparameters as Dhariwal and Nichol (2021) for the ImageNet and define ours for MNIST. The specific hyperparameters for diffusion and classification models follow Tab. 1. We train all of our models using Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We train in 16-bit precision using loss-scaling, but maintain 32-bit weights, EMA, and optimizer state. We use an EMA rate of 0.9999 for all experiments.

We use DDIM sampling for all experiments with 1000 timesteps. The same noise schedule is used for training. Even though DDIM allows faster sampling, we found that it does not work well for counterfactuals.

dataset	ImageNet 256	ImageNet 256	MNIST	MNIST
model	diffusion	classifier	diffusion	classifier
Diffusion steps	1000	1000	1000	1000
Model size	554M	54M	2M	500K
Channels	256	128	64	32
Depth	2	2	1	1
Channels multiple	1,1,2,2,4,4	1,1,2,2,4,4	1,2,4	1,2,4,4
Attention resolution	32,16,8	32,16,8	-	-
Batch size	256	256	256	256
Iterations	$\approx$ 2M	$\approx$ 500K	30K	3K
Learning Rate	1e-4	3e-4	1e-4	1e-4

Table 1: Hyperparameters for models.

Appendix E. Sampling from The Interventional Distribution

In this section, we make sure that our method complies with the second level of *Pearl's Causal Hierarchy* (details in Appendix B). Diff-SCM can be used for efficiently sampling from the interventional distributions $p_{\mathfrak{G}_{\text{image}}}(\mathbf{x}^{(1)} \mid do(\mathbf{x}^{(2)} = x^{(2)}))$. Sampling from the interventional distribution can be done by using the second part ("Generation with Intervention") of Alg. **??** but sampling $u^{(k)}$ from a Gaussian prior, instead of inferring the latent space (using "Abduction of Exogenous

Noise"). This formulation is identical to Dhariwal and Nichol (2021) with guided DDIM (Song et al., 2021) (details in appendix C). Dhariwal and Nichol (2021) achieves state-of-the-art image quality results in generation while providing faster sampling than DDPM. Since its capabilities in image synthesis compared to other generative models are shown in Dhariwal and Nichol (2021), we restrict ourselves to present qualitative results on ImageNet 256x256.

Experimental Setup. Our experiment, depicted in Fig. 1, consists in sampling a single latent space $u^{(1)}$ from a Gaussian distribution and generating samples for different classes. Since all images are generated from the same latent, this allows visualization of the effect of the classifier guidance for different classes. This setup differs from experiments in Dhariwal and Nichol (2021), where each image presented was a different sample $u^{(1)} \sim \mathbf{u}^{(1)}$. Here, by sampling $\mathbf{u}^{(1)}$ only once, we isolate the contribution of the causal mechanism from the sampling of the exogenous noise $\mathbf{u}^{(1)}$. We use the scale hyperparameter $s = 5$ for these experiments.

Figure 1: Sampling ImageNet images from the interventional distribution. All images originate from the same initial noise $u^{(k)}$ but different interventions are applied at inference time.

Appendix F. IM1 and IM2

Van Looveren and Klaise (2021) propose IM1 and IM2 for measuring the realism and closeness to the data manifold. These metrics are based on the reconstruction losses of auto-encoders trained on specific classes:

$$\text{IM1}(x_{\text{CF}}^{(1)}, x_{\text{F}}^{(2)}, x_{\text{CF}}^{(2)}) = \frac{\left\| x_{\text{CF}}^{(1)} - \text{AE}_{x_{\text{CF}}^{(2)}}(x_{\text{CF}}^{(1)}) \right\|_2^2}{\left\| x_{\text{CF}}^{(1)} - \text{AE}_{x_{\text{F}}^{(2)}}(x_{\text{CF}}^{(1)}) \right\|_2^2 + \epsilon} \tag{8}$$

$$\text{IM2}(x_{\text{CF}}^{(1)}, x_{\text{CF}}^{(2)}) = \frac{\left\| \text{AE}_{x_{\text{F}}^{(2)}}(x_{\text{CF}}^{(1)}) - \text{AE}(x_{\text{CF}}^{(1)}) \right\|_2^2}{\left\| x_{\text{CF}}^{(1)} \right\|_1 + \epsilon} \tag{9}$$

where $\text{AE}_{x^{(2)}}$ denotes an autoencoder trained only on instances from class $x^{(2)}$, and AE is an autoencoder trained on data from all classes. IM1 is the ratio of the reconstruction loss of an autoencoder trained on the counterfactual class divided by the loss of an autoencoder trained on all classes. IM2 is the normalized difference between the reconstruction of the CF under an autoencoder trained on the counterfactual class, and one trained on all classes.

Appendix G. More MNIST Counterfactuals

Here, we show in Fig. 2 that we can generate counterfactuals of all MNIST classes, given factual image. We use the scale hyperparameter $s = 0.7$ for these experiments.

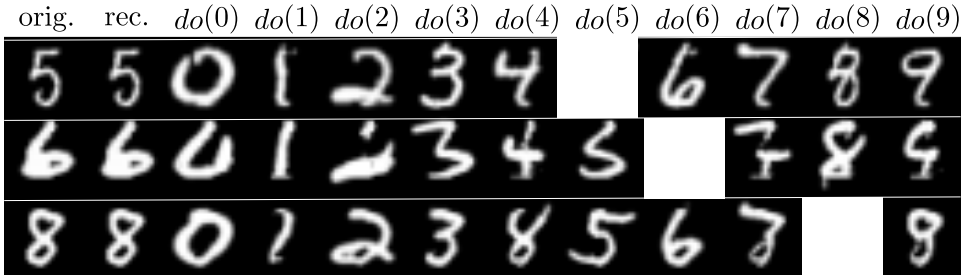

Figure 2: MNIST counterfactuals. From the left to right, one can observe the original image (*orig.*), the reconstruction (*rec.*, which entails in running the algorithm **??** without the anti-causal predictor) and the resulting counterfactuals for each of the digit classes in the dataset.

References

E. Bareinboim, J. Correa, D. Ibeling, and T Icard. On Pearl's Hierarchy and the Foundations of Causal Inference. Technical report, Causal AI Lab, Columbia University, 2020.

Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances on Neural Information Processing Systems*, 2020.

Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. *arxiv pre-print*, 2 2021.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009.

Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic books, 2018.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference*. MIT Press, 2017.

O Ronneberger, P.Fischer, and T Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proc. of Medical Image Computing and Computer-Assisted Intervention*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *Proc. of International Conference on Learning Representations*, 2021.

Arnaud Van Looveren and Janis Klaise. Interpretable Counterfactual Explanations Guided by Prototypes. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, volume 1907.02584, 7 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.