

MENG INDIVIDUAL PROJECT

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Counterfactual Content-Based Image Retrieval via Variational Twin Networks and VQ-VAEs

Author:
Matteo Bilardi

Supervisors:
Athanasios Vrontzos,
Bernhard Kainz

Second Marker:
Mark van der Wilk

Monday 20th June, 2022

Abstract

Counterfactual reasoning about an image allows one to hypothesise how that image would have looked like, had something about the world in which it was produced been different: ‘How would have my passport photo looked like, had I had blue eyes?’ or ‘How would this MRI scan have looked like, had the patient in it been 10 years older?’. Methods for counterfactual image generation enable sampling from the distribution over such counterfactual images. This work proposes, in addition to a generative method, a framework to retrieve the images which have the highest probability of being valid counterfactuals from a pre-existing dataset. Then, the two previous questions become ‘Which photo, within a dataset, would I be most likely to have on my passport, had I had blue eyes?’ and ‘Which MRI scan, within a medical database, would be the most likely to be produced for the patient in this MRI scan, had they been 10 years older?’.

To tackle image retrieval queries of a similar form, the variational twin network framework is first proposed. It is a generative model that successfully combines the variational approach of DeepSCM [1] with a deep twin network architecture [2] in order to produce credible and diverse factual-counterfactual image pairs for a synthetic dataset based on Morpho-MNIST [3]. Then, *ELBO rank* and *sample-similarity rank*, two different methods that rely on a variational twin network for performing counterfactual image retrieval, are proposed and evaluated: on more than 98% of the submitted test queries both methods correctly identified the ground truth counterfactual within a dataset. Additionally, the effects of a variety of training methods for VQ-VAEs are investigated in the context of standard image retrieval, yielding a more than doubled mAP@All score compared to a vanilla VQ-VAE on a popular image retrieval task.

Acknowledgements

I am grateful to my supervisors, Athanasios Vrontzos and Bernhard Kainz, for their invaluable help and for our discussions during the course of the project. I would also like to thank Hadrien Reynaud for answering my questions about certain technical details of their work [4].

I am thankful to my friends and to my family, both immediate and extended, for their affection and support.

Finally, I would like to thank Mark van der Wilk for asking me, at the time of the interim report meeting, a simple but paramount question: ‘How are you going to train it?’.

Contents

1	Introduction	4
1.1	Motivation	4
1.2	Contributions and report outline	5
2	Background	6
2.1	Deep Learning	6
2.1.1	FFNN	6
2.1.2	CNN	6
2.2	Causality	7
2.2.1	Structural causal model	7
2.2.2	The Ladder of Causation	8
2.2.3	Computing counterfactuals	10
2.2.4	Identifiability	11
2.2.5	Deep twin network	11
2.3	Variational generative models	12
2.3.1	Generative latent variable models	12
2.3.2	Variational inference	13
2.3.3	VAE	15
2.3.4	VQ-VAE	16
2.4	CBIR	17
2.4.1	Feature engineering	17
2.4.2	Feature embedding	18
2.4.3	Vector quantisation for CBIR	18
2.4.4	Feature learning	19
2.4.5	Evaluation Metrics	20
2.5	Ethical considerations	22
3	Counterfactual image generation	23
3.1	Definition	23
3.2	Image generation task	24
3.2.1	Dataset	24
3.2.2	Causal graph	25
3.2.3	Task	26
3.3	Method: variational twin network	26
3.3.1	DeepSCM	26
3.3.2	DTN	27

3.3.3	VTN	29
3.4	Experiments	32
3.4.1	Models	32
3.4.2	Methodology	33
3.4.3	Results	35
4	Counterfactual content-based image retrieval	37
4.1	CCBIR	37
4.1.1	Definition	37
4.1.2	Method 1: ELBO rank	38
4.1.3	Method 2: Sample-similarity rank	39
4.2	Experiments	40
4.2.1	Dataset composition	40
4.2.2	Models	41
4.2.3	Methodology	41
4.2.4	Results	42
5	VQ-VAE training for image retrieval	44
5.1	Retrieval task	44
5.1.1	Dataset	44
5.1.2	Dimensionality constraints	44
5.2	Training techniques	45
5.2.1	Normalised codewords	45
5.2.2	PQ-VAE	45
5.2.3	Noisy codeword assignment	46
5.2.4	Random restarts	49
5.3	Experiments	50
5.3.1	Models	50
5.3.2	Methodology	50
5.3.3	Comparison	51
5.3.4	Ablation Study	52
5.3.5	CCBIR	55
6	Conclusion	57
6.1	Main Results	57
6.2	Future work	58

+

Chapter 1

Introduction

1.1 Motivation

The ability to formulate and answer causal queries such as ‘What happens to Y if I do X ?’ (interventional) and ‘Would Y have been different, had I done X ?’ (counterfactual) is fundamental to human reasoning and is widely exercised in a number of fields, such as medical practice, economics, epidemiology, and artificial intelligence [5]. In recent years, the application of causality to generative image modelling has been a growing direction of research [6, 1, 7, 8, 4], thus permitting to imagine how an image would have looked like under a forceful change to a background variable. Examples include, how a portrait picture of a person with blond hair would appear if the ‘hair colour’ variable was forcefully set to be black [8], or how a brain MRI scan of a patient would change if their age was altered [1].

Orthogonally to the field of causality, content-based image retrieval (CBIR) [9] pipelines enable their user to submit a given query image, on the basis of which, relevant images from an existing dataset are returned. A defining property of many CBIR systems is that they rank returned images according to some notion of similarity with the query image. While this has clear utility, it also imposes a significant limitation on the kind of queries that are supported by the system: one must already be in possession of an image that is similar to the one they wish to find. However, one may instead only have access to a different image for which a modification is imagined on the basis of some background variables, e.g. ‘How would this daylight picture look like if it had been taken at night?’, and then wish to retrieve real images – as opposed to artificially generated ones – that have in fact been taken at night *and* look (as much as possible) like valid counterfactual images for the daylight picture. A more immediately impactful application could be that of a doctor imagining how a CT scan of one of their patients would have looked like under a different treatment regiment, and then retrieve the scans of other real patients who both received the alternative treatment and are likely counterfactuals for the patient, so that their cases may be compared to aid in the diagnostic process.

In the light of the above, the addition of causality to a CBIR framework could extend the usefulness of the system, by making it capable of assessing the relevance of a dataset image to a counterfactual of the query image, while simultaneously ‘grounding’ the retrieved

counterfactual candidates to the real world, as a dataset image must exist to be retrieved. Thus, the objective of this project was to design and implement a system for counterfactual content-based image retrieval.

1.2 Contributions and report outline

In chapter 2, the background literature is surveyed, covering the basics of deep learning, causality, variational generative models, and content based-image retrieval; section 2.5 documents the ethical considerations for the project.

In chapter 3, the framework of variational twin networks is proposed in order to approximate via a deep learning approach the generative process described by a structural causal model and to permit the generation of new counterfactual samples. The presented neural architecture leverages the variational inference strategy of a DeepSCM [1] for the principled abduction of exogenous variables during training, but becomes a deep twin network [2] after training, which enables the sampling of counterfactuals via the twin network [10] method.

In chapter 4, the task of counterfactual content-based image retrieval is formulated as the act of ranking images in a gallery dataset by their probability of being a valid counterfactual for a user-submitted factual image, given the respective treatments and shared covariates. Two different strategies are introduced to tackle the task, both of which make use of a variational twin network. The first method ranks each image in the dataset by decreasing order of a lower bound to the probability that the joint factual-counterfactual image pair was generated by the model, i.e. the ELBO. The second method follows the approach of sampling a single counterfactual image and ranking each gallery image by decreasing order of similarity with the counterfactual sample, where the measure of visual (dis)similarity is the distance between the encodings of two images within the continuous latent space of a VQ-VAE.

In chapter 5, a number of existing methods for the improvement of the training process for VQ-VAEs are studied with the aim of establishing their effect on retrieval performance when using the latent vectors produced by the resulting VQ-VAEs as feature embeddings for both standard and counterfactual image retrieval. A generalisation of the noise injection method of [11] is proposed. The semantics of the minimum usage hyperparameter for the random restarts' technique [12] are clarified in order to better guide its selection when random restarts and product quantisation [13] are used in conjunction.

At the end of chapters 3, 4, and 5, an 'Experiments' section presents the evaluation results relevant to each chapter. The main results of the project are summarised in chapter 6, which concludes the report by presenting possible avenues for future work.

Chapter 2

Background

2.1 Deep Learning

2.1.1 FFNN

The artificial neural network (NN) [14] is a powerful and prominent mathematical model that aims to approximate a real-valued vector function $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ by determining the optimal parameters (or weights) θ of a model f_θ . The most common type of NN is the *feed-forward neural network* (FFNN), which is represented as the composition of multiple functions, called layers, so that $f_\theta = f_{(D)} \circ \dots \circ f_{(1)}$, where D is the *depth* of the network, the output dimension of $f_{(i)}$ is the *width* of layer i , and each layer $f_{(i)}$ is an affine transformation of the output \mathbf{z} of the previous layer $f_{(i-1)}$ followed by an element-wise non-linear activation function $\phi_{(i)}$, i.e. $f_{(i)}(\mathbf{z}) = \phi_{(i)}(\mathbf{W}_{(i)}\mathbf{z} + \mathbf{b}_{(i)})$. Thus, the learnable weights of the networks are $\theta = \{\mathbf{W}_{(i)}, \mathbf{b}_{(i)}\}_{i=1}^D$, while the activation functions, the depth of the network, and the widths of hidden layers (all but the output and input ones) are hyperparameters. A scalar value for a particular layer, such as the j -th entry of the output of the i -th layer, $f_{(i)}(\mathbf{z})_j$, is called a neuron, and is conceptually reminiscent of the biological neuron as its output signal is a function of the output signals of other input neurons; in fact, in FFNNs, all the neurons in a given layer are typically fully-connected to all the neurons in the previous layer.

For fixed network hyperparameters, training the network is an attempt to learn an optimal θ so as to minimise a loss criterion $L : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$ between the output of the network and the true value of f on a representative training dataset. The minimisation objective is achieved via the process of gradient descent, which gradually shifts θ in the direction opposite to the gradient of L so that at each optimisation step $\theta_{t+1} := \theta_t - \alpha \nabla_\theta L(f_\theta(\mathbf{x}), \mathbf{y})$, where α is a learning rate. After a forward pass computing $f_\theta(\mathbf{x})$, $\nabla_\theta L(f_\theta(\mathbf{x}), \mathbf{y})$ is determined efficiently by repeated application of the chain rule starting from the output layer and computing intermediate partial derivatives up to the input layer through a process known as back-propagation.

2.1.2 CNN

The *convolutional neural network* (CNN) [14] is a type of NN that has been immensely successful in modelling relationships on high-dimensional data. CNNs are defined by their

use of convolutional layers and often include pooling layers, as well as traditional fully-connected layers of FFNN. A convolutional layer replaces the matrix multiplication (Wz) of a fully connected layer with a convolution operation ($I * K$) in which the input data I from the previous layer is convolved with a lower dimensional kernel K whose weights are learnt during training and shared by all neurons; the output of the operation is called a *feature map*. Practically, this implies that each neuron is only connected to a local subset of the neurons in the previous layer and is only influenced by a portion, known as the *receptive field*, of the original network input. As opposed to a FFNN, the structure of the input and outputs of each layer need not be flattened to a vector but is, in general, a tensor. Additionally, pooling layers are frequently used in CNNs to quickly reduce the dimensions of the input feature map by computing an aggregate operation, such as *max*, *average*, or *min*, over a sliding window of the feature map.

In the light of the above, important advantages provided by CNNs over FFNN have been identified in the literature [14], including a substantial reduction in the number of parameters that have to be learnt, thanks to the weight sharing property of the kernels; a reduction in computation time, due to the lower number of operations needed to perform a convolution over a smaller kernel rather than multiplying a matrix with the image; equivariance to translation, so that a translation of the input will be reflected in the output feature map, which is particularly useful for video processing as movement of an object in the video over time corresponds to its movement in the output feature maps.

2.2 Causality

2.2.1 Structural causal model

Structural causal models (SCM) [15, 16] or *causal theories* [17] are a formalism that models the generating process of certain variables in an environment. They are particularly useful for communicating clearly the assumptions that have been made about the generating process and for systematically guiding the *inference* of variables of interest, i.e. the prediction of their values, or of their probability distributions, given those of other variables [14].

An SCM \mathcal{M} is defined as a tuple $\langle U, P(U), V, F \rangle$, where U is a set of mutually independent, unobserved variables that are exogenous to the model (i.e. they are determined by outside forces), have joint distribution given by $P(U) = \prod_{i=1}^n P(U_i)$, and model the noise affecting the determination of V . V is a set of observed variables that are endogenous, i.e. whose values are determined on the basis of other variables $U \cup V$ in the model. Specifically, the value of each observed variable V_i is dependent, not allowing for cycles, on a set of variables $Pa_i \subseteq V$ (called *parents*) and the noise variables U_i according to a function $f_i \in F$ (called *mechanism*) so that $V_i := f_i(Pa_i, U_i)$. Any SCM \mathcal{M} thus defined induces

1. a directed acyclic graph (DAG) $G_{\mathcal{M}}$, called the *causal graph* induced by \mathcal{M} , where each variable is a node and such that there is a directed edge between any two nodes $X \rightarrow Y$ if $X \in Pa_Y$, i.e. if X is a direct cause of Y ;
2. a joint distribution $P(V)$ over the observed variables that, under the Markov assumption ($G_{\mathcal{M}}$ is a DAG and the unobserved variables are mutually independent), factorizes as $P(V) = \prod_i P(V_i | Pa_{V_i})$.

When the mechanisms F or the distribution over exogenous variables $P(U)$ are omitted but the DAG G_m is available, such *probabilistic graphical models* (PGMs) are more commonly referred to as *Bayesian networks* [18].

2.2.2 The Ladder of Causation

SCMs permit to answer a variety of increasingly insightful and cognitively more demanding questions about the world they describe; such questions have been classified by Pearl [15] into associational, interventional, and counterfactual, from lower levels to higher ones. Pictorially [19] and metaphorically, each level can be thought of as a rung on a ladder (the Ladder of Causation), where each rung subsumes and extends the capabilities of the one beneath.

Seeing

The lowest rung of the Ladder of Causation is occupied by observational queries, which are interested in determining how the observation of one variable changes the probability of another; for example, given that a person has bought sunscreen, what is the probability that they will also buy a swimsuit? Answering such queries involves observing the environment, witnessing the values of the variables of interest, and modelling how they are associated or correlated with each other, e.g. via deep learning techniques.

In the SCM framework, observational queries such as $P(a|b)$ can be evaluated systematically for a model \mathcal{M} by noting that a joint distribution $P(Y)$ over any subset of observed variables $Y \subseteq V$ is induced by \mathcal{M} , which can in turn be used to calculate any conditional or marginal quantity of interest [16]. For discrete U and V ,

$$P(Y = y) = \sum_{\{u|Y(u)=y\}} P(u) \quad (2.1)$$

That is, the probability of observing $Y = y$ is the probability of being in any state of the world u (also known as unit or situation) in which the evaluation of the mechanisms in F assigns to Y the value y , denoted as $Y(u) = y$. Operationally, Equation 2.1 is justified because one can obtain i.i.d. samples $Y_1, \dots, Y_n \sim P(Y)$ by first sampling noise variables $U_1, \dots, U_n \sim P(U)$ and then observing the values of Y produced by evaluating F for each noise sample [16].

Doing

On the second rung of the Ladder, interventional queries seek to predict what will happen to a variable if an action is taken to modify the state of the world; for example, will a patient with asthma experience fewer attacks if they adopt a new diet? Crucially, this differs from simply observing that the patient experienced fewer attacks when they adopted the diet. In fact, a third event, such as the patient moving to a different country with less air pollution, might have been the cause behind both the change in diet and the drop in frequency of asthma attacks, acting as a *confounder* of the causal effect of the diet on asthma attacks. To study true causal effects, it is therefore essential that the intervention only modifies the intervened upon variable, without affecting the mechanisms of other variables. The *do*-operator was introduced by Pearl [15] to indicate precisely such a manipulation, as $do(X := x)$ denotes the intervention that forcibly sets only the variables X to take values x regardless of the

natural mechanism for X . Hence, $P(Y = y|do(X := x))$ is the probability that $Y = y$ after the intervention $do(X := x)$.

In the context of an SCM \mathcal{M} , the action $do(X := x)$ is captured by a *submodel* \mathcal{M}_x identical to \mathcal{M} other than for its updated set F_x of mechanisms. In particular, F_x is obtained by replacing only the mechanism for each variable $V_i \in X$ with their corresponding constant assignment: $F_x = \{f_i|V_i \notin X\} \cup \{f_i(Pa_i, U_i) := x_i|V_i \in X\}$ [16]. The induced graph $G_{\mathcal{M}_x}$ reflects such change as no incoming edges into X will be present, which has been separated from its causal parents. The utility of the submodel thus defined stems from the fact that the *post-interventional* distribution (after $do(X := x)$) of any $Y \subseteq V$ in model \mathcal{M} is identical to its distribution in \mathcal{M}_x , i.e.

$$P_{\mathcal{M}}(Y|do(X := x)) \triangleq P_{\mathcal{M}}(Y_x) = P_{\mathcal{M}_x}(Y) \quad (2.2)$$

where the values that Y takes under model \mathcal{M}_x in situation u is called the *potential response* of Y to the intervention and is denoted as $Y_x(u)$, i.e. $Y_x(u) = Y_{\mathcal{M}_x}(u)$ [16].

In the light of the above, an SCM \mathcal{M} can be used to answer systematically interventional queries such as $P(Y = y|do(X := x))$ (denoted equivalently as $P(Y_x = y)$) because, by Equation 2.1 and Equation 2.2,

$$P(Y|do(X := x)) \triangleq P(Y_x) = \sum_{\{u|Y_x(u)=y\}} P(u) \quad (2.3)$$

That is, the query can be evaluated by first performing an intervention to obtain the submodel \mathcal{M}_x , and then summing the probabilities of all the situations in which the evaluation of the mechanisms F_x assigns y to Y .

Imaging

At the top of the Ladder of Causation, counterfactual queries seek to establish what the state of the world would have been in the present, had different actions been taken in the past, taking into account the present state of the world; for example, would the student have passed the module, had they attended all the lectures? This counterfactual query differs from the interventional query asking whether the student will pass the module if they attend all the lectures because the former updates the probability distribution over the possible states of the world by taking into account that the particular student did in fact fail the module and did not attend all lectures, while the latter ignores these observations.

Counterfactual sentences (also known as *potential outcomes* or simply *counterfactuals*) such as ‘ Y would be y in situation u , had X been x ’ mean that the potential response of Y to the intervention $do(X := x)$ takes on value y , written $Y_x(u) = y$ [15]. It is useful to clarify that the occurrence of a counterfactual sentence in a probabilistic query does not necessarily make the query counterfactual as, for a query to be counterfactual, it needs to be counter-to [2] facts that occur in different worlds, i.e. worlds which have non-identical SCMs. For example $P(Y_x = y)$, $P(Y_x = y, Z_x = z)$, and $P(Y_x = y|Z_x = z)$ are interventional queries, while $P(Y_x = y, X = x')$, $P(Y_x = y, Y_{x'} = y)$, and $P(Y_x = y|Z = z)$ are counterfactual queries.

Once again, the powerful framework of a fully specified SCM enables the evaluation of queries at this rung of the Ladder by noting [16] that it induces a joint distribution over the arbitrary

counterfactuals $Y_x = y, \dots, Z_w = z$, where $Y, Z, \dots, X, W \subseteq V$:

$$P(Y_x = y, \dots, Z_w = z) = \sum_{\{u | Y_x(u)=y \wedge \dots \wedge Z_w(u)=z\}} P(u) \quad (2.4)$$

That is, the query can be evaluated by first producing the appropriate submodel for each counterfactual and then summing the probabilities of all the situations in which the counterfactual statements are simultaneously satisfied in their respective submodel. The distribution is well-defined even if the counterfactuals appear contradictory (such as $Y_x = y$ and $Y = y'$ with $y \neq y'$) as the submodels in which they need to be satisfied are different [20]. In practice, the counterfactual queries of interest are usually less general than Equation 2.4, which supports statements about multiple counterfactual worlds, and instead only involve a counterfactual statement about a single hypothetical world given evidence $E = e$ from the real one. For example, $P(Y_x = y | E = e)$, which simplifies as

$$P(Y_x = y | E = e) = \sum_u P(Y_x = y) P(u | e) \quad (2.5)$$

2.2.3 Computing counterfactuals

Procedurally reflecting Equation 2.5, [15] proposed a three-step process to infer counterfactual quantities of the form $P(Y_x = y | E = e)$ in the context of an SCM \mathcal{M} :

Abduction-action-prediction

1. **Abduction:** Update the prior belief $P(U)$ about the past state of the world u in the light of the current observation e to obtain $P(U | e)$.
2. **Action:** Perform the intervention $do(X := x)$, simulating a minimal change in the history of events, to obtain the model $\mathcal{M}' = \langle U, P(U | e), V, F_x \rangle$.
3. **Prediction:** Imagine what the present would have been under that change by computing $P_{\mathcal{M}'}(Y = y)$.

Twin network

The abduction step in the abduction-action-prediction process described above is particularly resource intensive, as a large amount of memory is potentially required [15, 20] to store $P(U | e)$. This is because, despite the assumption of mutual independence of the noise variables U that makes $P(U)$ efficiently representable, mutual independence is no longer ensured when one conditions on e (e.g. $U_1 \not\perp U_2 | E$ in the causal graph $U_1 \rightarrow E \leftarrow U_2$) and the full joint distribution $P(U | e)$ needs to be described. The method of *twin networks* [10] can address this issue, as it permits the inference of a counterfactual quantity by translating it to an equivalent observational quantity which is computed via standard Bayesian inference on an augmented causal model called twin network, thus bypassing the need to explicitly describe $P(U | e)$.

The twin network \mathcal{M}^{TN} of $\mathcal{M} = \langle U, P(U), V, F \rangle$ is constructed by duplicating each observed *factual* variable $V_i \in V$ to obtain a new corresponding *counterfactual* variable $V_i^* \in V^*$ and by setting the functional form $f_i^* \in F^*$ of its mechanism to be the same as the corresponding

$f_i \in F$, except that the parents of V_i^* are the counterfactual variables (Pa_i^*) corresponding to the factual parents (Pa_i) of V_i , i.e. $V_i^* := f_i^*(Pa_i^*, U_i) \triangleq f_i(Pa_i^*, U_i)$. Thus $m^{TN} \triangleq \langle U, P(U), V \cup V^*, F \cup F^* \rangle$.

Given the twin network for an SCM, any counterfactual query of the type $P(Y_x = y|E = e)$ can be evaluated by first performing the intervention $do(X^* := x)$ on the twin network, and then computing the observational quantity $P(Y^* = y|E = e)$ via Bayesian inference on the network, i.e.

$$\underbrace{P_m(Y_x = y|E = e)}_{\text{counterfactual}} = \underbrace{P_{m^{TN}}(Y_{X^*:=x}^* = y|E = e)}_{\text{interventional}} = \underbrace{P_{m_{X^*:=x}^{TN}}(Y^* = y|E = e)}_{\text{observational}} \quad (2.6)$$

2.2.4 Identifiability

So far in the discussion, complete knowledge of the underlying data generation process captured by an SCM m has been assumed. In the real world, however, it is rarely the case that m is fully specified [21]; typically $P(U)$ as well as the functional form of each $f_i \in F$ are unknown, while the induced causal graph G_m is assumed via background knowledge or otherwise discovered [22]. Consequently, there might be distinct SCMs that satisfy a set of made assumptions and induce the same distributions over observational and interventional quantities, but that differ on particular counterfactual quantities, so that these cannot be determined uniquely. If all possible models consistent with the background knowledge agree on a counterfactual quantity, then that quantity is said to be *identifiable* [23] and can be computed from the available data.

Due to the desirability of identifiability, it is often useful to make further reasonable assumptions on top of the background knowledge if they bring about the identifiability of a quantity of interest. Examples of common assumptions relating to the causal effect of a variable X on Y , with X and Y binary, include the *unconfoundedness* assumption (also known as *ignorability*), that all the confounders of the causal effect $P(y|do(x))$ are observed, or the *monotonicity* assumption, that $P(Y_{X:=0} = 1 \wedge Y_{X:=1} = 0) = 0$ [2].

2.2.5 Deep twin network

The framework of the *deep twin network* [2] applies deep learning to the estimation of counterfactual quantities, leveraging the efficiency of the twin network approach for counterfactual inference. For a counterfactual query in which variables $X \subseteq V$ are being intervened upon and assuming that the causal graph G_m of the model is known, a deep twin network is a neural network whose high level structure mirrors the causal graph $G_{m_{X^*:=x}^{TN}}$ of the twin network. Concretely, $V \cup V^*$ can be partitioned into sets $Caus$ and Obs such that $Caus$ contains all causal descendents of X or X^* and $Obs \triangleq (V \cup V^*) \setminus Caus$; noting $\{X, X^*\} \subseteq Obs$. Then, for each $C \in Caus$, one seeks to learn the functional form $f_C \in F \cup F^*$ of its mechanism $C := f_C(Pa_C, U_C)$ by representing f_C as an architectural component (embedded in the NN) which takes as inputs the outputs of the respective NN components for Pa_C and U_C . Notably, to ease sampling, learning the distribution over U_C is reframed as learning a function g_C (also represented as a NN component) from an easy-to-sample distribution U'_C , such as a standard Gaussian, so that $U_C = g_C(U'_C)$ and U'_C is an input to the network.

The deep twin network thus obtained is a function $dtn : \mathcal{Obs} \times \mathcal{U}'_{Caus} \rightarrow \mathcal{Caus}$ which, after training, enables sampling from an approximation of the conditional distribution over the causal descendants of X and X^* , i.e.

$$u'_{Caus} \sim P(U'_{Caus}) \wedge caus = dtn(obs, u'_{Caus}) \implies caus \sim \tilde{P}_{m_{X^*:=x^*}^{TN}}(Caus|Obs = obs) \quad (2.7)$$

For instance, the implication of Equation 2.7 for the causal model with diagram $(X \rightarrow Y \leftarrow Z)$, is that one can sample from $P(Y_{X:=x^*} = y^* | X = x, Y = y, Z = z)$ by first sampling $u'_y \sim U'_y$, then computing $\langle \tilde{y}, \tilde{y}^* \rangle = dtn(\langle x, x^*, z \rangle, u'_y)$, and finally retaining the predicted counterfactual outcome \tilde{y}^* only if its predicted factual outcome \tilde{y} matches the observed one y [2].

2.3 Variational generative models

2.3.1 Generative latent variable models

Maximum likelihood estimation

In many cases, one wishes to construct a *generative model* [24] with parameters θ that permits the creation of new samples from the distribution $P_\theta(X)$ induced by the model and such that $P_\theta(X)$ best approximates a real-world distribution $P_{data}(X)$ from which the items in a given training set were independently sampled. *Maximum likelihood estimation* (MLE) is a popular method that can be used to find the values of the model's parameters which obtain such approximation. In particular, the optimal parameters θ^* , referred to as *maximum likelihood estimate*, should be chosen so that the expected value taken by the *likelihood function* $\theta \mapsto P_\theta(x)$ for a sample x from $P_{data}(X)$ is maximised:

$$\theta^* = \operatorname{argmax}_{\theta} \mathbb{E}_{P_{data}(X)}[P_\theta(X)] \quad (2.8)$$

Intuitively, this amounts to picking the parameters that maximise, on average, the probability that a datapoint from the training set was generated by the model.

Latent variable models

Directly pursuing optimisation objective 2.8 requires being able to evaluate $P_\theta(X)$ or estimate it by sampling; this may not be feasible, for example, if the distribution is arbitrarily complicated and high dimensional, as is typically the case for images, video, or speech. To ease the estimation of complex distributions, *latent variable models* (LVMs) [25] posit the existence of a *latent space* \mathcal{Z} and of a deterministic mapping $g : \mathcal{Z} \rightarrow \mathcal{X}$ such that, for any sample $x \sim P_{data}(X)$, there exists an unobserved or *latent variable* $z \in \mathcal{Z}$ that satisfies $x = g(z)$ and $P(Z)$ is *tractable*, i.e. easy to evaluate and sample from. Therefore, an LVM implicitly provides a recipe for constructing a generative model [24]:

1. Place a tractable prior $P(Z)$ over the latent variable.
2. Learn a mapping $g_\theta : \mathcal{Z} \rightarrow \mathcal{X}$, known as a *generator*, to approximate g .
3. Sample $z \sim P(Z)$, then apply $g_\theta(z)$ to generate a new sample from $P_\theta(X)$.

Since g could be arbitrarily complex, it can be desirable for g_θ to be a powerful function approximator, such as a neural network. Note also that because g_θ will generally be only an approximation of g and the training data might itself contain noise, an output distribution (e.g. Gaussian) $P_\theta(X|z)$ parameterized by g_θ and continuous in θ is chosen to express the associated uncertainty while permitting optimisation with respect to θ [26].

Evidence intractability

Under such a setup, a joint distribution $P_\theta(X, Z)$ is induced by the LVM. Assuming continuous z and by the law of total probability, the *evidence* or *marginal likelihood* of a single datapoint x under the model can be computed by marginalizing out the latent variable:

$$P_\theta(x) = \int_{z \in \mathcal{Z}} P_\theta(x, z) dz = \int_{z \in \mathcal{Z}} P_\theta(x|z) P(z) dz = \mathbb{E}_{P(Z)}[P_\theta(x|Z)] \quad (2.9)$$

The optimisation objective 2.8 can then be restated as

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \mathbb{E}_{P_{data}(X)}[\mathbb{E}_{P(Z)}[P_\theta(X|Z)]] \quad (2.10)$$

Given a differentiable estimate of $P_\theta(x)$, g_θ could be optimised by gradient ascent towards the maximum likelihood estimate. However, while both $P_\theta(x|z)$ and $P(z)$ in the integrand above can be computed efficiently, the choice to parametrize $P_\theta(X|z)$ with a neural network causes the integral in Equation 2.9 to not have a closed-form solution in general; moreover, accurately approximating the evidence by sampling is computationally intractable when \mathcal{Z} is a high-dimensional space [26]. Therefore, efficient optimisation of the LVM constructed is non-trivial as the evidence is intractable.

2.3.2 Variational inference

The field of *variational inference* (VI) [27, 28] provides an approximate route to optimise the parameters of an LVM despite the intractability of the integral in Equation 2.9. To this end, a tractable optimisation objective is pursued instead of 2.10 and it consists in the maximisation of a lower bound to the logarithm of the evidence; this is descriptively known as the *evidence lower bound* (ELBO).

ELBO

The ELBO can be derived by noticing that, for a datapoint x_i and any distribution $Q_i(Z)$ (called *variational distribution*) that is supported at least wherever $P_\theta(Z|x_i)$ is, the following holds due to Bayes rule, the properties of logarithms, and linearity of expectation:

$$\begin{aligned} \log P_\theta(x_i) &= \mathbb{E}_{Q_i(Z)}[\log P_\theta(x_i)] \\ &= \mathbb{E}_{Q_i(Z)} \left[\log \left(\frac{P_\theta(x_i|Z)P(Z)}{P_\theta(Z|x_i)} \cdot \frac{Q_i(Z)}{Q_i(Z)} \right) \right] \\ &= \mathbb{E}_{Q_i(Z)}[\log P_\theta(x_i|Z) - (\log Q_i(Z) - \log P(Z)) + \log Q_i(Z) - \log P_\theta(Z|x_i)] \\ &= \underbrace{\mathbb{E}_{Q_i(Z)}[\log P_\theta(x_i|Z)] - KL[Q_i(Z)||P(Z)]}_{\triangleq \mathcal{L}_{Q_i, \theta}(x_i)} + KL[Q_i(Z)||P_\theta(Z|x_i)] \end{aligned} \quad (2.11)$$

where $\mathcal{L}_{Q_i, \theta}(x_i)$ is the ELBO and $KL[\cdot||\cdot]$ is an asymmetric and non-negative measure of divergence between two distributions (called the Kullback–Leibler divergence) which is 0 when the distributions are identical and is defined as follows for two distributions $P_1(A)$ and $P_2(A)$:

$$KL[P_1(A)||P_2(A)] \triangleq \mathbb{E}_{P_1(A)} \left[\log \frac{P_1(A)}{P_2(A)} \right] \quad (2.12)$$

Inspection of Equation 2.11 permits the following remarks [29, 24]:

1. $\mathcal{L}_{Q_i, \theta}(x_i)$ must be a lower bound to $\log P_\theta(x_i)$, since any KL-divergence is non-negative.
2. $KL[Q_i(Z)||P(Z|x_i)]$ is intractable, since $P_\theta(Z|x_i)$ is a function of $P_\theta(x_i)$, which is itself intractable.
3. For a given setting of θ , $\log P_\theta(x_i)$ is constant, with the effect that maximising $\mathcal{L}_{Q_i, \theta}(x_i)$ with respect to Q_i simultaneously minimises $KL[Q_i(Z)||P(Z|x_i)]$.
4. As $KL[Q_i(Z)||P(Z|x_i)]$ gets closer to 0, the *tightness of the bound* increases, i.e. $\mathcal{L}_{Q_i, \theta}(x_i)$ better approximates $\log P_\theta(X)$ and the approximation becomes exact when $Q_i(Z)$ and $P_\theta(Z|x_i)$ are identical.
5. $\mathbb{E}_{Q_i(Z)}[\log P_\theta(x_i|Z)]$ is a *reconstruction term* whose maximisation with respect to $Q_i(Z)$ and θ increases both the degree to which the chosen variational distribution places probability mass on latents that are likely to generate the given datapoint, as well as the capacity of the generator to reconstruct that datapoint from the sampled latents.
6. $-KL[Q_i(Z)||P(Z)]$ is a *regularisation term* whose maximisation with respect to $Q_i(Z)$ pushes the variational distribution towards the sampling prior so that the model is incentivised to place the latents in the subset of the latent space that one wishes to sample from with high probability for the purpose of generating new data.

From these remarks (particularly 1, 3, and 4), it can be concluded [29] that maximising the ELBO with respect to both $Q_i(Z)$ and θ will maximise an approximation to the evidence for the data under the model (thus improving its generative quality) while minimising the error of that approximation. The objective in Equation 2.10 can then be replaced by

$$\begin{aligned} \theta^* &= \operatorname{argmax}_{\theta} \mathbb{E}_{P_{data}(X)} \left[\max_Q \mathcal{L}_{Q, \theta}(X) \right] \\ &= \operatorname{argmax}_{\theta} \mathbb{E}_{P_{data}(X)} \left[\max_Q \left[\mathbb{E}_{Q(Z)}[\log P_\theta(X|Z)] - KL[Q(Z)||P(Z)] \right] \right] \end{aligned} \quad (2.13)$$

Tractability of the ELBO

Both optimisation objectives 2.10 and 2.13 contain expectations of the data likelihood taken with respect to distributions over the latent variable; these are respectively $\mathbb{E}_{P(Z)}[\log P_\theta(x_i|Z)]$ and $\mathbb{E}_{Q_i(Z)}[\log P_\theta(x_i|Z)]$. In both cases, the integrals required to compute such expectations do not have a closed-form solution in general. The crucial difference, however, is that $\mathbb{E}_{P(Z)}[\log P_\theta(x_i|Z)]$ cannot be estimated efficiently while $\mathbb{E}_{Q_i(Z)}[\log P_\theta(x_i|Z)]$ can. Doersch

[26] notes that this is because a random sample from $P(Z)$ is highly unlikely to have generated a particular x_i , so that a high number of samples might be necessary to produce an accurate estimate of $\mathbb{E}_{P(Z)}[\log P_\theta(x_i|Z)]$. On the other hand, remark 3 indicates that the optimisation process brings $Q_i(Z)$ closer to $P(Z|x_i)$, with the effect that samples from $Q_i(Z)$ will have a high probability under the model to have generated the particular x_i : since the subset of \mathcal{Z} of high probability latents encoding a particular x_i is likely to be much smaller than that of latents encoding any x that the model can produce, then fewer samples will be required to estimate $\mathbb{E}_{Q_i(Z)}[\log P_\theta(x_i|Z)]$ accurately [26], thus making the ELBO computationally tractable.

2.3.3 VAE

From objective 2.13, it is clear that any optimisation process attempts to pick the best Q_i for each datapoint x_i in a training set. Kingma and Welling note that the per-datapoint sampling loop used for such optimisation is resource intensive and does not scale well to large datasets; tackling this limitation, they proposed the framework of *variational autoencoders* (VAEs) [30, 31].

Amortised VI

VAEs introduce a *recognition model* $Q_\psi(Z|X)$ parametrized by a neural network e_ψ (called *inference network*) that amortises the cost of VI over the whole dataset. Concretely, instead of running a per-datapoint optimisation algorithm to find the best variational distribution (or its parameters) for that datapoint, one assumes the existence of a distribution $R(Z; \pi)$ with parameters $\pi \in \Pi$ and of a function $e_\psi : \mathcal{X} \rightarrow \Pi$ that can predict its parameters so that an approximation to the best variational distribution for any x can be computed efficiently as $Q_\psi(Z|x) \triangleq R(Z; e_\psi(x))$ [32]. Notably, the weights ψ optimised during training are shared across all datapoints so that optimisation objective 2.13 can be replaced by

$$\begin{aligned} \psi^*, \theta^* &= \operatorname{argmax}_{\psi, \theta} \mathbb{E}_{P_{data}(X)} [\mathcal{L}_{\psi, \theta}(X)] \\ &= \operatorname{argmax}_{\psi, \theta} \mathbb{E}_{P_{data}(X)} [\mathbb{E}_{Q_\psi(Z|X)} [\log P_\theta(X|Z)] - KL[Q_\psi(Z|X) || P(Z)]] \end{aligned} \quad (2.14)$$

Moreover, VAEs can also be constructed and trained to generate class-conditional samples by passing the class information to the encoder and decoder networks [33].

Stochastic backpropagation

Since all the model parameters are neural networks' weights (ψ and θ), these can be optimised jointly via backpropagation to pursue the objective above. In practice, the expectations and their gradients are estimated by Monte-Carlo sampling [34, 14]. This poses a challenge when differentiating with respect to ψ , as sampling from $Q_\psi(Z|x)$ is an operation that depends on ψ while not being continuous in ψ . The *reparametrisation trick* [30] bypasses the issue by sampling an auxiliary variable from a distribution independent of ψ and then applying to it a function differentiable in ψ to obtain samples distributed like $Q_\psi(Z|x)$. For instance, in the typical case that $Q_\psi(Z|x)$ is chosen to be an independent Gaussian $N(Z; \boldsymbol{\mu}, \operatorname{diag}(\boldsymbol{\sigma}^2))$:

$$\underbrace{\boldsymbol{\epsilon} \sim N(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})}_{\text{independent of } \psi} \wedge \underbrace{\langle \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \rangle = e_\psi(x) \wedge \mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}}_{\text{differentiable w.r.t. } \psi} \implies \mathbf{z} \sim N(Z; \boldsymbol{\mu}, \operatorname{diag}(\boldsymbol{\sigma}^2))$$

Moreover, the frequent choice to assume both $Q_\psi(Z|x)$ and $P(Z)$ independent Gaussians has the advantage that $KL[Q_\psi(Z|x)||P(Z)]$ can be computed analytically, thus reducing the variance of the ELBO estimator.

Autoencoders

The stochastic framework of variational autoencoders takes its name from, and is reminiscent of, the deterministic framework of deep *autoencoders* (AEs) [14]. AEs impose the dimensionality of \mathcal{Z} to be significantly less than that of \mathcal{X} (also a frequent assumption in VAEs) and have the *encoder network* $e_\psi : \mathcal{X} \rightarrow \mathcal{Z}$ map directly to the latent space and the *decoder network* $g_\theta : \mathcal{Z} \rightarrow \mathcal{X}$ map directly to the data space, as opposed to distributions over those spaces, so that the autoencoder is the composition $ae_{\psi,\theta} \triangleq g_\theta \circ e_\psi$. Parameters are then optimised via backpropagation towards the identity function so $ae_{\psi,\theta}(x) \approx x$, e.g. by minimising the mean squared error (MSE) between the input and its reconstruction. This is with the aim of learning to produce a compact representation z for any datapoint, but without being able to generate new samples.

2.3.4 VQ-VAE

The *vector quantised-variational autoencoder* (VQ-VAE) [35] is a deep-learning framework for data compression and generation that architecturally resembles an autoencoder but which forces the latent space to be discrete by placing a *vector quantiser* [36] between the output of the encoder and the input to the decoder.

Vector quantisation

Given an order- N real-valued tensor $x \in \mathbb{R}^{* \times D}$, where $*$ denotes the dimensions of the first $N - 1$ axes of x , and given an arbitrary tuple of indices i^* into such axes (e.g. if x is a $32 \times 32 \times 3$ image then $*$ = 32×32 , $D = 3$, and $i^* \in \{0 \dots 31\} \times \{0 \dots 31\}$), a vector quantiser is a function vq_C mapping every D -dimensional vector x_{i^*} of the input x to the nearest *codeword* $C_k \in \mathbb{R}^D$ out of K possible vectors from a codebook $C \in \mathbb{R}^{K \times D}$:

$$\begin{aligned} vq_C : \mathbb{R}^{* \times D} &\rightarrow \mathbb{R}^{* \times D} \\ vq_C &\triangleq \text{lookup}_C \circ \text{nearest}_C \end{aligned}$$

where

$$\begin{aligned} \text{nearest}_C : \mathbb{R}^{* \times D} &\rightarrow \{0 \dots K - 1\}^* \\ \text{nearest}_C(x)_{i^*} &\triangleq \underset{k}{\operatorname{argmin}} \|x_{i^*} - C_k\|^2 && (\text{for all } i^*) \\ \text{lookup}_C : \{0 \dots K - 1\}^* &\rightarrow \mathbb{R}^{* \times D} \\ \text{lookup}_C(z)_{i^*} &\triangleq C_k \text{ where } k = z_{i^*} && (\text{for all } i^*) \end{aligned}$$

Notably, nearest_C maps the continuous input to a compressed discrete representation z – a tensor of indices into the codebook – which reduces the dimensionality of the input by D times. z can then be decoded into an approximation of the original input via lookup_C so that $vq_C(x) \approx x$.

Training

Using the definition of vector quantiser presented above, a VQ-VAE with encoder network $e_\psi : \mathcal{X} \rightarrow \mathbb{R}^{* \times D}$ and decoder network $g_\theta : \mathbb{R}^{* \times D} \rightarrow \mathcal{X}$ is the function $vqvae_{\psi, \theta, C} \triangleq g_\theta \circ vq_C \circ e_\psi$, which can be viewed as a AE with a vector quantiser as a non-linear activation between the two networks [35]. However, training by backpropagation is not as straightforward as in AEs because the nearest-neighbour step in vector quantisation, similarly to sampling in VAEs, is not differentiable. In this case, the issue is bypassed using the gradient of the loss function with respect to the quantisation output to approximate the gradient with respect to its input, i.e. $\nabla_{vq_C(e_\psi(x))} L \approx \nabla_{e_\psi(x)} L$.

In addition to a standard reconstruction loss (e.g. MSE), VQ-VAEs minimise a *commitment loss* to move the encoder outputs towards the current codewords in C and, symmetrically, a *quantisation loss* to learn the optimal codewords by making them closer to the current encoder outputs. The resulting loss function is

$$L \triangleq \underbrace{L_{rec}(\hat{x}, x)}_{\text{reconstruction loss}} + \beta \underbrace{\|z_e - sg[z_q]\|^2}_{\text{commitment loss}} + \underbrace{\|sg[z_e] - z_q\|^2}_{\text{quantisation loss}} \quad (2.15)$$

where $\hat{x} = vqvae_{\psi, \theta, C}(x)$, $z_e = e_\psi(x)$, $z_q = vq_C(e_\psi(x))$, β is a scalar hyperparameter that weighs the contribution of the commitment loss, and $sg[\cdot]$ is an identity function which stops the gradient of its argument from being accounted for during backpropagation. In practice, the quantisation loss term used to learn the codebook is often dropped in favour of setting each codeword C_k to an exponential moving average (EMA) [35] of the encoder outputs that are closest to C_k , akin to an online K -means. Notably, unlike VAEs, sampling images from a VQ-VAE latent space cannot be performed immediately after training but rather, a prior is usually learnt over the discrete latent space of the VQ-VAE. [37]

2.4 CBIR

Content-based image retrieval (CBIR) [38] pipelines allow the user to submit a query image and obtain in response a list of images relevant to the query image and selected from an existing gallery dataset. Ideally, the returned list should only contain relevant images and, especially if the potential number of relevant results is very large, the list should be presented in descending order of relevance, with the most relevant results at the top. While the notion of ‘relevance’ is specific to the use case, one typically seeks to retrieve images that are similar in some regard to the query image.

2.4.1 Feature engineering

In its early stages, the field of CBIR largely relied on measuring low-level global features of an image [38], including colours, textures, shapes, and their spatial positioning within the image, in order to characterise the scene and assess its similarity to other images. However, any quantitative measure of such features tends to vary widely when illumination, viewpoint, or scale differ across images, even if their scenes and content remain semantically unchanged, so that similarity measurements are not robust and retrieval performance is suboptimal [39].

To address the shortcomings of low-level global features and achieve invariance to a variety of visual changes, researchers engineered representations of features which are local and relevant

only to a specific image region or patch. Most prominently, the *scale-invariant feature transform* (SIFT) [40] is a method that detects the salient keypoints, assigns each keypoint the orientation which is dominant amongst the gradient directions in its neighbourhood, and then considers a local patch of 4×4 subregions centred at a keypoint, producing, for each subregion, an 8-bin histogram of the quantised gradient directions in that subregion. Finally, the 16 resulting 8-bin histograms are flattened to produce, for each keypoint, a real-valued feature vector with 128 entries known as a *SIFT descriptor*. By combining the properties of scale invariance from the selected keypoints, rotation invariance from the histograms, and robustness to changes in illumination from the gradient directions, SIFT provided a robust local feature descriptor that historically was used to great success in the image retrieval setting [39].

2.4.2 Feature embedding

The SIFT procedure outlined above can produce thousands of descriptors [40] for a single image. Therefore, any method that seeks to retrieve images similar to a query image by exhaustively comparing its descriptors with those of all the images already present in a large database is bound to be computationally infeasible. In practise, many CBIR systems achieve satisfactory retrieval speed by separating the process in two stages [41]: firstly, the database is filtered down to a small set of candidate images by comparison with a compact and fixed-size representation (a *feature embedding*) of each image’s local descriptors; then, the resulting images are re-ranked using their local descriptors for further similarity measurement (e.g. by performing spatial consistency checks [42]).

Consequently, the aggregation of an arbitrary number of local features into a highly discriminative feature embedding becomes important for retrieval precision, and a number of embeddings have been devised following the *bag of visual words* (BoW [42]) strategy. In particular, for descriptors of size D , a *codebook* of K *visual words* (i.e. D -dimensional vectors) is learnt through clustering algorithms, such as k -means, by training on a representative set of feature descriptors. Then, an embedding for an arbitrary number of descriptors can be produced by one of multiple encoding strategies of increasing accuracy. For instance, the vanilla version of BoW simply returns a K dimensional embedding where entry k is the frequency of feature descriptors that are closest to k th visual word; while the more accurate *vector of locally aggregated descriptors* (VLAD [43]) stacks the sums of the residual vectors between each visual word and their closest descriptors. There also exist embeddings that rely on fuzzy clustering of descriptors, such as the *fisher vector* (FV [44]).

2.4.3 Vector quantisation for CBIR

Efficient search

Alongside the development of compact feature embeddings, the necessity to perform efficient CBIR on large datasets lead to applications of vector quantisation techniques to the task of image compression and retrieval [45, 46]. The utility of VQ to CBIR is twofold. Firstly, it can be used to compress the input data, as outlined in section 2.3.4; additionally, it permits efficient search of dataset samples that are similar to an input query by avoiding the multiplication operations which would otherwise be necessary to compute the squared Euclidean distance (or some other similarity metric) between the query and each dataset image. In particular, given a vector quantiser vq_C with codebook $C \in \mathbb{R}^{K \times D}$ and using

the same notation and definitions of section 2.3.4, the distance $d(x_q, x_g)$ between a query image embedding $x_q \in \mathbb{R}^{* \times D}$ and a gallery image embedding $x_g \in \mathbb{R}^{* \times D}$ can be quickly approximated by looking up the distances between codewords in a precomputed table, i.e.

$$d(x_q, x_g) \approx d(vq_C(x_q), vq_C(x_g)) = \sum_{i^*} lt_C(nearest_C(x_q)_{i^*}, nearest_C(x_g)_{i^*})$$

where

$$lt_C : \{0 \dots K - 1\} \times \{0 \dots K - 1\} \rightarrow \mathbb{R}$$

$$lt_C(k_1, k_2) \triangleq \|C_{k_1} - C_{k_2}\|^2$$

where $lt_C(\cdot, \cdot)$ is precomputed in a $K \times K$ lookup table, and $nearest_C(x_g)$ is precomputed for all gallery images; note that such precomputations can happen before a query is submitted, because both the codebook and the gallery images are available.

Product quantisation

While standard vector quantisation is computationally and memory efficient for codebooks with a small number K of codewords, larger values can become infeasible because, as K doubles, so does the computational complexity of the nearest neighbour search step $nearest_C$, while the memory occupied by the lookup table of precomputed distances quadruples, and the number of bits required to represent the discrete latent of indices into the codebook grows by the number of indices (1 bit for each scalar index), which can be an issue if one requires very compressed embeddings. These properties can become problematic because a large number of codewords is highly desirable in order to reduce the quantisation error $\|x - vq_C(x)\|^2$.

Product quantisation (PQ) [47] is a quantisation method that simultaneously mitigates all of the issues above, as the product quantiser $pq_C : \mathbb{R}^{N \times D} \rightarrow \mathbb{R}^{N \times D}$ splits the input x into M equally sized chunks $x^{(m)} \in \mathbb{R}^{(N/M) \times D}$, quantises each chunk using a standard vector quantiser $vq_{C^{(m)}}$ (in this case called a *subquantiser*) with codebook $C^{(m)}$ specific to that subquantiser, and finally concatenates back the resulting quantised chunks so that $pq_C(x) \triangleq [vq_{C^{(1)}}(x^{(1)}) \dots vq_{C^{(M)}}(x^{(M)})]$. Thus, the effective codebook of the product quantiser is the Cartesian product of the codebooks of its subquantisers $C = C^{(1)} \times \dots \times C^{(M)}$. The number of codewords is typically set to be the same for all subquantisers, but the particular codewords learnt (e.g. via k -means) in each codebook are allowed to vary.

Effectively, such a setup permits to represent a large number of codewords (MK), so as to obtain a lower reconstruction error, but under the constraint that each subquantiser has only access to K of them, thus reducing the space and computational complexity compared to a standard vector quantiser with a codebook of size MK . Importantly, distances can still be computed efficiently by precomputing M lookup tables of size K^2 (one for each subquantiser), as opposed to the single lookup table of size M^2K^2 for the standard vector quantiser.

2.4.4 Feature learning

In the last decade, the state-of-the-art performance of CNNs in a variety of computer vision tasks was established, starting from the AlexNet [48] architecture applied to image

classification in 2012, and subsequently inspiring breakthroughs in image segmentation [49] and object detection [50]. In following with this trend, the use of CNNs and deep learning in the image retrieval setting has become widespread and currently underlies most CBIR frameworks [41].

Due to their hierarchical structure, CNN architectures can be interpreted [39] as pipelines that recognize, layer by layer, features of increasing semantic significance, with the shallower layers recognizing low-level features and deeper layers learning higher-level semantic concepts about the image content. In view of such an interpretation and of CNNs' capacity to generalise to different datasets [51], Babenko et al. [52] made one of the first attempts at exploiting CNNs for feature extraction by producing a global feature vector (a *neural code*) from the neurons of the fully-connected layers before the output of a CNN trained on ImageNet [53], thus obtaining competitive performance on an image retrieval task for an unrelated dataset. Additionally, the authors demonstrated how fine-tuning the parameters of the pre-trained network on the retrieval dataset could yield further increases in retrieval precision.

Later research [54] improved performance by considering the feature maps produced by intermediate convolutional layers instead of the output of fully connected ones. Therefrom, compact feature embeddings can be obtained through the aggregation of convolutional activations across feature maps by either classical BoW embeddings [55] or by various pooling strategies, including channel-wise max-pooling (MAC [56]), sum-pooling (SPoC [54]), and region-wise max-pooling (R-MAC [56]). The resulting embeddings have demonstrated increased robustness to image transformations compared to the activations of fully connected layers. Intuitively, such robustness has been attributed [54] to the role played by each neuron in a feature map, which can be understood as that of a local descriptor characterizing the image region which corresponds to the neuron's receptive field, thus mimicking a SIFT descriptor for that region. Under such intuition, it can be noted that the aggregation of multiple local descriptors is found to be more robust than a single global descriptor both for hand-crafted descriptors and for feature vectors extracted from deep learning architectures.

2.4.5 Evaluation Metrics

Precision, recall, and F1

In the information retrieval setting, precision (P) measures the proportion of relevant results amongst the results retrieved. Recall (R) measures the proportion of results retrieved amongst all the relevant results.

$$P \triangleq \frac{\text{\#relevant results retrieved}}{\text{\#results retrieved}} \qquad R \triangleq \frac{\text{\#relevant results retrieved}}{\text{\#relevant results}}$$

The $F1$ score is often used to combine information from recall and precision into a single measure. The more general F_β score can also be employed for finer-grained control.

$$F1 \triangleq \frac{2PR}{P + R} \qquad F_\beta \triangleq \frac{(1 + \beta^2)PR}{\beta^2 P + R}$$

Top- k metrics

Precision, recall, and F1 are appropriate for the unranked retrieval setting, with the ordering of the retrieved results being immaterial. However, the IR literature [57] has highlighted how such metrics are unsuitable for assessing the performance of a ranked retrieval system, as the end user may be interested only in the top results returned by the system (e.g. in a web search engine), rather than in a simpler, unordered set of relevant results. Hence, corresponding metrics have been devised (precision@ k , recall@ k , and F1@ k) that are evaluated only up to the top- k items retrieved.

AP and mAP

The average precision (AP) [57] for a single query measures the average precision at rank k ($P@k$) over all the r relevant results amongst the n retrieved results for the query, i.e.

$$AP \triangleq \frac{1}{r} \sum_{k=1}^n P@k \cdot rel(k)$$

where $rel(k)$ is 1 if result k is relevant and 0 otherwise. Intuitively, high values of AP indicate that most of the relevant results appear at the top of the ranked list of results for that query. The mean average precision (mAP) over a number of queries is normally provided in the literature as a benchmark metric.

Mean reciprocal rank

The *reciprocal rank* [58] for a list of results returned in response to a query is the reciprocal of the position (i.e. rank) in the list of the first relevant result returned, or 0 if no relevant results are returned. The mean reciprocal rank (MRR), is the average of the reciprocal ranks over a number q of queries:

$$MRR \triangleq \frac{1}{q} \sum_{i=1}^q \frac{1}{rank_i}$$

where $rank_i$ is the position of the first result relevant to the i -th query. The higher the MRR, the earlier a user is expected to find a relevant result when scrolling down the result list for their submitted query.

NDCG

In many cases, the relevance of a retrieved result to a query is not a binary notion, as some results may be more relevant than others. When a *graded relevance* score which indicates the usefulness of a particular document to the query exists, the ideal retrieval system should rank results in decreasing order of relevance. In order to compare the performance of different pipelines on the same retrieval task, one could sum the graded relevance of the returned documents to obtain the *cumulative gain* (CG). However, two retrieval systems that simply return all documents would obtain an identical CG regardless of the ordering of the produced list, even though such ordering is crucial to the user's experience. To address this, the *discounted cumulative gain* (DCG) [59] better reflects the notion that the utility to the user or *gain* of a highly relevant document is reduced if that document is placed further down

the list. For a returned list of length N :

$$DCG \triangleq \sum_{i=1}^N \frac{rel(i)}{\log_2(i+1)}$$

where $rel(i)$ is the graded relevance score of the document at position i in the returned list; i.e. the graded relevance of an item is discounted more heavily the greater its position in the list. To obtain a metric that is not proportional to the length of the list returned, the DCG is often normalised by the *ideal discounted cumulative gain* (IDCG) computed on the list of all documents in the datasets in descending order of their ground-truth relevance. The resulting metric is the *normalised discounted cumulative gain* ($NDCG \triangleq DCG/IDCG$) [59].

2.5 Ethical considerations

An important point of ethical concern stems from the potential use cases of the CCBIR framework, particularly those that involve high stakes decisions being made on the basis of its results. For example, a doctor might be using a CCBIR system as an aid in medical diagnosis. It is necessary to exercise great care in such a situation, noting that the tool may be an aid in – and not a substitute for – the doctor’s best judgement. This recommendation of prudence is technically grounded in the difficulty of assessing the accuracy of a real-world counterfactual inference system, as true counterfactuals are generally unobservable, and strong assumptions that do not hold in the real world may have been made.

Despite the lack of confidence that the above might entail, it should be noted that, compared to a generative model, the CCBIR framework would return images of real patients rather than synthetic ones: it is then easier for a doctor to establish whether a result is relevant by studying the specifics of the real case on their own, rather than having to trust a generative model to have synthesised an accurate counterfactual patient. This discussion applies beyond the medical field and is related to the fact that the CCBIR system must return images from a real dataset. On the other hand, precisely the requirement of data access may be a cause for concern: while purely generative models do not require an underlying dataset after training, the CCBIR pipeline described does. If the data is sensitive, such as in the case of patient’s data, or of private financial information, controlling access to the system would be essential.

Notably, though the medical application is likely one of the most prominent potential use cases for the CCBIR pipeline, the dataset used for evaluation purposes during the project is synthetic and did not involve medical data.

Lastly, one cause of concern may be environmental, as the ELBO-based approach for CCBIR is particularly computationally expensive for large datasets. Indeed, given that an alternative sample-similarity based method to the ELBO approach has been provided and that it is less computationally expensive, it may be preferred for tasks and datasets for which the retrieval performance of the two methods is similar.

Chapter 3

Counterfactual image generation

In this chapter, a new framework is presented to approximate via neural networks the generative process described by a structural causal model and generate new counterfactual samples. It applies a variational inference strategy for the tractable abduction of exogenous variables – originally proposed by the DeepSCM [1] approach – to the training of a deep twin network [2] architecture. A synthetic dataset is then used to evaluate the framework’s capacity to generate credible samples by means of twin network counterfactual inference (section 2.2.3).

3.1 Definition

Given

1. an SCM $\mathcal{M} = \langle U, P(U), V, F \rangle$ that describes the generative process for images from the distribution $P_{\mathcal{M}}(I)$, where $I \in V$;
2. an observed image $I = i$, referred to as the *factual image*, produced by the mechanisms of \mathcal{M} in a (unobserved) state of the world $U = u$;
3. two sets of values, c and t , that were observed for two sets of variables, C and T respectively, in the state of the world that produced the factual image i ; the values in c are referred to as the *covariates*; the set t is called the *factual treatment*; C , T and $\{I\}$ are pairwise disjoint, and $C, T \subseteq V$;
4. a set t^* of hypothetical values for the variables T , possibly different from the factual treatment t and referred to as the *counterfactual treatment*;

the task of *counterfactual image generation* consists in the production of samples from the probability distribution over the image $I = i^*$ that one would have expected to see in a state of the world that produced the factual image $I = i$ and covariates $C = c$, had T been t^* instead of t , i.e.

$$i^* \sim P_{\mathcal{M}}(I_{T:=t^*} = i^* | T = t, C = c, I = i) \quad (3.1)$$

where any sampled i^* is a possible *counterfactual image* for i .

3.2 Image generation task

This section presents the concrete dataset and associated counterfactual image generation task tackled by the chapter.

3.2.1 Dataset

Morpho-MNIST

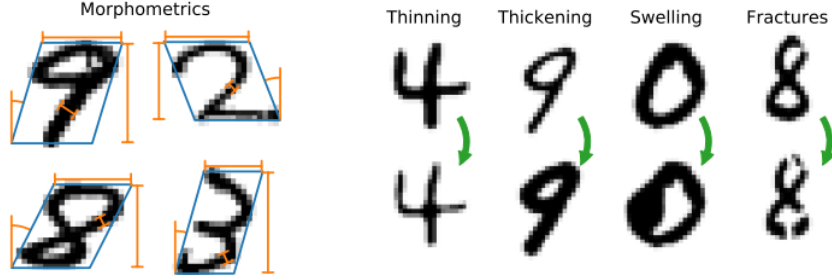


Figure 3.1: Taken from [3]. The morphometrics shown on the left include thickness, slant, height, width, and area. The effect of each perturbation on an unperturbed image is shown on the right.

Partly following [4], the dataset used for the image generation task was produced by applying a set of image perturbations to the MNIST dataset [60], a well-known collection of 70,000 greyscale images of handwritten digits measuring 28×28 pixels each. The perturbations were performed using the Morpho-MNIST [3] library, which implements a set of transformations on images from the original MNIST dataset, including thickening, thinning, swelling, and fracturing. Additionally, it provides a number of image measurement methods to compute the area, width, height, slant, thickness, and intensity of a digit within an image: such metrics are referred to as *morphometrics*. From Figure 3.1[3], it is clear that Morpho-MNIST simultaneously allows imposing a known causal structure on the data, as the application of a perturbation can be seen as a cause for how the perturbed image looks, while at the same time permitting the extraction of expressive covariates about an image by measuring its morphometrics. Such conveniences simplify the training and evaluation processes considerably compared to real world data, and thus motivated the dataset choice.

Generation

To generate the synthetic dataset, for each original image i_n^{orig} from the MNIST dataset, a swelling and a fracturing perturbation with fixed parameters (e.g. size of swelling and number of fractures) but randomly sampled locations are applied to i_n^{orig} to produce an image i_n^{swell} with a swollen digit and an image i_n^{frac} with a fractured digit respectively; two vectors t_n^{swell} and t_n^{frac} are constructed by concatenating the one-hot encoding of the perturbation (either swelling or fracturing) with the relevant pixel locations which have been sampled for the perturbation, and padding with -1 as necessary to obtain identically-sized vectors. Then, the one-hot encoding for the digit label of i_n^{orig} is concatenated with the morphometrics of i_n^{orig} to produce a metrics vector c_n of the original image, after each metric was standardised across the dataset to have zero mean and unit standard deviation (with original mean and standard deviation being computed only on the basis of the training dataset split). Lastly, i_n^{swell} and i_n^{frac} are made continuous by scaling pixel values to the range $[-1, 1]$. The final result is a

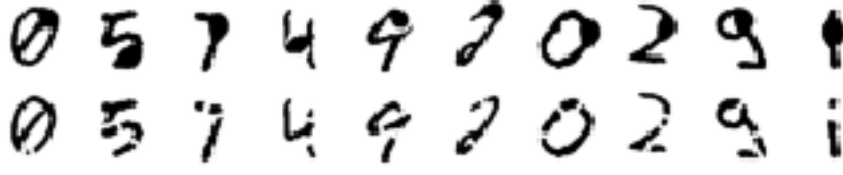


Figure 3.2: A sample of swollen-fractured pairs (vertically stacked) from the constructed synthetic dataset.

dataset of tuples $D = \{\langle i_n^{swell}, i_n^{frac}, t_n^{swell}, t_n^{frac}, c_n \rangle\}_{n=1}^{70,000}$, which was split by allocating 54,000 items to training, 6,000 to validation, and 10,000 to testing. Notably, such a dataset can easily be augmented¹ by applying A different swelling and fracturing perturbations to each original image, which lead to a dataset $D^A = \{\langle i_n^{swell}, i_n^{frac}, t_n^{swell}, t_n^{frac}, c_n \rangle\}_{n=1}^{70,000 \times A}$.

3.2.2 Causal graph

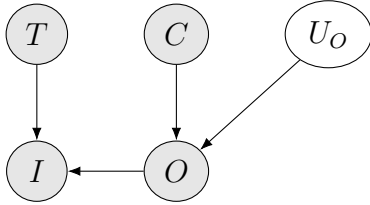


Figure 3.3: Causal graph modelling the full generative process of an image from the constructed dataset. O models the original unperturbed image with associated exogenous variable U_O .

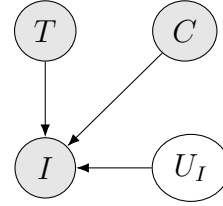


Figure 3.4: Simplified causal graph for the image generation process. The original unperturbed image O is no longer modelled directly compared to Figure 3.3.

Consider an image i taken from any item in D and the corresponding treatment t , be it either a swelling or a fracturing, that produced i by perturbing the original image i^{orig} with associated covariates c . The generative process for i can be modelled using the causal graph in Figure 3.3: it encodes that the covariates $C = c$ together with some unobservable random noise $U_O = u_O$ determine the pixel values of the original image $O = i^{orig}$; on the other hand, once the original image and the treatment vector $T = t$ are known, the pixel values $I = i$ are immediately established in a fully deterministic manner. Since I is fully determined when T and O are, and O is fully determined when C and U_O are, transitively I is fully determined when T , C , and U_O are; this permits omitting O from the graphical representation, and reinterpreting U_O as a noise variable U_I affecting the determination of I . These steps lead to the simpler causal graph in Figure 3.4, thus avoiding the direct modelling of the original image pixels, which are not of interest to the counterfactual inference tasks explored in this project.

¹If performing similar augmentations, care should be taken in avoiding data leakage by first splitting the original MNIST dataset into training, validation, and test portions, and only then applying the necessary perturbations, as no two dataset items produced from the same original image should appear in different dataset splits.

3.2.3 Task

The constructed synthetic dataset naturally leads to two symmetrical counterfactual image generation tasks for each dataset item, while simultaneously making evaluation relatively straightforward. In fact, given an item $\langle i_n^{swell}, i_n^{frac}, t_n^{swell}, t_n^{frac}, c_n \rangle$ and without loss of generality, one can take the swollen image i_n^{swell} to be the factual outcome under treatment t_n^{swell} and sample a counterfactual i^* , under the hypothetical treatment of a fracturing perturbation specified by t_n^{frac} , via some parametric generative model for the true SCM, so that $i^* \sim P_\theta(I_{T:=t_n^{frac}} = i^* | T = t_n^{swell}, C = c_n, I = i_n^{swell})$. Evidently, the fractured image i_n^{frac} must be one sample from the true counterfactual distribution, as it is the very image that was generated under the counterfactual treatment in the same state of the world (i.e. starting from the same original image i_n^{orig}) in which the swollen image was generated. Furthermore, visual inspection of the dataset samples (Figure 3.2) helps to support the argument that, once the factual (swollen) image is known, the possible original images that could have generated it under the specified factual treatment is severely restricted to images that must be extremely similar to the original image i_n^{orig} that has in fact generated i_n^{swell} ; this is because a perturbed image shares a large number of pixels with the original image. Lastly, for a fixed treatment, one can expect that similar original images will also lead to similar perturbed images, so that samples from the true counterfactual distribution of interest are likely to be close to i_n^{frac} too. Such a chain of reasoning justifies using the similarity between i_n^{frac} and i^* as an approximate measure of the model’s performance, as models that produce samples which are very dissimilar from i_n^{frac} are unlikely to be good approximations of the ground truth SCM. Notably, the argument is symmetric for the case where fractured images are taken to be factual and swollen images are taken to be counterfactuals.

3.3 Method: variational twin network

In this section, a new method for counterfactual generative modelling is proposed in connection to the defined image generation task. The strategy combines ideas from the counterfactual inference methods in [1] and [4], which are outlined next, before presenting the new method.

3.3.1 DeepSCM

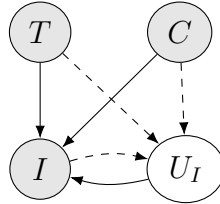


Figure 3.5: DeepSCM for the causal diagram of the counterfactual image generation task: a solid directed edge represents that the source variable is both a cause of the destination variable and an input to the decoder network of the conditional VAE producing the destination variable; a dashed directed edge indicates that the source variable is an input to the encoder network used to approximately abduce the conditional distribution over the destination variable.

Pawlowski et al. [1] propose a generative method for abduction-action-prediction counterfactual inference (section 2.2.3) that makes the model’s optimisation tractable even on high-

dimensional data, such as images, by representing each mechanism F_i for a given variable V_i as a function that takes as input the parents Pa_i of V_i and that is invertible, at least stochastically [61], with respect to the corresponding noise variable $U_i \in Pa_i$. Let $Pa'_i \triangleq Pa_i \setminus \{U_i\}$. Such invertibility can be achieved by making F_i identical to the decoder network $d_\theta(U_i, Pa'_i)$ of a conditional VAE and stochastically inverting² its output by sampling from the variational distribution parametrised by the encoder network $e_\psi(V_i, Pa'_i)$. For example and for the purposes of the counterfactual image generation task of interest, the achieved invertibility enables using the factual image i , its treatment t , and covariates c as evidence that is passed through the encoder network to abduce an approximation of the conditional distribution $P(U_I|T = t, C = c, I = i)$ over possible states of the noise variable; then, one can sample u_I from such approximation and pass it through the decoder alongside the counterfactual treatment t^* and covariates to obtain a counterfactual sample $i^* = d_\theta(u_I, t^*, c)$. A diagram for the setup is shown in Figure 3.5.

3.3.2 DTN

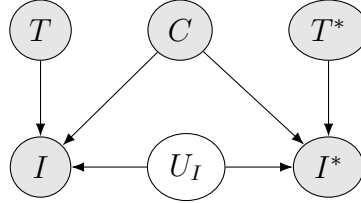


Figure 3.6: Causal graph of the twin network for the model in Figure 3.4. The node C for the covariates does not need to be duplicated, since C is not a causal descendant of the treatment. A deep twin network for this causal graph will model the mechanisms $I := F_I(U_I, T, C)$ and $I^* := F_{I^*}(U_I, T^*, C)$ as neural networks.

Before attempting counterfactual video generation, Reynaud et al. [4] tackle a counterfactual image generation task on a Morpho-MNIST based dataset similar to the one constructed in this project and with the same causal diagram (Figure 3.4). The method they proposed uses a deep twin network [2] architecture generally matching the causal graph of the twin network (shown in Figure 3.6) and reduces the dimensionality of the images to be generated, by using a VQ-VAE trained on both fractured and swollen images from the dataset. Let $e : \mathcal{G} \rightarrow \mathbb{R}^{N \times D}$, $vq : \mathbb{R}^{N \times D} \rightarrow \mathbb{R}^{N \times D}$, and $g : \mathbb{R}^{N \times D} \rightarrow \mathcal{G}$ be respectively the encoder network, the vector quantiser and the decoder network of the trained VQ-VAE with frozen weights. Reynaud et al. [4] define a DTN:

$$\begin{aligned}
 dtn_\theta : \mathcal{U}_I \times \mathcal{T} \times \mathcal{T} \times \mathcal{C} &\rightarrow \mathbb{R}^{N \times D} \times \mathbb{R}^{N \times D} \\
 dtn_\theta(u_I, t, t^*, c) &\triangleq \langle i, i^* \rangle
 \end{aligned} \tag{3.2}$$

where

$$i \triangleq nn_\theta^{(4)}(u_I \cdot nn_\theta^{(3)}([nn_\theta^{(1)}(t), nn_\theta^{(2)}(c)])) \tag{3.3}$$

$$i^* \triangleq nn_\theta^{(4)}(u_I \cdot nn_\theta^{(3)}([nn_\theta^{(1)}(t^*), nn_\theta^{(2)}(c)])) \tag{3.4}$$

²Interestingly, methods that provide exact invertibility were also used by Pawlowski et al. [1] to better model lower dimensional exogenous noise variables by representing F_i as a composition of (conditional) invertible functions, i.e. a conditional normalising flow [62, 63].

$\{nn_{\theta}^{(k)}\}_{k=1}^{(4)}$ are neural networks with appropriate input-output dimensions, and $[\cdot, \cdot]$ denotes vector concatenation. Training of the DTN is then performed so that, for each item $\langle i_n^{swell}, i_n^{frac}, t_n^{swell}, t_n^{frac}, c_n \rangle$ in a training batch, an associated noise variable $u_n \sim [N(0, 0.25) \bmod 1 + 1]$ is sampled, $\langle z_q, z_q^* \rangle = dt_{n\theta}(u_n, t_n^{swell}, t_n^{frac}, c_n)$ is computed, and the parameters θ are optimised by gradient descent to minimise the reconstruction errors between the quantised latent vectors obtained by encoding the ground truth images via the VQ-VAE and the produced DTN outputs, i.e. $MSE(vq(e(i_n^{swell})), z_q)$ and $MSE(vq(e(i_n^{frac})), z_q^*)$. After training, counterfactual image generation is performed by

1. sampling a large number of u_I ;
2. computing $\langle z_q, z_q^* \rangle = dt_{n\theta}(u_I, t_n^{swell}, t_n^{frac}, c_n)$ for each u_I ;
3. picking the pair $\langle z_q, z_q^* \rangle$ such that the error between the quantised latent $vq(e(i_n^{swell}))$ of the observed factual image and the sampled latent z_q of the factual output of the DTN is minimised³;
4. decoding the corresponding z_q^* via the VQ-VAE decoder as $i^* = g(z_q^*)$ and returning it as the counterfactual image sample.

Shortcomings of DTN training

While the DTN method described above has been empirically observed to produce visually convincing counterfactuals on an image generation task [4], a few theoretical shortcomings may be identified with the procedure.

Since the sampled u_I have no correlation with the ground truth images, a DTN trained by concatenating u_I as another input may largely ignore u_I for the purposes of producing an output image, thus reducing sample diversity at inference. In fact, Reynaud et al. [4] find it necessary to multiply the noise with an intermediate encoding for the treatment and covariates (see definitions 3.3 and 3.4); the authors claim that ‘This ensures that the network cannot easily disentangle the noise, as it would be the case with a concatenation’ [4]. Since multiplication involves the loss of information about the two original inputs, the learnt model for the mechanism $I := F_I(U_I, T, C)$ stops being a universal function approximator and is forced into a particular form, without justification from the training data or from background knowledge about the generative process to be modelled. Furthermore, the desirable increase in sample diversity of such a procedure comes at the cost of perturbing a representation of the correct treatment and covariates, rather than being brought about by additional image information encoded in u_I . Indeed, a DTN trained according to the strategy in [4] can be interpreted as a generator for the single most likely factual-counterfactual pair of images (conditional on the treatments and covariates), which is regularised through the injection of random noise during both training and inference to achieve sample diversity.

An alternative DTN training process was followed by Vlontzos et al. [2] on different counterfactual inference tasks, including on one semisynthetic and multiple real world datasets. In particular, a randomly sampled value u_I for the exogenous noise variable was preassigned to

³Since z_q and z_q^* are continuous, this process differs from the discrete case discussed in subsection 2.2.5, when one can simply pick the sampled pairs in which the factual outcome matches exactly.

each dataset item before training; the assigned noise becomes then a deterministic feature of the input data at training time, which is passed through a neural network to increase the flexibility of the noise distribution and then concatenated to the rest of the input variables. Since such an assignment forcibly correlates u_I to the ground truth outcomes, one can reasonably hypothesise that the DTN will be less likely to ignore u_I , and thus produce diverse samples at inference time even without an explicit entangling operation such as multiplication. In fact, in the case of a semisynthetic dataset in which the factual and counterfactual ground truth outcomes are generated as a function of the sampled u_I , the (causal) correlation from u_I to the data samples truly exists in the dataset. However, the procedure arguably appears theoretically problematic when it is used on real world data as the ‘true’ u_I that generated the counterfactual sample is not known, but one random u_I is nonetheless sampled and associated to each item, without the possibility to alter during training its position in the noise space \mathcal{U}_I .

Most fundamentally, the optimisation objectives of both training algorithms, which are based on reconstruction error, are partly lacking for the purposes of training a latent variable generative model with truly unknown latent variables, as they don’t permit the learning of a placement of latent vectors (u_I) in the latent space (\mathcal{U}_I) that is consistent with the sampling prior $P(U_I)$, i.e. which assigns high probability outcomes to high probability latent vectors: in the case of the algorithm from Reynaud et al. [4], this is because the sampling of a latent vector at each training step is independent of the factual and counterfactual outcomes; in the case of the procedure of Vlontzos et al. [2], this is because the position of a latent vector for an outcome cannot change during training.

3.3.3 VTN

The outlined issues with the training of deep twin networks are completely absent in the abduction-action-prediction inference system of a DeepSCM, as variational inference provides a probabilistically principled framework to train the neural networks modelling the mechanism of an SCM in the presence of unobserved exogenous variables. Noting that the twin network \mathcal{M}^{TN} for an SCM \mathcal{M} is itself an SCM, it is thus amenable to an adapted DeepSCM approach. Hence, this project proposes a generative model for counterfactual inference, the *variational twin network* (VTN), which is trained as a DeepSCM but is interpreted as a deep twin network for the purposes of counterfactual inference.

Construction

Given the causal graph G_m of an SCM $\mathcal{M} = \langle U, P(U), V, F \rangle$ where the functional form of the mechanisms in F is not known, a variational twin network for \mathcal{M} can be produced via the following steps:

1. construct the causal graph $G_{\mathcal{M}^{TN}}$ of the twin network $\mathcal{M}^{TN} = \langle U, P(U), V \cup V^*, F \cup F^* \rangle$ for \mathcal{M} ;
2. for each pair of factual and corresponding counterfactual variable $V_i, V_i^* \in V \cup V^*$, with respective sets of parents Pa_i, Pa_i^* , mechanisms F_i, F_i^* and common noise variable U_i , instantiate a conditional VAE with one encoder network $e_i(\cdot)$ and two decoder

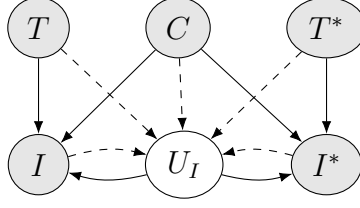


Figure 3.7: VTN for the causal graph in Figure 3.4. Much like the DeepSCM in Figure 3.5, the nodes with outgoing solid arrows directed into I are input variables to the decoder network $d_I(U_i, T, C)$ modelling the mechanism for I ; symmetrically, nodes with outgoing solid arrows going into I^* are input variables to the decoder network $d_{I^*}(U_i, T^*, C)$ modelling the mechanisms for I^* . The nodes with outgoing dashed arrows into U_I are input variables to the encoder network $e(I, I^*, T, T^*, C)$ which parametrises the variational distribution used to abduce U_i during training. Note that the encoder network is not used at inference and the VTN becomes identical to the DTN following the causal graph in Figure 3.6

networks $d_i(\cdot), d_i^*(\cdot)$; use the decoder networks to approximate the two mechanisms as

$$\tilde{V}_i := d_i(U_i, Pav_i) \quad \tilde{V}_i^* := d_i^*(U_i, Pav_i^*)$$

where $Pav_i \triangleq Pa_i \setminus \{U_i\}$ and $Pav_i^* \triangleq Pa_i^* \setminus \{U_i\}$ are the parents with the exogenous noise variable excluded; and use the encoder network to parametrise a single variational distribution Q over the noise variable U_i :

$$Q(U_i | V_i, V_i^*, Pav_i, Pav_i^*) \triangleq Q(U_i; e_i(V_i, V_i^*, Pav_i, Pav_i^*))$$

3. compose the decoder networks for every endogenous variable, following the topological order indicated by the causal graph $G_{m^{TN}}$, so as to build a function that, given concrete values for all the noise variables U (i.e. given a fully defined state of the world), returns the values taken by all the endogenous variables $V \cup V^*$; in other words, use all the instantiated decoders to construct the deep twin network $dtn : \mathcal{U} \rightarrow \mathcal{V} \times \mathcal{V}^*$.

Training objective

Training of the VTN can be performed independently on each conditional VAE: given a ground truth tuple $\langle pav_i, pav_i^*, v_i, v_i^* \rangle$ of values for the relevant variables $Pav_i, Pav_i^*, V_i, V_i^*$ from a training dataset, one seeks to optimise the parameters θ, θ^* of the decoder networks $d_\theta(\cdot), d_{\theta^*}^*(\cdot)$ and ψ of the encoder network $e_\psi(\cdot)$ so as to maximise the ELBO:

$$\begin{aligned} & \mathcal{L}_{\theta, \theta^*, \psi}(v_i, v_i^*, pav_i, pav_i^*) \\ & \triangleq \mathbb{E}_{Q_\psi(U_i | v_i, v_i^*, pav_i, pav_i^*)} [\log P_\theta(v_i | U_i, pav_i) + \log P_{\theta^*}(v_i^* | U_i, pav_i^*)] \\ & \quad - KL[Q_\psi(U_i | v_i, v_i^*, pav_i, pav_i^*) || P(U_i)] \end{aligned} \quad (3.5)$$

Inference

After training, counterfactual samples are produced in the same manner as in a standard DTN (detailed in subsection 2.2.5): the inputs to the mechanisms for the variables that

are children of observed or intervened-upon variables are forcibly set, while the remaining endogenous variables are determined by sampling the exogenous variables and evaluating the model’s mechanisms in topological order. Lastly, one conditions on the factual outcomes to select the counterfactual outcomes of interest, as in [2]. Notice that this is in stark contrast to the abduction-action-prediction of a DeepSCM: while the encoder networks in a DeepSCM are needed at inference to abduce the noise, the encoder networks of a VTN are used only during training. In fact, the inputs to the encoder network of a VTN require both a factual and counterfactual outcome, but the latter is typically not available when one is interested precisely in its generation, so that a VTN becomes identical to a DTN at inference time. Nonetheless, it will become apparent in chapter 4 that the encoder network of a VTN can still serve a useful purpose, beyond that of guiding the training process.

Hybrid weight sharing

Reynaud et al. [4] use identical neural networks to model both the factual and counterfactual mechanism for a given variable, this is evident in Equation 3.3 and Equation 3.4, and serves to incentivise the sharing of general visual features between the factual and counterfactual image. On the other hand, Vlontzos et al. [2] setup their DTN implementation using separate networks for each branch; indeed, when the treatment is partly or fully categorical (e.g. swelling vs. fracturing or treated vs. untreated patient), the architectural separation permits specialising a single network branch to each treatment. Since both treatment specialisation and sharing of common features appear to be desirable properties for the particular image generation task tackled in this chapter, a hybrid approach was adopted in the implementation of the proposed VTN. It is inspired by the structure of the TARNET [64] architecture, which was designed by Shalit et al. to perform the distinctively causal task of estimating a *conditional average treatment effect* (CATE)⁴ [65]: it uses a single neural network for processing the shared covariates and then forwards its output through two distinct branches, one for each treatment, in order to generate the final outcomes; additionally, it is easily generalisable to more than two treatments by adding more branches, as done in [66]. Thus, a similar structure was used in the VTN implementation of this project, by defining the decoder networks d_I, d_{I^*} for the image variables in Figure 3.7 as:

$$I := d_I(U_I, T, C) \triangleq \text{branch}([T, \text{trunk}([U_I, C])]) \quad (3.6)$$

$$I^* := d_{I^*}(U_I, T^*, C) \triangleq \text{branch}^*([T^*, \text{trunk}([U_I, C])]) \quad (3.7)$$

where trunk , branch , and branch^* are arbitrary neural networks of appropriate input-output dimensions, and $\text{trunk}([U_I, C])$ is only computed once per forward pass.

VQ-VAE prior

Much like the DTN in Reynaud et al. [4], the VTN implemented for the generative task of interest leverages the excellent image compression capacity of a trained VQ-VAE architecture [37] to reduce by an order of magnitude the dimensionality of the images to be generated, thus easing the sampling process [67] and allowing for the computation of distances between the observed factual image and the sampled factual images to occur in the latent space during the conditioning step. From a probabilistic point of view, it can be argued here that the role

⁴In a medical trial, the CATE corresponds to the expected difference in outcome between the treated group and the untreated control group, conditional on certain covariates being shared amongst individuals in both groups.

played by the twin network in such an approach is that of a (joint) counterfactual sampling prior over the VQ-VAE latent space, which is similar to the role of the observational sampling priors typically learnt over a trained VQ-VAE for unconditional or class-conditional image generation [67], but which belongs to a higher rung on the Ladder of Causation. In particular Reynaud et al. [4] train the DTN to target the continuous latent space $\mathbb{R}^{N \times D}$, where the VQ-VAE codewords lie, by minimising MSE; however, motivated by the view of the twin network as a sampling prior and given that the typically trained priors for VQ-VAEs are categorical [37, 67], as they enable sampling from the even more compressed discrete latent space of indices $\{0 \dots K - 1\}^N$ within the VQ-VAE codebook, both the continuous and the categorical alternatives are benchmarked in the evaluation section for this chapter to establish the more performant approach for the task of interest.

Requirements, assumptions, and limitations

Some important requirements and assumptions that underlie the training process outlined so far can limit the applicability of the VTN approach. In fact, the training of a VTN – just like that of a DTN [2] – requires a dataset which contains ground truth counterfactual samples. While these may be available if one constructs a synthetic dataset, as it has been done in this project, real world counterfactuals are generally unobservable. Nonetheless, under certain assumptions, modelling of a real world process for counterfactual inference is still possible even if only observational data is available by imputing appropriate counterfactual samples from the available data. For example, Vlontzos et al. [2] use the Perfect Match [66] algorithm to perform these imputations and successfully train a DTN on observational datasets.

Additionally, the modelling of SCMs via neural networks is reliant on the *unconfoundedness* assumption [1, 2, 66] that there are no unobserved confounders of the causal effect of the treatment on the outcome (e.g. no unobserved variable can be a parent of both the treatment and the image in the causal diagram for the image generation problem being modelled). Moreover, the identifiability of counterfactual quantities from observational ones may not be guaranteed in many real world settings, so that the level of trustworthiness of the produced counterfactuals cannot be easily established other than by expert knowledge.

Finally, while a VTN (or a DTN) trained for a low-dimensional categorical task can easily condition on the factual outcome by rejecting sampled factual-counterfactual pairs with a factual outcome which does not match the observed one [2], when the data is continuous or categorical but high-dimensional, such as is the case for the task of interest, the implemented VTN follows [4] in the simple heuristic of sampling a large number of pairs and picking the pair with the closest factual sample. Future work may therefore wish to investigate the use of a more principled sampling method to establish when the factual side of a sampled pair is sufficiently close to be accepted as a valid sample from the conditional distribution.

3.4 Experiments

3.4.1 Models

All models in this project are implemented via the PyTorch [68] machine learning library for Python. The vector quantiser employed by the VQ-VAE is extended from the vector-quantize-pytorch [69] library, while the VTN is implemented in Pyro [70], a probabilistic

programming framework for PyTorch.

Architecture

Both the VQ-VAE and the VTN use fully convolutional architectures, leveraging throughout the powerful version of residual blocks [71] proposed for the encoder network in [72] and altered to use the Mish [73] activation function. The version of the VTN prior over the VQ-VAE continuous latent space is trained by setting the two output distributions P_θ, P_{θ^*} in Equation 3.5 to be independent Gaussians, each of which has its mean parametrised by the corresponding decoder network’s output and variance equal to the MSE across the entire mini-batch between the ground truth latent vectors and the decoder outputs, thus bypassing the need for manual tuning of the variance hyperparameter and without reducing performance, as demonstrated in Rybkin et al. [74]. On the other hand, the categorical version of the VTN prior simply has a Softmax layer added to the output of each decoder network, with the output distributions in Equation 3.5 being made categorical. In both versions, the noise variable U_I is assumed to lie in \mathbb{R}^{16} , and the variational distribution Q to abduce it from the other variables is a Gaussian whose mean and variance are parametrised by the encoder network.

Hyperparameters

The VQ-VAE is first trained for 1,000 epochs with a batch size of 512, using the Adam [75] optimiser set to a learning rate of 5e-4 (as seen in [76]); the number of codewords in the codebook is set to 256 while their dimensionality is 8 so the codebook $C \in \mathbb{R}^{256 \times 8}$; the size of a discrete latent vector encoding an image is 64. After the VQ-VAE is trained, the VTN prior trains for 5,000 epochs with a batch size⁵ of 256, using the Adam optimiser with a learning rate of 2e-4. For both models, only the checkpoint with the lowest validation loss is kept. When performing counterfactual image generation at inference, the number of joint samples for the VTN is set to 2^{14} .

3.4.2 Methodology

To quantitatively assess the proposed VTN model for counterfactual image generation, the generative task detailed in subsection 3.2.3 is performed over the entirety of the 10,000 items test set: for each item $\langle i_n^{swell}, i_n^{frac}, t_n^{swell}, t_n^{frac}, c_n \rangle$, the swollen image is taken as the factual side, and a fractured counterfactual image is generated by sampling joint factual-counterfactual VQ-VAE latent vectors pairs z, z^* from the VTN-implied counterfactual distribution as described in section 3.3.3, picking the pair z_{pred}, z_{pred}^* with z_{pred} closest to the latent vector for i_n^{swell} . Similarity between images is measured using the well-established structural similarity index measure (SSIM) [77]. As in [4], the SSIM between the ground truth image i_{gt} ($i_{gt} = i_n^{swell}$ for the factual side and $i_{gt} = i_n^{frac}$ for the counterfactual) and its reconstruction i_{rec} through the VQ-VAE is reported to provide a measure of the reconstruction quality on both swollen and fractured images, denoted as $\text{SSIM}(i_{gt}, i_{rec})$; notice that such metric is independent of the VTN and only measures the VQ-VAE performance. Then, the quality of the images $i_{pred}^{swell}, i_{pred}^{frac}$ decoded from the predicted pair z_{pred}, z_{pred}^* is evaluated by measuring their respective similarity with the ground truths images i_n^{swell}, i_n^{frac} , denoted as

⁵Every item in a batch for the VTN contains both a factual and a counterfactual image, so that the number of images seen on a forward pass by each model is identical.

$\text{SSIM}(i_{gt}, i_{pred})$, and with the VQ-VAE reconstructed ground truths, denoted as $\text{SSIM}(i_{rec}, i_{pred})$. $\text{SSIM}(i_{rec}, i_{pred})$ gives an indication of the capacity of the VTN prior to sample credible factual-counterfactual pairs over the VQ-VAE latent space, while the $\text{SSIM}(i_{gt}, i_{pred})$ is given to provide a more complete picture of the loss in quality relative to the ground truths, as it implicitly accounts for the performance loss due the VQ-VAE reconstruction error.

Qualitative evaluation was performed by visually inspecting the generated samples for a number of tuples $\langle i_n^{swell}, i_n^{frac}, t_n^{swell}, t_n^{frac}, c_n \rangle$, displaying both the sampled factual-counterfactual pair with the factual side closest to the ground truth factual outcome, together with a number of samples from the joint distribution which are not conditioned on the factual side.

3.4.3 Results

Ground truth (factual above counterfactual)	Sample with closest factual outcome	Random samples from joint
		
		
		
		
		
		

Table 3.1: Samples from VTN: each factual-counterfactual pair is laid out vertically, with the factual (swollen) sample immediately above the counterfactual (fractured) sample. The VTN inference algorithm only sees the ground truth swollen images from the first column, the treatments, and the covariates; the goal is to generate a sample with the closest factual outcome (second column) such that its fractured image is very similar to the fractured image of the ground truth pair.

From Table 3.1, it is evident that the samples with the closest factual outcome show credible factual and counterfactual images, which are very similar to their respective ground truths. Despite not being surprising that the factual image of the best sample is similar to the

ground truth factual image, as such similarity is precisely the selection criterion for the sample, it is however not trivial that the corresponding sampled counterfactual also closely matches the (unobserved) counterfactual ground truth. One reason for the apparent success in the generation of credible counterfactuals likely resides in the capacity to sample factual-counterfactual pairs that appear to have been generated by a perturbation of the same original image. In fact, such property can be observed throughout each pair of random samples from the joint distribution, with the factual image above always following a very similar structure to the counterfactual below it. Moreover, the diversity that still exists between different pairs of random samples (see e.g. the distinct styles in the row for digit 4) reinforces the hypothesis that the similarity between pairs is not exclusively brought about by their shared covariates, but is instead accounted for by the exogenous state of the world u_I at the moment of sampling. Lastly, it can be seen that the information encoded within treatment vectors is largely respected, with the locations of the swelling and fractures being similar across the pairs of samples and consistent with the ground truth pair.

Metric	SSIM(i_{gt} , i_{rec})		SSIM(i_{rec} , i_{pred})		SSIM(i_{gt} , i_{pred})	
Side	factual	counterfact.	factual	counterfact.	factual	counterfact.
DTN [4]	0.9308	0.9308	0.6759	0.6759	0.6707	0.6705
Continuous-VTN	0.9929	0.9923	0.7447	0.7339	0.7438	0.7331
Categorical-VTN	0.9929	0.9923	0.7987	0.7909	0.7979	0.7900

Table 3.2: Quantitative results of VTN evaluation. The semantics of each metric are described in subsection 3.4.2. Results for the DTN model are taken from [4]. Direct performance comparisons should only be made between the two VTN models: the DTN’s metrics for a similar task are only provided for reference.

The quantitative results shown in Table 3.2 reinforce the suitability of the VTN to the image generation task of interest, with SSIM scores between ground truth pairs and predictions (SSIM(i_{gt} , i_{pred})) of 73.31% in the continuous case and 79% in the categorical one; such scores also demonstrate the superiority of the categorical approach compared to the continuous one when a VTN is used as a prior over the VQ-VAE latent space: this could be a result of both the reduced dimensionality and the fundamentally categorical nature of the VQ-VAE latent space. Additionally, the high reconstructions scores (SSIM(i_{gt} , i_{rec})) of the VQ-VAE on which the VTN is trained indicate that the loss of information caused by the compression step of the VQVAE is minimal. Finally, it is important to note that the scores obtained by Reynaud et al. [4] are given in Table 3.2 solely for the purposes of exemplifying what a reasonable benchmark score on a similar task may be but they should not be used to definitively assert the superiority of the VTN model.

Chapter 4

Counterfactual content-based image retrieval

In this chapter, the task of counterfactual content-based image retrieval (CCBIR) is defined. To tackle it, two methods for the execution of CCBIR queries are proposed and evaluated on a synthetic dataset.

4.1 CCBIR

4.1.1 Definition

Given

1. an SCM $\mathcal{M} = \langle U, P(U), V, F \rangle$ that describes the generative process for images from the distribution $P_{\mathcal{M}}(I)$, where $I \in V$;
2. a gallery dataset of images $\mathcal{G} = \{i_n\}_{n=1}^N$;
3. a user-submitted query, in the form of a tuple of assignments

$$q = \langle I := i, T := t, T := t^*, C := c \rangle$$

where i is the factual image, t is the factual treatment, t^* is the counterfactual treatment and c are the covariates, and they satisfy the same conditions as in section 3.1;

the ideal system for counterfactual content-based image retrieval (CCBIR) should return the list of gallery images sorted in descending order of some relevance measure $rel_q(\cdot)$ that induces the same ordering over dataset images as the ground truth relevance $rel_q^{gt}(\cdot)$. In particular $rel_q(\cdot)$ must be any function such that, for any two images i_a, i_b from \mathcal{G} ,

$$rel_q(i_a) \geq rel_q(i_b) \iff rel_q^{gt}(i_a) \geq rel_q^{gt}(i_b) \quad (4.1)$$

where the ground truth relevance of a dataset image i_n to the query q is the probability that the image I , produced when $C = c$, would have been i_n (instead of i), had T been t^* (instead

of t), i.e. the counterfactual quantity

$$rel_q^{gt}(i_n) \triangleq P_m(I_{T:=t^*} = i_n | T = t, C = c, I = i) \quad (4.2)$$

4.1.2 Method 1: ELBO rank

Equivalent relevance scores

Noting that

1. an exact solution to the CCBIR task defined above does not necessarily require the estimation of $rel_q^{gt}(i_n)$, as any relevance assigning function $rel_q(\cdot)$ that induces the same ordering on dataset images as $rel_q^{gt}(\cdot)$ will yield a correct ranking;
2. the ground truth relevance 4.2, which is a counterfactual quantity in the SCM \mathcal{M} , is identical to the observational quantity $P_{m_{T^*:=t^*}}^{TN}(I^* = i_n | T = t, C = c, I = i)$ in the submodel $m_{T^*:=t^*}^{TN}$ of \mathcal{M} 's twin network;
3. the ordering over counterfactual images implied by $i^* \mapsto P_{m_{T^*:=t^*}}^{TN}(I^* = i^* | T = t, C = c, I = i)$ is identical to that implied by $i^* \mapsto P_{m_{T^*:=t^*}}^{TN}(I = i, I^* = i^* | T = t, C = c)$, as for any two images i_a, i_b from \mathcal{G}

$$\begin{aligned} P(I^* = i_a | T = t, C = c, I = i) &\geq P(I^* = i_b | T = t, C = c, I = i) \\ \iff \frac{P(I = i, I^* = i_a | T = t, C = c)}{P(I = i | T = t, C = c)} &\geq \frac{P(I = i, I^* = i_b | T = t, C = c)}{P(I = i | T = t, C = c)} \\ \iff P(I = i, I^* = i_a | T = t, C = c) &\geq P(I = i, I^* = i_b | T = t, C = c) \end{aligned}$$

where all distributions are taken in the submodel $m_{T^*:=t^*}^{TN}$;

then, setting $rel_q(i^*) \triangleq P_{m_{T^*:=t^*}}^{TN}(I = i, I^* = i^* | C = c, T = t)$ leads to an ordering identical to that implied by $rel_q^{gt}(\cdot)$ so that, if one could compute $rel_q(\cdot)$ on each dataset image, the CCBIR task would be solved exactly.

ELBO rank

Recall that a variational twin network permits sampling from a distribution $P_\theta(I = i, I^* = i^* | C = c, T = t)$ that approximates $P_{m_{T^*:=t^*}}^{TN}(I = i, I^* = i^* | C = c, T = t)$. While estimation of the value of the distribution at a point is intractable (for similar reasons to those discussed in section 2.3.1), the estimation of the ELBO is both tractable and a required step during the training process of the VTN. Therefore, an approximate solution to the CCBIR task is to rank the images in the dataset by decreasing order of a Monte-Carlo estimate of their ELBO:

$$\widetilde{rel}_q(i^*) \triangleq \mathcal{L}_{\theta, \theta^*, \psi}(i, i^*, t, t^*, c) \quad (4.3)$$

where $\mathcal{L}_{\theta, \theta^*, \psi}$ is defined like in Equation 3.5. The quality of such an approximation depends on a number of factors, which may introduce errors in the ELBO-implied ordering:

1. how well the VTN’s distribution approximates that of $m_{T^*:=t^*}^{TN}$;
2. the tightness of the bound obtained by the ELBO;
3. the error of the Monte-Carlo estimate for the ELBO.

While the first 2 factors are dependent on the VTN’s architecture and training setup, the error of the Monte-Carlo estimate can be minimised at query time by taking more samples. During training, a single sample is usually sufficient, as the ELBO is averaged across a batch, which reduces the variance. However, when performing CCBIR using 4.3, a sufficiently accurate estimate of the ELBO is needed for each data point: depending on the dataset and architectural setup, a higher number of Monte-Carlo samples may therefore be beneficial. Further, note that the ELBO estimation step of this method requires for the encoder networks used during the training of the VTN to be available after the training, which is not a requirement if one is only interested in counterfactual image generation.

4.1.3 Method 2: Sample-similarity rank

Sample and retrieve

The outlined ELBO-based ranking strategy can be exceedingly computationally expensive, as the ELBO estimation process has to be run on every item in the dataset for each new query, which increases the latency experienced by the end-user of the CCBIR pipeline. While the logic is easily parallelisable to a multi-node GPU cluster, where the VTN model is replicated and the gallery dataset is sharded across the memory of multiple GPUs and nodes, such a setup may not be viable for use cases with more limited resources or stricter power consumption targets. A more efficient alternative method to tackle the defined CCBIR task is to split it into two subtasks for which the literature has already provided effective solutions: counterfactual image generation and content-based image retrieval. In particular, given a query $q = \langle I := i, T := t, T := t^*, C := c \rangle$, one can first leverage a counterfactual generative model to sample a high probability counterfactual image $i^* \sim P_m(I_{T:=t^*} | C = c, T = t, I = i)$ and then use a standard content-based image retrieval pipeline to rank images in the gallery dataset \mathcal{G} according to a graded relevance score with respect to i^* , $rel_{i^*}(\cdot)$, based on a measure of visual similarity with i^* .

Visual similarity and counterfactual similarity

Under the assumption that the more visually similar two images are, the closer to each other their probabilities of being a counterfactual to q will be, one expects that, for any two images i_a, i_b from \mathcal{G} ,

$$rel_{i^*}(i_a) \geq rel_{i^*}(i_b) \implies |rel_q^{gt}(i^*) - rel_q^{gt}(i_a)| \leq |rel_q^{gt}(i^*) - rel_q^{gt}(i_b)| \quad (4.4)$$

In other words, the more relevant an image i_a is to the sampled counterfactual i^* , according to the simple CBIR framework, the smaller will be the difference between the relevance scores of i^* and i_a to the query q , according to the full CCBIR framework. If i^* is indeed likely to be a counterfactual for i , such reasoning motivates ranking images in \mathcal{G} on the basis of their similarity to i^* , as highly similar images will also be likely counterfactuals. Effectively, the sample-similarity ranking method presented also assumes that two visually dissimilar images will not both have high probability of being valid counterfactuals; if such assumption were

not satisfied, a strategy would have to be devised in order to appropriately weigh the different relevance scores induced by multiple highly probable but visually dissimilar samples, which is left as future work.

VQ-VAE latent vectors as feature embeddings

The proposed sample-similarity ranking method has been implemented by using a variational twin network as the generative model to obtain a high probability counterfactual i^* in response to a query q . The subsequent similarity-based retrieval step could have been performed by submitting i^* as a query image to an existing off-the-shelf CBIR framework. However, noting that

1. the latent vectors produced at the bottleneck of autoencoding architectures are frequently used as compact feature embeddings for self-supervised image retrieval [78, 79, 80];
2. the VTN pipeline described in chapter 3 already relies on a VQ-VAE to reduce the dimensionality of the input images;
3. the outputs of the VTN lie precisely in the latent space of the VQ-VAE bottleneck;

it can be more computationally and memory efficient to use those same latents as feature embeddings for image retrieval, relying on the similarity between embeddings as a measure of content similarity between the images they encode, as opposed to decoding the latent into image space and then using a separate system for retrieval with its associated additional overhead. In the light of this and using the same notation of section 2.3.4, the relevance score was set to the negated squared Euclidean distance on the quantised latents of the VQ-VAE, i.e.

$$rel_{i^*}(x) \triangleq -||vq_C(e_\psi(i^*)) - vq_C(e_\psi(x))||^2$$

where $||vq_C(\cdot) - vq_C(\cdot)||^2$ is computed in the particularly efficient form discussed in section 2.4.3.

4.2 Experiments

4.2.1 Dataset composition

The evaluation procedure for a standard CBIR pipeline normally involves a strict separation between the query set and the gallery set, which are constructed to be non-overlapping. Consequently, when a query image is looked up in the gallery set, the CBIR system will never return an exactly identical image but only similar images. Indeed, it is not particularly useful to test the capacity of a CBIR system to retrieve a query image already contained in the gallery dataset, as any hashcode-based retrieval system will trivially find the gallery image with a hashcode that exactly matches the hashcode of the query image, and thus return that image as the top result. However, in the case of a CCBIR pipeline, even if a ground truth image with the highest probability of being the counterfactual to the query were contained within the gallery dataset, it would not necessarily be trivial to return it as the top result. This is because any real system that attempts to approximate the mechanisms of an SCM may incur an error in the estimation of the counterfactual relevance for the gallery images

(Equation 4.2) that is sufficiently large to produce an incorrect ranking. In fact, the capacity to consistently retrieve such a ground truth image as the top result would increase confidence in the quality of the relevance estimations performed by the CCBIR system. The practical consequence of such reasoning on the evaluation methodology followed in this project is that, while the factual image submitted in the query is not present in the gallery dataset, the corresponding counterfactual ground truth image is.

In particular, the capabilities of the CCBIR methods proposed have been tested using the same MorphoMNIST-based synthetic dataset constructed for the counterfactual image generation task in chapter 3: 70,000 pairs of images, where each pair contains 1 swollen image and 1 fractured image with associated treatment information (the locations and sizes) of the swelling and fracturing, alongside the metrics of the unperturbed original image, i.e. $\mathcal{D} = \{\langle i_n^{swell}, i_n^{frac}, t_n^{swell}, t_n^{frac}, c_n \rangle\}_{n=1}^{70,000}$. Such dataset has been adapted for the CCBIR task by constructing 2 gallery datasets $\mathcal{G}^{swell} = \{i_n^{swell}\}_{n=1}^{70,000}$ and $\mathcal{G}^{frac} = \{i_n^{frac}\}_{n=1}^{70,000}$, containing respectively all the swollen images and all the fractured images from \mathcal{D} , and 1 query dataset $\mathcal{Q} = \{\langle i_n^{swell}, i_n^{frac}, t_n^{swell}, t_n^{frac}, c_n \rangle\}_{n=1}^{1,000}$, built by randomly choosing 100 items for each digit from the portion of \mathcal{D} that was reserved as a test set for the variational twin network.

4.2.2 Models

During the evaluation process, the same categorical-VTN model trained in section 3.4 with its associated VQ-VAE was employed to test both the ELBO-ranking method and the sample-similarity method. The number of Monte-Carlo samples per dataset image during the estimation of the ELBO required by the proposed ELBO method was set to 4. While larger sample sizes may be beneficial, this hyperparameter has a multiplicative effect on the latency time of a query, and small sample sizes were observed to be sufficient to obtain good performance on the CCBIR task. On the other hand, when employing the sample-similarity based ranking, the number of joint samples used to generate a single counterfactual image conditioned on the factual image was left to the much larger value of 2^{14} , as the cost of sampling is only paid once per query instead of once per dataset image on every query, and, unlike the ELBO-based method, the overhead of using the encoder network is not present.

4.2.3 Methodology

The two CCBIR methods proposed were tested both in their capacity to retrieve counterfactual fractured images from \mathcal{G}^{frac} , given a query containing a factual swollen image, and in their capacity to retrieve counterfactual swollen images from \mathcal{G}^{swell} , given a query containing a factual fractured image. In the following discussion, only the former case is considered, as in the latter case the evaluation process is completely symmetric.

The CCBIR pipelines for gallery dataset \mathcal{G}^{frac} have been benchmarked by submitting, for each item $\langle i_n^{swell}, i_n^{frac}, t_n^{swell}, t_n^{frac}, c_n \rangle$ in the query dataset \mathcal{Q} , the query $q = \langle I := i_n^{swell}, T := t_n^{swell}, T := t_n^{frac}, C := c_n \rangle^1$. The ranked list returned in response to each query q was used to compute the following metrics:

1. the hit rate (HR) at different positions in the list, where $\text{HR}@k$ is 1 if i_n^{frac} appears

¹The exclusion of i_n^{frac} from query q is deliberate. i_n^{frac} is the ground truth counterfactual to i_n^{swell} , i.e. the image to be retrieved and, accordingly, it is used to compute the evaluation metrics for the query.

before or at position k in the returned list, and is 0 otherwise;

2. the reciprocal rate (section 2.4.5), with i_n^{frac} being taken as the only relevant result;
3. the $\text{NDCG}_{\text{SSIM}}$ metric², where the SSIM values necessary to produce a reference ranking are computed between each image in \mathcal{G}^{frac} and the ground truth counterfactual image i_n^{frac} , with the NDCG being measured between the SSIM-implied ranking and the predicted ranking;
4. the standard AP@All and AP@1000 metrics, where a returned image is considered relevant to the query if it has the same digit.

Such metrics were then averaged across queries to obtain the results in Table 4.2. It is important to note that the mAP metrics are not indicative of performance on the CCBIR task, as the notion of relevance used to compute them is not necessarily related to the counterfactual probability in Equation 4.2; nonetheless, they have been included in the evaluation to give a more complete picture of the retrieval characteristics of the benchmarked methods.

4.2.4 Results



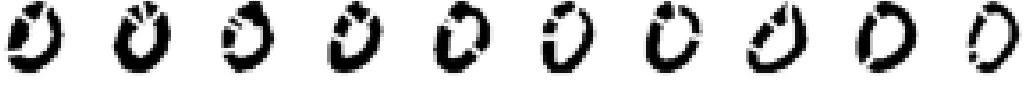


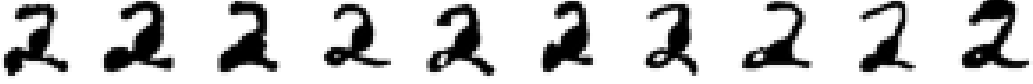












Submitted factual image	Ground truth counterfactual	Top-10 retrieved images
		
		
		
		
		
		

Table 4.1: Examples of queries submitted to a CCBIR pipeline using the CBIR-VQ-VTN model. The left column contains the factual image of the query tuple that was submitted by the user; the central column shows the ground truth counterfactual image that should be returned as the top result; the top-10 retrieved results are displayed on the right.

Table 4.1 shows the top-10 results for a number of queries submitted to the CCBIR system: the correct counterfactual gets consistently retrieved as the top result, while remaining results

²The $\text{NDCG}_{\text{SSIM}}$ metric is introduced in more detail and its purpose is further clarified in section 5.3.4.

in the top-10 lists look very similar to the ground truth counterfactual, increasing confidence in the performance of the system, beyond the capacity to retrieve the ground truth as the first result.

Setup				
Method	CBIR-VQ-VTN		ELBO-VQ-VTN	
Factual	swell	fracture	swell	fracture
Counterfactual	fracture	swell	fracture	swell
Results (%)				
HR@1	98.9	98.3	99.6	99.1
HR@5	99.3	99.4	99.8	99.8
HR@10	99.6	99.5	99.8	100.0
MRR	99.13	98.82	99.71	99.45
NDCG _{SSIM}	99.87	99.89	99.61	99.71
mAP@All	26.74	25.21	24.78	23.71
mAP@1000	75.59	69.86	74.86	68.20

Table 4.2: Benchmark results of different CCBIR methods: the similarity based method is labelled CBIR-VQ-VTN while the ELBO based method is denoted as ELBO-VQ-VTN. The factual row indicates which image is submitted as the factual query image, while the counterfactual row indicates both the counterfactual treatment and the gallery dataset used for retrieval.

From the hit rates and mean reciprocal ranks collected in Table 4.2, it is evident that all the proposed methods are largely successful at retrieving the ground truth counterfactuals for the query images, with hit rates at 1 being above 97% across all models. The hit rates and MRR of the ELBO based model are superior to the sample-similarity method, which is unsurprising, considering that ELBO estimation is performed for each item. However, it is striking how close the sample-similarity method is to the performance of the more resource-intensive ELBO method. Such results are encouraging for use cases with limited computational resources, as they indicate that a substantial drop in computational requirements may be obtained for only a small loss in retrieval performance. Furthermore, it can be observed that the similarity ranking method achieves slightly higher scores on the NDCG_{SSIM}, mAP@All, and mAP@1000 metrics; while these are not indicative of CCBIR performance, they suggest that relevance estimation based on the similarity of extracted features embeddings remains a more appropriate proxy to the classical definitions of image relevance. It is left as future work to ascertain whether the above insights transfer to more diverse and challenging CCBIR tasks.

Chapter 5

VQ-VAE training for image retrieval

The sample-similarity based ranking method for CCBIR proposed in subsection 4.1.3 relies on the ‘meaningfulness’ of distance computations in the VQ-VAE latent space in order to establish similarity between images and, ultimately, relevance to a query. Noting that number of improvements and tricks [81, 76, 82] in the training of VQ-VAE architectures have been proposed in recent years mainly to improve reconstruction quality and training dynamics, this chapter investigates whether a selection of such techniques can also be used to produce more meaningful distance computations on the embeddings generated by VQ-VAEs for a boost in image retrieval performance.

5.1 Retrieval task

5.1.1 Dataset

Since the results of the investigations in this chapter may have generic applicability beyond the context of CCBIR and potentially benefit any image retrieval method reliant on vector quantisation, it was deemed appropriate to first conduct benchmarks on a standard CBIR task, so as to permit comparison with existing solutions, and then assess whether any performance improvement gained thereby would also transfer to a CCBIR task. Given the clear similarity between the MorphoMNIST dataset, which this project employs for counterfactual inference, and the original MNIST dataset, which is often used to benchmark unsupervised CBIR frameworks, the retrieval performance for standard CBIR was initially optimised and assessed on MNIST.

5.1.2 Dimensionality constraints

An important constraint imposed by researchers [83, 79, 13] when benchmarking unsupervised CBIR pipelines on simple datasets (such as MNIST or CIFAR-10 [84]) consists in severely limiting the size of the feature embeddings by enforcing that the number of bits they encode be, for example, 32, 48, or 64; for this reason, such embeddings are often referred to as *hash codes*, as opposed to the generally unconstrained embeddings discussed in subsection 2.4.4 and typically used for supervised image retrieval on more challenging datasets. For the purpose of comparative evaluation, the immediate consequence of such a

size constraint is that the VQ-VAE architecture defined in chapter 3 should not be benchmarked as is, because the 64 dimensional discrete latent vector of indices into a codebook of 256 codewords contains $64 \times \log_2 256 = 512$ bits of data, which would constitute an unfair comparison with solutions using smaller embeddings. The straightforward remedy is to alter the architecture to reduce the size of the latent space, e.g. to an 8-dimensional vector with a codebook of 256 codewords for the 64-bit setting, at the cost of increasing the reconstruction error.

5.2 Training techniques

5.2.1 Normalised codewords

In the context of generative image models, Yu et al. [82] have recently observed improved training stability and higher reconstruction quality when continually performing l_2 -normalisation on both the codewords $\{C_k\}_{k=1}^K$ in the codebook $C \in \mathbb{R}^{K \times D}$ and the encoder output $e_\psi(x)$ before the nearest neighbour search step. This is equivalent to guaranteeing that the points represented by such vectors are placed on a D -dimensional hypersphere of unit radius, with the effect that minimising the squared Euclidean distance also minimises the cosine distance. Since the cosine distance is itself inversely proportional to the cosine similarity, minimising both the commitment error and the quantisation error (defined in section 2.3.4) under l_2 -normalisation becomes equivalent to maximising the cosine similarity between the encoder outputs and the closest codewords. Such reasoning is easy to confirm as, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$,

$$\begin{aligned} \left\| \frac{\mathbf{x}}{\|\mathbf{x}\|} - \frac{\mathbf{y}}{\|\mathbf{y}\|} \right\|^2 &= \left(\frac{\mathbf{x}}{\|\mathbf{x}\|} - \frac{\mathbf{y}}{\|\mathbf{y}\|} \right)^T \left(\frac{\mathbf{x}}{\|\mathbf{x}\|} - \frac{\mathbf{y}}{\|\mathbf{y}\|} \right) \\ &= \frac{\mathbf{x}^T \mathbf{x}}{\|\mathbf{x}\|^2} + \frac{\mathbf{y}^T \mathbf{y}}{\|\mathbf{y}\|^2} - 2 \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \\ &= 2 \underbrace{\left(1 - \overbrace{\cos(\angle(\mathbf{x}, \mathbf{y}))}^{\text{cosine similarity}} \right)}_{\text{cosine distance}} \end{aligned}$$

where $\angle(\mathbf{x}, \mathbf{y})$ denotes the angle between \mathbf{x} and \mathbf{y} .

Other than the reduction in reconstruction error that this technique can bring – which may on its own be sufficient to positively affect retrieval precision –, there is a second potential advantage specific to image retrieval: the cosine distance becomes both the metric optimised during training and the measure of distance between the quantised latent of the query image and that of each image in the dataset. Since it has been observed [85, 86, 87] that the cosine distance can outperform Euclidean distance in retrieval precision, l_2 -normalisation may constitute an additional improvement for some datasets and retrieval pipelines. Therefore, its effect on the CBIR pipeline of this project has been explored.

5.2.2 PQ-VAE

The insertion of a product quantiser as a bottleneck to autoencoding architectures has seen frequent adoption in recent years [78, 79, 80], especially for the purposes of unsupervised image retrieval via compact hashcodes, due to its capacity to generate a very large codebook at a fraction of the embeddings size and the memory cost that would be required for a

standard vector quantiser (as detailed in section 2.4.3). Building on the vanilla VQ-VAE model for image generation, Wu and Flierl [13] proposed learning the codebook of a product quantiser online, via EMA updates to the codebook of each subquantiser, thus replacing the standard vector quantiser and yielding a product quantised-variational autoencoder (PQ-VAE) fit for image retrieval via hashcodes. Given the dimensionality constraints discussed in subsection 5.1.2, the utility of this architectural change to the CBIR framework proposed has been investigated during the project.

5.2.3 Noisy codeword assignment

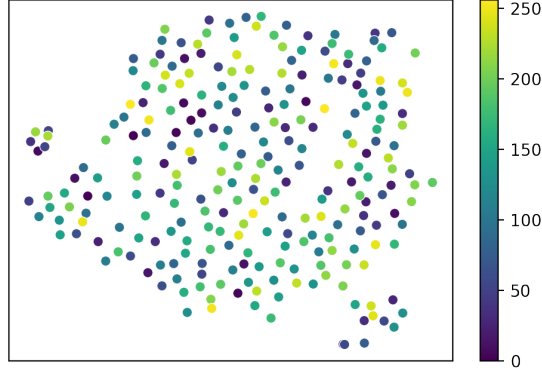


Figure 5.1: TSNE visualisation of all 256 codewords from a standard VQ-VAE: no discernible structure is present. The colour bar on the right indicates the codeword index.

Coppock and Schuller’s method

Coppock and Schuller [11] note that, while the effort to push the variational posterior towards the sampling prior during the training of a standard VAE promotes *continuity* in the latent space (i.e. the property that points which are close to each other in the latent space are decoded to similar instances in the data space), the vanilla VQ-VAE architecture does not exhibit such continuity and, accordingly, leads to the learning of codewords whose arrangement in the latent space appears to be largely random and lacking of a discernible semantic structure; see e.g. Figure 5.1.

To the end of increasing the accuracy of a classifier on a downstream classification task using the VQ-VAE as a feature extractor and the quantised latents as input features to the classifier, Coppock and Schuller proposed a noise injection method to promote continuity as well as a more structured arrangement of codewords in the quantised latent space. Given a discrete vector of indices $z = \text{nearest}_C(e_\psi(x)) \in \{0, \dots, K-1\}^N$ produced by the VQ-VAE on an input x , noise is injected by randomly reassigning each index $z_n \in \{0, \dots, K-1\}$ to a new index $z'_n = z_n + \epsilon$ where ϵ has been sampled from a discrete Gaussian $N_{\mathbb{Z}}(0, \sigma^2)$. This is equivalent to sampling $z'_n \sim N_{\mathbb{Z}}(z_n, \sigma^2)$ and means that the probability that z_n is assigned to a particular index z'_n decreases as the difference $|z_n - z'_n|$ between indices increases, so that the most likely value for z'_n is z_n (the index is unchanged), while the neighbours of z_n , i.e. $z_n \pm 1, z_n \pm 2, z_n \pm 3 \dots$, become progressively less likely values for z'_n .

The effect of Coppock and Schuller’s method is that codewords whose indices are close to each other in value will also have higher probability of being swapped with each other during training so that, for any two codewords $C_i, C_j \in \mathbb{R}^D$ such that $|i - j|$ is small, 1) the EMA

updates will bring C_i closer to C_j in the continuous latent space, as a consequence of the fact that the exponential moving average of the i -th codeword will be occasionally updated with encoder inputs which, despite being closer to C_j than they are to C_i , have nonetheless been assigned to the i -th codeword due to the noise injected; 2) the decoder network will be incentivised to produce decodings of C_i and C_j that are close in the data space, as a consequence of the effort to minimise the reconstruction error even when occurrences of i or j in the discrete latent vector of indices are swapped with each other by the noisy assignments. The hope is that these two processes will produce a correspondence between similarity in the continuous latent space and similarity in the data space, i.e. continuity.

Because the property of continuity appears aligned with the goal of more meaningful distance computations for the purposes of image retrieval, the effect of noisy codeword assignment during quantisation has been implemented and evaluated as a possible avenue for improvement during the development of the CBIR framework of this project.

Analogue clock

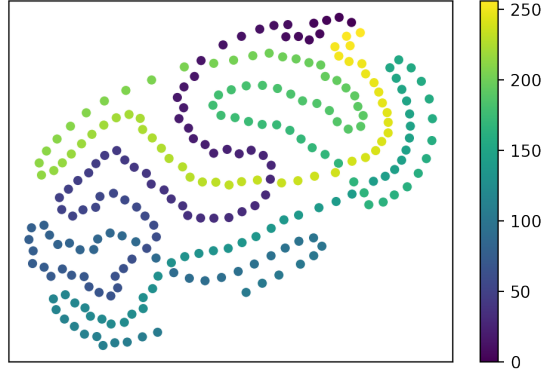


Figure 5.2: TSNE visualisation of all 256 codewords from a VQ-VAE trained via Coppock and Schuller’s noise injection method. The colour bar on the right indicates the codeword index.

Since the authors do not specify how to handle cases where the sampled ϵ implies $z'_n \notin \{0, \dots, K-1\}$, in the implementation of this project the choice was made to set $z'_n = \text{reassign}(z_n, \epsilon, K)$, where

$$\text{reassign}(z_n, \epsilon, K) \triangleq \left(z_n + \text{sgn}(\epsilon) \cdot \left(\epsilon \bmod \left\lfloor \frac{K}{2} \right\rfloor \right) \right) \bmod K \quad (5.1)$$

The intuition behind such reassignment formula is that it sees the discrete indices as equally-distanced points on the circumference of a circle and, for a given point z_n , it chooses the semicircle to the left $\{i \bmod K | i \in \{z_n, z_n-1, \dots, z_n - \lfloor \frac{K}{2} \rfloor\}\}$ or to the right $\{i \bmod K | i \in \{z_n, z_n+1, \dots, z_n + \lfloor \frac{K}{2} \rfloor\}\}$ of z_n depending on whether ϵ is negative or positive, and then reassigns z_n to a point on that semicircle; since ϵ is sampled from a Gaussian, the points on the semicircle that are further away from z_n are less likely to be chosen as replacements for z_n .

This has two theoretically pleasing properties: 1) z'_n is guaranteed to be a valid codeword as $\forall z_n \in \{0, \dots, K-1\}, \forall \epsilon \in \mathbb{Z} : z'_n \in \{0, \dots, K-1\}$; 2) the probability of z'_n being chosen as the replacement for z_n is negatively correlated to a measure of distance between z'_n and z_n :

such distance is the length of the shortest path between node z'_n and node z_n in the undirected graph with nodes $\{0, \dots, K-1\}$ and edges $\{\langle 0, 1 \rangle, \langle 1, 2 \rangle, \dots, \langle K-2, K-1 \rangle, \langle K-1, 0 \rangle\}$, which pictorially corresponds to an analogue clock with K hours. Note that simpler alternatives to Equation 5.1, such as resampling ϵ whenever $z'_n \notin \{0, \dots, K-1\}$, would satisfy property 1), but violate the ‘symmetry’ implied by 2); for instance, the probability of $z_n = K-1$ being randomly reassigned to 0 would be less than the probability of $z_n = K-2$ being reassigned to $K-1$, even though the distance between $K-2$ and $K-1$ is the same as that between $K-1$ and 0 on a K -hours clock. Such differences in distributions are absent if Equation 5.1 here proposed is used.

Inspecting the TSNE visualisation of the learnt codewords for a VQ-VAE trained with the implemented noisy codewords assignments, similar results to [11] have been obtained: a ‘string’ or ‘loop’ of codewords in the latent space ordered by their indices emerges in Figure 5.2 as codewords which are closer in index are pushed towards each other more strongly by the EMA updates.

Multi-ring codeword assignment

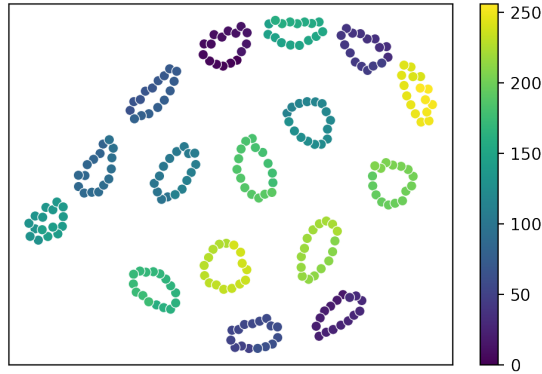


Figure 5.3: TSNE visualisation of all 256 codewords from a VQ-VAE trained with the proposed multi-ring noise injection method with a number of rings $R = 16$. The colour bar on the right indicates the codeword index.

Building on Coppock and Schuller’s method, an extension to the noisy assignment technique is proposed in this project. It is inspired by the TSNE visualisations of a number of performant solutions [79, 88] and it attempts to leverage noisy codeword assignments to artificially encourage (in addition to continuity) the *clustering* of codewords in the latent space – a common property that emerges in effective retrieval pipelines reliant on quantisation.

In particular, one partitions the set of codewords’ indices $\{0 \dots K-1\}$ into R subsets $\{G_i\}_{i=0}^{R-1}$ such that $G_i \triangleq \{iS + n | n \in \{0 \dots S-1\}\}$ where all subsets have the same size $S = K/R$ and R is a hyperparameter chosen amongst the divisors of K . Note that $\forall k \in \{0 \dots K-1\} : k \in G_{(k \bmod S)}$. Then, each scalar index z_n in a discrete latent vector of indices z can be randomly reassigned by picking a (possibly) different codeword index z'_n under the constraint that it belongs to the same subset $G_{(z_n \bmod S)}$ that z_n belongs to. In the same spirit of the implementation for the original noise injection method, such sampling is performed while enforcing that indices on the ‘analogue clock’ of S hours constructed from the elements in $G_{(z_n \bmod S)}$ are progressively less likely to be chosen as replacements for z_n the further away

they are from z_n . This can be achieved by setting

$$z'_n = gS + \text{reassign}(z_n - gS, \epsilon, S)$$

where $g \triangleq z_n \bmod S$ is the index of the subset that z_n belongs to, ϵ is sampled from $N_{\mathbb{Z}}(0, \sigma^2)$, and *reassign* is defined in Equation 5.1.

Inspecting the TSNE visualization (Figure 5.3) of the codewords resulting from a VQ-VAE trained according to the above strategy, exactly R clusters of S codewords each emerge in the latent space. Such clusters follow the same shape of a ring or loop seen in the original method, but replicate the shape in multiple locations and at a smaller scale. The original method from Coppock and Schuller can be considered a special case (obtained when $R = 1$) of this multi-ring approach.

5.2.4 Random restarts

Codebook collapse mitigation

Codebook collapse is an issue that can affect vanilla VQ-VAEs whereby a number of codewords become unused during training in favour of a restricted subset of codewords which are fully relied upon for quantisation. Consequently, the expressive capacity of the latent vector drops and reconstruction quality may be negatively impacted. The technique of *random restarts* proposed by Dhariwal et al. [12] addresses the issue by tracking the mean usages $\{U_i\}_{i=0}^{K-1}$, where U_i is an exponential moving average of the number of vectors outputted by the encoder network which have been assigned to the i -th codeword at each training step; if the mean usage U_i for any codeword falls below a pre-defined global threshold U_{min} , that codeword is evicted from the codebook and replaced with a vector randomly chosen from the output of the encoder network, thereby providing some confidence that the new codeword will see higher usage.

Threshold bounds

The original authors [12] pick a threshold value without a stated theoretical justification, presumably via trial-and-error experimentation. However, it is useful to further clarify the semantics of such threshold if one wishes to combine the technique with product quantisation, as such a combination may call for a more careful threshold selection. In particular, for a given VQ-VAE architecture and encoder network output $e_{\psi}(x) \in \mathbb{R}^{B \times N \times D}$, where B is the batch size and N the size of the discretised latent vector of a single sample, the standard vector quantiser will quantise $B \cdot N$ vectors, each of which will be assigned to one codeword, so that the maximum theoretical usage which can be simultaneously achieved across all codewords is $\frac{BN}{K}$. Therefore, the existence of at least one codeword with $U_i \leq \frac{BN}{K}$ is guaranteed in all training steps and U_{min} should be picked to be below $\frac{BN}{K}$ to avoid codewords being constantly replaced without the possibility of convergence. However, in the training of a PQ-VAE, each of the M subquantiser only acts on a chunk (in $\mathbb{R}^{B \times (N/M) \times D}$) of the encoder network output and thus quantises only $\frac{BN}{M}$ vectors, so the maximum theoretical usage simultaneously achievable is $\frac{BN}{KM}$ and U_{min} should be chosen to be below it. In summary, product quantisation calls for a lower threshold value than vector quantisation, all else being equal. In practise, such distinctions in the selection of U_{min} between vector quantisation and product quantisation can conveniently be avoided by normalising the usage

of each codeword so that it is divided by $\frac{BN}{K}$ and $\frac{BN}{KM}$ respectively; then, the constraint on the threshold becomes simply $U_{min} \leq 1$.

Noise-free codeword usage

Lastly, if one wishes to simultaneously use the random restarts technique with noisy codeword assignments, care should be taken to update mean usages on the basis of the codewords assignments before the injection of discrete noise. This allows the statistic to be a more accurate reflection of the outputs that the encoder network is truly capable of producing, as opposed to being affected by codewords that would not have seen any utilisation had there not been any noise injected.

5.3 Experiments

5.3.1 Models

The two best performing models trained on the basis of the described methods were obtained by using all techniques discussed. They differ in that the first one, henceforth referred to as PQ-RT-8 (product quantised regularisation trinity¹ with 8 rings), uses the multi-ring noise injection method with 8 rings and a random restart usage threshold set to 0.95, while the second, henceforth referred to as PQ-RT-1, uses the standard noise injection method, with a random restarts usage threshold of 0.85.

Product-quantisation and noisy codeword assignment (both single-ring and multi-ring) have been implemented by extending the `vector-quantize-pytorch` [69] library, which already provided support for l_2 -normalisation and random restarts. For ease of implementation, a rounded Gaussian was used instead of a discrete Gaussian when sampling discrete noise for the noisy codeword assignments.

With the aim of satisfying the dimensionality constraints of the CBIR task (subsection 5.1.2), the architecture of the VQ-VAE employed for CBIR benchmarks was altered by adding a linear layer immediately before the vector quantiser to reduce the dimensionality of the encoder network output from $\mathbb{R}^{64 \times 8}$ to $\mathbb{R}^{8 \times 8}$; a second corresponding linear layer was inserted immediately before the decoder network input to upscale the quantised latents from $\mathbb{R}^{8 \times 8}$ to $\mathbb{R}^{64 \times 8}$. Additionally, the channel size of the residual blocks of the VQ-VAE was halved, in order to avoid overfitting on MNIST, which is a smaller and slightly simpler dataset compared to the MorphoMNIST version used for CCBIR.

5.3.2 Methodology

To test the effect on performance of the surveyed techniques for VQ-VAE training while allowing for fair comparison with other retrieval methods, the standard setup for the MNIST CBIR task was followed: from the 70,000 images of the full MNIST dataset, 100 images per digit were randomly sampled from the test set to form a balanced query set of 1,000 images; the remaining 69,000 images were used as the pre-existing gallery dataset. For the purposes

¹The term ‘regularisation trinity’ is employed to describe the combination of l_2 -normalisation, random restarts, and noisy codeword assignment.

of computing benchmark metrics, an image in the ranked list returned by the model was considered relevant to a query image if both images contained the same digit.

5.3.3 Comparison

Method	Results (%)	
	mAP@All	mAP@1000
DeepBit [89]	-	44.53
UTH [90]	-	49.66
PCAH [91]	21.47	63.31
SpeH [92]	24.10	67.60
LSH [93]	31.71	66.23
SphH [94]	34.75	65.45
KMH [95]	35.78	67.62
ITQ [96]	45.37	80.23
DH [83]	46.74	-
DeepQuan [79]	52.54	-
HashGAN [88]	93.93	96.37
vanilla VQ-VAE	25.95	72.08
PQ-RT-1	49.74	85.10
PQ-RT-8	54.23	80.17

Table 5.1: Comparative results for a variety of methods on the MNIST CBIR task using 64-bit hash codes. The results for vanilla VQ-VAE, PQ-RT-1 and, PQ-RT-8 have been produced during this project, while the results for other models are taken from table 1 in [88] and table 1 in [79].

Table 5.1 reports the mAP@All and mAP@1000 scores on the MNIST CBIR task of a variety of alternative methods for unsupervised image retrieval via 64-bit hashcodes, alongside the two best-performing models (11 and 14 from Table 5.2) from the ablation study conducted. Unsupervised CBIR frameworks that do not provide mAP scores on MNIST, such as [97] and [80], have been excluded from the comparison. The proposed method substantially improves the retrieval performance of the vanilla VQ-VAE and surpasses all the alternative methods included other than HashGAN [88]. Notably, from a broad review of the literature, the PQ-RT-8 model appears to outperform in the mAP@All metric all product quantisation-based methods which report results for MNIST. Nonetheless, the state-of-the-art results achieved by HashGAN [88] (which does not employ vector quantisation or product quantisation) remain out of reach.

HashGAN

The excellent performance of HashGAN can in part be attributed to the enforcement of weight sharing between the discriminator network of a GAN [98] and the encoder network that produces the binary hashcode for an input image: the authors claim that the adversarial loss of the GAN acts as a form of data-dependant regularisation, preventing the encoder from overfitting the training data. Additionally, a comprehensive loss function contributes to retrieval performance by encouraging hashcode similarity under similar images and bits’ independence for more meaningful distance computations.

DeepQuan

DeepQuan [79] is perhaps the retrieval technique closest to the CBIR pipeline proposed in this project, both architecturally and in terms of final mAP results. The authors also use an autoencoding architecture with a product quantiser at the bottleneck. However, rather than learning the codebook online, it is updated by standard k -means at the end of each epoch. Moreover, DeepQuan does not employ l_2 -normalisation, random restarts, or noisy codeword assignments. Instead, it relies on an additional loss function term in order to induce a more discriminative structure in the latent space by maximising the distance between each vector $e_\psi(x)_i \in \mathbb{R}^D$ of the encoder output $e_\psi(x) \in \mathbb{R}^{N \times D}$ and all the codewords to which that output vector is not assigned by the quantisation step, i.e. vectors $\{C_j | j \neq \text{nearest}_C(e_\psi(x))_i \wedge j \in \{0, \dots, K-1\}\}$.

5.3.4 Ablation Study

An ablation study was conducted for the purposes of establishing the relative contribution of each alteration to the standard VQ-VAE training. In particular, retrieval performance has been assessed under different types of quantisation methods (vector quantisation or product quantisation), distance metrics between latents (Euclidean or cosine), methods of noisy codeword assignment (no noise, original method from Coppock and Schuller, or the proposed modification using 8 rings), and random restarts' minimum usage thresholds (no random restarts, 0.85, or 0.95). Results are shown in Table 5.2 and include the typically reported metrics: mean average precision over the full ranked list returned by a query (mAP@All) and over the top 1,000 results (mAP@1000).

NDCG_{SSIM}

In addition to mAP@All and mAP@1000 results, Table 5.2 shows, for each tested model, the average over all queries of the NDCG (section 2.4.5) computed between the ranked list returned by the model and a reference list obtained by ordering the gallery images by descending SSIM between each gallery image and the query image. Such benchmark metric, henceforth referred to as NDCG_{SSIM}, is introduced here in an attempt to evaluate the degree to which similarity in the latent space corresponds to similarity in the image space. In particular, noting that

1. the ranking list returned by the model for a query image depends on relative similarity between embeddings in the latent space,
2. the ranking list implied by SSIM values depends on the relative visual similarity between images,
3. the NDCG provides an indication of how well a predicted ranking list approximates a ground truth list,

then NDCG_{SSIM} seeks to provide an indication of how well the ordering implied by similarity in the latent space approximates the ordering implied by the visual similarity of the images. It is important to note that the concept of image relevance is task-dependant and, in general, is not identical to simple visual similarity; therefore, the SSIM-implied ranking list may be very different from the ideal ranking list. Thus, the utility of NDCG_{SSIM} lies not in being

another metric for evaluating retrieval performance on a specific task, but rather in providing a task-independent way of assessing the capacity of a distance measure on the latent space to be a proxy for general visual dissimilarity on images.

Results

Model					Results (%)		
ID	PQ/VQ	Distance	Noise rings	Rand. restarts threshold	mAP@All	mAP@1000	NDCG _{SSIM}
1	VQ	Euclidean	n/a	n/a	25.95	72.08	98.45
2	VQ	Cosine	n/a	n/a	23.47	66.59	98.67
3	PQ	Euclidean	n/a	n/a	26.57	71.92	98.54
4	PQ	Cosine	n/a	n/a	24.72	71.59	98.83
5	PQ	Euclidean	1	n/a	27.75	73.79	98.40
6	PQ	Cosine	1	n/a	27.83	76.51	98.60
7	PQ	Cosine	8	n/a	26.22	74.31	98.48
8	PQ	Euclidean	n/a	0.85	11.05	10.89	94.78
9	PQ	Cosine	n/a	0.85	24.35	70.44	98.77
10	PQ	Euclidean	1	0.85	18.88	28.91	96.42
11	PQ	Cosine	1	0.85	49.74	85.10	97.91
12	PQ	Cosine	8	0.85	47.36	80.12	98.01
13	PQ	Cosine	1	0.95	51.57	79.03	97.68
14	PQ	Cosine	8	0.95	54.23	80.17	97.71

Table 5.2: Ablation study results on the MNIST CBIR task. Models 11 and 14 are identical to PQ-RT-1 and PQ-RT-8.

Inspection of Table 5.2 permits a series of remarks.

Product quantisation

Comparison between pairs of models that differ only in the quantisation technique used (c.f. 1 and 3, 2 and 4) suggests that product quantisation provides a slight improvement in mAP compared to vector quantisation, which is seen more strongly for the cosine distance model. The performance advantage of product quantisation is not surprising, as a product quantiser is strictly more flexible than the corresponding vector quantiser in the arrangement of codewords in the latent space.

Distance metric

Inspecting pairs of models that differ only in the distance metric used (cf. 1 and 2, 3 and 4, 5 and 6, 8 and 9, 10 and 11) highlights that the NDCG_{SSIM} metric is always higher in the model that uses the cosine distance as opposed to the Euclidean distance, which suggests that the l_2 -normalisation step during training leads to distance computations in the latent space which are more reflective of visual dissimilarity compared to the unnormalised training. As mentioned in the previous section, higher NDCG_{SSIM} scores need not imply higher mAP scores (cf. models 1 and 2).

Noise injection

Comparing pairs of models that differ only in whether noisy codeword assignment is performed during training (cf. models 3 and 5, 4 and 6) indicates that noise injection improves mAP for both distance measures, with the cosine distance model seeing the greater performance increase.

Random restarts

Performance on pairs of models without noise injection that differ only on whether random restarts are used (cf. 3 and 8, 4 and 9) indicates that the addition of random restarts with a relatively high minimum usage threshold (0.85) is extremely detrimental to mAP when Euclidean distance is used but only slightly detrimental when l_2 -normalisation is performed. Since the random restarts’ technique can be seen as a form of noise injection with the potential to destabilise training dynamics, these results corroborate the increased training stability of the l_2 -normalised codebook observed by Yu et al. [82].

Regularisation trinity

Most strikingly, the simultaneous use of noisy codeword assignments with random restarts and l_2 -normalisation (models 11-14) doubles the mAP score compared to models that are missing any single one of these three features but have the other two (models 6, 9, and 10). This is especially surprising considering that random restarts without noisy codeword assignments are slightly detrimental to performance. One hypothesis is that the simultaneous efforts to learn a codebook where every codeword has a relatively high utilisation (via random restarts) and to induce a particular structure in the latent space (via noisy codeword assignment) help respectively to extract robust features that are present in most images (so as to yield high codeword utilisation) and to place far from each other in the latent space the codewords that codify said robust features (so as to reduce the reconstruction error under noisy codeword assignments); thus, two codewords that contribute to codifying two different digits – a rather robust image feature – are likely to be placed far away in the latent space, leading to higher precision when using distances in the latent space to rank results. Meanwhile, l_2 -normalisation plays a key role in stabilising the training dynamics (cf. models 10 and 11).

Multi-ring noise injection

The effect on retrieval precision of the multi-ring noise injection method proposed in section 5.2.3 is less clear. Comparing pairs of models that differ only in the noise injection method used, the alternative method using 8 rings is slightly detrimental to performance when random restarts are not used (cf. models 6 and 7) or when the random restarts’ threshold is set to 0.85 (cf. models 11 and 12), while being slightly beneficial when it is set to 0.95 (cf. models 13 and 14). Given these results, it seems possible that the main effect of the alternative method is an increase in training stability: as the number of rings grows, the number of codewords within a ring that are valid candidates for a random reassignment decreases, thus slightly reducing the amount of randomness during training. Such increased training stability might be unhelpful in the case of an already sufficiently stable setup with a lower random restarts threshold (models 6 and 7, 11 and 12); however, it may be able to compensate for the added training instability of a higher random restarts threshold (models 13 and 14), leading to improved performance.

Overall, the ablation study confirms that every surveyed technique has a positive contribution to the image retrieval performance of the PQ-RT-8 and PQ-RT-1 models on the studied CBIR task.

5.3.5 CCBIR

Given the improvements to a standard CBIR task brought about by the discussed training techniques, it seems natural to hypothesise that these methods could also improve the retrieval performance of the proposed strategies for CCBIR, especially in the case of the sample-similarity ranking method, which partly relies on the meaningfulness of distance computations in the latent space to establish relevance. To test such a hypothesis, a PQ-VAE with the same architecture as the VQ-VAE for the CCBIR experiments of section 4.2 was trained on the CCBIR dataset. It uses all the discussed methods for VQ-VAE performance improvements (a product quantiser with 8 subquantisers, l_2 -normalisation, random restarts with a minimum usage threshold of 0.4, and single-ring noisy codeword assignments)². Then, the quantitative experiments of section 4.2 were reproduced by fitting an identical categorical VTN to the newly trained PQ-VAE and testing its CCBIR performance using both the sample-similarity method (denoted as CBIR-PQ-VTN) and the ELBO method (denoted as ELBO-PQ-VTN). The previously and newly obtained results are shown side-by-side for comparison in Table 5.3.

Setup								
Method	CBIR-VQ-VTN		CBIR-PQ-VTN		ELBO-VQ-VTN		ELBO-PQ-VTN	
Factual	swell	fracture	swell	fracture	swell	fracture	swell	fracture
Counterfactual	fracture	swell	fracture	swell	fracture	swell	fracture	swell
Results (%)								
HR@1	98.9	98.3	97.8	98.3	99.6	99.1	99.4	99.0
HR@5	99.3	99.4	99.1	99.4	99.8	99.8	99.6	99.7
HR@10	99.6	99.5	99.2	99.4	99.8	100.0	99.8	99.8
MRR	99.13	98.82	98.42	98.81	99.71	99.45	99.50	99.29
NDCG _{SSIM}	99.87	99.89	99.86	99.89	99.61	99.71	98.44	98.97
mAP@All	26.74	25.21	27.70	25.78	24.78	23.71	17.91	19.64
mAP@1000	75.59	69.86	75.05	68.39	74.86	68.20	50.36	52.14

Table 5.3: CCBIR benchmark results comparing methods that use a vanilla VQ-VAE (CBIR-VQ-VTN and ELBO-VQ-VTN) with methods that use a PQ-VAE trained with the techniques discussed in this chapter (CBIR-PQ-VTN and ELBO-PQ-VTN). The results for CBIR-VQ-VTN and ELBO-VQ-VTN are identical to Table 4.2 and are included again here to ease comparison. The factual row indicates which image is submitted as the factual query image, while the counterfactual row indicates both the counterfactual treatment and the gallery dataset used for retrieval.

The combined effect of the surveyed techniques for improved VQ-VAE training was detrimental to almost all performance metrics on the benchmarked CCBIR task (except for a small increase in mAP@All for the sample-similarity based method), as can be seen by confronting model CBIR-VQ-VTN with CBIR-PQ-VTN and ELBO-VQ-VTN with ELBO-PQ-VTN. Such results are in contradiction with the initial hypothesis that an improvement in CBIR results would translate to an improvement in CCBIR results. One reason for this

²These hyperparameters are not identical to PQ-RT-1 or PQ-RT-8 since a new hyperparameter search had to be performed for the slightly different architecture and dataset of the CCBIR setting.

could be attributed to the fact that the criteria for relevance are distinct between the CBIR task and the CCBIR task: indeed, CBIR-PQ-VTN has slightly higher mAP@All score but a slightly lower MRR than CBIR-VQ-VTN, i.e. it has higher performance in the CBIR task but lower performance in the CCBIR one. A second reason for the drop in performance when using the product quantised methods could be an increase in difficulty for the VTN to fit the latent space of the product codebook compared to a standard vector quantiser: with 8 subquantisers, there are 8 times more codebooks, and associated latent spaces that the VTN must be able to target. The substantial drop in many of the metrics when comparing ELBO-VQ-VTN with ELBO-PQ-VTN may be in support of such interpretation. Investigating performance on a deeper and wider neural architecture could therefore be a starting point for future work wishing to improve on such results.

Chapter 6

Conclusion

This chapter concludes the project by summarising the main results and presenting avenues for future work.

6.1 Main Results

The qualitative evaluation performed in the experiments of chapter 3 indicates that the variational twin network (VTN) framework is capable of sampling highly credible pairs of factual-counterfactual images, which display a considerable degree of diversity while remaining consistent with the covariates and treatments. Quantitative evaluation reinforced these results, as the method achieved an average structural similarity index (SSIM) score of 79% between the ground truth counterfactual image and a generated counterfactual sample; additionally, setting the VTN output to directly target the categorical latent space of the VQ-VAE lead to a 7.76% improvement in SSIM over targeting its continuous counterpart.

A quantitative evaluation of the two strategies for counterfactual image retrieval proposed in chapter 4 demonstrated their suitability for the task. Both methods are largely successful in selecting as the top result a ground truth counterfactual image present within the gallery dataset: the ELBO-based method achieved HR@1 (hit rate at 1) scores strictly above 99.0%, while the sample similarity’s HR@1 scores were above 98.0%; the results also established the empirical superiority of the more mathematically principled ELBO approach. Visually judging the top-10 retrieved results for a sample of submitted queries, the images returned for each query appeared to be likely counterfactuals due to their close similarity with the counterfactual ground truth.

The ablation study conducted in chapter 5 to establish the relative effect on performance of l_2 -normalisation, product quantisation, random restarts, and noisy codeword assignment, indicates that all tested techniques contribute positively to retrieval performance, but some of them only have a positive effect when used in conjunction. The best trained model obtained a mAP@All score of 54.23% in a popular CBIR task for compact embeddings, which is more than double the score of a vanilla VQ-VAE, and is marginally superior to other methods that use product quantisation; however, it remains significantly below the performance achieved by adversarial approaches. Finally, the improvements to retrieval

performance produced by such methods did not transfer from the standard image retrieval setting to the counterfactual one, empirically highlighting the theoretical distinction in their respective definitions of relevance, and suggesting that it may be more challenging for the implemented VTN to fit a product quantised latent space compared to a vector quantised one.

Overall, the project has succeeded in its objective of building a system for counterfactual image retrieval by providing two strategies for estimating counterfactual relevance while introducing a principled framework for learning generative models with support for twin-network counterfactual inference. The hope is that these frameworks may serve both researchers wishing to extend the capabilities of a CBIR system through causality and those wishing to ‘ground’ to the real world the imaginative power of counterfactual image generators.

6.2 Future work

Datasets

During the project, the entire evaluation process for the counterfactual inference tasks was conducted on a single synthetic dataset of relatively limited complexity for which ground truth counterfactuals were available. While this setup was instrumental to ease the theoretical design of the framework and its concrete implementation, it is partly unsuitable for providing an assessment of the system’s performance on real world data, where ground truth counterfactuals may need to be imputed. Additionally, the ELBO based approach obtained a hit rate at 1 close to the theoretical maximum of 100%, which suggests that the benchmark may not be sufficiently difficult to fully showcase where the performance limits of the framework lie. Therefore, future work should both construct more difficult synthetic datasets for CCBIR and investigate performance on real world datasets. For example, performing CCBIR on medical image data, such as on the MRI brain scans from the UK Biobank [99] seen in [1], would be an interesting real world application with the potential for practical utility to medical practitioners.

Counterfactual information retrieval

While the project has focused on counterfactual image retrieval, the designed theoretical framework should be sufficiently generic to support information retrieval on a variety of datasets, including those in video form or in simpler tabular data. Indeed, even the architectural choice to use the VTN as a categorical prior on a VQ-VAE latent space may have applications beyond image retrieval, such as in user recommendation systems [100], which could then be imbued with counterfactual capabilities.

Normalising flows

Pawlowski et al. [1] use both VAEs and normalising flows to abduce the exogenous noise variables in the environment. However, the virtual twin network architecture proposed in this project only includes VAEs. While VAEs were found to be sufficient for the generation of convincing counterfactuals in the setting of interest, other datasets and image generation tasks may benefit from the exact invertibility of the mechanisms granted by the use of normalising flows, especially for lower dimensionality data.

Bibliography

- [1] Nick Pawlowski, Daniel C. Castro, and Ben Glocker. Deep Structural Causal Models for Tractable Counterfactual Inference. *arXiv:2006.06485 [cs, stat]*, October 2020. URL <http://arxiv.org/abs/2006.06485>.
- [2] Athanasios Vlontzos, Bernhard Kainz, and Ciaran M. Gilligan-Lee. Estimating the probabilities of causation via deep monotonic twin networks. *arXiv:2109.01904 [cs]*, September 2021. URL <http://arxiv.org/abs/2109.01904>.
- [3] Daniel C. Castro, Jeremy Tan, Bernhard Kainz, Ender Konukoglu, and Ben Glocker. Morpho-MNIST: Quantitative Assessment and Diagnostics for Representation Learning, October 2019. URL <http://arxiv.org/abs/1809.10780>.
- [4] Hadrien Reynaud, Athanasios Vlontzos, Mischa Dombrowski, Ciarán Lee, Arian Beqiri, Paul Leeson, and Bernhard Kainz. D’ARTAGNAN: Counterfactual Video Generation, June 2022. URL <http://arxiv.org/abs/2206.01651>.
- [5] Ruocheng Guo, Lu Cheng, Jundong Li, P. Richard Hahn, and Huan Liu. A Survey of Learning Causality with Data: Problems and Methods. *ACM Computing Surveys*, 53(4):1–37, July 2021. ISSN 0360-0300, 1557-7341. doi: 10.1145/3397269. URL <http://arxiv.org/abs/1809.09337>.
- [6] Murat Kocaoglu, Christopher Snyder, Alexandros G. Dimakis, and Sriram Vishwanath. CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training. In *International Conference on Learning Representations*, February 2018. URL <https://openreview.net/forum?id=BJE-4xWOW>.
- [7] Axel Sauer and Andreas Geiger. Counterfactual Generative Networks. In *International Conference on Learning Representations*, September 2020. URL <https://openreview.net/forum?id=BXewfAYMmJw>.
- [8] Saloni Dash, Vineeth N. Balasubramanian, and Amit Sharma. Evaluating and Mitigating Bias in Image Classifiers: A Causal Perspective Using Counterfactuals. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 915–924, 2022. URL https://openaccess.thecvf.com/content/WACV2022/html/Dash_Evaluating_and_Mitigating_Bias_in_Image_Classifiers_A_Causal_Perspective_WACV_2022_paper.html.
- [9] Ibtihal M. Hameed, Sadiq H. Abdulhussain, and Basheera M. Mahmmod. Content-based image retrieval: A review of recent trends. *Cogent Engineering*, 8(1):1927469, January 2021. ISSN 2331-1916. doi: 10.1080/23311916.2021.1927469. URL <https://www.tandfonline.com/doi/full/10.1080/23311916.2021.1927469>.

- [10] Alexander Balke and Judea Pearl. Probabilistic evaluation of counterfactual queries. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 1)*, AAAI '94, pages 230–237, USA, 1994. American Association for Artificial Intelligence. ISBN 0-262-61102-3.
- [11] Harry Coppock and Bjorn Schuller. Vector Quantised-Variational Autoencoders (VQ-VAEs) for Representation learning. URL <https://harrycoppock.com/publication/2020-09-01-masterthesis>.
- [12] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A Generative Model for Music, April 2020. URL <http://arxiv.org/abs/2005.00341>.
- [13] Hanwei Wu and Markus Flierl. Learning Product Codebooks Using Vector-Quantized Autoencoders for Image Retrieval. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5, November 2019. doi: 10.1109/GlobalSIP45357.2019.8969272.
- [14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts, 2016. ISBN 978-0-262-03561-3.
- [15] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, U.K. ; New York, second edition, 2009. ISBN 978-0-521-89560-6.
- [16] Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. On Pearl’s Hierarchy and the Foundations of Causal Inference. Technical report, 2021. URL <https://causalai.net/r60.pdf>.
- [17] Judea Pearl. Graphical Models for Probabilistic and Causal Reasoning. In Philippe Smets, editor, *Quantified Representation of Uncertainty and Imprecision*, Handbook of Defeasible Reasoning and Uncertainty Management Systems, pages 367–389. Springer Netherlands, Dordrecht, 1998. ISBN 978-94-017-1735-9. doi: 10.1007/978-94-017-1735-9_12. URL https://doi.org/10.1007/978-94-017-1735-9_12.
- [18] J. Pearl. *Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning*. Report. University of California (Los Angeles). Computer Science Department, 1985. URL https://ftp.cs.ucla.edu/pub/stat_ser/r43-1985.pdf.
- [19] Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, New York, first edition edition, 2018. ISBN 978-0-465-09760-9.
- [20] Logan Graham, Ciarán M Lee, and Yura Perov. Copy, paste, infer: A robust analysis of twin networks for counterfactual inference. page 9. URL https://cpb-us-w2.wpmucdn.com/sites.coecis.cornell.edu/dist/a/238/files/2019/12/Id_65_final.pdf.
- [21] Ilya Shpitser and Judea Pearl. What counterfactuals can be tested. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, UAI’07, pages 352–359, Arlington, Virginia, USA, July 2007. AUAI Press. ISBN 978-0-9749039-3-4.

- [22] Ramalingam Shanmugam. Elements of causal inference: Foundations and learning algorithms. *Journal of Statistical Computation and Simulation*, 88(16), November 2018. ISSN 0094-9655, 1563-5163. doi: 10.1080/00949655.2018.1505197. URL <https://www.tandfonline.com/doi/full/10.1080/00949655.2018.1505197>.
- [23] Junzhe Zhang, Jin Tian, and Elias Bareinboim. Partial Counterfactual Identification from Observational and Experimental Data, October 2021. URL <http://arxiv.org/abs/2110.05690>.
- [24] Lars Ruthotto and Eldad Haber. An Introduction to Deep Generative Modeling. *arXiv:2103.05180 [cs]*, April 2021. URL <http://arxiv.org/abs/2103.05180>.
- [25] Christopher M Bishop. LATENT VARIABLE MODELS. Learning in Graphical Models, M. I. Jordan (Ed.), MIT Press (1999), 371–403.
- [26] Carl Doersch. Tutorial on Variational Autoencoders. *arXiv:1606.05908 [cs, stat]*, January 2021. URL <http://arxiv.org/abs/1606.05908>.
- [27] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.2017.1285773. URL <http://arxiv.org/abs/1601.00670>.
- [28] Ankush Ganguly and Samuel W. F. Earp. An Introduction to Variational Inference. *arXiv:2108.13083 [cs, stat]*, November 2021. URL <http://arxiv.org/abs/2108.13083>.
- [29] Diederik P. Kingma and Max Welling. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8237, 1935-8245. doi: 10.1561/22000000056. URL <http://arxiv.org/abs/1906.02691>.
- [30] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, May 2014. URL <http://arxiv.org/abs/1312.6114>.
- [31] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *arXiv:1401.4082 [cs, stat]*, May 2014. URL <http://arxiv.org/abs/1401.4082>.
- [32] Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in Variational Inference. *arXiv:1711.05597 [cs, stat]*, October 2018. URL <http://arxiv.org/abs/1711.05597>.
- [33] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL <https://proceedings.neurips.cc/paper/2015/hash/8d55a249e6baa5c06772297520da2051-Abstract.html>.
- [34] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo Gradient Estimation in Machine Learning. *arXiv:1906.10652 [cs, math, stat]*, September 2020. URL <http://arxiv.org/abs/1906.10652>.

- [35] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. *arXiv:1711.00937 [cs]*, May 2018. URL <http://arxiv.org/abs/1711.00937>.
- [36] R. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, April 1984. ISSN 1558-1284. doi: 10.1109/MASSP.1984.1162229.
- [37] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. *arXiv:1711.00937 [cs]*, May 2018. URL <http://arxiv.org/abs/1711.00937>.
- [38] Afshan Latif, Aqsa Rasheed, Umer Sajid, Jameel Ahmed, Nouman Ali, Naeem Iqbal Ratyal, Bushra Zafar, Saadat Hanif Dar, Muhammad Sajid, and Tehmina Khalil. Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review. *Mathematical Problems in Engineering*, 2019:1–21, August 2019. ISSN 1024-123X, 1563-5147. doi: 10.1155/2019/9658350. URL <https://www.hindawi.com/journals/mpe/2019/9658350/>.
- [39] Liang Zheng, Yi Yang, and Qi Tian. SIFT Meets CNN: A Decade Survey of Instance Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5): 1224–1244, May 2018. ISSN 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2017.2709749. URL <https://ieeexplore.ieee.org/document/7935507/>.
- [40] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL <http://link.springer.com/10.1023/B:VISI.0000029664.99615.94>.
- [41] Wei Chen, Yu Liu, Weiping Wang, Erwin Bakker, Theodoros Georgiou, Paul Fieguth, Li Liu, and Michael S. Lew. Deep Learning for Instance Retrieval: A Survey. *arXiv:2101.11282 [cs]*, January 2022. URL <http://arxiv.org/abs/2101.11282>.
- [42] Josef Sivic and Andrew Zisserman. Video Google: Efficient Visual Search of Videos. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Jean Ponce, Martial Hebert, Cordelia Schmid, and Andrew Zisserman, editors, *Toward Category-Level Object Recognition*, volume 4170, pages 127–144. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-68794-8 978-3-540-68795-5. doi: 10.1007/11957959_7. URL http://link.springer.com/10.1007/11957959_7.
- [43] Herve Jegou, Matthijs Douze, Cordelia Schmid, and Patrick Perez. Aggregating local descriptors into a compact image representation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, San Francisco, CA, USA, June 2010. IEEE. ISBN 978-1-4244-6984-0. doi: 10.1109/CVPR.2010.5540039. URL <http://ieeexplore.ieee.org/document/5540039/>.
- [44] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision*, 105(3):222–245, December 2013. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-013-0636-x. URL <http://link.springer.com/10.1007/s11263-013-0636-x>.

- [45] Guojun Lu and Shyhwei Teng. A Novel Image Retrieval Technique based on Vector Quantization. In *International Conference on Computational Intelligence for Modelling, Control and Automation*, page 6, 1999.
- [46] Ajay H. Daptardar and James A. Storer. Content-Based Image Retrieval Via Vector Quantization. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, George Bebis, Richard Boyle, Darko Koracin, and Bahram Parvin, editors, *Advances in Visual Computing*, volume 3804, pages 502–509. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-30750-1 978-3-540-32284-9. doi: 10.1007/11595755_61. URL http://link.springer.com/10.1007/11595755_61.
- [47] H Jégou, M Douze, and C Schmid. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1): 117–128, January 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.57. URL <http://ieeexplore.ieee.org/document/5432202/>.
- [48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017. ISSN 0001-0782, 1557-7317. doi: 10.1145/3065386. URL <https://dl.acm.org/doi/10.1145/3065386>.
- [49] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *arXiv:1412.7062 [cs]*, June 2016. URL <http://arxiv.org/abs/1412.7062>.
- [50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497 [cs]*, January 2016. URL <http://arxiv.org/abs/1506.01497>.
- [51] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *arXiv:1310.1531 [cs]*, October 2013. URL <http://arxiv.org/abs/1310.1531>.
- [52] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural Codes for Image Retrieval. *arXiv:1404.1777 [cs]*, July 2014. URL <http://arxiv.org/abs/1404.1777>.
- [53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami, FL, June 2009. IEEE. ISBN 978-1-4244-3992-8. doi: 10.1109/CVPR.2009.5206848. URL <https://ieeexplore.ieee.org/document/5206848/>.
- [54] Artem Babenko Yandex and Victor Lempitsky. Aggregating Local Deep Features for Image Retrieval. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277, Santiago, Chile, December 2015. IEEE. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.150. URL <http://ieeexplore.ieee.org/document/7410507/>.

- [55] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale Orderless Pooling of Deep Convolutional Activation Features. *arXiv:1403.1840 [cs]*, September 2014. URL <http://arxiv.org/abs/1403.1840>.
- [56] Giorgos Tolias, Ronan Sifre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. *arXiv:1511.05879 [cs]*, February 2016. URL <http://arxiv.org/abs/1511.05879>.
- [57] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, July 2008. ISBN 978-1-139-47210-4.
- [58] Ellen M. Voorhees. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378, December 2001. ISSN 1351-3249, 1469-8110. doi: 10.1017/S1351324901002789. URL https://www.cambridge.org/core/product/identifier/S1351324901002789/type/journal_article.
- [59] Kalervo Järvelin and Jaana Kekäläinen. Discounted Cumulated Gain. In LING LIU and M. TAMER ÖZSU, editors, *Encyclopedia of Database Systems*, pages 849–853. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_478. URL https://doi.org/10.1007/978-0-387-39940-9_478.
- [60] MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. URL <http://yann.lecun.com/exdb/mnist/>.
- [61] Andreas Stuhlmüller, Jacob Taylor, and Noah Goodman. Learning Stochastic Inverses. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL <https://papers.nips.cc/paper/2013/hash/7f53f8c6c730af6aeb52e66eb74d8507-Abstract.html>.
- [62] Danilo Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1530–1538. PMLR, June 2015. URL <https://proceedings.mlr.press/v37/rezende15.html>.
- [63] Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning Likelihoods with Conditional Normalizing Flows, November 2019. URL <http://arxiv.org/abs/1912.00042>.
- [64] Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. *arXiv:1606.03976 [cs, stat]*, May 2017. URL <http://arxiv.org/abs/1606.03976>.
- [65] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. A Survey on Causal Inference. *ACM Transactions on Knowledge Discovery from Data*, 15(5):1–46, October 2021. ISSN 1556-4681, 1556-472X. doi: 10.1145/3444944. URL <https://dl.acm.org/doi/10.1145/3444944>.
- [66] Patrick Schwab, Lorenz Linhardt, and Walter Karlen. Perfect Match: A Simple Method for Learning Representations For Counterfactual Inference With Neural Networks. *arXiv:1810.00656 [cs, stat]*, May 2019. URL <http://arxiv.org/abs/1810.00656>.

- [67] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Abstract.html>.
- [68] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html>.
- [69] Phil Wang. Lucidrains/vector-quantize-pytorch, June 2022. URL <https://github.com/lucidrains/vector-quantize-pytorch>.
- [70] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, January 2019. ISSN 1532-4435.
- [71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.
- [72] Arash Vahdat and Jan Kautz. NVAE: A Deep Hierarchical Variational Autoencoder. In *Advances in Neural Information Processing Systems*, volume 33, pages 19667–19679. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/e3b21256183cf7c2c7a66be163579d37-Abstract.html>.
- [73] Diganta Misra. Mish: A Self Regularized Non-Monotonic Neural Activation Function. URL <https://arxiv.org/abs/1908.08681>.
- [74] Oleh Rybkin, Kostas Daniilidis, and Sergey Levine. Simple and Effective VAE Training with Calibrated Decoders. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9179–9189. PMLR, July 2021. URL <https://proceedings.mlr.press/v139/rybkin21a.html>.
- [75] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1412.6980>.
- [76] Adrian Łańcucki, Jan Chorowski, Guillaume Sanchez, Ricard Marxer, Nanxin Chen, Hans J.G.A. Dolfing, Sameer Khurana, Tanel Alumäe, and Antoine Laurent. Robust Training of Vector Quantized Bottleneck Models. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, July 2020. doi: 10.1109/IJCNN48605.2020.9207145.

- [77] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. ISSN 1941-0042. doi: 10.1109/TIP.2003.819861.
- [78] Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen. Deep Quantization Network for Efficient Image Retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), March 2016. ISSN 2374-3468. doi: 10.1609/aaai.v30i1.10455. URL <https://ojs.aaai.org/index.php/AAAI/article/view/10455>.
- [79] Junjie Chen, William K. Cheung, and Anran Wang. Learning Deep Unsupervised Binary Codes for Image Retrieval. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 613–619, Stockholm, Sweden, July 2018. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-2-7. doi: 10.24963/ijcai.2018/85. URL <https://www.ijcai.org/proceedings/2018/85>.
- [80] Young Kyun Jang and Nam Ik Cho. Self-Supervised Product Quantization for Deep Unsupervised Image Retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12085–12094, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Jang_Self-Supervised_Product_Quantization_for_Deep_Unsupervised_Image_Retrieval_ICCV_2021_paper.html.
- [81] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A Generative Model for Music, April 2020. URL <http://arxiv.org/abs/2005.00341>.
- [82] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized Image Modeling with Improved VQGAN, March 2022. URL <http://arxiv.org/abs/2110.04627>.
- [83] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. Deep hashing for compact binary codes learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2475–2483, June 2015. doi: 10.1109/CVPR.2015.7298862.
- [84] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. URL <http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>.
- [85] K. Kavitha, B. Sandhya, and B. Thirumala. Evaluation of Distance Measures for Feature based Image Registration using AlexNet. *International Journal of Advanced Computer Science and Applications*, 9(10), 2018. ISSN 21565570, 2158107X. doi: 10.14569/IJACSA.2018.091034. URL <http://thesai.org/Publications/ViewPaper?Volume=9&Issue=10&Code=ijacsa&SerialNo=34>.
- [86] Divya M O and Vimina E R. Performance Analysis of Distance Metric for Content Based Image Retrieval. *International Journal of Engineering and Advanced Technology*, 8(6):2215–2218, August 2019. ISSN 22498958. doi: 10.35940/ijeat.F8610.088619. URL <https://www.ijeat.org/portfolio-item/F8610088619/>.

- [87] Qian Bao and Ping Guo. Comparative studies on similarity measures for remote sensing image retrieval. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, volume 1, pages 1112–1116, The Hague, Netherlands, 2004. IEEE. ISBN 978-0-7803-8567-2. doi: 10.1109/ICSMC.2004.1398453. URL <http://ieeexplore.ieee.org/document/1398453/>.
- [88] Kamran Ghasedi Dizaji, Feng Zheng, Najmeh Sadoughi Nourabadi, Yanhua Yang, Cheng Deng, and Heng Huang. Unsupervised Deep Generative Adversarial Hashing Network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3664–3673, Salt Lake City, UT, June 2018. IEEE. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00386. URL <https://ieeexplore.ieee.org/document/8578484/>.
- [89] Kevin Lin, Jiwen Lu, Chu-Song Chen, and Jie Zhou. Learning Compact Binary Descriptors with Unsupervised Deep Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1183–1192, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.133. URL <http://ieeexplore.ieee.org/document/7780502/>.
- [90] Shanshan Huang, Yichao Xiong, Ya Zhang, and Jia Wang. Unsupervised Triplet Hashing for Fast Image Retrieval, February 2017. URL <http://arxiv.org/abs/1702.08798>.
- [91] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for scalable image retrieval. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3424–3431, San Francisco, CA, USA, June 2010. IEEE. ISBN 978-1-4244-6984-0. doi: 10.1109/CVPR.2010.5539994. URL <http://ieeexplore.ieee.org/document/5539994/>.
- [92] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral Hashing. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL <https://papers.nips.cc/paper/2008/hash/d58072be2820e8682c0a27c0518e805e-Abstract.html>.
- [93] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity Search in High Dimensions via Hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases, VLDB '99*, pages 518–529, San Francisco, CA, USA, September 1999. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-615-9.
- [94] Jae-Pil Heo, Youngwoon Lee, Junfeng He, Shih-Fu Chang, and Sung-Eui Yoon. Spherical hashing. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2957–2964, June 2012. doi: 10.1109/CVPR.2012.6248024.
- [95] Kaiming He, Fang Wen, and Jian Sun. K-Means Hashing: An Affinity-Preserving Quantization Method for Learning Binary Compact Codes. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2938–2945, June 2013. doi: 10.1109/CVPR.2013.378.
- [96] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR 2011*, pages 817–824, June 2011. doi: 10.1109/CVPR.2011.5995432.

- [97] Jinpeng Wang, Ziyun Zeng, Bin Chen, Tao Dai, and Shu-Tao Xia. Contrastive Quantization with Code Memory for Unsupervised Image Retrieval. URL <https://arxiv.org/abs/2109.05205>.
- [98] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks, June 2014. URL <http://arxiv.org/abs/1406.2661>.
- [99] Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, Bette Liu, Paul Matthews, Giok Ong, Jill Pell, Alan Silman, Alan Young, Tim Sprosen, Tim Peakman, and Rory Collins. UK biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12(3):e1001779, March 2015. ISSN 1549-1676. doi: 10.1371/journal.pmed.1001779.
- [100] Jan Van Balen and Mark Levy. PQ-VAE: Efficient Recommendation Using Quantized Embeddings. *Late-breaking Results, 13th ACM Conference on Recommender Systems*, page 5, 2019.