Creativity and Machine Learning: A Survey

GIORGIO FRANCESCHELLI, Alma Mater Studiorum Università di Bologna, Bologna, Italy
MIRCO MUSOLESI, University College London, London, United Kingdom of Great Britain and Northern Ireland and Alma Mater Studiorum Università di Bologna, Bologna, Italy

There is a growing interest in the area of machine learning and creativity. This survey presents an overview of the history and the state of the art of computational creativity theories, key machine learning techniques (including generative deep learning), and corresponding automatic evaluation methods. After presenting a critical discussion of the key contributions in this area, we outline the current research challenges and emerging opportunities in this field.

CCS Concepts: • **Computing methodologies** → **Machine learning** • **Applied computing**;

Additional Key Words and Phrases: Computational creativity, machine learning, generative deep learning, creativity evaluation methods

1 INTRODUCTION

The connection between creativity and machines is as old as computer science itself. In 1842, Lady Lovelace, an English mathematician and writer, who is recognized by many as the first computer programmer, issued what is now known as "Lovelace's Objection" [274]. She stated that the Analytical Engine (the digital programmable machine proposed by Charles Babbage [9]) *"has no pretensions to originate anything. It can do whatever we know how to order it to perform"* [184]. Indeed, in the centuries that followed, numerous projects and studies have been undertaken with the aim of designing machines capable of "originating something" [51, 57, 120, 155, 183, 214]. We have witnessed the emergence of a specialized field in computer science, namely Computational Creativity [41], which concerns the study of the relationship between creativity and artificial systems [56, 289].

In this context, the adoption of **deep learning (DL)** techniques has led to substantial breakthroughs in recent years. Vast computational power and very large amounts of available data are at the basis of the increasing success of deep generative models (i.e., generative models based on DL [83]). Indeed, generative deep learning technologies have been used to write newspaper

Authors' Contact Information: Giorgio Franceschelli, Alma Mater Studiorum Università di Bologna, Bologna, Emilia-Romagna, Italy; e-mail: giorgio.franceschelli@unibo.it; Mirco Musolesi, University College London, London, London, United Kingdom of Great Britain and Northern Ireland and Alma Mater Studiorum Università di Bologna, Bologna, Emilia-Romagna, Italy; e-mail: m.musolesi@ucl.ac.uk.

articles,[1] generate human faces [144] and voices [166], design drugs and proteins [139], and even create artworks sold for a hundred thousand dollars.[2] While it is apparent that current technologies are able to generate impressive outputs, at the same time it is also possible to argue that they cannot be considered *creative* in general [27]. In fact, the goal of generative deep learning is to produce synthetic data that closely resemble real ones fed in input [83]. On the other hand, creativity involves novelty and diversity [195]. While for some problems mere content generation [282] might be sufficient, for other tasks, e.g., in the Arts, the ability to create different (but still valuable) outputs is essential: a creative model can find practical applications in the arts and industrial design as support for artists, content creators, designers and researchers, just to name a few. Moreover, generating more diverse data might mitigate legal and ethical issues related to content reproduction [112, 287].

Goal and contributions of the survey. The goal of this survey is to present and critically discuss the state of the art in generative deep learning from the point of view of machine creativity. Moreover, to the best of our knowledge, this is the first survey that explores how current DL models have been or can be used as a basis for both generation (i.e., producing creative artifacts) and evaluation (i.e., recognizing creativity in artifacts). The contribution of this survey can be summarized as follows. After a brief overview of the meaning and definitions of creativity in Section 2, Section 3 presents an in-depth analysis of generative deep learning through the introduction of a new taxonomy and a critical analysis of machine creativity. Then, several **machine learning (ML)**-based methodologies for evaluating creativity are presented and discussed in Section 4. Finally, Section 5 concludes the paper, outlining open questions and research directions for the field.

Related surveys. We now provide an overview of other surveys in areas related to the present work. For readers interested in a survey on deep generative models, we recommend [24, 110]; for an analysis of the state of the art in evaluation in computational creativity, [153] is an essential reading; for a review on AI and creativity in general, we recommend [233]; for a practical view of generative deep learning, we suggest [83]; finally, for an in-depth examination of artistic AI works (also in human-computer co-creations [108]), [187] is a very comprehensive source of information.

2 DEFINING CREATIVITY

Creativity has been studied for decades, and yet, there is no agreement about its definition. More than one hundred definitions have been provided [5, 271], and the number is still growing. In other words, we can say that creativity is a suitcase word, i.e., people conflate multiple meanings into it [188]. Nonetheless, some concepts are nowadays widely accepted. One of them is the possibility of studying creativity from four different perspectives: *person*, *press*, *process*, and *product* [227]. These have also been studied in computational creativity [138]. However, the focus has traditionally been on the *product* dimension. Indeed, the idea is that we study creativity without considering the inner being of the creator (person), the relation with the environment (press), or if the process maps to the steps a human goes through in the act of creation (e.g., [6]). Even if they are important, we focus on aspects of creative work in relation to the output (product) itself, and on aspects of the generative process that may or may not lead to a creative product.

For this reason, we consider Boden's three criteria for studying machine creativity, defined as *"the ability to come up with ideas or artifacts that are* new, surprising and valuable*"* [22]. In particular, value refers to utility, performance, and attractiveness [179]; it is related with both the quality of the production, and its acceptance by the society. Novelty refers to the dissimilarity between

---

the produced artifact and other examples in its class [229]. Finally, surprise refers to the degree to which a stimulus disagrees with expectation [17]. We underpin our analysis on Boden's three criteria since they have been widely adopted. Boden also suggests that three forms of creativity can be identified [22] to describe how a novel and surprising product is obtained. The three forms of creativity are *combinatorial*, *exploratory* and *transformational*, ordered by increasing rarity and produced surprise. Combinatorial creativity is about making unfamiliar combinations of familiar ideas, e.g., analogies in textual forms or collages in the visual arts. Exploratory creativity involves the exploration of the conceptual space defined by the cultural context considered, e.g., inventing a new type of cut for fries. Transformational creativity involves changing that space in a way that allows new and previously inconceivable thoughts to become possible, as it has been for free verse in poetry or abstract painting in art.

Finally, it is worth noting that Boden also identified four different questions that emerge when studying computational creativity. These are referred to as Lovelace questions, because many people would respond to them by using Lovelace's objection. The first question is whether computational ideas can help us understand how human creativity works. The second is whether computers could ever do things that at least appear to be creative. The third is whether a computer could ever appear to recognize creativity. Finally, the fourth is whether computers themselves could ever really be creative, i.e., with the originality of the products only deriving from the machine itself [22]. While the first one is studied in Boden's work, we will provide the reader with an overview of the techniques that can possibly be used to answer the second (Section 3) and the third (Section 4). With respect to the fourth, Boden states that *"it is not a scientific question as the others are, but in part a philosophical worry about "meaning" and in part a disguised request for a moral political decision"*. We agree with this position, but we hope that our survey will provide the reader with elements for answering the fourth one as well.

3 GENERATIVE MODELS

A generative model can be defined as follows: given a dataset of observations X, and assuming that X has been generated according to an unknown distribution p_{data}, a generative model p_{model} is a model able to mimic p_{data}. By sampling from p_{model}, observations that appear to have been drawn from p_{data} can be generated [83]. Generative deep learning is just the application of deep learning techniques to form p_{model}.

At first glance, this definition appears to be incompatible with those presented in Section 2. Indeed, mimicry is the opposite of novelty. However, what a generative model should aim at mimicking is the underlying distribution representative of the artifacts, and not the specific artifacts themselves; in other words, it should aim at learning the conceptual space defined by the cultural context considered. A generative model can be said to exhibit combinatorial creativity if it can sample new and valuable works that are combinations of real data and exploratory creativity if the works actually differ from real ones. Vice versa, transformational creativity emerges if and only if the distribution for sampling diverges in some way from the underlying one (e.g., due to a different training process, by altering the distribution after learning, or by changing the sampling technique). In summary, what matters is how the space of solutions is learned and how artifacts are sampled from it.

In this section, we aim at studying the level of creativity of existing generative deep learning models. Following the discussion above, we analyze how the models learn their spaces of solutions and how the observations are generated from them. A new generative deep learning taxonomy is then introduced based on the different training and sampling techniques at the basis of each method. Figure 1 provides a summary of the seven generative classes considered in this survey. Since our focus is on machine creativity, we do not discuss the implementation details of each
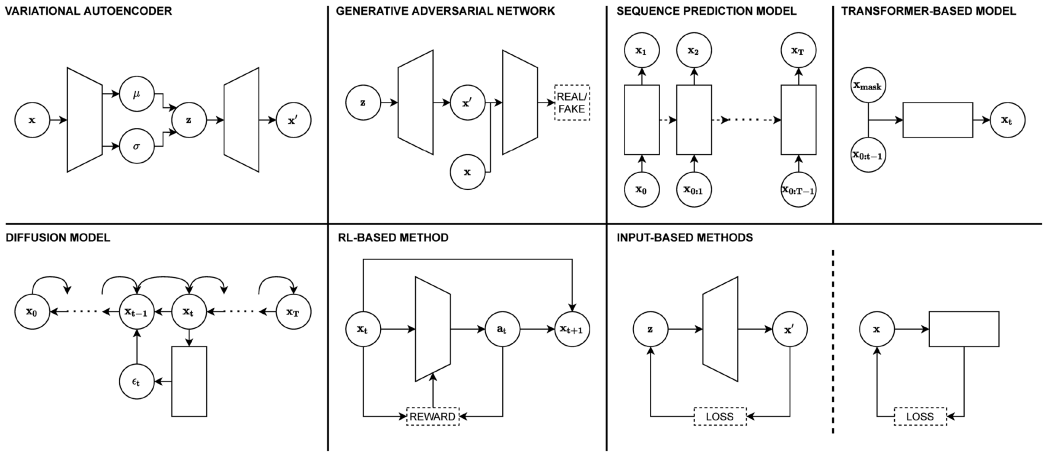
Fig. 1. A schematic view of the seven classes of generative learning methods presented in this survey. Top, left to right: Variational Autoencoder (3.1), with a decoder generating $\mathbf{x}'$ given a latent vector $\mathbf{z}$, and an encoder representing $\mathbf{x}$ into a latent distribution; Generative Adversarial Network (3.2), with a generator to produce $\mathbf{x}'$, and a discriminator to distinguish between real $\mathbf{x}$ and synthetic $\mathbf{x}'$; Sequence prediction model (3.3), with a generator to output $\mathbf{x}$ one token after the other given in input previous tokens; Transformer-based model (3.4), with a Transformer outputting $\mathbf{x}$ one token after the other given in input previous tokens, or a masked version of $\mathbf{x}$. Bottom, left to right: Diffusion model (3.5), with a model to learn an error ϵ, which is used to incrementally reconstruct $\mathbf{x_0}$; Reinforcement Learning (RL)-based method (3.6), with a generative model acting (i.e., progressively generating $\mathbf{x}$) to maximize a given reward function; Input-based methods (3.7), with an input optimized by a given loss. The input can be a vector $\mathbf{z}$ given to a generative model to obtain the desired output, or directly a product $\mathbf{x}$ becoming the desired output.

class of methods. We instead present the *core concepts* at the basis of each class; some relevant *examples of models*; potential *applications*; and a *critical discussion* evaluating the level of machine creativity considering the definitions above. As a final remark, it is worth noting that we limit our examples to the Arts (e.g., poems, music, or paintings). Indeed, generative learning can be applied to design [95, 180]; game content generation (see [172] for a comprehensive survey); recipes [193, 280]; scientific discovery [52, 241]; and in general to any activity, which has a non-trivial solution [34].

3.1 Variational Auto-Encoders

3.1.1 Core Concepts. A **Variational Auto-Encoder (VAE)** [150, 226] is a learning architecture composed of two models: an encoder (or recognition model) and a decoder (or generative model). The former compresses high-dimensional input data into a latent space, i.e., a lower-dimensional space whose features are not directly observable, yet provide a meaningful representation. The latter decompresses the representation vector back to the original domain [83]. Classic autoencoders directly learn to represent each input in a latent representation vector. Conversely, VAEs learn a (Gaussian) distribution over the possible values of the latent representation, i.e., the encoder learns the mean and the (log of the) variance of the distribution.

VAEs are trained by optimizing two losses: the reconstruction loss and the regularization loss. The former is the log-likelihood of the real data $\mathbf{x}$ from the decoder given their latent vectors $\mathbf{z}$, i.e., it is the error of the decoder in reconstructing $\mathbf{x}$. The latter is the **Kullback-Leibler (KL)** divergence between the distribution learned by the encoder and a prior distribution, e.g., a Gaussian. Notably, the latent vector $\mathbf{z}$ in input to the decoder is obtained by means of the so-called

reparameterization trick, i.e., by sampling from the distribution defined by the mean and the variance. Without it, sampling would induce noise in the gradients required for learning [151].

The mathematical derivation of the whole loss has its roots in variational inference [135]. Indeed, VAEs can be seen as an efficient and stochastic variational inference method, in which **neural networks (NNs)** and stochastic gradient descent are used to learn an approximation (i.e., the encoder) of the true posterior [87]. In VAEs, similar high-dimensional data are mapped to close distributions. This makes it possible to sample a random point **z** from the latent space, and still obtain a comprehensible reconstruction [83]. On the other hand, VAE tends to produce blurred images [307]. It may also happen that high-density regions under the prior have a low density under the approximate posterior, i.e., these regions are not decoded to data-like samples [7]. Finally, the objective can lead to overly simplified representations without using the entire capacity, obtaining only a sub-optimal generative model [37].

3.1.2   Examples of Models. Several models based on VAEs have been proposed [151] in recent years. In the following, we focus on those relevant to our discussion on machine creativity. In β-VAE [113], a parameter β is used to scale the magnitude of the regularization loss, which allows a better disentanglement of the latent space [38]. Another example is VAE-GAN [156], which merges VAE and **Generative Adversarial Networks (GAN**; see Section 3.2) [98]. This is done by treating the decoder as the generator of the GAN, thus training it by means of the GAN loss function. This leads to the generation of substantially less blurred images. Similarly, **Adversarially Learned Inference (ALI)** [71] merges VAE and GAN by asking the discriminator to distinguish between pairs of real data (and their latent representations) and pairs of sampled representations and synthetic data. Instead, **Adversarial Autoencoders (AAE)** [181] substitute the regularization loss with a discriminative signal, where the discriminator has to distinguish between random latent samples and encoded latent vectors. Another way to address the problem of "sample blurriness" is with PixelVAE [105], where the autoregressive PixelCNN [276, 277] is used as the decoder. In [26], to deal with sequential data such as texts, where generation requires more steps, the encoder learns to produce a latent representation of a sentence, while the recurrent neural network RNN-based decoder learns to reproduce it word after word. However, VAE can also generate text by means of convolution and deconvolution [243]. To solve the problem of low-density regions, the authors of [7] propose an energy-based model called **noise contrastive prior (NCP)**, trained by contrasting samples from the aggregate posterior to samples from a base prior. Finally, another interesting model is **Vector Quantised-VAE (VQ-VAE)** [278]; in this case, the encoder outputs discrete, rather than continuous, codes, and the prior is learned rather than static.

3.1.3   Applications. VAEs can be used for semi-supervised classification to provide an auxiliary objective, improving the data efficiency [149, 175]; to perform iterative reasoning about objects in a scene [77]; to model the latent dynamics of an environment [286]. Of course, VAEs have also been used to generate synthetic data, including for conditional generation. For example, a layered foreground-background generative model can be used to generate images based on both the latent representation and a representation of the attributes [296]. In [109] the latent space of a VAE is trained on chemical structures by means of gradient-based optimization toward certain properties (see Section 3.7). AAEs have also been applied to the same problem [140]. Finally, another interesting application of VAE is **Deep Recurrent Attentive Writer (DRAW)** [101]. DRAW constructs scenes in an iterative way, by accumulating changes emitted by the decoder (then given to the encoder in input). This allows for iterative self-corrections and a more natural form of image construction. RNNs and attention mechanism are used to consider previous generations and to decide at each time step where to focus attention, respectively.

3.1.4   Critical Discussion. Models based on VAEs can be considered as an example of exploratory creativity. The latent space is learned with the goal of representing data in the most accurate way. The random sampling performed during generation is therefore an exploration of that space: regions not seen during training can be reached as well, even though they can lead to poor generation [7] and some more complex variants may be needed, as discussed. On the other hand, there is no guarantee that the results will be valuable, novel, or surprising. There is no guarantee that the generation from random sampling is of good quality or diverse from training data. Indeed, given their characteristics, VAEs discourage novelty in a sense. In particular, diversity could be achieved in theory using VAEs and gradient-based optimization techniques, such as those presented in [109], with novelty and surprise as target properties. We will discuss these aspects in Section 3.8.

3.2   Generative Adversarial Networks

3.2.1   Core Concepts. A Generative Adversarial Network [98] is an architecture composed by two networks: a generative model and a discriminative model. The latter learns to distinguish between real samples and samples generated by the former. In parallel, the former learns to produce samples from random noise vectors such that they are recognized as real by the latter. This competition drives both models to improve until the generated samples are indistinguishable from the original ones.

The adversarial training allows the generator to learn to produce seemingly real samples from random noise without being exposed to data. The simplicity of the idea and the quality of results are at the basis of the success of GANs. However, few limitations exist. For instance, GAN can suffer from mode collapse, where the generator only learns to produce a small subset of the real samples [186]. In addition, the latent space of random inputs is typically not disentangled and it is necessary to introduce constraints in order to learn an interpretable representation [147].

3.2.2   Examples of Models. The number of proposed variants is still growing. An in-depth survey on GANs is [103]. Indeed, several refinements have been proposed in the past years, such as using deep convolutional networks [217] or self-attention [301], incrementally growing the networks [143], or scaling the model parameters [31]. In the following, we present examples that are relevant to the issue of machine creativity.

The problem of non-meaningful representation has been addressed in different ways. For instance, InfoGAN [46] adds a latent code c to z. An auxiliary model learns to predict c given the sample generated by means of it. In this way, it can learn disentangled representations in a completely unsupervised manner. Another possibility is **Bidirectional GAN (BiGAN)** [67]. In order to include an inverse mapping from data to latent representation, an encoder is added to the architecture. The discriminator is then trained to distinguish between pairs of random noise and synthetic data and pairs of real data and latent encoding. It is possible to condition the generation by means of a target content [204], a text [224], or even an image [125]. In order to do so, it is sufficient to use the conditional information as input for both generator and discriminator [189]. Similarly, image-to-image translation is possible also without paired datasets. CycleGAN [310] trains two generators (from one domain to another, and vice versa) so that each of them produces images both from the target domain and correctly reconstructed by the counterpart.

In StyleGAN [145, 146], the generator architecture is re-designed in order to control the image synthesis process. The style of the image is adjusted at each layer based on the latent code (the specific intermediate code to control each layer is provided by a non-linear mapping network). This allows for the automatic separation of high-level attributes from stochastic variations in

the generated images. It also allows for mixing regularization, where two latent codes are used alternatively to guide the generator. StyleGAN-V [248] builds on top of it to learn to produce videos by only using a few frames of it. To generate longer and more realistic motions, a two-stage approach can be used as well: first, a low-resolution generator is adversarially trained on long sequences; then, a high-resolution generator transforms a portion of the produced, low-resolution video in a high-resolution one [32].

Finally, it is also worth mentioning variants that adapt GANs to sequential tasks (e.g., text generation). Since GANs require the generator to be differentiable, they cannot generate discrete data [97]. However, several techniques have been proposed to avoid this problem. One possibility is to transform the discrete generation into a continuous one. Music can be processed like an image by considering its waveform (as in WaveGAN [66] and GANSynth [76]) or its musical score composed of tracks and bars (as in MuseGAN [68]). Music in a desired style can be obtained through conditional inputs. Another possibility is to consider a soft-argmax function as an approximation of the inference for each step [304]. TextGAN [305] uses it together with feature matching to learn the production of sentences. In place of the discriminative signal, it uses the difference between the latent feature distributions of real and synthetic sentences learned by the discriminator. Another solution is to transform the GAN into a **Reinforcement Learning (RL)** framework, as in **Sequential GAN (SeqGAN)** [300]. The generative model is the agent; the tokens generated so far form the state; the selection of the next token to be generated is the action to be performed; and the discriminative signal is the reward. The REINFORCE algorithm [290] can then be used to adversarially train the generative model. Other policy gradient methods can be used as well [79]. On the other hand, the learning signal (i.e., the reward) might be very sparse. A way to solve this issue is to use inverse RL [311]. For example, the authors of [245] use inverse RL to learn a reward function able to associate positive rewards to real state-action pairs, and non-positive rewards to synthetic state-action pairs. Notably, this can help solve mode collapse too. Another variant is LeakGAN [107]. Here, a hierarchical generator composed of a Manager and a Worker is used. The Worker produces a sentence conditioned by a goal vector provided by the Manager. The Worker and the discriminative model are trained following SeqGAN; the Manager is trained to predict goal vectors that lead to the identification of advantageous directions in the discriminative feature space. More specifically, the Manager receives a feature vector from the discriminator, i.e., its last convolutional layer, at each generated token. By means of this *leaked* information and the hierarchical architecture, LeakGAN produces longer and higher-quality texts. Finally, another possibility is to use Gumbel-softmax relaxation [126, 177], as in **Relational GAN (RelGAN)** [201]. **Controlled TExt generation Relational Memory GAN (CTERM-GAN)** [20] builds on the latter by also conditioning the generator on an external embedding input. In addition, it uses both a syntactic discriminator to predict whether a sentence is correct and a semantic discriminator to infer if a sentence is coherent with the external input.

3.2.3   Applications. GANs have been applied to a variety of practical problems in several application scenarios. They have been widely used for semi-supervised learning [203]; for generating adversarial examples [294] to better train image classifiers [178]; and, in general, in computer vision (see [285] for a detailed discussion). The generative power of GANs has also found its place in recommender systems (see [60]) to generate fashion items; in science and chemistry [185, 194]. Of course, its ability to generate high-quality samples has been exploited in many other areas, from anime design [134] and 3D object modeling [292] to photo-realistic consequences of climate change [242]. Conditional inputs also allow the production of artistic works by controlling stylistic properties such as genre [266] or influencer [49]. Finally, the most famous example of the artistic

power of GAN is the collection of paintings by Obvious, a French art collective [283]; one of their works has been sold for more than 400,000 dollars.[3]

3.2.4 Critical Discussion. GANs are difficult to evaluate from a machine creativity perspective. The generator does not receive the original works as input, so it samples from a conceptual space that is built only indirectly from them. In rare cases, this can also lead to a different conceptual space (with respect to the original one) and so to transformational creativity, but it typically leads to exploratory creativity. In fact, since the goal is to learn to generate seemingly real artifacts from a latent distribution, it is likely that it will approximate the real one. Still, it is possible to identify potential creative solutions among those generated by the model.

An advantage of GANs is the presence of a *recognition* network, i.e., the discriminator, trained to recognize real (valuable) works. This is important for two reasons. It suffices for being able to define GANs *appreciative* [52], which is a central sub-task of creativity [6, 91]. In addition, it allows us to consider their products as valuable, as it is in a sense their intrinsic objective. However, there is no guarantee that they will also be new and surprising. Nevertheless, it seems possible to extend a GAN objective to include such properties as well (see Section 3.8 for a discussion).

3.3 Sequence Prediction Models

3.3.1 Core Concepts. A sequence prediction model is a generative model that considers generation as a sequential process. It works in an autoregressive fashion: it predicts the future outcome of the sequence (i.e., the next token) from the previously observed outcomes of that sequence, usually by means of an internal state that encodes information from the past. It is trained to minimize the prediction error for each token in the dataset. At inference time, this simple yet effective approach only requires to produce one token after the other, feeding back to the model what has been produced so far [142]. It makes it possible to learn dependencies between tokens in real data so that the same dependencies can be exploited when generating synthetic data. However, this causes the generation to be highly dependent on real data, e.g., there is the risk of potentially reproducing portions of the training set.

3.3.2 Examples of Models. Several models have been proposed, most of them based on RNN, and especially on **long short-term memory (LSTM)** [119]. The reason is that RNNs use internal states based on previous computation: inputs received at earlier time steps can affect the response to the current input. However, RNNs tend to perform worse with longer sequences [15]. LSTM is a specific RNN architecture that addresses the problem of long-term dependencies through the use of additional gates determining what to remember and what to forget at each step.

RNNs can be used to model joint probabilities of characters (Char-RNN) [142]; words [213]; phonemes [121]; syllables [312]; and even tokens from transcriptions of folk music (Folk-RNN) [261]. They can also receive conditional inputs like the encoding of the previous lines [303]. Richer architectures that combine models focusing on different properties can be used to generate more complex text, e.g., poetry based on pentameter and rhymes [157]. Finally, sequence modeling can also be combined with reinforcement learning. For example, the authors of [129] use a Note-RNN model (based on single notes) trained using Deep Q-Network [191]; as rewards, they consider both the classic loss of sequence prediction models and a reward based on rules of music theory. In this way, the model learns a set of composition rules, while still maintaining information about the transition probabilities of the training data. The advantages of adopting an RL-based approach are described in Section 3.6.

---

[3]Fun fact: the sold painting is called *Portrait of Edmond De Belamy* because Belamy sounds like *bel ami*, a sort of French translation of... *Goodfellow*.

Due to the difficulties in working with long sequences, results in tasks like narrative generation are affected by a lack of coherence [231]. Many approaches have been proposed to address this problem. For instance, stories can be generated in terms of events [182] (i.e., tuples with subject, verb, object, and an additional *wildcard*) by an encoder-decoder RNN (also known as Sequence-to-Sequence, see [263]); events are modeled by another encoder-decoder RNN. Instead of events, it is also possible to focus on entities (i.e., vectors representing characters) [50].

Sequence prediction models are also used for domains not commonly modeled as sequences, like images. Image modeling can be defined in a discrete way by means of a joint distribution of pixels: the model learns to predict the next pixel given all the previously generated ones. It starts at the top left pixel and then proceeds towards the bottom right. The two seminal architectures for sequence prediction of images are PixelRNN and PixelCNN [276]. The former is a two-dimensional RNN (based on rows or diagonals). The latter is a **convolutional neural network (CNN)** with an additional fixed dependency range (i.e., the convolution filters are masked in order to only use information about pixels above and to the left of the current one). To obtain better results, gated activation units can be used in place of rectified linear units between the masked convolutions; conditional inputs encoding high-level image descriptions can be used as well [277]. Notably, the Gated PixelCNN architecture can also be used for other types of data: WaveNet [275] implements it to generate audio based on the waveform, possibly guiding the generation with conditional inputs.

While intuitive in terms of architecture, RNNs are limited by the vanishing gradient problem and non-parallelizability in the time dimension [158]. Very recent works explore solutions to tackle these issues by means of structured state spaces [102] and a combination of RNNs and Transformers [211] (see Section 3.4).

3.3.3 Applications. As discussed, sequence prediction models have been used to learn to write poems or stories (by predicting a character, syllable, or word after the other); to compose music (by predicting a note or a waveform after the other); to draw images (by predicting a pixel after the other). In general, they can be used for any kind of time series forecasting [170]. They can also be used for co-creativity, as in Creative Help [231]. Despite their simplicity, sequence prediction models are one of the most successful generative techniques. An interesting example is *Sunspring*. It might be considered as the first AI-scripted movie: it was generated by a Char-RNN trained on thousands of sci-fi scripts [187]. The quality of the result is demonstrated by the fact that it was able to reach the top ten at the annual Sci-Fi London Film Festival in its 48-Hour Film Challenge.[4]

3.3.4 Critical Discussion. Sequence prediction models generate outputs that have characteristics of both exploratory and combinatorial creativity. They are based on probabilistic predictions and they are able to generate new outputs in the induced space, but they can also reuse sequences of tokens from different works, combining them together. There is no guarantee that the results will be valuable or novel, and classic methods such as RNNs lack surprise [35]. It is worth noting that the possibility of using conditional inputs and being able to work at different levels of abstraction might indirectly lead to creative outputs, but creativity should then be attributed to the higher-level component (or human if the input is provided by the user) that is guiding the generation for specific elements and characteristics of the result.

3.4 Transformer-Based Models

3.4.1 Core Concepts. Transformer-based models are neural networks based on the Transformer architecture [281]. They represent the main example of foundation models [23], because of the

---

[4]Quite interestingly, the AI that wrote *Sunspring* declared that its name was Benjamin, probably in honor of Walter Benjamin, the German philosopher who, already in 1935 [16], understood that new mechanical techniques related to art can radically change the public attitude to art and artists.

leading role they have been assuming in language, vision, and robotics. A Transformer is an architecture for sequential modeling that does not require recurrent or convolutional layers. Instead, it only relies on a self-attention mechanism [11] that models long-distance context without a sequential dependency. Each layer consists of multi-head attention (i.e., several self-attention mechanisms running in parallel), a feed-forward network, and residual connections. Since self-attention is agnostic to token order, a technique called positional embedding is used to capture the ordering [281].

In principle, a Transformer is nothing more than an autoregressive model: it works by predicting the current token given the previous ones (see Section 3.3). However, a few fundamental differences exist. A Transformer can also be trained by means of masked modeling: some of the input tokens are randomly masked, and the model has to learn how to reconstruct them from the entire context, and not only from the previous portions [61]. The possibility of dealing with very long sequences allows for prompting. By providing a natural language prompt in input, the model can generate the desired output, e.g., the answer to a question, a class between a set of classes for a given text, or a poem in a particular style [34]. This is done by simply passing the prompt in input as a text, and then leveraging the model to predict what comes next (e.g., the answer to a question). These advantages, together with the very large amount of data available, the increasing computational power, and the parallelism induced by their architecture, have led Transformer-based models to become the state of the art for several tasks. Nevertheless, the computational costs of the architecture from [281] grow quadratically with the input size.

3.4.2 Examples of Models. Several Transformer-based approaches have been proposed in recent years. The design of specific Transformers for a variety of applications is presented in several surveys (e.g., [23, 148]) and books (e.g., [273]).

The domain mostly influenced by Transformers is **natural language processing (NLP)**. **Bidirectional Encoder Representations from Transformers (BERT)** [61] is a Transformer-based encoder trained for both predicting the next sentence (in an autoregressive fashion) and reconstructing masked tokens from the context. Several enhanced variations of the original model have been proposed, such as, for instance, solutions that remove the next-sentence pre-training objective [173], use inter-sentence coherence as an additional loss [154], or employ distillation [114] to train a smaller model [238]. The other main approach is that used by the **Generative Pretrained Transformer (GPT)** family [34, 215, 218]. Here, a Transformer-based decoder is trained in an autoregressive way by additionally conditioning on the task of interest. After training, it can be used to perform a wide range of tasks by providing a description or a few demonstrations of the task. The effectiveness of this text-to-text generative approach has then been explored by T5 [220]. Many other large language models [246, 269, 302] have been proposed to achieve better results by means of more parameters and computation [249], or more qualitative data [106]. Mixture of Experts [244] can be used as well in place of the feed-forward network to train a larger but lighter model (since only portions of it are used per task), as done by **Generalist Language Model (GLaM)** [70]. Finally, **Bidirectional and Auto-Regressive Transformer (BART)** [164] ideally merges together a BERT-encoder (trained by corrupting text with an arbitrary noising function) and a GPT-decoder (trained to reconstruct the original text autoregressively). Such an encoder-decoder architecture is able to achieve state-of-the-art results in machine translations, as well as other text-to-text tasks.

Transformer-based models have been used in domains different from **language modeling (LM)**. Few have been proposed for music generation. One of the first examples was Music Transformer [122], which can generate one-minute music in Bach's style with internal consistency; another remarkable one is Musenet [210], which is able to produce 4-minute musical

composition with a GPT-2 architecture; and, finally, it is worth mentioning Jukebox [62], which can generate multiple minutes of music from raw audio by training a Sparse Transformer [47] (i.e., a Transformer with sparse factorization of the attention matrix in order to reduce from quadratic to linear scaling) over the low-dimensional discrete space induced by a VQ-VAE. Conditioning is always considered by means of genre, author, or instruments. MusicLM [3] additionally allows to generate music from text descriptions by aligning text and audio representation from different state-of-the-art models [25, 123]. Another important application domain is video-making. **Video Vision Transformer (ViViT)** [8] generates videos using classic Transformer architectures; **Video Transformer (VidTr)** [306] achieves state-of-the-art performance thanks to the standard deviation-based pooling method; and VideoGPT [295] does so by learning discrete latent representations of raw video with VQ-VAE, and then training a GPT autoregressively.

Transformers have been highly influential in computer vision too. The first model was Image Transformer [209]. It restricts the self-attention mechanism to attend to local neighborhoods, so larger images can be processed. Class-conditioned generation is also supported, by passing the embedding of the relative class in input. To avoid restricting self-attention to local neighborhoods, Vision Transformer [69] divides an image into fixed-size patches, linearly embeds each of them, adds position embeddings, and then feeds the resulting sequence of vectors to a standard Transformer encoder. **Masked Autoencoders (MAE)** [111] instead uses an encoder-decoder architecture based on Transformers trained with masked image modeling (i.e., to reconstruct randomly masked pixels). A BERT adaptation to images called **Bidirectional Encoder representation from Image Transformers (BEiT)** [14] has also been proposed. Masked image modeling has also been used together with classic autoregressive loss [44]. Conversely, **Vector Quantised-GAN (VQ-GAN)** [78] allows a Transformer to be based on vector quantization. A GAN learns an effective codebook of image constituents. To do so, the generator is implemented as an auto-encoder; vector quantization is applied over the latent representation returned by the encoder. It is then possible to efficiently encode an image in a sequence corresponding to the codebook indices of their embeddings. The Transformer is finally trained on that sequence to learn long-range interactions. These changes also allow us to avoid quadratic scaling, which is intractable for high-resolution images. Finally, DALL-E [216] takes advantage of a discrete VAE. To generate images based on an input text, it learns a discrete image encoding; it concatenates the input text embedding with the image encoding; it learns autoregressively on them. CogView implements a similar architecture [65].

Finally, Transformer-based models have also been used in multimodal settings, in which data sources are of different types. A survey can be found in [264]. The first examples of these systems consider text and images together as the output of the Transformer architecture. By aligning their latent representations, images and texts can be generated by Transformer-based decoders given a multimodal representation. For instance, **Contrastive Language-Image Pre-training (CLIP)** [216] has an image encoder pre-trained together with a text encoder to generate a caption for an image. **A Large-scale ImaGe and Noisy-text embedding (ALIGN)** [132], based on similar mechanisms, can achieve remarkable performance through training based on a noisier dataset. In [272] the authors propose a frozen language model for multimodal few-shot learning: a vision encoder is trained to represent each image as a sequence of continuous embeddings, so that the frozen language model prompted with this embedding can generate the appropriate caption. In [80] the authors present **Bridging-Vision-and-Language (BriVL)**, which performs multimodal tasks by learning from weak semantic correlation data. Finally, there is a trend toward even more complex multimodal models. For example, **Video-Audio-Text Transformer (VATT)** [4] learns to extract multimodal representations from video, audio, and text; instead, Gato [225] serializes all data (e.g., text, images, games, other RL-related tasks) into a flat sequence of tokens that is then embedded and passed to a standard large-scale language model. Similarly, Gemini [93] achieves

state-of-the-art performance in multimodal tasks by working on interleaved sequences of text, image, audio, and video as inputs; [94] extends it to Mixture-of-Experts setting. Finally, NExT-GPT [293] handles any combination of four modalities (text, audio, image, and video) by connecting a language model with multimodal adaptors and diffusion decoders (see Section 3.5).

3.4.3 Applications. Transformer-based large language models can be used for almost any NLP task, including text summarization, generation, and interaction. In order to do so, the model can be used as frozen (i.e., to provide latent representations in input to other models); can be fine-tuned for the specific objective; can be exploited with zero-shot, one-shot or few-shot setting by prompting the task or few demonstrations in input. Transfer learning can instead be used to perform image classification by means of Transformer-based models trained on images. Other domain-specific techniques can be used as well: for instance, PlotMachines [223] learns to write narrative paragraphs not by receiving prompts, but by receiving plot outlines and representations of previous paragraphs. From a generative learning perspective, Transformers have shown impressive performance in producing long sequences of texts and music or speech [284], as well as in generating images based on input text. Their application has not been limited to these data sources. For instance, AlphaFold uses a Transformer architecture to predict protein structure [139]; RecipeGPT employs it to generate recipes [160]; and GitHub Copilot relies on it to support code development [45].

3.4.4 Critical Discussion. Given the fact that the Transformers can be considered as an evolution of sequence prediction models, the observations made for that class of models (see Section 3.3) apply also to them. However, the inherent characteristics of their architecture allow for larger models and higher-quality outputs, which are also able to capture a variety of dependencies of text across data sources. More in general, a broader conceptual space is induced. This means that domain-specific tasks might be addressed by means of solutions outside or at the boundary of the sub-space linked with that domain. Moreover, possibly also through careful use of inputs (see Section 3.7), their adoption might lead to transformational creativity. As far as Boden's criteria are concerned, there is no guarantee that the output of the Transformer architecture would be valuable, novel, or surprising, even though current state-of-the-art **large language models (LLMs)** achieve almost human-like performance in creative tests [259, 308].

3.5 Diffusion Models

3.5.1 Core Concepts. Diffusion models are a family of methods able to generate samples by gradually removing noise from a signal [253]. The most representative approach is the **Denoising Diffusion Probabilistic Model (DDPM)** [115]. An input x_0 is corrupted by gradually adding noise until obtaining an x_T from a pre-defined distribution; the model then has to reverse the process. Each timestep t corresponds to a certain noise level; x_t can be seen as a mixture of x_0 with some noise ϵ whose ratio is determined by t. The model learns a function ϵ_θ to predict the noise component of x_t by minimizing the mean-squared error. x_{t-1} is then obtained from a diagonal Gaussian with mean as a function of $\epsilon_\theta(x_t, t)$, and with a fixed [115] or learned [200] variance. In other words, it learns to associate points from a predefined random distribution with real data through iterative denoising. Because of this, at inference time, a diffusion model can generate a new sample by starting from pure random noise. The generation can also be conditioned by simply modifying the noise perturbation so that it depends on the conditional information. However, this iterative sampling process might potentially lead to slow generation; a proposed solution is to induce self-consistency, i.e., ensuring that points on the same trajectory map to the same initial ones [254]. In this way, the output can be obtained in a single step.

The aforementioned diffusion process is similar to the one followed by score-based generative models [255, 256]. Instead of noise, here a model is trained to learn the score, i.e., the gradient of the log probability density with respect to real data. The samples are then obtained using Langevin dynamics [288]. Despite the differences, both of them can be seen as specific, discrete cases of Stochastic Differential Equations [257].

3.5.2 Examples of Models. Diffusion models have been primarily used for image generation. In order to generate higher-quality images and to allow text-to-image generation, a variety of effective methods for conditioning have been proposed. A possibility is to use classifier guidance [63]: the diffusion score (i.e., the added noise) includes the gradient of the log-likelihood of an auxiliary classifier model. An alternative is classifier-free guidance [117]: to avoid learning an additional model, a single neural network is used to parameterize two diffusion models, one conditional and one unconditional; the two models are then jointly trained by randomly setting the class for the unconditional model. Finally, the sampling is performed using a linear combination of conditional and unconditional score estimates. **Guided Language to Image Diffusion for Generation and Editing (GLIDE)** [198] demonstrates how classifier-free guidance can be effectively used to generate text-conditional images. In addition, it shows how diffusion models can be used for image editing by fine-tuning in order to reconstruct masked regions. Performance improvement can be obtained by means of a cascade of multiple diffusion models performing conditioning augmentation [116]. Notably, the diffusion model can operate on latent vectors instead of real images. Stable Diffusion [232] employs a diffusion model in the latent space of a pre-trained autoencoder. Similarly, DALL-E 2 [221] generates images by conditioning with image representations. At first, it learns a prior diffusion model to generate possible CLIP image embeddings from a given text caption, i.e., conditioned by its CLIP text embedding. Then, a diffusion decoder produces images conditioned by the image embedding. The generation quality can be further improved by means of generated captions for the images in the training set [19]. Imagen [235] uses instead a cascaded diffusion decoder, together with a frozen language model as a text encoder to increase the quality of output.

Although the approach is particularly suitable for images, applications to other data sources have been developed as well. DiffWave [152] and WaveGrad [45] use diffusion models to generate audio. They overcome the continuous-discrete dichotomy by working on waveform. Another possibility is to use an auto-encoder like MusicVAE [230] to transform the sequence into a set of continuous latent vectors, through which a diffusion model is trained [190]. Resembling image generators, **Contrastive Language-Audio Pretraining (CLAP)** embeddings [75] can be used to generate audio by conditioning on text descriptions [171]. Diffusion-LM [167] employs diffusion models to write text by denoising a sequence of Gaussian vectors into continuous word vectors (then converted into discrete words by a rounding step); DiffuSeq [96] performs sequence-to-sequence generation tasks by embedding source and target sequences in the same embedding space through a Transformer architecture. Diffusion models have been used for 3D generation as well [199]. Finally, diffusion models for video have also been proposed, based on gradient-based conditioning [118], and on processing latent spacetime patches. In particular, with respect to the latter, Sora [33] first turns videos into sequences of patches and then uses a diffusion Transformer to predict the original patches from random noise (and conditioning inputs like text prompts), improving sample quality and flexibility.

3.5.3 Applications. Despite their recent introduction, diffusion models have been used to generate audio, music, and video, as well as to generate and edit images conditioned on input text, e.g., with in-painting [174] or subject-driven generation [234]; we refer to [298] for a comprehensive survey of this area. Indeed, they lead to higher-quality outputs than the previous state-of-the-art

models. In particular, DALL-E 2 and Stable Diffusion have been able to produce images from textual instructions with superior fidelity and variety.

3.5.4 Critical Discussion. Diffusion models learn a mapping between real images and a Gaussian latent space. Because of this, they are an example of exploratory creativity: they randomly sample from that space, and then they possibly navigate it in the direction imposed by conditional inputs. There is no guarantee that the results will be valuable, novel, or surprising, even though these approaches are able to generate outputs characterized by a high variety. As already argued, novelty and surprise may only arise due to the conditioning input (for example, a human describing a novel combination of elements), i.e., the model is not imaginative on its own.

3.6 Reinforcement Learning-Based Methods

3.6.1 Core Concepts. With RL-based methods, we aim to indicate all the generative models whose training relies on maximizing a reward. These models are based on the architectures introduced so far, e.g., they can be GANs or autoregressive models. The difference is that they are not (only) trained to fool the discriminative part or to reduce prediction error. The typical framework considers the generative model as the agent; each action causes a modification to the current product, i.e., the state; and the agent learns a policy that maximizes the cumulative reward. Therefore, the reward can be used to impose desired behavior on the generative model. The RL-based approach can be implemented for the entire training or for fine-tuning a pre-trained model. The sampling scheme remains the same and depends on the chosen generative model.

3.6.2 Examples of Models. A first example is **Objective-Reinforced GAN (ORGAN)** [104]. Here, RL is used not only to adapt GANs to sequential tasks but also to provide additional learning signals, such as rewards from specific-domain objectives (e.g., tonality and ratio of steps for music generation). Also [299] follows this path by using rewards like fluency, coherence, meaningfulness, and overall quality to generate poems. Another possibility is to use the metrics used at test time (e.g., BLEU or ROUGE) [10, 222]. Instead, RL-DUET [133] casts online music accompaniment generation as an RL problem with an ensemble of reward models, i.e., autoregressive models trained with or without the whole context, and with or without the human-produced counterpart. In this way, inter-coherence (between humans and machines) and intra-coherence can be obtained. Finally, Intelli-Paint [247] can paint in human style by using a sequential planner that learns a painting policy to predict vectorized brushstroke parameters from the current state.

RL can also be used to fine-tune a pre-trained generative model. Doodle-SDQ [309] first learns to draw simple strokes using supervised learning; then, it improves its generation by means of rewards about similarity, color, and line movement. Conversely, the authors of [265] suggest to consider a pre-trained LSTM language model as a policy model. Fine-tuning then aims at maximizing the probability that a given event occurs at the end of the narrative. RL Tuner [130] uses RL to fine-tune a Note-RNN [72] to produce music that follows a set of music theory rules. To avoid forgetting note probabilities learned from the data, the probability value returned by a copy of the pre-trained Note-RNN can be used as an additional reward. Sequence Tutor [128] generalizes this idea of learning a policy that trades off staying close to the data distribution while improving performance on specific metrics. A comprehensive critical discussion of the rewards for RL-based generative models can be found in [85]. Finally, RL can be used to help models follow human preferences [48] or feedback. The latter technique is referred to as **Reinforcement Learning from Human Feedback (RLHF)** [260]. For example, ChatGPT [206] is an interactive version of GPT-3 [34] (initially) and GPT-4 [207] (at the time of writing), fine-tuned to maximize a learned reward of human values. RLHF improves its conversational skills while mitigating mistakes and biases; because of this, it has become a standard de facto for fine-tuning large language models, e.g., in

[270]. It is also possible to use AI and not human feedback [12, 159]. However, RLHF has some limitations [42], and alternative RL-free strategies are increasingly popular (e.g., [219]).

3.6.3 Applications. As seen, RL-based models can be used to fully train or fine-tune generative models for different tasks; ideally, for any task that could benefit from domain-specific objectives. This is the case of music and molecule generation [104, 128], but also of dialogue generation [165], image generation [21] and painting [124]. In addition, the sequential nature of RL can help as well in all the tasks requiring to deal with new directives during generation (e.g., music interaction). Finally, RLHF can be used to directly optimize models for creative tasks, e.g., poetry [208].

3.6.4 Critical Discussion. The evaluation of the creativity of RL-based models depends on how the agent is implemented and which rewards are considered. The learned space of solutions depends on the used rewards (and on the pre-training technique in case of fine-tuning). They typically contain an adversarial signal or a likelihood with respect to the training data; thus, combinatorial or exploratory creativity is obtained. However, additional rewards can have the effect of transforming that space (see Section 3.8). As far as Boden's criteria are concerned, value is typically ensured by some qualitative domain-specific reward or by approaching human preferences. This also allows us to consider them as appreciative, as discussed in Section 3.2.4. Novelty and surprise might be achieved as well by means of specific rewards; however, this is not the case for current models.

3.7 Input-Based Methods

3.7.1 Core Concepts. The last class of methods we consider in our analysis is not about a different generative model. On the contrary, it is about a different way to sample results from (pretrained) generative models, namely by means of its inputs. Two different approaches can be used. The first is about carefully selecting or optimizing the input to a generative model (e.g., the latent vector or the text prompt) so that to obtain the desired output. The second approach is about optimizing the input so that it directly becomes the desired output. They rely on losses that are usually based on features learned by neural networks. While the two approaches are technically different, both of them aim at obtaining better outputs by exploiting the knowledge of the pre-trained model through the optimization of the inputs.

3.7.2 Examples of Models. The first approach consists of carefully modifying the input of a generative model until the output matches the desired properties. The main example is VQGAN-CLIP [58]. Given a text description, VQGAN produces a candidate image from a random latent vector; the vector is then optimized by minimizing the distance between the embeddings of the description and the candidate image. Both embeddings are computed using CLIP [216]. Variants can be implemented as in Wav2CLIP [291], where an audio encoder is learned to match the CLIP encoders so that VQGAN-CLIP can be used from raw audio; or as in music2video [127], where videos are generated from audio a frame after the other by both minimizing the distance between subsequent frames, and the distance between image and music segment embedded by Wav2CLIP. In addition to the random latent vector, the text or audio description can be optimized as well. This can be performed by the users through many iterations of careful adjustments, or by means of an automated procedure. The latter is commonly known as prompt tuning. Prompt tuning is about producing prompts via backpropagation; the optimized prompts can then condition frozen language models in order to perform specific tasks without having to fine-tune them [162]. An additional model can also be trained to output the desired prompt [163]. Finally, image generators such as VQGAN can also be exploited in other ways, i.e., with binary-tournament genetic algorithm [82] or more complex evolution strategies [267]. Another possibility is to optimize the input so that

the generated output maximizes a target neuron of an image classifier [197]. This helps generate what that neuron has learned. The desired latent vector can also be produced by an additional model [196].

The second approach consists of optimizing the inputs to transform them into the desired outputs. DeepDream [192] generates "hallucinated" images by modifying the input to maximize the activation of a certain layer from a pre-trained classifier. Artistic style transfer is based on the same idea. Given an input image and a target image, the former is modified by means of both style and content losses thanks to a pre-trained classifier. The content loss is minimized if the current and the original input images generate the same outputs from the hidden layers. The style loss is minimized if the current and target images have the same pattern of correlation between feature maps in the hidden layers [89]. Control over results can be improved by considering additional losses about color, space, and scale [90].

3.7.3   Applications. Input-based methods can be used with any generative model to produce the desired output. With language models, they can exploit their generality in several specific tasks without fine-tuning them. For instance, prompt tuning can be used by writers for co-creation [43] or to force LLMs to *brainstorm* [262]. With image generators, they can obtain drawings adherent to given descriptions, or high-quality but yet peculiar paintings like colorist [82], abstract [267] or alien [250] artworks. We believe applications to other domains are yet to come. Both types of input-based methods can be used not only to produce desired outputs or to transfer styles; they can also be used to better analyze what is inside the network [197, 205].

3.7.4   Critical Discussion. Since input-based methods are applied to pre-trained generative models, the space of solutions in which they work is the one induced by those models, i.e., the common spaces we can derive from real data. Nonetheless, some techniques may be able to cause productions that are outside that space or at its boundaries, i.e., to cause transformational creativity. This might happen if the model is general, and the output for a specific task is not only sampled from the sub-space of solutions for that task (e.g., with prompt tuning over a language model). Input-based methods are also valuable: the input optimization itself is typically guided by some sort of qualitative loss. On the other hand, they are not explicitly novel or surprising (although the results might seem so). However, nothing prevents optimizing the loss in such directions (see Section 3.8).

3.8   Practical Assessment of Creativity-Oriented Methods

We conclude this analysis of generative models with a discussion of how they might increase their *creativity* according to Boden's definition. We have discussed how the presence of a recognition model (e.g., a discriminative model or a reward model) helps ensure the value of the products. In the same way, novelty and surprise can be fostered by the integration of other components. A straightforward way to obtain novel and surprising outputs is to train a generative model by means of novelty and surprise objectives. This is the core idea behind **Creative Adversarial Network (CAN)** [73, 239]. In addition to the classic discriminative signal, i.e., a value loss, the generator is also trained to optimize a novelty loss. This is defined as the deviation from style norms, i.e., the error related to the prediction of the style of the generated image. The sum of the two training signals helps the model learn to produce artworks that are different (in style) from the training data. The same approach has been used to develop a creative StyleGAN, i.e., StyleCAN [131]. Another, very simple way to augment the training signal of a generative model with *creativity-oriented* objectives is by means of RL-based methods. The choice of the reward structure is the fundamental element in the design of effective generative reinforcement learning systems. Rewards should teach the model to generate an output with a high level of novelty and surprise. An example is ORGAN

[104], where appropriate reward functions can be used. For instance, statistical measures (e.g., Chi-squared) or metrics of distance between distributions (e.g., KL divergence) might be used to ground ideas of novelty and surprise.

Another possibility is the development of an input-based method where the input is optimized to obtain a product that is valuable, novel, and surprising. This may be achieved either by forcing a further exploration of the latent space (e.g., by means of evolutionary search [81]), or by defining appropriate loss functions to perform gradient descent over the input. All these methodologies are also called *active divergence* [18] since they aim to generate in ways that do not simply reproduce training data. A survey on active divergence can be found in [30]. A different output can also be obtained by carefully altering the probability distribution of the model, e.g., by scaling its probabilities with learned functions to maximize target properties [59, 251, 297].

A different approach is followed by the Composer-Audience architecture [35]. Two models are considered: the Audience, a simple sequence prediction model trained on a given dataset; and the Composer, another sequence prediction model trained on a different dataset. In addition, the Composer also receives the next-token expectations from the Audience, and it learns when to follow its guidance and when to diverge from expectations, i.e., when to be surprising. For instance, it can learn to produce jokes by considering non-humorous texts to train the Audience, and humorous texts to train the Composer. Even though this approach is useful for learning how to generate valuable and surprising output, it is only applicable when paired datasets are available.

As far as the type of creativity is concerned, there can be ways to achieve a better exploration or even transformation of the space of solutions. For example, since CAN novelty loss is used during training, it learns to diverge from the distribution of real data. The same is true for RL-based methods with novelty and surprise rewards (especially if the training happens from scratch). Finally, a more explored or transformed space may be reached using RL-based methods driven by curiosity [36]: an agent can learn to be creative and discover new patterns thanks to intrinsic rewards to measure novelty, interestingness, and surprise. This can be done by training a predictive model of the growing data history and by using its learning progress as the reward. In this way, the agent is motivated to make things the predictor does not yet know. If an external qualitative reward is considered as well, the agent should in theory learn to do things that are new, but still valuable [240]. The same idea can also be applied to different techniques like evolutionary strategies [176]. **Deep Learning Novelty Explorer (DeLeNoX)** [168] uses a denoising autoencoder to learn low-dimensional representations of the last generated artifacts. Then, a population of candidate artifacts (in terms of their representation) is evolved through a feasible-infeasible novelty search [169] in order to maximize the distances between them, i.e., to increase their novelty, while still considering qualitative constraints. Other evolutionary strategies might be considered as well to search the space of artifacts for novel [161] and surprising [100] results. Instead of relying on manually crafted metrics, **Quality Diversity through Human Feedback (QDHF)** [64] uses human feedback for computing quality and distance in learned latent projection for computing diversity. **Quality-Diversity through AI Feedback (QDAIF)** [28] makes the model more independent in searching and innovating by totally relying on its own feedback for both quality and diversity.

Table 1 summarizes all the generative approaches discussed in this section, highlighting their characteristics from a machine creativity perspective.

4 CREATIVITY MEASURES

In this section, we present different methodologies to evaluate the creativity of artifacts generated by artificial agents. These can typically be extended to human-generated artifacts. For each of them, we explore the core concepts, the dimensions of creativity that are considered, the evaluation protocol, and, finally, we critically assess them. The presence of several different proposals can be

Table 1. Summary of all the Methods Explained so far, Considering their Type of Creativity as
Discussed in the Corresponding Subsections; the Possible Presence of Boden's Criteria
(✓ if Induced by the Training Process; ∼ if not Considered; ✗ if Excluded); and
Some Practical Suggestions to Achieve a Higher Degree of Creativity

Generative family	Type of creativity	Boden's criteria	Creative suggestions
VAE	Exploratory	∼ Value ∼ Novelty ∼ Surprise	Creativity-oriented input-based methods
GAN	Exploratory	✓ Value ∼ Novelty ∼ Surprise	CAN; Creativity-oriented input-based methods
Sequence prediction model	Combinatorial, Exploratory	∼ Value ∼ Novelty ✗ Surprise	Composer-Audience; Creativity-oriented RL-based methods
Transformer-based models	Combinatorial, Exploratory, Transformational	∼ Value ∼ Novelty ✗ Surprise	Creativity-oriented prompt tuning or RL-based methods
Diffusion models	Exploratory	∼ Value ∼ Novelty ∼ Surprise	Creativity-oriented input-based methods
RL-based methods	Combinatorial, Exploratory, Transformational	✓ Value ∼ Novelty ∼ Surprise	Intrinsic rewards; Novelty-based rewards
Input-based methods	Exploratory, Transformational	✓ Value ∼ Novelty ∼ Surprise	Evolutionary search; Novelty-based optimization

associated with the fact that it is not always straightforward to determine the "right" question
to ask in an evaluation of a creative artifact [88]. For instance, the **Generation, Evaluation, and
Metrics (GEM)** Benchmark for natural language generation [92] does not contain any creativity
evaluation measure. This is due to the inadequacy of the creative metrics proposed to date to
correctly capture and measure all the necessary dimensions of creativity. The focus of our overview
is on measures that are based on (or associated with) machine learning techniques. It is worth
noting that some of them might be calculated without using machine learning, but we will refer to
an implementation based on the latter. For an in-depth overview of creativity measures not strictly
related to machines, we refer to [236]. Table 2 reports all the evaluation methods considered in this
section, highlighting the dimensions they try to capture, their applicability, and their limitations.
We will discuss these aspects in the remainder of this section.

4.1 Lovelace 2.0 Test

4.1.1 Overview. The **Lovelace Test (LT)** [29] was proposed in 2001 as a *creativity-oriented* al-
ternative to the world-famous Turing test [274]. More formally, LT is defined as follows:

Definition 4.1. An artificial agent *A*, designed by *H*, passes LT if and only if:
(1) *A* outputs *o*;
(2) *A*'s outputting *o* is not the result of a fluke hardware error but of processes *A* can repeat;

Table 2. Summary of Creativity Evaluation Methods and their Characteristics

Name	What evaluates	How evaluates	Applicability	Limits
Lovelace 2.0 Test	Evaluators' creativity definition	Mean number of challenges per evaluator	General	• Requires a substantial human intervention
Ritchie's criteria	Quality, novelty, typicality	Human opinions (then elaborated through 18 criteria)	General	• Requires human evaluation • Requires to state thresholds • No innovation definition
FACE	Tuples of generative acts	Volume of acts, number of acts, quality (through aesthetic measure)	General	• Abstract method • Definition of aesthetic measure left to the user
SPECS	What we state creativity is	Identification and test of standards for the creativity components	General	• More a framework for evaluation method definition than a real method
Creativity implication network	Value, novelty	Similarity between works (considering subsequent works for value and previous works for novelty)	General	• Not possible to accurately measure the creativity of the most recent works • Wrong creativity and time-positioning correlation
Chef Watson (assessment part)	Novelty, quality	Bayesian surprise, smell pleasantness regression	Specific (recipes)	• Requires human ratings of pleasantness
DARCI	Art appreciation	Neural network to associate image features and description	Specific (visual art)	• Not based on product • Considers just one of the creative tripod
PIERRE - Evaluation part	Novelty, quality	Count of new combinations; user ratings	Specific (recipes)	• Requires user ratings over ingredients
EVE'	Feelings, meanings	Negative log of prediction and posterior probability	General	• Requires a way to explain • Value only through meaning
Common model of creativity for design	Novelty, value, surprise	K-Means on a description space and a performance space; degree of violation of anticipated patterns	Specific (design)	• Requires to define attribute-value pairs • Requires to define clustering parameters
Unexpectedness	Novelty, surprise, transformativity	Possibility to update, and degree of violation of expectations	General	• Does not take care of value
Essential criteria of creativity	Value, novelty, surprise	Sum of performance variables, distance between artifacts and between real and expected artifact	General	• Requires to define performance variables • Requires to define clustering parameters
Computational metrics for storytelling	Novelty, rarity, recreational effort, surprise	Distance between dominant terms, consecutive fragments/clusters of terms	Specific (storytelling)	• Requires to define domination • Requires to define clustering parameters

(3) H (or someone who knows what H knows, and has H's resources) cannot explain how A produced o by appealing to A's architecture, knowledge-base, and core functions.

LT provides several insights for understanding and quantifying machine creativity, but it is rather abstract. For these reasons, a 2.0 version has been proposed [228]. The so-called Lovelace 2.0 Test is defined as:

Definition 4.2. Artificial agent A is challenged as follows:
(1) A must create an artifact o of type t;
(2) o must conform to a set of constraints C where $c_i \in C$ is any criterion expressible in natural language;

(3) a human evaluator *h*, having chosen *t* and *C*, is satisfied that *o* is a valid instance of *t* and meets *C*;

(4) a human referee *r* determines the combination of *t* and *C* to not be unrealistic for an average human.

4.1.2 Dimensions of Creativity Considered. Since the evaluation depends on the tests performed by human evaluators, the dimensions of creativity considered by it might vary greatly. This allows for considering value, novelty, and surprise, as well as domain-specific dimensions.

4.1.3 Protocol for Evaluation. The Lovelace 2.0 Test can be used to quantify the creativity of an artificial agent - by means of its artificial productions - considering a set *H* of human evaluators. With n_i as the first test performed by evaluator $h_i \in H$ not passed by the agent, the creativity of the artificial agent can be expressed as the mean number of challenges-per-evaluator passed: $\sum_i \frac{(n_i)}{|H|}$.

4.1.4 Critical Examination. This methodology represents an effective way to measure creativity since it is ideally applicable to any field and it is quantitative. Although the latter is not based on machine learning, it may be used in principle for performing (some of) the tests. However, this methodology requires considerable human intervention in order to define all the tests.

4.2 Ritchie's Criteria

4.2.1 Overview. Ritchie's Criteria [229] is a set of criteria for evaluating the extent to which a program has been creative (or not) in generating artifacts. These criteria are based on three main factors: novelty, quality, and typicality. [229] contains a proposed series of criteria, but according to the authors they should only be intended as a "repertoire".

4.2.2 Dimensions of Creativity Considered. Ritchie's Criteria are based on three factors: quality, typicality, and novelty. Quality measures how much an item is a high-quality example of its genre. Typicality measures how much an item is an example of the artifact class in question. Novelty measures the dissimilarity of an item with respect to existing examples in its class. Quality and typicality are collected using human opinions about the produced artifacts or using "measurable" characteristics about, for instance, syntax and metric (for poetry generator). On the other hand, novelty is intended as the sum of "untypicality" (the opposite of typicality) and innovation.

4.2.3 Protocol for Evaluation. The computation of the criteria is based on the analysis of the result set of produced artifacts, along with the inspiring set (composed by artifacts of that field used during training and/or generation). It also requires the definition of quality and typicality indicators for the artifacts considered. More specifically, the proposed criteria are: the average of typicality or quality over the result set; the proportion of items with good typicality score, which is also characterized by high quality; the proportion of the output that falls into the category of untypical but high-valued; the ratio between untypical high-valued items and typical high-valued items; the proportion of the inspiring set that has been reproduced in the result set. The assessment of these criteria is performed on both the entire result set and on the subset that does not contain any item from the inspiring set.

4.2.4 Critical Examination. These measures represent a promising way to evaluate creativity, but their application is not straightforward. In fact, they do not clearly specify how to measure novelty in terms of innovation. Furthermore, all measures require a large number of thresholds to be set (and the results are very sensitive to such thresholds [212]). The criteria for the selection of these thresholds are not trivial per se. It is difficult to identify a general methodology for

setting these values. Finally, the collection of correct human opinions (in terms of consistency of measurement methodology, audience selection, etc.) is not a trivial task either.

4.3 FACE

4.3.1 Overview. In [54] the authors introduce FACE as a descriptive model of the creative act of artificial systems. A creative act is considered a non-empty tuple of generative acts. FACE is designed to provide assessors with both quantitative and qualitative evaluations.

4.3.2 Dimensions of Creativity Considered. While FACE can be used to evaluate a product as the consequence of a creative act, its focus is on the process. The qualitative evaluation is therefore left to the aesthetic function; the dimensions considered depend in turn on how it has been defined (or generated).

4.3.3 Protocol for Evaluation. More specifically, the FACE model considers a creative act as a non-empty tuple of generative acts of eight possible types: an expression of a concept, i.e., an instance of an *(input, output)* pair produced by the system; a method for generating expressions of a concept; a concept; a method for generating concepts; an aesthetic measure; a method for generating aesthetic measures; an item of framing information, i.e., a piece of additional information or description regarding the generative act; a method for generating framing information. It is then possible to use it in a quantitative way (how many acts are produced); in a cumulative way (how many types of acts are considered); and in a qualitative way (by means of the aesthetic measure taken into consideration).

4.3.4 Critical Examination. The FACE model represents a very comprehensive set of concepts and methodologies for assessing machine creativity. However, one of the most challenging aspects of FACE is the definition of the aesthetic measure, which is not specified; potentially, it might be defined by the system itself, counting as a potential creative act. This may award systems performing self-evaluation, i.e., guiding their generation based on learned objectives. This in theory might mean the systems are incentivized to develop their own tastes, which is an important part of human creativity.

4.4 SPECS

4.4.1 Overview. The **Standardized Procedure for Evaluating Creative Systems (SPECS)** [136] is a framework for the evaluation of creative systems, which can easily be adapted to many different potential domains. The framework is based on the definition of fourteen "components" used to evaluate machine creativity.

4.4.2 Dimensions of Creativity Considered. The 14 key components of SPECS are: active involvement and persistence; dealing with uncertainty; domain competence; general intellect; generation of results; independence and freedom; intention and emotional involvement; originality; progression and development; social interaction and communication; spontaneity/subconscious processing; thinking and evaluation; value; and variety, divergence, and experimentation. However, SPECS does not restrict researchers to use all of them; moreover, domain-specific components can be added as well.

4.4.3 Protocol for Evaluation. SPECS is composed of three steps. The first one requires providing a definition of creativity that the system should satisfy, using the suggested components, and potentially other domain-specific ones. The second requires to specify the standards for evaluating such components. The third requires to test the system against the standards and report the results.

4.4.4 Critical Examination. This is an effective framework for working with computational creativity, but it cannot be considered as a practical evaluation method. Its effectiveness is strongly dependent on which components are considered and how they are evaluated for each specific task. In [137] the author discusses how SPECS satisfies Ritchie's criteria and it is more comprehensive and expressive than the FACE model and the creative tripod [52] (according to which a creative system should exhibit skills, appreciation, and imagination) in a meta-evaluation test. They also use a human opinion survey based on five criteria: correctness, usefulness, faithfulness (as a model of creativity), usability (of the methodology), and generality. SPECS is evaluated considering music improvisation generators; it is also judged by the developers of the generative systems. In particular, the authors show that SPECS can help obtain additional insights into how a generative model works and how it can be improved.

4.5 Creativity Implication Network

4.5.1 Overview. A different method of quantifying creativity is based on building an art graph called Creativity Implication Network [74]. Given a set of artworks, a directed graph can be defined by considering a vertex for each artwork. More specifically, an arc connecting artwork p_i to p_j is inserted if p_i has been created before p_j. A positive weight w_{ij} quantifying the similarity score between the two artworks under consideration is associated with each arc. The creativity of an artwork is then derived by means of computations on the resulting graph.

4.5.2 Dimensions of Creativity Considered. This method captures both value and novelty. The value is defined as the influence on future artworks. The novelty is defined as the dissimilarity between the artwork and the previous ones.

4.5.3 Protocol for Evaluation. The derivation of the Creativity Implication Network requires a similarity function to compute the similarity scores; its definition is left to the researchers, but it can be based on ML techniques (as in the original paper, where computer vision techniques were used). Given the network, the creativity of artwork p_i depends on the similarity with the previous artworks (the higher the similarity, the lower the creativity) and with the subsequent artworks (the higher the similarity, the higher the creativity).

4.5.4 Critical Examination. The Creativity Implication Network represents an effective way to deal with the creativity of sets of artworks. It considers both value and novelty, and it allows for using automated techniques in the computation of similarity. On the other hand, two potential drawbacks should be highlighted. The first one is related to artworks that occupy the position of "leaves" in the graph: if there are no subsequent works in the graph, their creativity would only be based on novelty, and not on value. The second one is more subtle, and it is about time-positioning. As demonstrated by [74], moving back an artwork has the effect of increasing its creativity; however, this appears conceptually wrong. As discussed in [22], the time location of an artwork is fundamental in the quantification of its creativity. It may happen that, due to the surprise component of creativity, an artwork that appears too *early* might not be considered as surprising because observers are not able to truly understand it; on the contrary, if it appears too *late*, it might be considered as obvious and not surprising at all. In conclusion, even if this approach is able to correctly capture value and novelty, it cannot capture the concept of surprise.

4.6 Generate-and-Test Setting

4.6.1 Overview. Generate-and-test setting [268] is a family of methods based on the separation of the generative process into two phases: generation and evaluation. First, the system generates a candidate artifact. Then, it evaluates its degree of creativity and outputs the artifact if the

evaluation is passed. For example, the authors of [280] use this approach to develop a computational creativity system for generating culinary recipes and menus called Chef Watson. [55] describes an augmentation of Painting Fool [53] with **Digital ARtist Communicating Intention (DARCI)** [202] in a generate-and-test setting (i.e., by using The Painting Fool for generation, and DARCI for evaluation). **Pseudo-Intelligent Evolutionary Real-time Recipe Engine (PIERRE)** [193] is also based on two models, one for generating recipes with a genetic algorithm, and one for evaluating them.

4.6.2 Dimensions of Creativity Considered. The dimensions of creativity considered depend on the specific implementation of the evaluation function. Different evaluation functions have been designed to evaluate, for example, quality and novelty [193, 280] and art appreciation [55]. It is worth noting that, given the generality of the approach, other evaluation functions could be designed to capture other aspects of machine creativity.

4.6.3 Protocol for Evaluation. The evaluation protocol strictly depends on the specific implementation of the evaluation. For instance, Chef Watson [280] uses two measures: flavorfulness for quality, and Bayesian surprise for novelty. Flavorfulness is computed by means of a regression model, built on olfactory pleasantness considering its constituent ingredients. Bayesian surprise [13] is a measure of surprise in terms of the impact of a piece of data that changes a prior distribution into a posterior distribution, calculated applying Bayes' theorem. The surprise is then the distance between the posterior and prior distributions. It is worth noting that it has been demonstrated that there exists a mathematical limit in the maximization of quality and novelty when novelty is expressed in terms of Bayesian surprise [279]. On the other hand, The Painting Fool [53] uses DARCI [202] in place of the evaluation function. DARCI is able to make associations between image features and descriptions of the images learned using a series of NNs as the basis for the appreciation. It has therefore been used as a sort of artificial art critic to complement The Painting Fool, allowing it to assess the validity of its own creations. Finally, PIERRE [193] evaluates the generated recipes again using novelty and quality. Novelty is computed by counting new combinations of ingredients used. Quality is based on two NNs that perform a regression of user ratings based on the amount of different ingredients, working at two levels of abstraction.

4.6.4 Critical Examination. The advantage of the generate-and-test setting is that a variety of evaluation functions can be defined. This allows, for instance, to evaluate the generative system by means of Boden's three criteria, while still considering the specific characteristics of the domain of interest. However, its applicability is not general: as we have seen in the previous section, many generative systems do not follow the proposed setting (e.g., input-based methods use the evaluation to guide the generation, thus merging the two stages).

4.7 Atoms of EVE'

4.7.1 Overview. [39] proposes an approach to measure aesthetic experience called Atoms of EVE', which is based on a probabilistic model of the world to derive expectations and explanations. The authors state that aesthetic arises in two ways: by forming E (*expectation*) while avoiding V (*violation*); and by forming E' (*explanation*) while resolving V.

4.7.2 Dimensions of Creativity Considered. Even if not explicitly considered, the three grounding concepts of Atoms of EVE' strongly intertwine with creativity. Expectation is close to value: it measures how much we are able to understand the object of interest. Violation is close to surprise: it is the unexpectedness of an object at a certain moment. Explanation is again close to value: it measures intelligibility (i.e., its usefulness). These same considerations have been expressed by

[40], where the author uses EVE' to define a creativity measure based on feelings (i.e., surprise), computed by means of violation, and meanings (i.e., value), computed by means of explanation.

4.7.3 Protocol for Evaluation. Expectation is computed as the posterior probability after the occurrence of a given object: it measures how much the prior belief can help explain that object. Violation is instead computed as the unexpectedness of that object. Together with apprehension, which is the unpredictability of the next object (before seeing it), violation returns the *tension*, one of the two fundamental measures of aesthetics. Finally, explanation measures how much the encountered violation can be explained by the posterior belief. Together with expectation, explanation returns a quantification of *pleasure*, the other fundamental measure of aesthetics.

4.7.4 Critical Examination. As observed before, while such a computation for surprise is common, this is not true for value. The focus on explanation provides an interesting way to mathematically define value. However, value is not only about finding meaning but also about utility, performance, and attractiveness [179]; this is not possible through this measure. Finally, novelty is not considered.

4.8 Common Model of Creativity for Design

4.8.1 Overview. The authors of [180] propose a model to evaluate creativity in the specific domain of design. They consider creativity as a relative measure in a conceptual space of potential and existing designs. Designs are represented by attribute-value pairs; and novelty, value, and surprise capture different aspects of creativity in their space. A similar approach is at the basis of the regent-dependent model [86], according to which artifacts are described through sets of pairs with a regent (i.e., an action or an attribute) and a dependent (i.e., the specific target or value of the regent).

4.8.2 Dimensions of Creativity Considered. The model proposed by [180] considers all three dimensions suggested by [22], i.e., novelty, value, and surprise. Novelty is considered as a matter of comparing objects in a descriptive space; it is the degree of difference. Value is related to performance, i.e., utility preferences associated with the attributes of an object. Finally, surprise is linked to violated expectations.

4.8.3 Protocol for Evaluation. The model is based on the analysis of the conceptual space of potential and existing designs defined by all the potential attribute-value pairs. Novelty is evaluated with respect to a description space, i.e., by considering each product as the set of its descriptive attributes. Value is considered with respect to a performance space, i.e., by considering attributes that have utility preferences associated to them. Finally, surprise is based on finding violations of patterns that are possible to anticipate in the space of both current and possible designs. The K-means clustering algorithm is used to organize known designs by means of their attributes. Then, novelty, value, and surprise measures of a new design are obtained by looking at the distance to the nearest cluster centroid.

4.8.4 Critical Examination. Even if it explicitly targets the design domain, this approach is able to combine the three dimensions of creativity by Boden. Nonetheless, it is limited by the fact that artifacts have to be described through an attribute-value pair representation. In particular, a large number of features might be needed. Otherwise, we might lose aspects of the artifacts that are fundamental to correctly quantify creativity. Since it is not possible to know the fundamental features in advance, the method requires one to enumerate as many features as possible. However, the risk is to define an excessive number of non-informative attributes, making the computation of the metrics too computationally expensive. In fact, the data points become increasingly "sparse"

as dimensionality increases; many techniques (especially clustering) are based on distance, and therefore they may suffer from the curse of dimensionality [258]. Finally, as for classic machine learning techniques, there is the need to manually define and extract the chosen features from unstructured data, which is a time-consuming and potentially prone-to-error activity. A possible way to overcome the problems related to feature extraction and the curse of dimensionality might be to adopt deep learning techniques, given their effectiveness with unstructured data.

4.9 Unexpectedness Framework

4.9.1 Overview. The authors of [99] suggest that a framework of unexpectedness (i.e., violation of an observer's expectations) can deal with novelty, surprise, and domain transformation (also called transformativity). Although they do not claim it can be a measure of creativity on its own, and that value should be added as well, they suggest it can become a vital component in computational creativity evaluation.

4.9.2 Dimensions of Creativity Considered. The authors of [99] suggest that unexpectedness can be used to compute three dimensions of creativity: novelty, surprise, and transformativity. Indeed, novelty is about the possibility of violating the observer's expectations about the continuity of a domain; if the current model of belief is not applicable to the current artifact, it can be considered novel. Surprise is instead about the possibility of violating the observer's expectations about an artifact. Finally, transformativity is about the possibility of violating the observer's expectations about the conceptual space itself (i.e., finding that the rules governing it were not accurate).

4.9.3 Protocol for Evaluation. The unexpectedness framework should allow one to model expectation. Notably, expectation should be linked with the socio-cultural context of the observer, since it is the observer that forms expectation, not the domain itself. In particular, an expectation is generated by a prediction about the predicted (i.e., the dependent variables of the artifact) given a condition (i.e., a relationship between the predicted property and some other property of the object) that applies within a scope (i.e., the set of possible artifacts to which the expectations apply). An observation that falls within that scope can then be measured for congruence with respect to that expectation.

4.9.4 Critical Examination. The unexpectedness measure appears to be able to provide researchers and practitioners with a way to derive novelty and surprise. Notably, it also captures transformativity, clarifying at the same time how *simple* surprise differs from it, i.e., that surprise is related to expectations about a single artifact, while transformativity is related to expectations about the entire domain. However, it requires defining its conceptual space in terms of explicit rules, which can be violated (and in a way that allows a violation to be detected). In addition, it does not include value in the assessment of machine creativity.

4.10 Essential Criteria of Creativity

4.10.1 Overview. The metric proposed by [179] is based on three components: novelty, value, and surprise. It relies on the idea that a creativity metric has to be independent not only from the domain but also from the producer.

4.10.2 Dimensions of Creativity Considered. The criteria of creativity defined by [179] cover exactly Boden's three criteria. In particular, novelty is intended here as a measure of how different the artifact is from known artifacts in its class. Value is quantified by considering how the potentially creative artifact compares in utility, performance, or attractiveness to other artifacts in its class. Finally, unexpectedness is defined as the variation from expectation developed for the next new artifact in its class.

4.10.3   Protocol for Evaluation. Novelty is calculated as the distance between the artifact of interest and the other artifacts in its class. The partition into classes is obtained by means of a clustering algorithm. Surprise is calculated by considering whether or not the artifact agrees with the expected next artifact in the pattern extracted from recent artifacts. More specifically, it is calculated as the difference between the expected next artifact and the real next artifact. Such a pattern is predicted by a self-supervised neural network; predictions are refined using reinforcement learning to correct the learned trajectory in case of sequential data. Finally, value is calculated as the weighted sum of the performance variables of the artifact. The weights depend on a co-evolutionary algorithm with a fitness function that can change over time in case the current population of artifacts changes.

4.10.4   Critical Examination. The method considers all three of Boden's criteria; it is not linked to a specific domain, or the producer itself; it deals with the evolution of creativity, capturing its volatile nature at the same time. However, in our opinion, it is limited in terms of applicability by the fact that it requires the definition of performance variables (similarly to other approaches based on attribute-value pairs, see Section 4.8.4). Moreover, the setting of the parameters of the clustering algorithms at the basis of this method and the definition of distances among artifacts require human fine-tuning.

4.11   Computational Metrics for Storytelling

4.11.1   Overview. For the specific case of storytelling, the authors of [141] propose a set of computational metrics to compute the evaluation of novelty, surprise, rarity, and recreational effort.

4.11.2   Dimensions of Creativity Considered. Novelty and surprise are evaluated according to the standard Boden's definition, while rarity is intended as the presence of rare combinations of properties and recreational effort as the difficulty in achieving a specific result.

4.11.3   Protocol for Evaluation. Novelty is computed as the average semantic distance between the dominant terms included in the textual representation of the story, compared to the average semantic distance of the dominant terms in all stories. Surprise is computed as the average semantic distance between the consecutive fragments of each story. Rarity is computed as the distance between the individual clusters of each term in each story and those in the story set. Finally, recreational effort is computed as the number of different clusters each story contains.

4.11.4   Critical Examination. Although value is not considered, the proposed metrics appear to be appropriate to evaluate novelty and surprise. Nonetheless, they suffer from two problems: they are intrinsically domain-specific and they require that all the types of clusters are defined correctly, which is very difficult to ensure in practice.

5   OUTLOOK AND CONCLUSION

In this survey, we have provided the reader with an overview of the state of the art at the intersection between creativity and machine learning. Firstly, we have introduced the concept of machine creativity, including key concepts and definitions. Secondly, we have described a variety of generative learning techniques, considering their potential applications and limitations. Finally, we have discussed several evaluation frameworks for quantifying machine creativity, highlighting their characteristics and the dimensions they are able to capture.

Even if the field of machine creativity has witnessed increasing interest and popularity in recent years, there are still several open challenges. First of all, an interesting direction is the exploration of *creativity-oriented* objective functions, to directly train models to be creative or to navigate the induced latent space to find creative solutions. Another open problem is the definition of

techniques to explore or transform the space of solutions. A fundamental area is the definition of novel and robust evaluation techniques for both generated and real artifacts. As discussed in Section 4, deep learning might be used as a basis for the definition of metrics for machine creativity. It should be noted that there is also an ongoing debate about the role of human versus machine evaluation [153]. Another promising research direction concerns the machine interpretation of art [1]. Moreover, machine learning techniques might also be used to investigate psychological dimensions of creativity [2]. There are also foundational questions related to generative deep learning and copyright [84]. For example, it is not clear if machine-generated works could be protected by Intellectual Property, and, if they are, who should be the owner of the related rights [237]. In addition, other problems concerning copyright should be considered, such as if and when training over protected work is permitted [252]. Another important ongoing debate is about authorship and the human role in creative fields in the era of AI.[5]

The models and frameworks discussed in this work show the remarkable potential of generative learning for machine creativity. We hope that this survey will represent a valuable guide for researchers and practitioners working in this fascinating area.

REFERENCES

[1] Panos Achlioptas, Maks Ovsjanikov, Kilichbek Haydarov, Mohamed Elhoseiny, and Leonidas Guibas. 2021. ArtEmis: Affective language for art. In *Proc. of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*.

[2] Sergio Agnoli, Laura Franchin, Enrico Rubaltelli, and Giovanni Emanuele Corazza. 2019. The emotionally intelligent use of attention and affective arousal under creative frustration and creative success. *Personality and Individual Differences* 142 (2019), 242–248.

[3] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. MusicLM: Generating Music From Text. (2023). arXiv:2301.11325

[4] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. 2021. VATT: Transformers for multimodal self-supervised learning from raw video, audio and text. In *Advances in Neural Information Processing Systems (NeurIPS'21)*.

[5] Andrei G. Aleinikov, Sharon Kackmeister, and Ron Koenig. 2000. *Creating Creativity: 101 Definitions (what Webster Never Told You)*. Alden B. Dow Creativity Center Press, Midland, MI.

[6] Teresa M. Amabile. 1983. The social psychology of creativity: A componential conceptualization. *Journal of Personality and Social Psychology* 45, 2 (1983), 357–376.

[7] Jyoti Aneja, Alexander G. Schwing, Jan Kautz, and Arash Vahdat. 2021. A contrastive learning approach for training variational autoencoder priors. In *Advances in Neural Information Processing Systems (NeurIPS'21)*.

[8] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. 2021. ViViT: A video vision transformer. In *Proc. of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV'21)*.

[9] Charles Babbage. 1864. Of the analytical engine. In *Passages from the Life of a Philosopher*. Vol. 3. Longman, Green, Longman, Roberts, & Green, 112–141.

[10] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of the 3rd International Conference on Learning Representations (ICLR'15)*.

[12] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional AI: Harmlessness from AI Feedback. (2022). arXiv:2212.08073

---

[5]This is one of the ethical dilemmas highlighted by UNESCO in its Preliminary study on the Ethics of Artificial Intelligence.

[13] Pierre Baldi and Laurent Itti. 2010. Of bits and wows: A Bayesian theory of surprise with applications to attention. *Neural Networks* 23 (2010), 649–666.

[14] Hangbo Bao, Li Dong, and Furu Wei. 2022. BEiT: BERT pre-training of image transformers. In *Proc. of the 10th International Conference on Learning Representations (ICLR'22)*.

[15] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* 5, 2 (1994), 157–166.

[16] Walter Benjamin. 2008. *The Work of Art in the Age of Mechanical Reproduction*. Penguin Books Ltd, London, UK.

[17] Daniel E. Berlyne. 1971. *Aesthetics and Psychobiology*. Appleton-Century-Crofts, New York, NY.

[18] Sebastian Berns and Simon Colton. 2020. Bridging generative deep learning and computational creativity. In *Proc. of the 11th International Conference on Computational Creativity (ICCC'20)*.

[19] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. 2024. Improving Image Generation with Better Captions. (2024). Retrieved April 30, 2024 from https://cdn.openai.com/papers/dall-e-3.pdf

[20] Federico Betti, Giorgia Ramponi, and Massimo Piccardi. 2020. Controlled text generation with adversarial learning. In *Proc. of the 13th International Conference on Natural Language Generation (INLG'20)*.

[21] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. 2023. Training diffusion models with reinforcement learning. In *ICML'23 Workshop on Efficient Systems for Foundation Models*.

[22] Margaret A. Boden. 2003. *The Creative Mind: Myths and Mechanisms*. Routledge, London, UK.

[23] Rishi Bommasani, Drew Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney Arx, Michael Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Davis, Dora Demszky, and Percy Liang. 2021. On the Opportunities and Risks of Foundation Models. (2021). arXiv:2108.07258

[24] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. 2021. Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 11 (2021), 7327–7347.

[25] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2022. AudioLM: A Language Modeling Approach to Audio Generation. (2022). arXiv:2209.03143

[26] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proc. of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNNL'16)*.

[27] Oliver Bown. 2021. *Beyond the Creative Species*. The MIT Press, Cambridge, MA.

[28] Herbie Bradley, Andrew Dai, Hannah Teufel, Jenny Zhang, Koen Oostermeijer, Marco Bellagente, Jeff Clune, Kenneth Stanley, Grégory Schott, and Joel Lehman. 2024. Quality-diversity through AI feedback. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*.

[29] Selmer Bringsjord, Paul Bello, and David Ferrucci. 2001. Creativity, the Turing test, and the (better) Lovelace test. *Minds and Machines* 11 (2001), 3–27.

[30] Terence Broad, Sebastian Berns, Simon Colton, and Mick Grierson. 2021. Active divergence with generative deep learning - a survey and taxonomy. In *Proc. of the 12th International Conference on Computational Creativity (ICCC'21)*.

[31] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale GAN training for high fidelity natural image synthesis. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*.

[32] Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei Efros, and Tero Karras. 2022. Generating long videos of dynamic scenes. In *Advances in Neural Information Processing Systems (NeurIPS'22)*.

[33] Tim Brooks, Bill Peebles, Connor Homes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Wing Yin Ng, Ricky Wang, and Aditya Ramesh. 2024. Video generation models as world simulators. (2024). Retrieved April 30, 2024 from https://openai.com/research/video-generation-models-as-world-simulators

[34] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS'20)*.

[35] Razvan C. Bunescu and Oseremen O. Uduehi. 2019. Learning to surprise: A composer-audience architecture. In *Proc. of the 10th International Conference on Computational Creativity (ICCC'19)*.

[36] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. 2019. Large-scale study of curiosity-driven learning. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*.

[37] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. 2016. Importance weighted autoencoders. In *Proc. of the 4th International Conference on Learning Representations (ICLR'16)*.

[38] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding Disentangling in β-VAE. (2018). arXiv:1804.03599

[39] Kevin Burns. 2006. Atoms of EVE': A Bayesian basis for esthetic analysis of style in sketching. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 20 (2006), 185–199.

[40] Kevin Burns. 2015. Computing the creativeness of amusing advertisements: A Bayesian model of Burma-Shave's muse. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 29 (2015), 109–128.

[41] Amílcar Cardoso, Tony Veale, and Geraint A. Wiggins. 2009. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine* 30, 3 (2009), 15.

[42] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Bıyık, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. 2023. Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback. (2023). *Transactions on Machine Learning Research.*

[43] Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 2023. Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing. In *Proc. of the AAAI'23 Workshop on Creative AI Across Modalities.*

[44] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *Proc. of the 37th International Conference on Machine Learning (ICML'20).*

[45] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidi Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. (2021). arXiv:2107.03374

[46] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS'16).*

[47] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating Long Sequences with Sparse Transformers. (2019). arXiv:1904.10509

[48] Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS'17).*

[49] Eric Chu. 2018. Artistic influence GAN. In *Proc. of the NeurIPS'18 Workshop on Machine Learning for Creativity and Design.*

[50] Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. 2018. Neural text generation in stories using entity representations as context. In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).*

[51] Harold Cohen. 1988. How to draw three people in a botanical garden. In *Proc. of the 7th AAAI National Conference on Artificial Intelligence (AAAI'88).*

[52] Simon Colton. 2008. Creativity versus the perception of creativity in computational systems. In *Proc. of the 2008 AAAI Spring Symposium.*

[53] Simon Colton. 2012. The painting fool: Stories from building an automated painter. In *Computers and Creativity.* Springer, Berlin, 3–38.

[54] Simon Colton, John William Charnley, and Alison Pease. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proc. of the 2nd International Conference on Computational Creativity (ICCC'11).*

[55] Simon Colton, Jakob Halskov, Dan Ventura, Ian Gouldstone, Michael Cook, and Blanca Pérez-Ferrer. 2015. The painting fool sees! New projects with the automated painter. In *Proc. of the 6th International Conference on Computational Creativity (ICCC'15).*

[56] Simon Colton and Geraint A. Wiggins. 2012. Computational creativity: The final frontier?. In *Proc. of the 20th European Conference on Artificial Intelligence (ECAI'12).*

[57] David Cope. 1989. Experiments in musical intelligence (EMI): Non-Linear Linguistic-Based composition. *Interface* 18 (1989), 117–139.

[58] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. 2022. VQGAN-CLIP: Open domain image generation and editing with natural language guidance. In *Proc. of the 17th European Conference on Computer Vision (ECCV'22)*.

[59] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*.

[60] Yashar Deldjoo, Tommaso Di Noia, and Felice Antonio Merra. 2021. A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks. *Comput. Surveys* 54, 2 (2021), 1–38.

[61] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

[62] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A Generative Model for Music. (2020). arXiv:2005.00341

[63] Prafulla Dhariwal and Alexander Quinn Nichol. 2021. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS'21)*.

[64] Li Ding, Jenny Zhang, Jeff Clune, Lee Spector, and Joel Lehman. 2023. Quality diversity through human feedback. In *Proc. of the NeurIPS'23 Workshop ALOE*.

[65] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. 2021. CogView: Mastering text-to-image generation via transformers. In *Advances in Neural Information Processing Systems (NeurIPS'21)*.

[66] Chris Donahue, Julian McAuley, and Miller Puckette. 2019. Adversarial audio synthesis. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*.

[67] Jeff Donahue, Philipp Krahenbuhl, and Trevor Darrell. 2017. Adversarial feature learning. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*.

[68] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. 2018. MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*.

[69] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*.

[70] Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proc. of the 39th International Conference on Machine Learning (ICML'22)*.

[71] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. 2017. Adversarially learned inference. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*.

[72] Douglas Eck and Jurgen Schmidhuber. 2002. Finding temporal structure in music: Blues improvisation with LSTM recurrent networks. In *Proc. of the 12th IEEE Workshop on Neural Networks for Signal Processing*.

[73] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. 2017. CAN: Creative adversarial networks, generating "art" by learning about styles and deviating from style norms. In *Proc. of the 8th International Conference on Computational Creativity (ICCC'17)*.

[74] Ahmed Elgammal and Babak Saleh. 2015. Quantifying creativity in art networks. In *Proc. of the 6th International Conference on Computational Creativity (ICCC'15)*.

[75] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2023. CLAP: Learning audio concepts from natural language supervision. In *Proc. of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'23)*.

[76] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. 2019. GAN-Synth: Adversarial neural audio synthesis. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*.

[77] S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. 2016. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems (NeurIPS'16)*.

[78] Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proc. of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*.

[79] William Fedus, Ian Goodfellow, and Andrew M. Dai. 2018. MaskGAN: Better text generation via filling in the _____. In *Proc. of the 6th International Conference on Learning Representations (ICLR'18)*.

[80] Nanyi Fei, Zhiwu Lu, Yizhao Gao, Guoxing Yang, Yuqi Huo, Jingyuan Wen, Haoyu Lu, Ruihua Song, Xin Gao, Tao Xiang, Hao Sun, and Ji-Rong Wen. 2022. Towards artificial general intelligence via a multimodal foundation model. *Nature Communications* 13, 1 (2022), 3094.

[81] Pablo Fernandes, Joao Nuno Correia, and Penousal Machado. 2020. Evolutionary latent space exploration of generative adversarial networks. In *Proc. of the 2020 International Conference on the Applications of Evolutionary Computation (Part of EvoStar'20)*.

[82] Chrisantha Fernando, S. M. Ali Eslami, Jean-Baptiste Alayrac, Piotr Mirowski, Dylan Banarse, and Simon Osindero. 2021. Generative Art Using Neural Visual Grammars and Dual Encoders. (2021). arXiv:2105.00162

[83] David Foster. 2019. *Generative Deep Learning*. O'Reilly, Sebastopol, CA.

[84] Giorgio Franceschelli and Mirco Musolesi. 2022. Copyright in generative deep learning. *Data & Policy* 4 (2022), e17.

[85] Giorgio Franceschelli and Mirco Musolesi. 2024. Reinforcement learning for generative AI: State of the art, opportunities and open research challenges. *Journal of Artificial Intelligence Research* 79 (2024), 417–446.

[86] Celso França, Luís Fabrício Wanderley Góes, Alvaro Amorim, Rodrigo C. O. Rocha, and Alysson Ribeiro Da Silva. 2016. Regent-dependent creativity: A domain independent metric for the assessment of creative artifacts. In *Proc. of the 7th International Conference on Computational Creativity (ICCC'16)*.

[87] Ankush Ganguly and Samuel W. F. Earp. 2021. An Introduction to Variational Inference. (2021). arXiv:2108.13083

[88] Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61, 1 (2018), 65–170.

[89] Leon Gatys, Alexander Ecker, and Matthias Bethge. 2016. A neural algorithm of artistic style. *Journal of Vision* 16, 12 (2016), 326.

[90] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. 2017. Controlling perceptual factors in neural style transfer. In *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*.

[91] Berys Gaut. 2003. Creativity and imagination. In *The Creation of Art: New Essays in Philosophical Aesthetics*. Cambridge University Press, 148–173.

[92] Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proc. of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM'21)*.

[93] Gemini Team and Google. 2023. Gemini: A Family of Highly Capable Multimodal Models. (2023). arXiv:2312.11805

[94] Gemini Team and Google. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. (2024). Retrieved April 30, 2024 from https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report.pdf

[95] John Gero. 2000. Computational models of innovative and creative design processes. *Technological Forecasting and Social Change* 64, 2-3 (2000), 183–196.

[96] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. DiffuSeq: Sequence to sequence text generation with diffusion models. In *Proc. of the 11th International Conference on Learning Representations (ICLR'23)*.

[97] Ian Goodfellow. 2017. NIPS 2016 Tutorial: Generative Adversarial Networks. (2017). arXiv:1701.00160

[98] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS'14)*.

[99] Kazjon Grace and Mary Lou Maher. 2014. What to expect when you're expecting: The role of unexpectedness in computationally evaluating creativity. In *Proc. of the 5th International Conference on Computational Creativity (ICCC'14)*.

[100] Daniele Gravina, Antonios Liapis, and Georgios Yannakakis. 2016. Surprise search: Beyond objectives and novelty. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO'16)*.

[101] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. 2015. DRAW: A recurrent neural network for image generation. In *Proc. of the 32nd International Conference on Machine Learning (ICML'15)*.

[102] Albert Gu, Karan Goel, and Christopher Re. 2022. Efficiently modeling long sequences with structured state spaces. In *Proc. of the 10th International Conference on Learning Representations (ICLR'22)*.

[103] Jie Gui, Z. Sun, Yonggang Wen, Dacheng Tao, and Ye Jie-ping. 2021. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2021), 3313–3332.

[104] Gabriel L. Guimaraes, Benjamin Sanchez-Lengeling, Pedro Luis Cunha Farias, and Alan Aspuru-Guzik. 2017. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. (2017). arXiv:1705.10843

[105] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. 2017. PixelVAE: A latent variable model for natural images. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*.

[106] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are All You Need. (2023). arXiv:2306.11644

[107] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*.

[108] Matthew Guzdial and Mark O. Riedl. 2019. An Interaction Framework for Studying Co-Creative AI. (2019). arXiv:1903.09709

[109] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. 2018. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science* 4, 2 (2018), 268–276.

[110] G. M. Harshvardhan, Mahendra Kumar Gourisaria, Manjusha Pandey, and Siddharth Swarup Rautaray. 2020. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review* 38 (2020), 100285.

[111] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*. 16000–16009.

[112] Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A. Lemley, and Percy Liang. 2023. Foundation Models and Fair Use. (2023). arXiv:2303.15715

[113] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*.

[114] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *Proc. of the NeurIPS'15 Deep Learning and Representation Learning Workshop*.

[115] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS'20)*.

[116] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. 2022. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research* 23, 47 (2022), 1–33.

[117] Jonathan Ho and Tim Salimans. 2021. Classifier-free diffusion guidance. In *Proc. of the NeurIPS'21 Workshop on Deep Generative Models and Downstream Applications*.

[118] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. 2022. Video diffusion models. In *Proc. of the ICLR'22 Workshop on Deep Generative Models for Highly Structured Data*.

[119] Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[120] Douglas R. Hofstadter and Melanie Mitchell. 1994. The copycat project: A model of mental fluidity and analogy-making. In *Advances in Connectionist and Neural Computation Theory, Vol. 2. Analogical Connections*. Ablex Publishing, 31–112.

[121] Jack Hopkins and Douwe Kiela. 2017. Automatically generating rhythmic verse with neural networks. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[122] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2019. Music transformer. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*.

[123] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. 2022. MuLan: A joint embedding of music audio and natural language. In *Proc. of the 23rd International Society for Music Information Retrieval Conference (ISMIR'22)*.

[124] Zhewei Huang, Shuchang Zhou, and Wen Heng. 2019. Learning to paint with model-based deep reinforcement learning. In *Proc. of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV'19)*.

[125] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*.

[126] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with Gumbel-Softmax. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*.

[127] Joel Jang, Sumin Shin, and Yoonjeon Kim. 2022. Music2Video: Automatic Generation of Music Video with Fusion of Audio and Text. (2022). arXiv:2201.03809

[128] Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E. Turner, and Douglas Eck. 2017. Sequence tutor: Conservative fine-tuning of sequence generation models with KL-control. In *Proc. of the 34th International Conference on Machine Learning (ICML'17)*.

[129] Natasha Jaques, Shixiang Gu, Richard E. Turner, and Douglas Eck. 2016. Generating music by fine-tuning recurrent neural networks with reinforcement learning. In *Proc. of the NeurIPS'16 Deep Reinforcement Learning Workshop*.

[130] Natasha Jaques, Shixiang Gu, Richard E. Turner, and Douglas Eck. 2017. Tuning recurrent neural networks with reinforcement learning. In *Proc. of the ICLR'17 Workshop*.

[131] Divyansh Jha, Hanna Chang, and Mohamed Elhoseiny. 2021. Wolfflin's affective generative analysis for visual art. In *Proc. of the 20th International Conference on Computational Creativity (ICCC'21)*.

[132] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proc. of the 38th International Conference on Machine Learning (ICML'21)*.

[133] Nan Jiang, Sheng Jin, Zhiyao Duan, and Changshui Zhang. 2020. RL-Duet: Online music accompaniment generation using deep reinforcement learning. In *Proc. of the 34th AAAI Conference on Artificial Intelligence, the 32nd Innovative Applications of Artificial Intelligence Conference, the 10th AAAI Symposium on Educational Advances in Artificial Intelligence*.

[134] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. 2017. Towards the Automatic Anime Characters Creation with Generative Adversarial Networks. (2017). arXiv:1708.05509

[135] Michael I. Jordan, Zoubin Ghahrmamani, Tommi S. Jaakkola, and Lawrence K. Saul. 1999. An introduction to variational methods for graphical models. *Machine Learning* 37 (1999), 183–233.

[136] Anna Jordanous. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4 (2012), 246–279.

[137] Anna Jordanous. 2014. Stepping back to progress forwards: Setting standards for meta-evaluation of computational creativity. In *Proc. of the 5th International Conference on Computational Creativity (ICCC'14)*.

[138] Anna Jordanous. 2016. Four PPPPerspectives on computational creativity in theory and in practice. *Connection Science* 28, 2 (2016), 294–216.

[139] John M. Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zidek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andy Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596 (2021), 583–589.

[140] Artur Kadurin, Sergey Nikolenko, Kuzma Khrabrov, Alex Aliper, and Alex Zhavoronkov. 2017. druGAN: An advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Molecular Pharmaceutics* 14, 9 (2017), 3098–3104.

[141] Pythagoras Karampiperis, Antonis Koukourikos, and Evangelia Koliopoulou. 2014. Towards machines for measuring creativity: The use of computational tools in storytelling activities. In *Proc. of the 2014 IEEE 14th International Conference on Advanced Learning Technologies (ICALT'14)*.

[142] Andrej Karpathy. 2015. The Unreasonable Effectiveness of Recurrent Neural Networks. (2015). Retrieved April 30, 2024 from https://karpathy.github.io/2015/05/21/rnn-effectiveness

[143] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. of the 6th International Conference on Learning Representations (ICLR'18)*.

[144] Tero Karras, Miika Aittala, Samuli Laine, Erik Harkonen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS'21)*.

[145] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proc. of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*.

[146] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of StyleGAN. In *Proc. of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*.

[147] Hadi Kazemi, Seyed Mehdi Iranmanesh, and Nasser Nasrabadi. 2019. Style and content disentanglement in generative adversarial networks. In *Proc. of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV'19)*.

[148] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in vision: A survey. *Comput. Surveys* 54, 10s (2022), 1–41.

[149] Durk P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems (NeurIPS'14)*.

[150] Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *Proc. of the 2nd International Conference on Learning Representations (ICLR'14)*.

[151] Diederik P. Kingma and Max Welling. 2019. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning* 12, 4 (2019), 307–392.

[152] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2021. DiffWave: A versatile diffusion model for audio synthesis. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*.

[153] Carolyn Lamb, Daniel G. Brown, and Charles L. A. Clarke. 2018. Evaluating computational creativity: An interdisciplinary tutorial. *Comput. Surveys* 51, 2 (2018), 1–34.

[154] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*.

[155] Pat Langley, Herbert A. Simon, Gary L. Bradshaw, and Jan M. Zytkow. 1987. *Scientific Discovery: Computational Explorations of the Creative Process*. The MIT Press, Cambridge, MA.

[156] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. 2016. Autoencoding beyond pixels using a learned similarity metric. In *Proc. of the 33rd International Conference on Machine Learning (ICML'16)*.

[157] Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[158] Phong Le and Willem Zuidema. 2016. Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs. In *Proc. of the 1st Workshop on Representation Learning for NLP*.

[159] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. (2023). arXiv:2309.00267

[160] Helena H. Lee, Ke Shu, Palakorn Achananuparp, Philips Kokoh Prasetyo, Yue Liu, Ee-Peng Lim, and Lav R. Varshney. 2020. RecipeGPT: Generative pre-training based cooking recipe generation and evaluation system. In *Companion Proceedings of the Web Conference 2020*.

[161] Joel Lehman and Kenneth O. Stanley. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation* 19, 2 (2011), 189–223.

[162] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proc. of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*.

[163] Yoav Levine, Itay Dalmedigos, Ori Ram, Yoel Zeldes, Daniel Jannai, Dor Muhlgay, Yoni Osin, Opher Lieber, Barak Lenz, Shai Shalev-Shwartz, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2022. Standing on the Shoulders of Giant Frozen Language Models. (2022). arXiv:2204.10019

[164] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*.

[165] Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*.

[166] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019. Neural speech synthesis with transformer network. In *Proc. of the 33rd AAAI Conference on Artificial Intelligence and 31st Innovative Applications of Artificial Intelligence Conference and 9th AAAI Symposium on Educational Advances in Artificial Intelligence*.

[167] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. 2022. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems (NeurIPS'22)*.

[168] Antonios Liapis, Hector P. Martinez, Julian Togelius, and Georgios N. Yannakakis. 2013. Transforming exploratory creativity with DeLeNoX. In *Proc. of the 4th International Conference on Computational Creativity (ICCC'13)*.

[169] Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. 2013. Enhancements to constrained novelty search: Two-population novelty search for generating game content. In *Proc. of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO'13)*.

[170] Bryan Lim and Stefan Zohren. 2021. Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A* 379 (2021), 20200209.

[171] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. 2023. AudioLDM: Text-to-audio generation with latent diffusion models. In *Proc. of the 40th International Conference on Machine Learning (ICML'23)*. 21450–21474.

[172] Jialin Liu, Sam Snodgrass, Ahmed Khalifa, Sebastian Risi, Georgios N. Yannakakis, and Julian Togelius. 2021. Deep learning for procedural content generation. *Neural Computing and Applications* 33 (2021), 19–37.

[173] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. (2019). arXiv:1907.11692

[174] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*.

[175] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. 2016. Auxiliary deep generative models. In *Proc. of the 33rd International Conference on Machine Learning (ICML'16)*.

[176] Penousal Machado, Juan Romero, Antonino Santos, Amílcar Cardoso, and Alejandro Pazos. 2007. On the development of evolutionary artificial artists. *Computers and Graphics* 31, 6 (2007), 818–826.

[177] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*.

[178] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proc. of the 6th International Conference on Learning Representations (ICLR'18)*.

[179] Mary Maher. 2010. Evaluating creativity in humans, computers, and collectively intelligent systems. In *Proc. of the 1st DESIRE Network Conference on Creativity and Innovation in Design*.

[180] Mary Maher and Doug Fisher. 2012. Using AI to evaluate creative designs. In *Proc. of the 2nd International Conference on Design Creativity (ICDC'12)*.

[181] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. 2016. Adversarial autoencoders. In *Proc. of the 4th International Conference on Learning Representations (ICLR'16)*.

[182] Lara J. Martin, Prithviraj Ammanabrolu, William Hancock, Shruti Singh, Brent Harrison, and Mark O. Riedl. 2018. Event representations for automated story generation with deep neural nets. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*.

[183] James R. Meehan. 1977. TALE-SPIN, an interactive program that writes stories. In *Proc. of the 5th International Joint Conference on Artificial Intelligence - Volume 1 (IJCAI'77)*.

[184] Luigi Federico Menabrea and Ada Lovelace. 1843. Sketch of the analytical engine invented by Charles Babbage. In *Scientific Memoirs*. Vol. 3. Richard and John E. Taylor, 666–731.

[185] Oscar Mendez-Lucio, Benoit Baillif, Djork-Arné Clevert, David Rouquié, and Joerg Wichard. 2020. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nature Communications* 11, 10 (2020), 1–10.

[186] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. 2017. Unrolled generative adversarial networks. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*.

[187] Arthur I. Miller. 2019. *The Artist in the Machine*. The MIT Press, Cambridge, MA.

[188] Marvin Minsky. 2006. *The Emotion Machine*. Simon & Schuster, New York, NY.

[189] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. (2014). arXiv:1411.1784

[190] Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. 2021. Symbolic music generation with diffusion models. In *Proc. of the 22nd Int. Society for Music Information Retrieval Conf. (ISMIR'21)*.

[191] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518 (2015), 529–533.

[192] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. 2015. Inceptionism: Going Deeper into Neural Networks. (2015). Retrieved April 30, 2024 from https://blog.research.google/2015/06/inceptionism-going-deeper-into-neural.html

[193] Richard G. Morris, Scott H. Burton, Paul Bodily, and Dan Ventura. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proc. of the 3rd International Conference on Computational Creativity (ICCC'12)*.

[194] Saman Motamed, Patrik Rogalla, and Farzad Khalvati. 2021. RANDGAN: Randomized generative adversarial network for detection of COVID-19 in Chest X-Ray. *Scientific Reports* 11 (2021), 8602.

[195] Allen Newell, J. C. Shaw, and Herbert A. Simon. 1962. The processes of creative thinking. In *Contemporary Approaches to Creative Thinking: A Symposium Held at the University of Colorado*. Atherton Press, 63–119.

[196] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. 2017. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*.

[197] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems (NeurIPS'16)*.

[198] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proc. of the 39th International Conference on Machine Learning (ICML'22)*.

[199] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. 2022. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. (2022). arXiv:2212.08751

[200] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *Proc. of the 38th International Conference on Machine Learning (ICML'21)*.

[201] Weili Nie, Nina Narodytska, and Ankit Patel. 2019. RelGAN: Relational generative adversarial networks for text generation. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*.

[202] David Norton, Derral Heath, and Dan Ventura. 2010. Establishing appreciation in a creative system. In *Proc. of the 1st International Conference on Computational Creativity (ICCC'15)*.

[203] Augustus Odena. 2016. Semi-Supervised Learning with Generative Adversarial Networks. (2016). arXiv:1606.01583

[204] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier GANs. In *Proc. of the 34th International Conference on Machine Learning (ICML'17)*.

[205] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature Visualization. (2017). *Distill*.

[206] OpenAI. 2022. Introducing ChatGPT. (2022). Retrieved April 30, 2024 from https://openai.com/blog/chatgpt

[207] OpenAI. 2023. GPT-4 Technical Report. (2023). arXiv:2303.08774

[208] Rafael Pardinas, Gabriel Huang, David Vazquez, and Alexandre Piché. 2023. Leveraging human preferences to master poetry. In *Proc. of the AAAI'23 Workshop on Creative AI Across Modalities*.

[209] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In *Proc. of the 35th International Conference on Machine Learning (ICML'18)*.

[210] Christine Payne. 2019. MuseNet. (2019). Retrieved April 30, 2024 from https://openai.com/blog/musenet

[211] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. 2023. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP'23*.

[212] Francisco C. Pereira, Mateus Mendes, Pablo Gervas, and Amilcar Cardoso. 2005. Experiments with assessment of creative systems: An application of Ritchie's Criteria. In *Proc. of the IJCAI'15 Second Computational Creativity Workshop*.

[213] Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. GhostWriter: Using an LSTM for automatic rap lyric generation. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*.

[214] Racter. 1984. *The Policeman's Beard is Half Constructed*. Warner Books, Inc., New York, NY.

[215] Alec Radford. 2018. Improving Language Understanding with Unsupervised Learning. (2018). Retrieved April 30, 2024 from https://openai.com/blog/language-unsupervised

[216] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proc. of the 38th International Conference on Machine Learning (ICML'21)*.

[217] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proc. of the 4th International Conference on Learning Representations (ICLR'16)*.

[218] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019). Retrieved April 30, 2024 from https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[219] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proc. of the 37th Conference on Neural Information Processing Systems (NeurIPS'23)*.

[220] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.

[221] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. (2022). arXiv:2204.06125

[222] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proc. of the 4th International Conference on Learning Representations (ICLR'16)*.

[223] Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*.

[224] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. In *Proc. of the 33rd International Conference on Machine Learning (ICML'16)*.

[225] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. 2022. A Generalist Agent. (2022). *Transactions on Machine Learning Research.*

[226] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. of the 31st International Conference on Machine Learning (ICML'14)*.

[227] Mel Rhodes. 1961. An analysis of creativity. *The Phi Delta Kappan* 42, 7 (1961), 305–310.

[228] Mark O. Riedl. 2014. The Lovelace 2.0 Test of Artificial Creativity and Intelligence. (2014). arXiv:1410.6142

[229] Graeme Ritchie. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17 (2007), 67–99.

[230] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. A hierarchical latent vector model for learning long-term structure in music. In *Proc. of the 35th International Conference on Machine Learning (ICML'18)*.

[231] Melissa Roemmele and Andrew S. Gordon. 2018. Automated assistance for creative writing with an RNN Language Model. In *Proc. of the 23rd International Conference on Intelligent User Interfaces Companion (IUI'18)*.

[232] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*.

[233] Jon Rowe and Derek Partridge. 1993. Creativity: A survey of AI approaches. *Artificial Intelligence Review* 7 (1993), 43–70.

[234] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'23)*.

[235] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems (NeurIPS'22)*.

[236] Sameh Said Metwaly, Wim Van den Noortgate, and Eva Kyndt. 2017. Approaches to measuring creativity: A systematic literature review. *Creativity. Theories – Research - Applications* 4, 2 (2017), 238–275.

[237] Pamela Samuelson. 1985. Allocating ownership rights in computer-generated works. *University of Pittsburgh Law Review* 47 (1985), 1185.

[238] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proc. of the NeurIPS'19 Workshop*.

[239] Othman Sbai, Mohamed Elhoseiny, Antoine Bordes, Yann LeCun, and Camille Couprie. 2019. DesIGN: Design inspiration from generative networks. In *Proc. of the Computer Vision - ECCV'18 Workshops*.

[240] Jürgen Schmidhuber. 2010. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development* 2, 3 (2010), 230–247.

[241] Michael D. Schmidt and Hod Lipson. 2009. Distilling free-form natural laws from experimental data. *Science* 324 (2009), 81–85.

[242] Victor Schmidt, Alexandra Sasha Luccioni, Mélisande Teng, Tianyu Zhang, Alexia Reynaud, Sunand Raghupathi, Gautier Cosne, Adrien Juraver, Vahe Vardanyan, Alex Hernandez-Garcia, and Yoshua Bengio. 2022. ClimateGAN:

Raising climate change awareness by generating images of floods. In *Proc. of the 10th International Conference on Learning Representations (ICLR'22)*.

[243] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*.

[244] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*.

[245] Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2018. Toward diverse text generation with inverse reinforcement learning. In *Proc. of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*.

[246] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. (2019). arXiv:1909.08053

[247] Jaskirat Singh, Cameron Smith, Jose Echevarria, and Liang Zheng. 2022. Intelli-paint: Towards developing more human-intelligible painting agents. In *Proc. of the 17th European Conference on Computer Vision (ECCV'22)*.

[248] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. 2022. StyleGAN-V: A continuous video generator with the price, image quality and perks of StyleGAN2. In *Proc. of the 2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'22)*.

[249] Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zhang, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model. (2022). arXiv:2201.11990

[250] Charlie Snell. 2021. Alien Dreams: An Emerging Art Scene. (2021). Retrieved April 30, 2024 from https://mlberkeley.substack.com/p/clip-art/

[251] Charlie V. Snell, Ilya Kostrikov, Yi Su, Sherry Yang, and Sergey Levine. 2023. Offline RL for natural language generation with implicit language Q learning. In *Proc. of the 11th International Conference on Learning Representations (ICLR'23)*.

[252] Benjamin Sobel. 2020. A taxonomy of training data: Disentangling the mismatched rights, remedies, and rationales for restricting machine learning. *Artificial Intelligence and Intellectual Property* (2020), 36.

[253] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. of the 32nd International Conference on Machine Learning (ICML'15)*.

[254] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. 2023. Consistency models. In *Proc. of the 40th International Conference on Machine Learning (ICML'23)*.

[255] Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS'19)*.

[256] Yang Song and Stefano Ermon. 2020. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS'20)*.

[257] Yang Song, Yascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-based generative modeling through stochastic differential equations. In *Proc. of the 9th International Conference on Learning Representations*.

[258] Michael Steinbach, Levent Ertöz, and Vipin Kumar. 2004. The challenges of clustering high dimensional data. *New Directions in Statistical Physics* 213 (2004), 273–309.

[259] Claire Stevenson, Iris Smal, Matthijs Baas, Raoul Grasman, and Han van der Maas. 2022. Putting GPT-3's creativity to the (alternative uses) test. In *Proc. of the 13th International Conference on Computational Creativity (ICCC'22)*.

[260] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS'20)*.

[261] Bob L. Sturm, João Felipe Santos, Oded Ben-Tal, and Iryna Korshunova. 2016. Music transcription modelling and composition using deep learning. In *Proc. of the 1st Conference on Computer Simulation of Musical Creativity (CSMC'16)*.

[262] Douglas Summers-Stay, Clare R. Voss, and Stephanie M. Lukin. 2023. Brainstorm, then select: A generative language model improves its creativity score. In *Proc. of the AAAI'23 Workshop on Creative AI Across Modalities*.

[263] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS'14)*.

[264] Masahiro Suzuki and Yutaka Matsuo. 2022. A survey of multimodal deep generative models. *Advanced Robotics* 36, 5-6 (2022), 261–278.

[265] Pradyumna Tambwekar, Murtaza Dhuliawala, Lara J. Martin, Animesh Mehta, Brent Harrison, and Mark O. Riedl. 2019. Controllable neural story plot generation via reward shaping. In *Proc. of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*.

[266] Wei Ren Tan, Chee Seng Chan, Hernán E. Aguirre, and Kiyoshi Tanaka. 2017. ArtGAN: Artwork synthesis with conditional categorical GANs. In *Proc. of the 2017 IEEE International Conference on Image Processing (ICIP'17)*.

[267] Yingtao Tian and David Ha. 2022. Modern evolution strategies for creativity: Fitting concrete images and abstract concepts. In *Artificial Intelligence in Music, Sound, Art and Design (EvoMUSART 2022)*.

[268] Hannu Toivonen and Oskar Gross. 2015. Data mining and machine learning in computational creativity. *WIREs Data Mining and Knowledge Discovery* 5, 6 (2015), 265–275.

[269] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. (2023). arXiv:2302.13971

[270] Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. (2023). arXiv:2307.09288

[271] Donald J. Treffinger. 1996. *Creativity, Creative Thinking, and Critical Thinking: In Search of Definitions*. Center for Creative Learning, Sarasota, FL.

[272] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. In *Advances in Neural Information Processing Systems (NeurIPS'21)*.

[273] Lewis Tunstall, Leandro von Werra, and Thomas Wolf. 2022. *Natural Language Processing with Transformers*. O'Reilly, Sebastopol, CA.

[274] Alan M. Turing. 1950. Computing machinery and intelligence. *Mind* LIX, 236 (1950), 433–460.

[275] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. In *Proc. of the 9th ISCA Workshop on Speech Synthesis*.

[276] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. In *Proc. of The 33rd International Conference on Machine Learning (ICML'16)*.

[277] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. 2016. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems (NeurIPS'16)*.

[278] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS'17)*.

[279] Lav R. Varshney. 2019. Mathematical limit theorems for computational creativity. *IBM Journal of Research and Development* 63, 1 (2019), 2:1–2:12.

[280] Lav R. Varshney, Florian Pinel, Kush R. Varshney, Debarun Bhattacharjya, Angela Schoergendorfer, and Y-Min Chee. 2019. A big data approach to computational creativity: The curious case of Chef Watson. *IBM Journal of Research and Development* 63, 1 (2019), 7:1–7:18.

[281] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS'17)*.

[282] Dan Ventura. 2016. Mere generation: Essential barometer or dated concept?. In *Proc. of the 7th International Conference on Computational Creativity (ICCC'16)*.

[283] Gauthier Vernier, Hugo Caselles-Dupré, and Peirre Fautrel. 2020. Electric dreams of ukiyo: A series of Japanese artworks created by an artificial intelligence. *Patterns* 1, 2 (2020), 100026.

[284] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers. (2023). arXiv:2301.02111

[285] Zhengwei Wang, Qi She, and Tomás E. Ward. 2021. Generative adversarial networks in computer vision: A survey and taxonomy. *Comput. Surveys* 54, 2 (2021), 1–38.

[286] Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. 2015. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems (NeurIPS'15)*.

[287] Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, Julia Haas, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2022. Taxonomy of risks posed by language models. In *Proc. of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT'22)*.

[288] Max Welling and Yee Whye Teh. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proc. of the 28th International Conference on International Conference on Machine Learning (ICML'11)*.

[289] Geraint A. Wiggins. 2006. Searching for computational creativity. *New Generation Computing* 24 (2006), 209–222.

[290] Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8 (1992), 229–256.

[291] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. 2022. Wav2CLIP: Learning robust audio representations from CLIP. In *Proc. of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'22)*.

[292] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. 2016. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Advances in Neural Information Processing Systems (NeurIPS'16)*, Vol. 29.

[293] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. 2023. NExT-GPT: Any-to-Any Multimodal LLM. (2023). arXiv:2309.05519

[294] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. In *Proc. of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*.

[295] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. 2021. VideoGPT: Video Generation using VQ-VAE and Transformers. (2021). arXiv:2104.10157

[296] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. 2016. Attribute2Image: Conditional image generation from visual attributes. In *Proc. of the 11th European Conference on Computer Vision (ECCV'16)*.

[297] Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'21)*.

[298] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion models: A comprehensive survey of methods and applications. *Comput. Surveys* 56, 4 (2023), 39.

[299] Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. 2018. Automatic poetry generation with mutual reinforcement learning. In *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*.

[300] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *Proc. of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*.

[301] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2019. Self-attention generative adversarial networks. In *Proc. of the 36th International Conference on Machine Learning (ICML'19)*.

[302] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. (2022). arXiv:2205.01068

[303] Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*.

[304] Yizhe Zhang, Zhe Gan, and Lawrence Carin. 2016. Generating text via adversarial training. In *Proc. of the NeurIPS'16 Workshop on Adversarial Training*.

[305] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *Proc. of the 34th International Conference on Machine Learning (ICML'17)*.

[306] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. 2021. VidTr: Video transformer without convolutions. In *Proc. of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV'21)*.

[307] Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017. Towards Deeper Understanding of Variational Autoencoding Models. (2017). arXiv:1702.08658

[308] Yunpu Zhao, Rui Zhang, Wenyi Li, Di Huang, Jiaming Guo, Shaohui Peng, Yifan Hao, Yuanbo Wen, Xing Hu, Zidong Du, Qi Guo, Ling Li, and Yunji Chen. 2024. Assessing and Understanding Creativity in Large Language Models. (2024). arXiv:2401.12491

[309] Tao Zhou, Chen Fang, Zhaowen Wang, Jimei Yang, Byungmoon Kim, Zhili Chen, Jonathan Brandt, and Demetri Terzopoulos. 2018. Learning to Sketch with Deep Q Networks and Demonstrated Strokes. (2018). arXiv:1810.05977

[310] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. of the 2017 IEEE International Conference on Computer Vision (ICCV'17)*.

[311] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum entropy inverse reinforcement learning. In *Proc. of the 23rd AAAI Conference on Artificial Intelligence (AAAI'08)*.

[312] Andrea Zugarini, Stefano Melacci, and Marco Maggini. 2019. Neural poetry: Learning to generate poems using syllables. In *Proc. of the 29th International Conference on Artificial Neural Networks (ICANN'19)*.