Generation and Evaluation of Creative Images from Limited Data: A Class-to-Class VAE Approach

Xiaomeng Ye, Ziwei Zhao, David Leake and David Crandall
Luddy School of Informatics, Computing, and Engineering
Indiana University
Bloomington IN 47408, USA
{xiaye,ziwei}@iu.edu, {leake, djcran}@indiana.edu

Abstract

Generating novel items with desired characteristics requires creativity. One method to achieve this is through creative transformations. Deep learning network methods provide an interesting potential substrate for this task. This paper presents a method for network-based generation of novel images by applying variational autoencoders (VAEs) to learn features, which are then perturbed based on a class-to-class (C2C) method for learning of inter-class similarity and difference information, enabling generating creative samples. Our method learns the pattern between classes, applies this pattern to samples of a source class, and generates new samples of a target class. This study also proposes a general approach to evaluating the creativity of sample generators for classification domains, by evaluating the samples generated by the generator trained in a one-shot setting. The evaluation approach requires only classification labels but not human assessments of creativity. An experiment in two image domains supports that the samples generated by our method satisfy two of Boden's creativity criteria: being valuable (falling into desired categories) and novel (samples show high variance).

Introduction

Advances in machine learning have yielded many successful deep generative models (Pan et al. 2019). Such models generate samples conforming to the distribution of the training data, leading to samples that are "authentic" in the sense of substantially sharing the properties of real examples. A surge of research in computational creativity is applying generative deep learning methods while inducing novelty—and even surprise—for the sake of creativity, as described in the surveys of Franceschelli and Musolesi (2021) and Broad et al. (2021). For example, the creative adversarial network proposed by Elgammal et al. (2017) is a generative adversarial network (GAN) that generates artwork that is realistic but also deviates from style norms. Similar work has induced creativity in GANs by introducing additional goals (loss functions) beyond the original adversarial loss (e.g. StyleGAN (Karras, Laine, and Aila 2018) and (Sbai et al. 2018)). Following StyleGan, Nobari, Rashad, and Ahmed (2021) proposed a systematical method to modify GANs to automatically generate novel designs without human intervention. Generally, variational auto encoder (VAE) methods are less suited for creativity tasks because their reconstruction loss aims to mimic the data distribution within a learned latent space and it is difficult to reflect other goals in the corresponding loss function. However, these latent spaces can be manipulated to induce creative results (e.g. MusicVAE (Roberts et al. 2018b) and sketchRNN (Ha and Eck 2017)).

Characterizing the creativity of AI systems requires criteria for assessing the creativity of a process or of a system's results within a task context. Developing such criteria is nontrivial and has received considerable attention (Wiggins 2021). Boden (1991) provides three criteria for assesssing the creativity of outputs of a process: value, novelty, and surprise. Many researchers have continued this school of thought, refining and expanding on these criteria (Wiggins 2006; Draper 2010).

This paper addresses creativity as it applies to generating new samples for a target class when training samples are limited. We consider a sample (generated or not) to be *valuable* if it fits in the target class, and *novel* if it is different from the observed samples of the target class. According to Boden, surprise can happen when the sample is unexpected (which requires a prior expectation entity). We do not consider surprise when evaluating our model.

This paper makes two contributions. First, we present an algorithm for generating creative samples in a classification domain. The algorithm uses a method we call a *class-to-class variational autoencoder* (C2C-VAE), which learns a latent space of the difference patterns between samples of all classes. The C2C-VAE then samples new differences from this latent space, and applies the difference to existing samples in the original conceptual space to generate new samples. Second, we address the general question of how to evaluate the creativity of sample generators for classification in a one-shot setting. We propose an approach which we call GOF/TOM—"Generated On Few, Tested On Many." A generator is trained in a zero, one, or few-shot setting where samples of a target class are trimmed from the training set. The generator is then used to generate new samples of the target class. Meanwhile, an oracle is trained on the *untrimmed* dataset to evaluate the generated samples. Because the generator has limited examples of the target class, its ability to generate satisfactory unseen samples of the target class can be used as measure for its creativity. This is used to evaluate the C2C-VAE.

We begin by discussing related techniques for generating and measuring computational creativity. We then present our C2C-VAE approach for generating creative samples, and introduce our GOF/TOM approach for evaluating creativity. Finally, we evaluate C2C-VAE on two data sets, MNIST (LeCun and Cortes 2010) and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), using the GOF/TOM approach. In the two data sets, C2C-VAE successfully generates samples that are valuable and novel with respect to its training data, making a case for the potential of C2C-VAE as a creative approach. We examine the limitations of C2C-VAE and propose methods for addressing them in future work.

Background

Active Divergence with Generative Deep Learning

In her seminal work, Boden (1991) identifies three forms of creativity: combinatorial, exploratory and transformational. Combinatorial creativity generates new ideas by combining old ones. Exploratory and transformational creativity both involve a conceptual space, where the former explores the conceptual space while the later alters it, potentially causing a paradigm shift (Wiggins 2006; Franceschelli and Musolesi 2021).

Franceschelli and Musolesi (Franceschelli and Musolesi 2021) consider VAEs and GANs to perform exploratory creativity, as they both sample from a conceptual space. GANs can be also be transformational. As an example, in CANs (Creative Adversarial Networks), the discriminator determines both whether a sample image is art or not and its artistic style, while the generator tries to generate art and also generates deviations from original style norms. GANs can even be combinatorial. For example, StyleGAN can achieve style mixing by combinig the latent codes of two samples at multiple different levels of detail.

A CycleGAN is a image-to-image translation technique that can translate an image of one class into an image of another, e.g. modifying the image of a horse h into that of a zebra z using a translation function Z (or conversely from a zebra into a horse, using a function H). A CycleGAN is trained with two loss functions: 1) An adversarial loss trains the generators Z and H to generate quality images (so that a horse h can be translated into a realistic zebra $Z(h)$); 2) A cycle-consistency loss ensures the transition can go back-and-forth (so that the horse-translated zerba $Z(h)$ can be translated back to a horse $H(Z(h))$ similar to the original horse h). The artist Helena Sarin uses CycleGAN to generate creativity-related artwork (NVIDIA 2021).

Our proposed model is based on variational autoencoders (VAEs) (Kingma and Welling 2013). A VAE is comprised of an encoder and a decoder, both implemented as neural networks. The encoder takes samples as inputs and compresses them into a Gaussian distribution of lower-dimension embedding vectors in a latent space. The decoder takes an embedding vector and recovers the original input sample. Regularity—the property of similar samples having similar representations—is encouraged in the latent space because a sample is encoded as a distribution of embeddings, instead of a single embedding as in autoencoders (the forerunners of VAEs).

Features extracted by VAE can be manipulated by perturbation and even vector arithmetic for creative results. For example, MusicVAE (Roberts et al. 2018b) has the ability to "adjust the number of notes in a melody by adding/subtracting a note density vector to/from the latent code" (Roberts et al. 2018a). Similarly, sketchRNN can "subtract the latent vector of an encoded pig head from the latent vector of a full pig, to arrive at a vector that represents a body. Adding this difference to the latent vector of a cat head results in a full cat (i.e. cat head + body = full cat)" (Ha and Eck 2017). The effects of such modification over VAE embeddings are not guaranteed and only partially understood. As noted by Ha and Eck (2017), such analogy is only possible when the embedding distribution is smooth and any interpolation between two embeddings is coherent. This study attempts to model the differences between pairs of embeddings extracted by VAE.

In the taxonomy of active divergence by Broad et al. (2021), this study proposes a method of chaining models. The method is a combination of a standard VAE with a secondary VAE (C2C-VAE) that explores the learned representation of feature differences.

Class-to-class Approach

Classification methods commonly consider the similarity of new instances to instances in a class. The Class-to-class (C2C) approach considers both similarity and difference. It assumes that there exist inter-class patterns between each pair of classes, and the samples from the two classes are consistently similar in some features and different in some other features. For example, zebras and horses have the similarity of both belonging to the Equidae family, and the difference that zebras have stripes while horses do not. The inter-class patterns, once learned, can be used to classify a query based on instances from another or multiple other classes (Ye 2018a; Ye et al. 2020; 2021).

We hypothesize that inter-class patterns can also be used in computational creativity. A system that learns inter-class difference patterns can intentionally apply the patterns to modify a sample. For example, knowing that zebras have stripes and horses do not, the system can modify a horse image by replacing its texture with black-and-white stripes and thus create a new zebra image.

The C2C approach is highly related to GAN methods. For example, CycleGAN is trained on unpaired image-to-image data from one class to another and can generate zebra images from horse images (Zhu et al. 2017). GAN methods are mostly end-to-end. For example, CycleGAN generates an output image from an input image, and the inter-class pattern is integrated into the procedure of the model and is applied automatically in the forward pass of the neural network. C2C methods work with the inter-class pattern directly. For example, the method to be presented in this study uses the feature differences between two samples as both inputs and expected outputs of an variational autoencoder. This difference provides more flexibility to introduce creativity. More specifically, our approach can choose an inter-

class pattern as the modification and also choose a sample to apply this modification.

Measurement of Creativity

Franceschelli and Musolesi (2021) survey multiple creativity measures implemented via machine learning algorithms. Our method, GOF/TOM ("generated on few, tested on many"), fits within the formalization of the generate and test framework (Toivonen and Gross 2015), in which the system uses a generative function to generate samples and an evaluation function to evaluate the samples. The authors describe three works (Varshney et al. 2013; Norton, Heath, and Ventura 2010; Morris et al. 2012) that fit in this framework.

Varshney et al. (2013) proposes a system that generates creative recipes. The novelty of a recipe is evaluated based on Bayesian surprise, the difference between a prior probability distribution of recipe and a posterior probability distribution after a new recipe is observed. The value of a recipe is evaluated by a model predicting pleasantness of scent from its ingredients and flavor compounds in those ingredients.

Ritchie (2007) describes that creativity can come from an *inspiring* set, which is a set of usually highly valuable samples used to train or configure the generator. Gervás (2011) expands on this by splitting an inspiring set into a learning set, which informs the construction of the generator, and a *reference* set, which is used to evaluate the novelty of generated samples. Similarly, Morris et al. (2012) uses an inspiring set (crockpot recipes) to generate samples via a genetic algorithm and to evaluate the quality of generated sample by training a multilayer perceptron to predict user ratings from a sample.

Creativity Inspired Zero-Shot Learning

The goal of a zero-shot learning task for classification is to train on seen classes and then predict the class label of a sample from an unseen class (or samples from seen and unseen classes in generalized zero-shot learning). Elhoseiny and Elfeki (2019) implemented a creativity inspired zero-shot learning algorithm. In that work, both visual and semantic information are available for seen classes but only semantic descriptions are available for unseen classes. The authors introduce a creativity inspired zero-shot learning method which trains a discriminator to differentiate between real and fake images and also classifies an image into seen classes. It also trains a generator to generate realistic images based on texts describing seen classes and realistic yet hard-to-classify (high entropy over seen classes) images from "hallucinated" texts. This training goal drives the generator to explore the latent space of texts with two objectives: 1) Generate samples that are realistic; 2) Generate samples from hallucinated texts. These properties enable the generator to generate realistic images based on the descriptions of unseen classes.

Creative Sample Generation with a Class-to-class Variational Autoencoder

This paper proposes a class-to-class variational autoencoder (C2C-VAE) approach to generating creative samples in the context of classification—that is, generating creative samples falling within desired categories (e.g., generating images for creative versions of a given letter of the alphabet). For this task, the C2C-VAE approach learns the difference pattern between pairs of samples of two different classes. Four spaces are involved in this task: the original sample space $L1$, the feature space $L2$, the space of feature differences $L2'$, and the space of feature difference embeddings $L3$. As a precondition, C2C-VAE relies on a means to transition between $L1$ and $L2$, more specifically, to extract a feature $f(s)$ from a sample s and also to recover a sample s from feature $f(s)$.

For our testbed system (illustrated in Figure 1), we train a traditional VAE on the training data and use the encoder f to extract features and the decoder f' to recover samples. Given a pair of samples from different classes, $s_1 \in C_1$ and $s_2 \in C_2$, a VAE can extract their features $f(s_1)$ and $f(s_2)$ in the space $L2$. The feature difference $f_\Delta(s_1, s_2)$ in the space $L2'$ can be calculated as $f_\Delta(s_1, s_2) = f(s_1) - f(s_2)$, an element-wise subtraction between two feature vectors. The space $L2'$ can be thought of as the complement of the space $L2$, whence the name $L2'$.

C2C-VAE is based on the class-to-class assumption that the feature differences (in space $L2'$) from one class to another class follow a consistent pattern. This pattern can be represented as an embedding vector (or a distribution of embeddings) in another latent space $L3$. The C2C-VAE is itself another variational autoencoder with an encoder g that encodes a feature difference f_Δ in $L2'$ to an embedding $g(f_\Delta)$ in $L3$ and a decoder g' that decodes an embedding $g(f_\Delta)$ back to a feature difference f_Δ. Note that both the VAE encoder f and the C2C-VAE encoder g are *variational encoders* that encode an input to a distribution of embeddings. This is to ensure regularity in the latent space $L2$ and $L3$. For simplicity, the encoder f (or g) can be thought of as extracting one feature (or a feature difference embedding) from an input.

Because there exist multiple pairs of classes and each pair $C_i - C_j$ has its own unique pattern, a C2C-VAE either learns only one pattern, reflecting a specific pair of classes $C_i - C_j$, or learns multiple patterns by conditioning its encoder and decoder with extra parameters indicating C_i and C_j. In our tests, we take the later approach. Therefore, the C2C-VAE presented here is actually a *conditional* variational autoencoder (Sohn, Lee, and Yan 2015).

Training a C2C-VAE

A C2C-VAE is trained using the following procedure:

- Train a traditional VAE with encoder f and decoder f'.
- Assemble training pairs: Randomly collect 10000 pairs of samples during every training epoch. For each sample pair s_i (of class C_i) and s_j (of class C_j), extract their features $f(s_i)$ and $f(s_j)$, calculate their feature difference $f_\Delta(s_i, s_j) = f(s_i) - f(s_j)$.

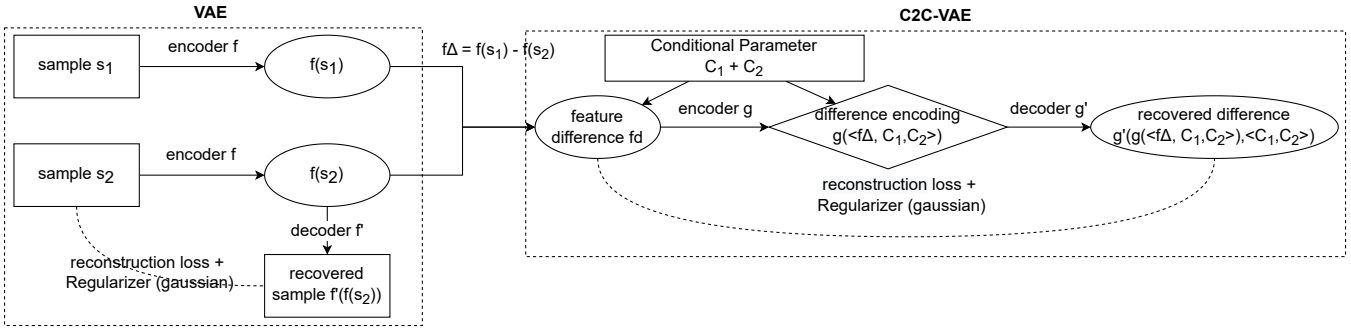


Figure 1: A VAE extracts (recovers) a feature from (to) a sample. A C2C-VAE extracts (recovers) an embedding from (to) a feature difference. $L1$ entities are marked with rectangles, $L2$ and $L2'$ entities with circles, and $L3$ entities with diamonds.

- Train C2C-VAE with the vector $< f_\Delta, C_i, C_j >$: The encoder g (conditioned on the class pairs) learns to encode the input to embedding $g(< f_\Delta, C_i, C_j >)$. The decoder g' (also conditioned on the class pairs) learns to decode the embedding back to $f'_\Delta = g'(g(< f_\Delta, C_i, C_j >), < C_i, C_j >)$. Both g and g' are trained to minimize the reconstruction loss and the KL-divergence between the prior distribution (in this case, a Gaussian distribution) and the distribution of embeddings $g(< f_\Delta, C_i, C_j >)$. The loss function for C2C-VAE is:

$$loss = ||g'(g(< f_\Delta, C_i, C_j >), < C_i, C_j >) - f_\Delta||^2 + KL[g(< f_\Delta, C_i, C_j >), N(0,1)]$$

Just as a VAE can generate new samples of the original space $L1$. A C2C-VAE can generate new feature differences in $L2'$, which in turn can be used to modify features in $L2$. The modified features can then be recovered as new samples in $L1$. More specifically, A C2C-VAE can be used to generate a new sample s_j of a target class C_j by adapting an existing sample (called as the source sample) using the following procedure:

- Choose a source sample s_i of class C_i in the space $L1$. Get its feature $f(s_i)$ in the space $L2$.
- Sample an embedding $g(< f_\Delta, C_i, C_j >)$ from the Gaussian distribution $N(0,1)$ in space $L3$. Decode this embedding to get a feature difference $f'_\Delta = g'(g(< f_\Delta, C_i, C_j >), < C_i, C_j >)$ in the space $L2'$.
- Apply the feature difference f'_Δ in $L2'$ to $f(s_i)$ in $L2$, to get the target sample feature $f(s_j) = f(s_i) - f'_\Delta$ in $L2$.
- Decode the target sample feature $f'(f(s_j))$ to the sample space $L1$

This procedure touches each of Boden's three proposed aspects of creativity. It is *exploring* the space $L3$ of difference patterns, *combining* a feature difference with another feature in $L2$, and *transforming* the sample in $L1$.

In Boden's original definition, the boundary between exploratory and transformational creativity is blurred. For this reason, Wiggins (2006) refines Boden's original framework by identifying two rule sets defining the conceptual space (in our previous terminology, this is $L1$, perhaps even $L2$.): the rule set $\mathscr{R}$ that constrains the space and the rule set $\mathscr{T}$ that traverses the space. Transformational creativity can emerge from either transforming $\mathscr{R}$, leading to a new conceptual space, or $\mathscr{T}$, leading to new traversal in the same conceptual space. Under Wiggins' definition, C2C-VAE is applying $\mathscr{T}$-transformation, where samples in the original conceptual space are modified by difference patterns sampled by C2C-VAE.

From the perspective Boden's three criteria of creativity, we hypothesize that C2C-VAE can generate realistic feature differences leading to valuable (within-category) samples, thanks to the regularity offered by both VAE and C2C-VAE. C2C-VAE provides three means for achieving novelty: 1) sampling in the space $L3$, allowing variation in the feature difference; 2) diversifying the source sample and adapting from different samples or classes; 3) sampling in the space $L2$, allowing variation in the feature of the source sample. Our testbed system focuses on the first means for creativity.

Evaluating Creativity for One-Shot Classification Domains

We consider automated evaluation of creativity of generated samples in a classification domain in which a generator can learn to generate samples from data and a classifier can learn to classify samples. We propose a creativity evaluation approach named GOF/TOM (for "generate on few, test on many"). The approach is applicable to multiple forms of generators (e.g. VAE, GAN). Although we describe it in a classification task domain (as we focus on evaluating C2C-VAE), this approach could be applied to regression or other task domains as well.

Given a data set, a class is chosen as the target class. Data samples of that class are called target samples. Some or all target samples are removed from the training set, creating a zero/one/few-shot setting (for simplicity, we will ignore the differences between the three settings and refer them as the one-shot setting), where the target data is not available to the generator in its entirety. Then the generator is trained on the trimmed training set. Meanwhile, a separate model (e.g. classifier) is trained on the untrimmed data set. Because the model sees target samples unknown to the generator, we call it the *oracle*. There can exist multiple oracles serving differ-

ent purposes, as in our experimental evaluation.

After training of both the generator and the oracle, the generator is used to generate new target samples. The oracle can facilitate the evaluation of the generated sample on the three aspects of creativity (Boden 1991). We assume that the oracle is implemented using deep learning or other techniques enabling extraction of features for the assessment of novelty and/or surprise.

Value The oracle classifies the generated samples. We consider the generator able to generate valuable samples if it can consistently generate samples of the target class. The accuracy of the generator is the percentage of the generated samples falling into the intended class. High accuracy means highly valuable generator.

Novelty The oracle can extract features of the samples into a latent space. The latent space needs to be smooth so that similar samples are close together and different samples are distant from each other. The generated samples are novel if their extracted features show variety. For our experiment, we use the activation of the embedding layer of an VAE as the feature values extracted for each sample. The variance of the features is used as the measure of the variety of the samples generated.

Surprise A generator achieves *surprising* results if it can consistently generate samples of the target class that are unexpected. Existing measures of surprise are surveyed in Franceschelli and Musolesi (2021). Surprise is beyond the scope of this paper and the capability for C2C-VAE to generate surprising samples is left for future research.

Uniqueness and Benefits

Our evaluation method differs from those mentioned earlier in a few ways: The untrimmed data set is not the same as the *inspiring set* (Ritchie 2007) because the data contained are not necessarily creative; The generator learns from the trimmed data set and its knowledge of the target class is deliberately limited. Even if the untrimmed data set is inspiring, its trimmed version may not be; Instead of using a reference set as in Gervás (2011), the oracle classifier is used to evaluate the samples; Unlike the many evaluations such as in Norton, Heath, and Ventura (2010) and Morris et al. (2012), the oracle classifier does not require user rating data or other human assessment of artistry or creativity. It requires only classification labels, which are more widely available.

Methods (Gervás 2011; Morris et al. 2012) that use some (portion of) inspiring set for the evaluation integrate evaluation within the creative system. The creative system filters generated samples by evaluation. However, GOF/TOM estimates is envisioned as a tool for after-the-fact evaluation of the system's performance.

Caveats

In GOF/TOM, the generator is trained in a one-shot setting and then its generated samples are evaluated by an oracle trained using the untrimmed data set. There are three implications: 1) If the task domain is truly one-shot, then the construction of the oracle is impossible, due to the lack of additional data. 2) GOF/TOM may be less suitable for generators with weaker one-shot learning capability. 3) The method assesses value based on classifications by the oracle and novelty based on features generated by the oracle; such features could potentially be used for assessing surprise as well. The usefulness of all three measures depends on how well the oracle has learned from the training data.

A concern for any automated evaluation of creativity is whether it truly captures the important characteristics. We believe that the use of accuracy as a proxy for value, and variance as a proxy for novelty, is reasonable in the domains used for the evaluation. However, these measures may miss important aspects of creativity for some domains. More work is needed on the measures to apply.

Evaluation

We carried out experiments on two data sets, using the GOF/TOM approach to evaluate the creativity of C2C-VAE. The first data set is the MNIST dataset of hand-written digits from 0 to 9. The second is the fashion-MNIST dataset of 10 classes of clothing and accessories. Each of MNIST and fashion-MNIST is provided with predetermined splits for training (60,000 samples) and testing (10,000 samples) data sets; these were used for training and validation. Each sample is a 28x28 grayscale image, associated with a label from 10 classes. The oracle classifier and the VAE are both trained with the full data set. The two data sets are chosen because a traditional VAE can extract features from them.

In all experiments, each class is successively chosen as the target class. For comparison with C2C-VAE, we also trained a conditional VAE (CVAE). The CVAE is trained to generate a sample conditioned on an additional parameter controlling the class of the sample generated. During testing, both C2C-VAE and CVAE generate samples of the target class.

System Design

The oracle for value is a resnet18 (not pretrained) network, of which the first layer is replaced with a convolutional layer with $(in\_channels = 1, out\_channels = 64, kernel\_size = (7,7), stride = (2,2), padding = (3,3), bias = False)$, and the last layer is replaced with a linear layer with 10 outputs for classifications.

The VAE follows a standard design. The encoder of the VAE is composite of two consecutive convolutional layers $(out\_channels = c, kernel\_size = 4, stride = 2, padding = 1)$ and $(out\_channels = c * 2, kernel\_size = 4, stride = 2, padding = 1)$, where $c = 64$. A linear layer for mean and another linear layer for log variance follow the convolutional layers and extract a distribution of features from the output of the convolutional layers. A feature is a vector of dimension 32. The decoder of the VAE reverses the design of the encoder: It consists of a linear layer and two consecutive convolutional layers. The input and output dimensions are the reverse of their corresponding encoder layers, other parameters being equal. The VAE is trained with standard reconstruction loss and KL-divergence loss: $Loss_{vae} = loss_{recon} + KL\text{-}divergence$

The VAE's encoder f serves as a feature extractor for the C2C-VAE and also as the oracle for novelty. The resnet18

(a) MNIST (b) Fashion-MNIST

Figure 2: Average Samples Constructed by the VAE

classifier can also extract features but the feature space is not as smooth as that of the VAE.

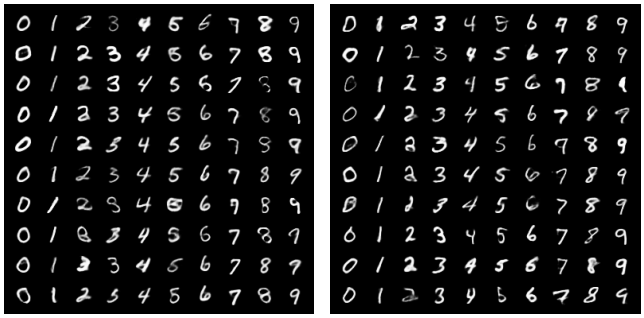
The C2C-VAE has its own encoder and decoder. Given a pair of samples, their features are extracted by the VAE and their class labels are one-hot encoded. The encoder takes the feature difference and the two class labels as input. The input is passed to a fully connected RELU layer with $(out\_features = 32)$. A linear layer for mean and another linear layer for log variance follow and extract a distribution of embeddings, which are of 10 dimensions. The decoder takes an embedding and the two class labels as input. The input is passed to two consecutive linear layers to recover a feature difference similar to the original.

The CVAE has a very similar architecture to the VAE, except it is modified to be conditioned on the class label. Specifically, the encoder and the decoder use the same convolutional layers, but their linear layers that interact with features also take in the class labels as extra inputs.

The C2C-VAE can only generate a new sample by adapting a source sample. We choose an average sample s_{avg} (see Figure 2) from each class C (other than the target class) as the source sample by the following procedure:

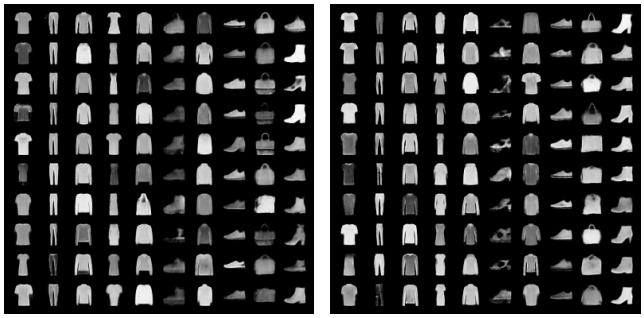
- Select all n samples s_1 - s_n of the class C;
- Calculate the average of their features $avg(\Sigma f(s_i)) = (\Sigma_{i=0}^{n} f(s_i))/n$;
- Use the decoder f' to recover the average sample $f'(avg(\Sigma f(s_i)))$.

In addition to the reconstruction loss, both the C2C-VAE and the CVAE are learning to minimize a KL-divergence loss with a Gaussian distribution ($\mu = 0, \sigma = 1$). This means that they are both trained to project their corresponding input to the Gaussian distribution. Although their inputs and embeddings carry different meanings (The CVAE takes input from $L1$ while the C2C-VAE takes input from $L2'$), we note that their embedding distributions are intended to be the same Gaussian. This also means that we could compare their performance with regard to the standard deviation std of the Gaussian. The experimenter can make either model produce more or less various samples by tuning std, controlling the distribution from which the model is sampling from. The higher std is, the wider the distribution becomes, and the generated samples lose value (accuracy) but gain variety. Note that the C2C-VAE can also introduce additional novelty by altering the source sample, but this comes at a further cost of stability of the results (discussed in the Discussion and Future Work section).



(a) C2C-VAE (b) CVAE

Figure 3: MNIST samples generated by the models trained under normal setting. $std = 1$. Both models demonstrate valuable and various samples.



(a) C2C-VAE (b) CVAE

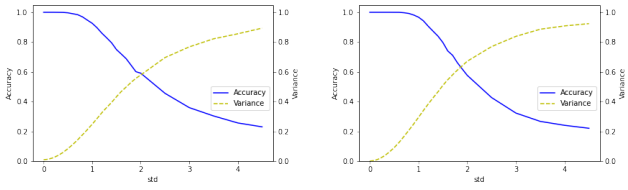
Figure 4: Fashion-MNIST samples generated by the models trained under normal setting. $std = 1$. Both models demonstrate valuable and various samples.

Comparison between the C2C-VAE and the CVAE

Under the Normal Setting Before examining the models under the GOF/TOM setting, we present some results under the normal setting to provide a backdrop for comparison. Under the normal setting, all models are trained on the untrimmed data set. In all figures of generated samples, column j represents the samples generated for class C_j. In all the figures of generated samples by the C2C-VAE, unless otherwise specified, the (i, j), where $i \neq j$, sample is a sample generated by choosing the average sample of class C_i, sampling a feature difference from class i to class j, and applying this feature difference to the chosen sample; Additionally, the (i, i) sample (on the diagonal) is an average sample of class C_i. In all the figures of samples by the CVAE, column j represents samples generated by random sampling in class C_j.

Figures 3 and 4 illustrate that both C2C-VAE and C-VAE produce valuable and varied samples. Because the models have seen various samples of the target class during training, the variety here is not equivalent to novelty (but it will be in GOF/TOM evaluation).

Figure 5 illustrates the tradeoff between value (measured



(a) C2C-VAE (b) CVAE

Figure 5: Under the normal setting in MNIST, both C2C-VAE and VAE trade off accuracy and variance as std is adjusted. Results are similar for the normal setting in fashion-MNIST

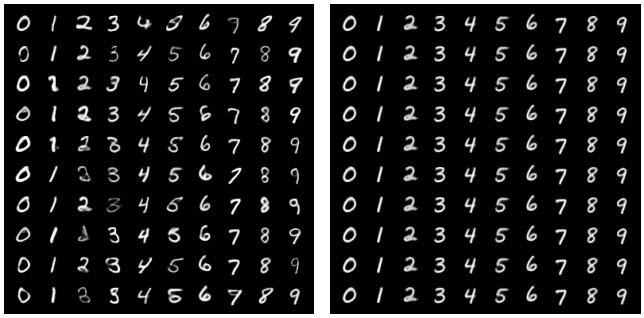
by the accuracy of samples generated judged by the oracle for value) and variety (measured by the variance of the features extracted by the oracle for novelty). Under the normal setting, the two models share similar tradeoffs.

Under the GOF/TOM Setting Our specific implementation of the GOF/TOM setting trims all samples of a target class except one average sample. We choose the average sample to better represent target class. Both CVAE and C2C-VAE are trained with the trimmed data set. During each training epoch, 10% of the training batch is this one-shot sample while 90% is randomly chosen from other samples (CVAE trains on the batch directly while C2C-VAE trains on pairs from the batch). This design counters the imbalanced classes caused by trimming. Therefore the CVAE learns about the target class from only the average sample, while the C2C-VAE learns from pairs of this sample and samples of other classes.

In contrast to the figures of generated samples under the normal setting, the figures presented in this section follow an additional rule: Column j is generated by models which are trained under the one-shot setting, where all samples except the average sample of class j are removed during training. Because the models have only seen a single sample of the target class during training, any variety of generated samples of the target class is equivalent to novelty.

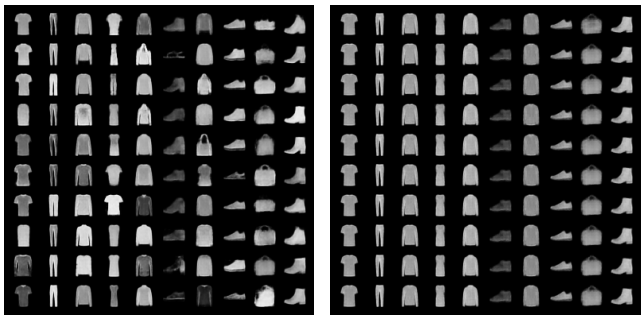
When $std = 1$, the C2C-VAE generates novel—thus creative—samples while the CVAE can only generate similar samples, as shown in Figures 6 and 7.

Figures 8 and 9 illustrate the tradeoff between value (measured by the accuracy of samples generated judged by the oracle for value) and novelty (measured by the variance of the features extracted by the oracle for novelty). C2C-VAE exhibits an accuracy and variance tradeoff as std is tuned. For a given level of $variance$, CVAE needs a bigger std change at a bigger cost of $accuracy$ than C2C-VAE. For example, CVAE needs $std = 4$ to gain the same variance as C2C-VAE for $std = 1$ in MNIST, and $std = 4.5$ to gain the same variance as C2C-VAE ($std = 1$) in fashion-MNIST. When std is so high, the quality of the images is very poor, as shown in Figure 10.



(a) C2C-VAE (b) CVAE

Figure 6: MNIST samples generated by the models trained under one-shot setting. $std = 1$. Intuitively, C2C-VAE generates more creative samples.



(a) C2C-VAE (b) CVAE

Figure 7: Fashion-MNIST samples generated by the models trained under one-shot setting. $std = 1$. Intuitively, C2C-VAE generates more creative samples.

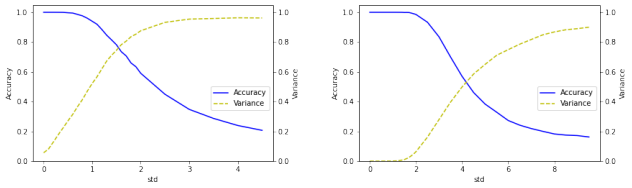
Discussion and Future Work

Conditions for Success of the C2C-VAE Method

The applicability of the C2C-VAE method depends on three conditions: (1) There exists a VAE that can extract features of samples, (2) there exists a C2C-VAE that can extract embeddings of feature differences of two classes, and (3) the generated feature differences can be applied to a chosen sample. Condition (1) depends on properties of VAEs and is beyond the scope of this paper.

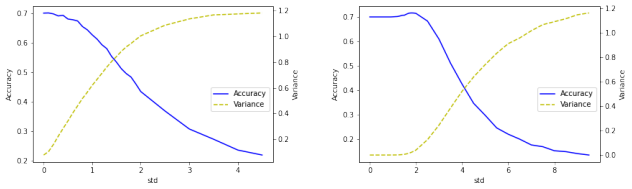
Condition (2) can fail if the feature differences between two classes do not conform to a single pattern. When two classes each have wide distributions, the difference between the two distribution can have very high variation, decreasing effectiveness of the C2C approach (Ye 2018b). For example, drawings in the Quick, Draw! dataset vary considerably within classes. For example, a cat or dog may be drawn with a head only, or with a body, or with limbs and a tail.

Even if condition (2) holds, C2C-VAE can be sensitive to the choice of source sample, causing the failure of condition (3). C2C-VAE can generate creative samples by generating from different source samples, while at the risk of generating bad samples (see Figure 11).



(a) C2C-VAE (b) CVAE

Figure 8: Under the one-shot setting in MNIST, C2C-VAE can trade off accuracy and variance when std is tuned. For a given level of $variance$, CVAE needs a bigger std change at a bigger cost of $accuracy$ than C2C-VAE.



(a) C2C-VAE (b) CVAE

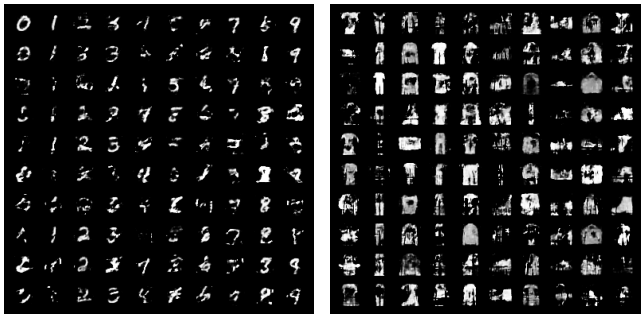
Figure 9: Under the one-shot setting in fashion-MNIST, C2C-VAE can trade off accuracy and variance when std is tuned. For a given level of $variance$, CVAE needs a bigger std change at a bigger cost of $accuracy$ than C2C-VAE.

In the procedure for generating new samples using C2C-VAE, the choice of a source sample s_i and the choice of a feature difference embedding and its subsequently induced feature difference f'_Δ are currently two independent choices. There could (and perhaps even should) exist some dependency between the two choices. As a future direction, both conditions (2) and (3) may be resolved by conditioning the C2C-VAE on the source sample.

Relationship to CycleGAN

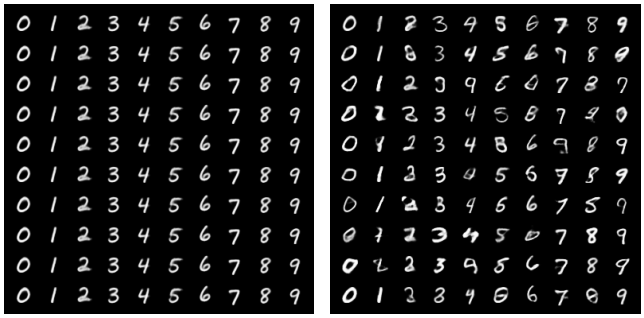
C2C-VAE and CycleGAN are completely different techniques but share many foundational assumptions. CycleGAN assumes a pattern between two classes and trains translation functions (generators) on all possible pairs between two classes to learn it. The reconstruction loss of C2C-VAE (that the feature difference can be recreated) corresponds to the cycle-consistency loss of CycleGAN (that the sample can be recovered). The reconstruction loss of the VAE which C2C-VAE depends upon for feature extraction (a realistic sample can be reconstructed from a feature) corresponds to the adversarial loss of CycleGAN (the recovered sample is realistic). The two models are similar in their foundations, but C2C-VAE works with the space $L2'$ while CycleGAN works with the space $L1$ (and arguably $L2$).

GOF/TOM can benefit from GAN. In its current design, the oracle for value often classifies a generated sample confidently with high activation score even if it is of poor quality by human perception. As a future direction, the oracle might be integrated with a discriminator of GAN to better distinguish poor samples.



(a) MNIST ($std = 4$) (b) Fashion-MNIST ($std = 4.5$)

Figure 10: To gain variance, CVAE requires greatly increases std, sacrificing quality of generated samples



(a) Generated by Modifying Average Samples (b) Generated by Modifying Random Samples

Figure 11: If the source samples are randomly selected, C2C-VAE might generate bad samples. Here $std = 0$, so variance is solely due to the choice of source samples.

Conclusion
Creativity from Inter-Class Patterns

Network-based models provide exciting mechanisms for modeling creativity in AI systems. Existing work on generative methods for creativity can be seen as oriented primarily towards the conceptual space of samples, while C2C-VAE exploits the relationship between samples. If existing approaches look at the foreground of conceptual space $L2$, C2C-VAE looks at the background $L2'$, in order to bring that background to bear as additional information to facilitate creative sample generation in one-shot settings. The presented experiments support that for a creative image generation task, C2C-VAE can achieve high novelty—variance in generated samples—while maintaining accuracy.

Acknowledgements

We acknowledge support from the Department of the Navy, Office of Naval Research (Award N00014-19-1-2655).

Author Contributions

The main tasks and contributing authors are listed below in order of their contribution in each task.

- Conceptualization: Xiaomeng Ye, David Leake, Ziwei Zhao, David Crandall
- Writing: Xiaomeng Ye, David Leake, Ziwei Zhao, David Crandall
- Algorithm Design: Xiaomeng Ye, Ziwei Zhao
- Programming: Ziwei Zhao, Xiaomeng Ye (earlier versions)
- Experimentation and Results: Ziwei Zhao, Xiaomeng Ye
- Review: David Leake, Xiaomeng Ye, David Crandall, Ziwei Zhao

References

Boden, M. A. 1991. *The Creative Mind: Myths and Mechanisms*. USA: Basic Books, Inc.

Broad, T.; Berns, S.; Colton, S.; and Grierson, M. 2021. Active divergence with generative deep learning - A survey and taxonomy. *CoRR* abs/2107.05599.

Draper, S. 2010. Creativity. https://www.psy.gla.ac.uk/~steve/best/creative.html. Accessed: 2022-05-08.

Elgammal, A. M.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017. CAN: creative adversarial networks, generating "art" by learning about styles and deviating from style norms. *CoRR* abs/1706.07068.

Elhoseiny, M., and Elfeki, M. 2019. Creativity inspired zero-shot learning. *CoRR* abs/1904.01109.

Franceschelli, G., and Musolesi, M. 2021. Creativity and machine learning: A survey. *CoRR* abs/2104.02726.

Gervás, P. 2011. Dynamic inspiring sets for sustained novelty in poetry generation. In Ventura, D.; Gervás, P.; Harrell, D. F.; Maher, M. L.; Pease, A.; and Wiggins, G. A., eds., *Proceedings of the Second International Conference on Computational Creativity, ICCC 2011, Mexico City, Mexico, April 27-29, 2011*, 111–116. computationalcreativity.net.

Ha, D., and Eck, D. 2017. A neural representation of sketch drawings. *CoRR* abs/1704.03477.

Karras, T.; Laine, S.; and Aila, T. 2018. A style-based generator architecture for generative adversarial networks. *CoRR* abs/1812.04948.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

LeCun, Y., and Cortes, C. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/.

Morris, R. G.; Burton, S. H.; Bodily, P.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *ICCC*.

Nobari, A. H.; Rashad, M. F.; and Ahmed, F. 2021. Creativegan: Editing generative adversarial networks for creative design synthesis. *CoRR* abs/2103.06242.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing appreciation in a creative system. In *ICCC 2010*.

NVIDIA. 2021. AI artist Helena Sarin. https://www.nvidia.com/en-us/research/ai-art-gallery/artists/helena-sarin/. Accessed: 2022-02-17.

Pan, Z.; Yu, W.; Yi, X.; Khan, A.; Yuan, F.; and Zheng, Y. 2019. Recent progress on generative adversarial networks (gans): A survey. *IEEE Access* 7:36322–36333.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Roberts, A.; Engel, J.; Raffel, C.; Simon, I.; and Hawthorne, C. 2018a. Musicvae: Creating a palette for musical scores with machine learning. https://magenta.tensorflow.org/music-vae. Accessed: 2022-02-17.

Roberts, A.; Engel, J. H.; Raffel, C.; Hawthorne, C.; and Eck, D. 2018b. A hierarchical latent vector model for learning long-term structure in music. *CoRR* abs/1803.05428.

Sbai, O.; Elhoseiny, M.; Bordes, A.; LeCun, Y.; and Couprie, C. 2018. DeSIGN: Design inspiration from generative networks. *CoRR* abs/1804.00921.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Toivonen, H., and Gross, O. 2015. Data mining and machine learning in computational creativity. *WIREs Data Mining and Knowledge Discovery* 5(6):265–275.

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Bhattacharjya, D.; Schörgendorfer, A.; and Chee, Y. 2013. A big data approach to computational creativity. *CoRR* abs/1311.1213.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458. Creative Systems.

Wiggins, G. A. 2021. Creativity and consciousness: Framing, fiction and fraud. In de Silva Garza, A. G.; Veale, T.; Aguilar, W.; and y Pérez, R. P., eds., *Proceedings of the Twelfth International Conference on Computational Creativity, México City, México (Virtual), September 14-18, 2021*, 182–191. Association for Computational Creativity (ACC).

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR* abs/1708.07747.

Ye, X.; Leake, D.; Huibregtse, W.; and Dalkilic, M. 2020. Applying class-to-class siamese networks to explain classifications with supportive and contrastive cases. In *International Conference on Case-Based Reasoning*, 245–260. Springer.

Ye, X.; Leake, D.; Jalali, V.; and Crandall, D. J. 2021. Learning adaptations for case-based classification: A neu-

ral network approach. In Sánchez-Ruiz, A. A., and Floyd, M. W., eds., *Case-Based Reasoning Research and Development*, 279–293. Cham: Springer International Publishing.

Ye, X. 2018a. The enemy of my enemy is my friend: Class-to-class weighting in k-nearest neighbors algorithm. In *Proceedings of the Thirty-First International Florida Artificial Intelligence Research Society Conference, FLAIRS 2018*, 389–394.

Ye, X. 2018b. The enemy of my enemy is my friend: Class-to-class weighting in k-nearest neighbors algorithm. In *FLAIRS Conference*.

Zhu, J.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR* abs/1703.10593.