Controllable Data Generation by Deep Learning: A Review

SHIYU WANG, Department of Biostatistics and Bioinformatics, Emory University, Atlanta, USA
YUANQI DU, Department of Computer Science, Cornell University, Ithaca, USA
XIAOJIE GUO, IBM Thomas J. Watson Research Center, Yorktown Heights, USA
BO PAN, Department of Computer Science, Emory University, Atlanta, USA
ZHAOHUI QIN, Department of Biostatistics and Bioinformatics, Emory University, Atlanta, USA
LIANG ZHAO, Department of Computer Science, Emory University, Atlanta, USA

Designing and generating new data under targeted properties has been attracting various critical applications such as molecule design, image editing and speech synthesis. Traditional hand-crafted approaches heavily rely on expertise experience and intensive human efforts, yet still suffer from the insufficiency of scientific knowledge and low throughput to support effective and efficient data generation. Recently, the advancement of deep learning has created the opportunity for expressive methods to learn the underlying representation and properties of data. Such capability provides new ways of determining the mutual relationship between the structural patterns and functional properties of the data and leveraging such relationships to generate structural data, given the desired properties. This article is a systematic review that explains this promising research area, commonly known as controllable deep data generation. First, the article raises the potential challenges and provides preliminaries. Then the article formally defines controllable deep data generation, proposes a taxonomy on various techniques and summarizes the evaluation metrics in this specific domain. After that, the article introduces exciting applications of controllable deep data generation, experimentally analyzes and compares existing works. Finally, this article highlights the promising future directions of controllable deep data generation and identifies five potential challenges.

CCS Concepts: • **Computing methodologies** → **Neural networks**; **Control methods;**

Additional Key Words and Phrases: Data generation, deep learning, deep generative models, property controllable generation

ACM Reference Format:
Shiyu Wang, Yuanqi Du, Xiaojie Guo, Bo Pan, Zhaohui Qin, and Liang Zhao. 2024. Controllable Data Generation by Deep Learning: A Review. *ACM Comput. Surv.* 56, 9, Article 228 (April 2024), 38 pages. https://doi.org/10.1145/3648609

S. Wang and Y. Du contributed equally to this research.

Authors' addresses: S. Wang and Z. Qin, Department of Biostatistics and Bioinformatics, Emory University, Atlanta, USA; e-mails: shiyu.wang@emory.edu, zhaohui.qin@emory.edu; Y. Du, Department of Computer Science, Cornell University, Ithaca, USA; e-mail: ydu6@gmu.edu; X. Guo, IBM Thomas J. Watson Research Center, Yorktown Heights, USA; e-mail: xiaojie.guo@ibm.com; B. Pan, Department of Computer Science, Emory University, Atlanta, USA; e-mail: bo.pan@emory.edu; L. Zhao (Corresponding author), Department of Computer Science, Emory University, Atlanta, USA; e-mail: liang.zhao@emory.edu.

ACM 0360-0300/2024/04-ART228
https://doi.org/10.1145/3648609

1 INTRODUCTION

Data generation is an important field that aims to capture the inherent distribution of data to generate similar yet new data. The field is a long-lasting, fast-growing one with broad applications in critical fields such as molecule design [1–3], image editing [4–6], text generation [7, 8] and speech synthesis [9–11]. Data generation requires exploring and manipulating the complex data structures which historically lead to high cost, intensive manpower, and rich domain knowledge in large (and usually discrete) searching space. Partially because of this, traditional methods for data generation are customized to specific domains so that domain-specific heuristics and engineering can be more easily applied [12–15]. For instance, the process of drug design, which is to generate new molecular structures, typically requires chemists to hand-craft candidate structures and then test for desired properties such as solubility and toxicity. Computational methods such as generic algorithms also may assist with combinatorial search for molecule structures by designing molecular mutation and crossover rules based on domain knowledge [16]. The molecule structure space is huge, however: for instance, the number of realistic drug-like molecules is estimated to be around 10^{33} [17], and they pose considerable difficulties to search and identify the structure of interest. Moreover, in many domains such as neuroscience, circuit design, and protein structure, our domain knowledge is still quite limited and incomplete. The paucity of understanding on the data generative process limits our capability in reproducing and even creating new ones with desired properties. Another example is the logic circuit design which aims to output the desired schematics of the integrated circuit. The traditional circuit design is a rather complex process that relies on a large amount of mathematical modeling of the behavior of circuit elements based on characteristics of charge [13, 18] and selecting appropriate materials for different circuit devices according to their properties [13, 19]. Noticeably, detailed reviews for traditional data generation techniques can be found individually in specific domains [13, 15, 20, 21].

In the recent years, the advancement of deep learning provides new opportunities to tackle the aforementioned challenges in data generation. Deep learning techniques have exhibited significant success in learning the representation of various data types including images, texts, sequences and graphs [22–28]. This success further empowers us to fit the mapping from data structures to their corresponding (latent) features, where the former can usually be discrete and unstructured while the latter are continuous vectors or matrices. Hence, instead of directly exploring the high-dimensional space of complex data structure with expensive combinatorial algorithms, the latent features of the data are explorable in the continuous vector space with efficient algorithms such as gradient-based ones. For instance, protein structure is formed by a sequence of amino acids and therefore the distribution of the sequential data can be captured and encoded by sequential deep learning models such as **recurrent neural networks (RNNs)** and transformers [29]. Then the new sequence of amino acids can be autoregressively generated from the learned latent space of protein structure [30]. Greater sequence diversity demonstrably has been achieved by those deep learning-based methods for protein design than with the traditional frameworks such as Rosetta [31]. Since deep learning extracts the latent features in end-to-end fashion, moreover, the dependency on domain knowledge may be largely reduced. For example, in the field of image synthesis, deep learning-based techniques can learn the latent semantic representations of paintings of specific artists and readily fit the distribution in latent space and hence synthesizing new paintings of the same artists is simply a sampling+decoding process [32]. Also because of this greater independence of domain knowledge, the data generation techniques based on deep learning could have better potential in being more easily generalized or cross-used in different data types or applications.

Despite the promise of black-box deep learning techniques in tackling the traditional hurdles in data generation, how to fill the gaps between the learned latent features and the real-world properties of interest is critical to ensure the alignment between the generated data structures and the desired properties. Generating data with desired properties is de facto prerequisite in typical real-world applications, ranging from medicine design [33, 34], to circuit obfuscation [35], art design [36, 37], and audio synthesis [38, 39]. For instance, chemists may not only generate novel **quaternary ammonium compounds (QACs)**, but also hope that generated QACs have strong solubility in the water and **minimum inhibitory concentration (MIC)** smaller than 4mg/L to ensure antibiocity [40]. The community of image captioning may expect to generate more human-like texts from images with the length less than ten words in a humorous style [41, 42]. As a result, to tackle this core crux in controlling the properties of the data generated by deep learning techniques, fast-growing demand, and research body in controllable deep data generation have been observed in recent years [1, 6, 43–45].

1.1 Challenges in Controllable Deep Data Generation

Despite its importance in making deep learning the de facto techniques for real-world, data-generation tasks, controllable deep data generation is technically nontrivial to achieve because of several major challenges:

- **Large search space with property constraints**. The structure of the data to be generated, such as graph, image and text, can be highly complex, discrete and unstructured. Controllable deep data generation, therefore, needs to search in a large, discrete, and unstructured data space to identify the structure of data that satisfies desired property constraints. For instance, the number of realistic drug-like molecules is large and estimated to be around 10^{33} [17] and poses significant challenges on the efficiency of the generation model to search for the molecular structure that possesses desired properties.
- **Interference of multi-property control due to correlation among properties**. Correlation among properties of data is ubiquitous in many real-world settings. For instance, a scenery image of “nighttime” is more likely to have low “brightness”. In addition, for molecules such as alkanes, increasing molecular size will likely result in a decrease in their water solubility [46]. Nevertheless, these correlated properties are difficult to control, since controlling one will constrain the others into some subspace such as a hyperplane or even a non-convex set, and that factor raises the need to design the corresponding optimization and generation strategies [47, 48].
- **Diversity in property types and heterogeneity in their control objectives**. The standard deep learning framework is typically designed for continuous data and trained via back-propagation. Nevertheless, discrete properties widely exist in real-world settings. For instance, linguists may want to generate a piece of audio speech with a specific emotion (e.g., anger, happiness, fear, sadness, neutral, etc.) while others also may want to synthesize a scenery image in the “morning”. Difficulty to formularize and control different types of properties as multiple objectives, including continuous or discrete property values, property values within a range or maximizing (or minimizing) those properties, calls for various strategies and frameworks for controllable deep data generation.
- **Efficiency issues in representation of complex data**. Deep data generation produces new data from the latent representation learned to cover the distribution of observed data. The structure of observed data can be rather complex, however. For example, the average length of proteins from eukaryote is 449, and each of these residues forms as any of 20 amino acids [49]. Complex molecules are typically made up of chains and rings that contain carbon, hydrogen, oxygen and nitrogen atoms and may have millions of those atoms linked together in specific arrays [50]. As properties of data are usually highly sensitive to small changes of structure [51], efficiently and expressively learning the representation of data is critical to generating new data with desired properties.

- **Difficulty in labeling ground-truth data properties**. To train and evaluate the controllable deep data generation models, comparing the properties preserved by the generated data and the ground-truth properties is important. Although one can extract certain properties of generated molecules relying on well-developed domain knowledge and tools such as chemoinformatics toolkits [52], this way is usually not the case for many domains where properties are difficult to be identified automatically for the data types such as text, audio, images, social networks, and so on. For instance, the model will not know what style a piece of text preserves or what the emotion a specific audio expresses unless it is annotated either by human labor or by pre-trained models.

1.2 Our Contributions

To date, a considerable amount of research has been devoted to controllable deep data generation to address the challenges. Understanding comprehensively the strengths and weaknesses of existing works is important for advancing the state-of-the-art techniques and in order to foresee potential research opportunities. Also, broad interests exist in controlling data generation of various domains. Although most proposed methods have been designed targeting individual application domains, it is beneficial and possible to generalize their techniques to other application domains. Hence, cross-referencing these methods that serve different application domains is tough and needs to be addressed. The quality of results of controllable deep data generation, moreover, requires specifically designed evaluation strategies in various application domains. Consequently, systematic standardization and summarization of different evaluation strategies across various domains is necessary. In addition, given the growing demand from both **Artificial Intelligence (AI)** scientists who are looking for new available datasets to test their models for controllable deep data generation and domain-specific communities who are looking for more powerful controlling techniques to generate complex structured data with desired properties, the absence of a systematic survey of existing techniques for controllable deep data generation limits the advance of data generation for both parties. To fill the gap, this survey will provide a systematic review of controllable deep data generation techniques to help interdisciplinary researchers understand basic principles of controllable deep data generation, choose appropriate techniques to solve problems in their related domains and advance the research frontiers with standardized evaluation scenarios. The major contributions of this survey are as follows.

- **A systematic summarization, categorization, and comparison of existing techniques**. Existing techniques for controllable deep data generation are thoroughly categorized based on how the generation process is triggered to form a novel taxonomy of a generic framework. Technical details, advantages, and disadvantages among different subcategories of the taxonomy are discussed and compared. The taxonomy is proposed to enable researchers from various application domains to locate the technique that best fits their needs.
- **Standardized evaluation metrics and procedures**. Historically, data generation methods and their evaluations usually are customized to individual domains and are not well unified despite their shared abstract problems and goals. To tackle this challenge, the paper summarizes common evaluation metrics and procedures for controllable deep data generation and standardizes them from the perspective of both generated data quality and their property controllability.
- **A comprehensive categorization and summarization of major applications**. Major applications including molecule synthesis and optimization, protein design, image editing and emotional speech generation are comprehensively introduced and summarized. A full comparison and discussion includes various techniques that have been applied to these application domains. The comprehensive categorization and summarization of these major applications will help the AI researchers explore extensive application domains and + guide researchers in those domains to generate data with appropriate techniques.

- **Systematic review of existing benchmark datasets and empirical comparison of existing techniques on them**. Benchmark datasets borrowed in various application domains are systematically summarized according to various data modalities. In addition, experimental results are conducted by peer-reviewed articles and us to compare representative models for controllable deep data generation on those benchmark datasets. The systematic review of the existing benchmark datasets and the empirical comparison of representative techniques will enable model developers to involve additional datasets to evaluate their models and compare the performance of their proposed models with benchmark results.
- **A helpful discussion of current research status and potential future directions**. Based on the comprehensive survey of various techniques for controllable deep data generation, standardized evaluation metrics, a broad spectrum of applications, systematic review of benchmark datasets and empirical comparisons of existing techniques, this paper concludes by offering profound insights into several open problems and outlining promising future directions in this domain.

1.3 Related Works

Previously published surveys have some relevancy to controllable deep data generation. These surveys can be classified into four topics: (1) deep learning, (2) deep representation learning, (3) deep data generation, and (4) controllable data generation in specific domains.

Deep learning is a general topic that covers a broad range of neural network-based techniques and includes two branches: (1) deep representation learning to embed complex data into expressive and informative low-dimensional representation and (2) deep data generation to recover the data from the low-dimensional representation. To date, the perspectives of deep learning have been thoroughly surveyed, including its history [25, 53–56], objectives [25, 57], models [25, 53–60] and so on [59, 61].

Deep representation learning has a huge impact on the downstream tasks, such as prediction [62, 63], classification [64, 65] and self-supervision [66, 67]. Deep representation learning has been surveyed from different perspectives, including history [68], motivation [69], learning strategies [68, 69], models [68, 69], training [68, 69], challenges [69] and future directions [68]. Other surveys instead summarize representation learning techniques intended for various types of domains. For instance, Zhang et al. [70] proposed new taxonomies to categorize and summarize the information network representation learning techniques, while Li and Pi [71] reviewed a large number of network representation learning algorithms from two clear points of view of homogeneous network and heterogeneous network. Chen et al. [72] reviewed a wide range of graph embedding techniques. Ridgeway [73] instead surveyed various constraints that encourage a learning algorithm to discover factorial representations that identify underlying independent causal factors of variation in data. Zhang et al. [74] continued to expand the topic by exploring the advancement of concept factorization-based representation learning ranging from shallow to deep or multilayer cases.

For deep data generation, Oussidi and Elhassouny [75] gave an overview of the building blocks of deep generative models, such as **variational autoencoders (VAEs)** and **generative adversarial networks (GANs)**, and pointed out issues that arise when trying to design and train deep generative architectures using shallow ones. Regenwetter et al. [76] further introduced key works that have introduced new techniques and methods, and identified key challenges and limitations currently seen in deep generative models across design fields, such as design creativity and handling constraints and objectives. Furthermore, Eigenschink et al. [77] proposed a data-driven evaluation framework for generative models for synthetic data. Harshvardhan et al. [78] instead covered deep generation as part of generation in machine learning and proposed future directions. In addition to surveys on general deep data generation, other surveys may focus on the deep data generation in specific domains including graph generation [79–81], image synthesis [82, 83], text generation [84, 85] and audio generation [86–88].

Surveys are available for controllable data generation techniques customized for specific domains. For instance, Huang et al. [20] surveyed computational methodologies for *de novo* protein design on the full sequence space guided by physical principles that underlie protein folding. Pederson [89] reviewed circuit analysis and its applications to computer-aided circuit design. Guihaire and Hao [90] reviewed crucial strategic and tactical steps of transit network design. In addition to shallow models, deep models designed for controllable data generation were also reviewed in various domains. For instance, Toshevska and Gievska [91] claimed to have systemically reviewed text style transfer methodologies by deep learning, which covers controllable style generation on text data. Jin et al. [92] is another survey that defined text style transfer aiming to control certain attributes in the generated text, such as politeness, emotion, humor, and many others. Another field related to controllable deep data generation is drug design, which usually controls properties of generated drug molecules [93]. Kim et al. [94] discussed the conditional control of generated compounds and the corresponding models for property control. Kell et al. [95] instead only focused on the common-used model for drug design, VAEs, and provided examples of recent success of VAEs in terms of drug design. A recent effort by Gao et al. [96] benchmarks molecule optimization task empowered by machine learning. In contrast, our survey reviews standardization, comparison, unification, categorization, and taxonomy of controllable deep data generation techniques spanning applications.

1.4 Outline of the Survey

We start the survey by first introducing the background of controllable deep data generation, challenges, our contributions, and the relationship between our survey and existing surveys in Section 1. We next formally define the problem of controllable deep data generation (Section 2.1) and propose the taxonomy according to various property control techniques (Section 2.2) in Section 2, followed by the summarization of evaluation metrics in the same section. In Section 3, we introduce techniques that have been developed for controllable deep data generation as well as detailed explanation of the concepts and representative works of controllable generation from scratch (Section 3.1) and controllable transformation from source data (Section 3.2) according to our taxonomy. In Section 4, we showcase applications of models reviewed in Section 3.1 and Section 3.2 for various domain-specific tasks, followed by popular datasets employed in these domains. Later in Section 5, we introduce the potential challenges, opportunities of the domain and limitations of the existing methods. We will conclude our survey in Section 6. In Appendix Section A, moreover, we introduce common frameworks for deep data generation as preliminaries. We also performed experimental comparison as well as analysis on popular molecular, image, text and audio datasets with common models for controllable deep data generation in Appendix Section B.

2 PROBLEM FORMULATION, TAXONOMY AND PERFORMANCE EVALUATIONS

This section firstly formulates the problem of controllable deep data generation in Section 2.1. Next we propose a taxonomy to summarize various techniques in this domain in Section 2.2. We then unify standard metrics to evaluate the performance of these techniques for controllable deep data generation in Section 2.3.

2.1 Problem Formulation

We formally define the controllable deep data generation as follows. Controllable deep data generation aims to generate data with desired properties based on latent features extracted by deep learning techniques. Define $x \in \mathcal{X}$ as a data object such as an image, a graph, a text document, and the like. Here $\mathcal{X}$ is the domain of the data. Also define c as a set of N properties of interest of x such that $c = \{c_1, c_2, \ldots, c_N\}$, where each c_i $(i = 1, \ldots, N)$ is the i-th property value of x. Moreover, we have $c^* = \{c_1^*, c_2^*, \ldots, c_N^*\}$ to hold the set of **desired** values (or ranges) of N properties. In some situations, c_i^* can be a binary value or real value. For example, when generating images of scenery, one may want to generate a nighttime (instead of daytime) image such that c_i^* = "nighttime". In another example, one may want to generate a molecule with weight of 100 (amu) so that $c_i = 100$. In some other situations, the desired property value c_i^* can be a range or a set, for example, one may want to generate a molecule with ClogP[1] between 2 and 4. c_i^* can have other forms depending on the applications. For example, when we do not have a specific desired value of the property but just want to maximize (or minimize) it, one could set it as (negative) infinite. Denote $z \in \mathbb{R}^{1\times K}$ as the latent representation of each data object x learned by deep learning models. The distribution $p(z)$ characterizes the semantic distribution of data objects in latent feature space.

Controllable deep data generation aims to learn a data generator $g()$ that can generate data based on the latent semantic features $z \sim p(z)$ with its property values c satisfying the user-desired c^*: $x \leftarrow g(z, x_0) \quad s.t.,\ c \in c^*$, where x_0 is an optional term when any "source data" is available for new data object x to derive and transfer from. For example, one can transfer a photo (x_0) into a painting of Van Gogh's style (x), or can edit a given molecule (x_0) toward a new one (x) with better properties. Such a scenario is called data transformation, which has wide, real-world applications and can be considered as an extension of data generation.

2.2 Taxonomy

To have an overall view of this field and allow comparison of various techniques in a qualitative way, we propose taxonomy to categorize the existing works of controllable deep data generation techniques. Specifically, according to how the generation process is triggered, the proposed taxonomy has two main classes of works: 1) *controllable generation from scratch* and 2) *transformation from source data*. Specifically, controllable generation from scratch aims to generate data where specific attributes or features of the generated data can be controlled or guided by the user. Transformation from source data, by contrast, aims to transform from the data in the source domain to the data in the target domain while preserving target properties [98]. Each class is further divided into subcategories based on how the property-control signal is introduced to the model. The detailed taxonomy including the subcategories of each aspect is presented in Figure 1. The comprehensive introduction of controllable generation from scratch and transformation from source data will be provided in Section 3.1 and Section 3.2, respectively. In addition to the proposed taxonomy, we summarize the peer-reviewed publications following the aforementioned taxonomy in Table 1.

2.3 Performance Evaluations

The evaluation of the performance of controllable deep data generation consists of two parts: (1) *data-quality evaluation* evaluates the general quality of generated data and (2) *property-controllability evaluation* evaluates how well the properties are controlled.

Data-quality evaluation of controllable deep data generation is adapted from the evaluation metrics for general deep generative models. The data-quality evaluation consists of two strategies: (1) self-quality-based metrics and (2) distribution-based metrics. Self-quality-based metrics are conducted to evaluate the overall quality of generated data including novelty, validity and uniqueness. Novelty measures how different the generated data is from the training data. For

[1] Here ClogP refers to the calculated value of partition coefficient (logP). logP of a material defines the ratio of its solubility in two immiscible solvents [97].

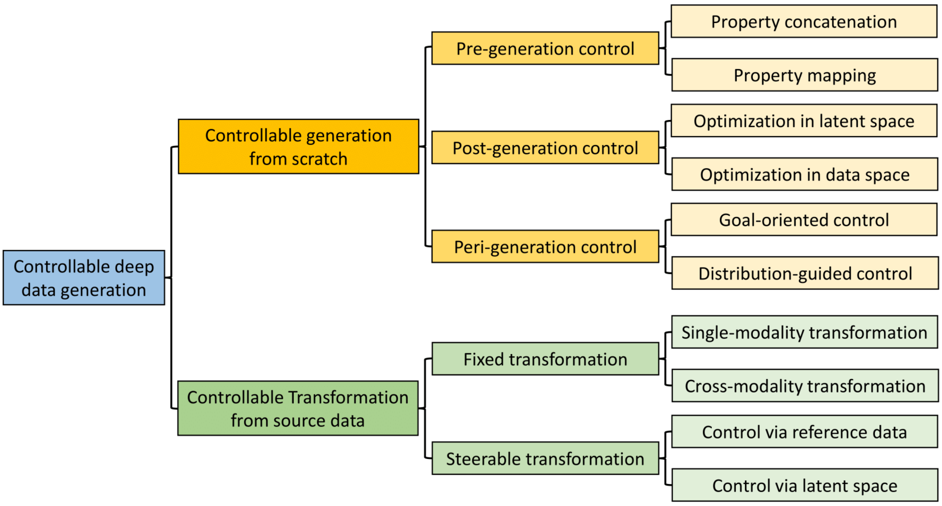

Controllable deep data generation
Controllable generation from scratch
Pre-generation control
Property concatenation
Property mapping
Post-generation control
Optimization in latent space
Optimization in data space
Peri-generation control
Goal-oriented control
Distribution-guided control
Controllable Transformation from source data
Fixed transformation
Single-modality transformation
Cross-modality transformation
Steerable transformation
Control via reference data
Control via latent space

Fig. 1. The proposed taxonomy of controllable deep data generation.

Table 1. Publications for each Category of the Taxonomy

Taxonomy			Representative publications
Controllable generation from scratch	Pre-generation control	Property concatenation	[5, 38, 45, 99–118]
		Property mapping	[2, 6, 43, 119–129]
	Post-generation control	Latent space optimization	[130–165]
		Data space optimization	[33, 130, 166–169]
	Peri-generation control	Goal-oriented control	[1, 3, 7, 170–178]
		Distribution-guided control	[10, 179, 180]
Controllable transformation from source data	Fixed transformation	Within-domain transformation	[168, 181–188]
		Cross-domain transformation	[182, 183, 185, 189, 190]
	Steerable transformation	Control via reference data	[39, 190–211]
		Control via latent space	[11, 212–222, 222–226]

instance, in the task of graph generation, novelty measures the percentage of generated graphs that are not sub-graphs of the training graphs. Validity, as its name suggests, measures whether generated data preserves specific characteristics. For example, validity for molecule generation tasks measures the percentage of generated molecules that are chemically valid based on domain-specific rules [1]. Since ideally the generated data should be diverse but not identical, uniqueness measures the diversity of generated data calculated by the percentage of unique data in all generated data. Distribution-based metrics measure the distance between the distribution of generated data and the distribution of real data. Lower value of these metrics corresponds to similar distribution preserved by the generated and real data and indicates better generation results. A few metrics are designed to measure the distance between distributions, such as **Kullback–Leibler divergence (KLD)** [146], **Maximum Mean Discrepancy (MMD)** [227, 228] and **Fréchet Inception Distance (FID)** [229]. Since data-quality evaluation can be applied to any generation tasks and has been concluded in a few surveys [78, 79, 230], we will not regard it as the focus of our survey, and details can be referred to these surveys.

Property-controllability evaluation that assesses how well the properties are controlled is a critical but challenging task since generated data usually does not explicitly tell us if those properties are preserved. For instance, we can surely tell if an image contains a specific pattern by our

Table 2. Metrics for Property-controllability Evaluation

Type			Metrics	Formula
Automated evaluation	Continuous properties	Distance-based metrics	MSE	$\frac{1}{M}\|\|c-c^*\|\|_2^2$
			MAE	$\frac{1}{M}\|\|c-c^*\|\|_1$
			Cosine similarity	$\frac{c \cdot c^*}{\|\|c\|\| \cdot \|\|c^*\|\|}$
			L2 norm	$\|\|c-c^*\|\|_2$
		Likelihood-based	CLL	$\log p(c\|x)$
			AMP	$AMP_i = \frac{1}{M}\sum_{m=1}^{M} p_m(x_i), i = 1, \ldots, N$
		Hypothesis testing-based	p-value	–
	Discrete properties	Classification-based	Accuracy	$\frac{TP+TN}{M}$
			Precision	$\frac{TP}{TP+FP}$
			Recall	$\frac{TP}{TP+FN}$
			F1 score	$2 \times \frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$
		Entropy-based	BCE	$-\frac{1}{M}\sum_{m=1}^{M}\{c_i \log p(c_i) + (1-c_i)\log(1-p(c_i))\}$
Manual validation	Score-based		MOS	$\frac{1}{S}\sum_{i=1}^{S} RATE_i$
			Accuracy	$\frac{TP+TN}{M}$
			Precision	$\frac{TP}{TP+FP}$
			Recall	$\frac{TP}{TP+FN}$
			F1 score	$2 \times \frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$
	Rank-based		Average human rank	$\frac{1}{S}\sum_{i=1}^{S} RANK_{ij}$

eyes, but we cannot tell directly the toxicity of generated molecules in this way. In general, metrics for property-controllability evaluation are designed to measure the distance between properties of generated data and target properties, which comprise two strategies: (1) automated evaluation-based metrics and (2) human annotation-based metrics. The typical evaluation metrics in evaluating controllable deep data generation are summarized in Table 2.

Automated evaluation-based metrics are distance measures formularized to compute the similarity of properties of the generated data and the target properties. Two steps are contained: property calculation and property validation, which can be broken down to the validation of continuous and discrete values. Specifically, the property values in this way are first extracted from the generated data either by directly calculation or via pre-trained models to be compared with the target properties [5, 45, 101, 104, 111, 112]. Then, for continuous properties, automated evaluation-based metrics include distance measures (e.g., **mean squared error (MSE)**, **mean absolute error (MAE)**, cosine similarity, and L2 norm) [5, 123, 138, 153, 213], likelihood-based metrics (e.g., **conditional log-likelihoods (CLL)** and **Attribute Mean Probability (AMP)**) if properties are assumed to follow a specific probabilistic distribution [6, 119, 133, 137] and hypothesis testing-based metrics (e.g., p-values from A/B testing) to test if properties of generated data and real data are significantly different [10, 103, 107] (Table 2). Note that to calculate AMP, M different classifiers are trained for M different properties in c. Then for each data point x_i in x, the classifiers could output the probabilities $p_m(x_i)$ as presented in Table 2. For discrete properties, automated evaluation-based metrics include classification-based metrics (e.g., accuracy, precision and F1 score) [181, 190, 231] and entropy-based metrics (e.g., **binary cross entropy (BCE)**) [141, 172]. Classification-based metrics are computed based on **true positive (*TP*)**, **true negative (*TN*)**, **false positive (*FP*)** and **false negative (*FN*)** when comparing predicted properties and true properties. Specifically, Accuracy is calculated as the percentage of properties from the generated data that are in the same class as the real properties. F1 score views the generator as a property classifier and measures how accurately the properties of generated data can be classified into the same class of the real properties [232]. BCE compares each of the predicted probabilities for properties of generated data to the real binary properties.

Human annotation-based metrics are obtained by recruiting human evaluators to score manually the properties of generated data [8, 104, 107, 120, 125, 233]. Strategies to obtain a human annotated score can be classified into two types: (1) score-based metrics and (2) rank-based metrics. Score-based metrics directly measure the quality of the properties of generated data. For instance, some works directly score the quality of generated speech such as the **mean opinion score (MOS)**, in which S evaluators are asked to rate the quality and naturalness of the stimulus i (i.e., $RATE_i$) on a Likert-type scale from 1 to 5 with scores labeled as Bad, Poor, Fair, Good and Excellent, respectively [234]. Other works recruit human evaluators to classify generated data according to their properties and calculate the classification-based score such as accuracy, precision, and F1 score compared to real properties [186, 221]. Rank-based metrics compare performance of controllable deep data generation across models. Human evaluators are asked to rank subjectively the data generated by various models according to the quality of properties of generated data. Then the average human rank is calculated for each model with $RANK_{ij}$ as the rank of the evaluator i on the model j. The model with the lower average human rank should have better performance.

3 TECHNIQUES FOR CONTROLLABLE DEEP DATA GENERATION

In this section, we introduce the techniques of controllable generation from scratch and the controllable transformation from source data in detail in Section 3.1 and Section 3.2, respectively.

3.1 Controllable Generation from Scratch

According to which phase of the generation process the control is involved, *controllable generation from scratch* can be divided into three scenarios: (1) *pre-generation control* feeds the property-control signal to the model before the data is generated; (2) *post-generation control* iteratively optimizes the properties of the data after the data is generated; (3) *peri-generation control* directly generates data with desired properties by sampling from the data distribution learned during the training process of the model.

These three tactics have their specific advantages and limitations. Pre-generation control simply manipulates the value of properties as the input of the generator before the generation process and is relatively easy to implement. Besides, the generator takes the property-related information along and disentangles it with the data representation as the input, preserving a high level of interpretability in the model. Pre-generation control cannot guarantee that the target properties are preserved by the generated data, however, since it does not evaluate and leverage the properties of generated data in the generation process. By contrast, post-generation control evaluates the properties after the data is generated and leverages the properties to guide the generation process for the next iteration, until the desired objective of properties is reached. Hence, the post-generation control usually guarantees that generated data preserves desired properties. Nevertheless, post-generation control requires that the properties are evaluated after the data is generated and serve as the control signal to improve the generation process. This usually leads to a more complex generation architecture compared to pre-generation control-based techniques. Peri-generation control directly generates data via the learned distribution, which owns a relatively simple architecture of generation. Peri-generation control usually requires a relatively large dataset to train the model, however, it entails that the model should learn the distribution of the data with desired properties during the training process.

3.1.1 Pre-generation Control. Pre-generation control, as shown in Figure 1 and Figure 2 (1), aims to control properties of generated data conditioning on target properties. In general, two different ways are available for incorporating the control signal of desired properties as the input:

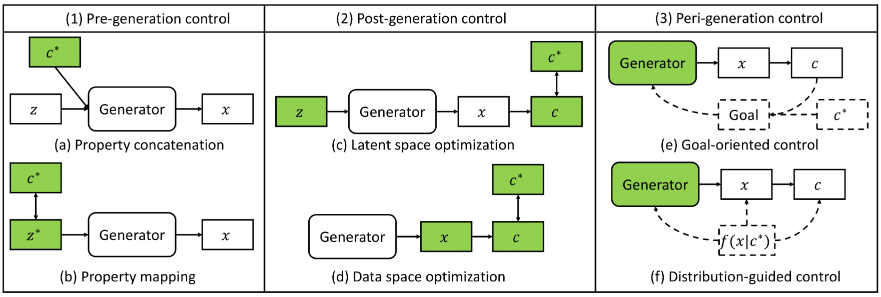
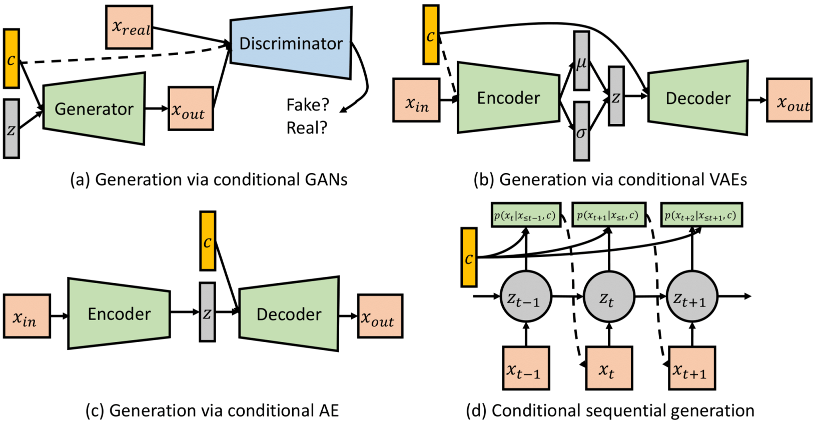

(1) Pre-generation control
(2) Post-generation control
(3) Peri-generation control
c^*
z
Generator
x
(a) Property concatenation
c^*
z^*
Generator
x
(b) Property mapping
c^*
z
Generator
x
c
(c) Latent space optimization
c^*
Generator
x
c
(d) Data space optimization
Generator
x
c
Goal
c^*
(e) Goal-oriented control
Generator
x
c
$f(x|c^*)$
(f) Distribution-guided control

Fig. 2. Architecture of control via controllable generation from scratch. Green modules are where the control signal is entailed.



(a) Generation via conditional GANs

(b) Generation via conditional VAEs

(c) Generation via conditional AE

(d) Conditional sequential generation

Fig. 3. Property concatenation-based models for controllable generation from scratch.

(a) *property concatenation* feeds the raw properties to the generator while (b) *property mapping* first encodes the properties into their embedding and feeds the embedding to the generator.

Compared with property concatenation techniques, the structure of property mapping-based models is usually more complex, since the embedding of the target properties is obtained by either encoding the target properties into the latent space via an encoder, or optimizing the latent space, based on the learned mapping between the latent space and properties. Nevertheless, different types of input properties (i.e., continuous or categorical properties) may cause difficulties in the design of the generator of property concatenation-based techniques, which is instead not a challenge for property mapping-based techniques since they only take continuous property embedding as the input.

Property concatenation conditions the generation process on the original target properties simply by feeding property codes along with the data representation into the generator, as shown in Figure 2(a) and Figure 3. As a result, given the target properties c^*, the generation process can be formulated as $x \sim p(x|c = c^*)$. In this process, the properties are independent of the latent representation of the data.

In recent years, a huge amount of effort has been devoted to controllable generation from scratch with the strategy of property concatenation. Techniques to achieve property concatenation have been developed under popular existing deep generation frameworks, such as adversarial training, variational inference, autoencoder and sequential model. For instance, the framework of adversarial training has been borrowed by works that control data generation by conditioning on the auxiliary information [5]. As presented in Figure 3(a), several works feed handcrafted properties into the generator of GANs to realize property control on generated data [45, 112, 114], which have the objective as follows:

$$\mathcal{L} = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z|c)))], \tag{1}$$

where D and G are discriminator and generator, respectively. Based on Equation (1), D and G are trained together, in which G is trained conditionally on desired properties to produce realistic data by having it compete against D in a game-theoretic framework. In addition to feeding properties into the generator, some works also input the properties into the discriminator [45, 112, 114]. In this case the $D(x)$ of the Equation (1) can be rewritten as $D(x|c)$ to condition the discrimination process on the properties.

In addition to training on the adversarial loss, another way to train the objective is to maximize the joint likelihood of data conditional on the target properties based on the variational inference. The variational inference is proposed to approximate the intractable posterior density of the latent variables for example, approximate the posterior $p(z|x, c)$ when training conditional VAEs that are broadly used for controllable generation from scratch-based frameworks. For implementation, several works feed the properties into the decoder of VAEs along with the data representation (Figure 3(b)) [38, 99, 107]. The objective can be formularized as follows:

$$\mathcal{L} = -\mathbb{E}_{z\sim p(z|x)}[\log p_\theta(x|z, c)] + KL[q_\phi(z|x)||p(z)], \tag{2}$$

where $q_\phi(z|x)$ is the approximated posterior distribution of $p(z|x)$ and ϕ is the parameter of the encoder. $p_\theta(x|z, c)$ is the decoder that generates data from latents parameterized by θ, where properties c are concatenated with the latent representation of data z. Under the same framework, other works feed properties both into the encoder with data and into the decoder with the data representation [107, 116]. For example, given target properties, RationaleRL encodes rational of properties and data into latent space by the encoder. The decoder is initialized with both rational and latent data representation to ensure that generated molecules contain the corresponding rational. In this case, $p(z|x)$ in the objective above becomes $p(z|x, c)$ and the approximated posterior is changed to $q_\phi(z|x, c)$.

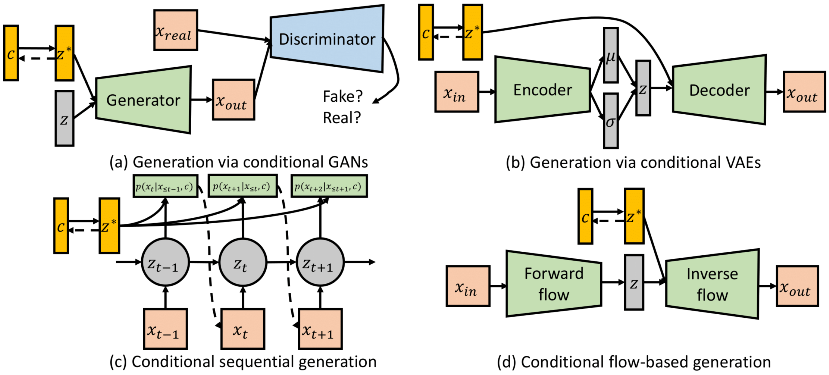
Besides techniques under the framework of adversarial training and variational inference, autoencoder serves as another framework for controllable generation from scratch (Figure 3(c)). Similar to the variational inference-based techniques mentioned above, several works directly feed the target properties into the decoder [100] of autoencoder to generate data with desired properties. The objective can be formulated as the reconstruction loss:

$$\mathcal{L} = -\mathbb{E}_{z\sim p(z|x)}[\log p_\theta(x|z, c)], \tag{3}$$

where $p_\theta(x|z, c)$ is the generator that is learned by optimizing Equation (3) and the concatenation of data representation and properties serve as the input. Another branch of property concatenation-based techniques is implemented by sequential generative models, particularly for generating sequential data. Sequential generative models condition the generation of the token at the current stage on both previously generated tokens and the target properties (Figure 3(d)) [102, 110, 113, 115]:

$$x_t \sim p_\theta(x_t|x_{\leq t-1}, c), \tag{4}$$

where x_t is generated at the t-th stage and $x_{\leq t-1}$ is the set of data generated before the t-th stage. The generator is parameterized by θ and properties are concatenated with data generated at the previous step. The condition sequential generation has a broad range of applications. For instance, several deep text generative models view a sentence as a sequence of words and generate the



(a) Generation via conditional GANs

(b) Generation via conditional VAEs

(c) Conditional sequential generation

(d) Conditional flow-based generation

Fig. 4. Property mapping-based models for controllable generation from scratch.

whole sentence word-by-word sequentially [102, 113, 115]. To realize the controllable data generation, the word to be generated at the current stage depends on both previously generated words and the target properties. Controllable graph generation can also be conducted in a sequential manner, where the node to be generated at the current stage relies on both previously generated nodes, edges, and the target properties [110]. In addition to the autoregressive model, generative diffusion is a rapidly emerging method for controllable data generation. Most strategies based on property concatenation rely on classifier-free diffusion guidance [235]. This guidance blends score estimates from both a conditional diffusion model and a concurrently trained unconditional diffusion model. Specifically, rather than learning $\epsilon_\theta(x_t, t)$, the classifier-free diffusion guidance learns $\epsilon_\theta(x_t, c, t)$ in which the property is concatenated with latents of the data (i.e., diffused data). Hoogeboom et al. [105] generate molecules conditional on their properties by concatenating the property values with the latents. Tevet et al. [108] generate human motion based on text descriptions similarly following the property-concatenation strategy. Nichol et al. [106] proposed GLIDE, which serves as a text-conditional diffusion model that has a superior performance by means of classifier-free guidance training strategy. Huang et al. [118] generate speech based on the diffusion model conditioning on the Mel-spectrogram at each step.

Property mapping first maps the target properties into latent space, as shown in Figure 2(b) and Figure 4, and then enables property control by feeding the embedding of the target properties into the generator. Suppose the mapping from the target properties to the corresponding latent variables is achieved by the function $f(c)$ and represented by $z^* = f(c^*)$, then given the target properties c^*, generated process can be formulated as $x \sim p(x|z^* = f(c^*))$.

Similar to property concatenation, property mapping techniques designed for controllable generation from scratch are mostly under the framework of adversarial training [123, 125], maximizing joint likelihood via the variational inference [2, 43, 119, 120, 124] and conditional sequential generation [121]. As illustrated in Figure 4(a), unlike the adversarial training-based models for property concatenation that can feed the target properties into both generator and discriminator, the conditional GANs for property mapping only feed the embedding of target properties into the generator since the discriminator of GANs aims to classify fake and real data rather than their embedding [123, 125]. The objective function of conditional GANs can be adapted from Equation (1):

$$\mathcal{L} = \mathbb{E}_{x \sim p(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z, z^*|c)))], \quad (5)$$

where z^* is the embedding of the input properties obtained from $z^* = f(c)$. Models learned based on the variational inference act in a similar way as in the property concatenation setting. As shown in Figure 4(b), the generation is controlled by feeding the embedding of the target properties into the decoder [2, 43, 119, 120, 124]. Specifically, the objective can be formularized as:

$$\mathcal{L} = -\mathbb{E}_{z^* \sim p(z^*|c), z \sim p(z|x)}[\log p_\theta(x|z, z^*)] + KL[q_{\phi_1}(z|x)||p(z)] + KL[q_{\phi_2}(z^*|c)||p(z^*)], \tag{6}$$

where ϕ_1 and ϕ_2 are parameters of posteriors to approximate $p(z)$ and $p(z^*)$, respectively. The decoder is parameterized by θ. The existing works have shown considerable diversity in embedding properties into latent space. For instance, some works pre-train a property encoder to extract and embed the target properties into the latent space. Then the data can be generated by the decoder that takes both the data and property embedding as the input [120, 124]. In some other works, the mutual dependence between the individual property and the specific latent variable can be realized via an invertible function [43]. Then in the generation process, latent variables can be backtracked given target properties via the invertible function and serve as the input of the decoder. In addition, prediction models such as sparse Gaussian process can also be used to predict properties from the data representation [2]. The corresponding latent variables can be optimized based on techniques like Bayesian optimization [236] given target properties and are then fed into the decoder to generate data. Particularly, for sequential data, as presented in Figure 4(c), conditional information of properties are projected onto the input layer along with the data from the previous stage to generate data at the current stage: $x_t \sim p(x_t|x_{\leq t-1}, z^* = f(c))$ [121]. In addition to generating sequential data, Choi et al. [129] employ the diffusion model to generate images while at each step the generation is conditioned on the control signal extracted from the reference image.

In addition to the models described above, as shown in Figure 4(d), conditional flow-based models relying on the invertible constraint serve as another tool for property mapping-based controllable generation from scratch by learning the mapping between data distribution $p(x)$ and prior distribution $p(z)$ using a bijective function $z = F(x)$ [6, 122, 126, 127]. Then the data is generated by the invertible mapping from the latent variables. One type of conditional flow-based generative model encodes properties into the latent space z^* via the encoder, and encodes the data into z via a forward flow [6]. The distance of the distribution of the property embedding and data embedding is minimized. Then the data can be generated from the embedding of the target properties. To train this conditional flow-based generative model, we maximize the likelihood:

$$\log p_\theta(x, c) = \log p_\theta(z, c) + \log \left| det \frac{\partial F(x)}{\partial x} \right|. \tag{7}$$

Another type of conditional flow-based generative model feeds properties as the parameter of forward function, which is usually a neural network [122, 126, 127]. During the generation process, the target properties are fed into the inverse function to generate corresponding data. In the training process, we modify $p_\theta(x, c)$ to $p_\theta(x|c)$ and $F(x)$ to $F(x; c)$ in objective Equation (7) while $z = F(x; c)$.

3.1.2 Post-generation Control. Post-generation control, as suggested in Figure 2 (2), achieves controllable generation from scratch by first generating data and then optimizing the properties of generated data towards the target property c^*. According to whether latent space is entailed during the process of controllable generation from scratch, two common tactics have been employed: (a) *latent space optimization* takes advantage of the continuous latent vectors learned by models to avoid the combinatorial search of data structures. Specifically, it manipulates the learned latent vectors by optimization methods or heuristics; by contrast, (b) *data space optimization* does not embed data or properties into latent space, but rather directly generates data with target properties by searching over the original data space which is often high-dimensional and sometimes discrete.

The key difference between latent space optimization and data space optimization is whether a latent space is entailed during the process of controllable generation from scratch. The benefit of latent space optimization is that the latent space learned by models is usually continuous and low-dimensional which makes the optimization phase an easier problem. In addition, regularization or structure prior can be easily imposed while learning the latent space, which is particularly true when the input data is highly discrete, high-dimensional and indifferentiable such as graphs, texts, and so on. By contrast, data space optimization usually optimizes the property-related objective based on the domain knowledge The objective function is usually easier to optimize than the optimization on the latent space which is even non-convex.

Latent space optimization often requires a latent space learned by models which have an encoder $q_\phi(z|x)$ that maps data from data space X to the learned latent space Z, and a decoder that maps the latent vector back to the data space. The encoder and decoder usually are parameterized by neural networks depending on the data type (e.g., GNN for graph data, CNN for image data, etc.). As shown in Figure 2(c), Latent space optimization refers to optimizing the latent vectors z by a heuristic function $f(z)$ in the latent space to obtain optimized latent vectors z' that can be decoded to x' with desired properties, as follows:

$$z = q_\phi(z|x); \quad z' = z + f(z); \quad x' = p_\theta(x'|z'), \tag{8}$$

where $f(z)$ can be in different formats and learned in different settings.

The first way to learn $f(z)$ is to find meaningful latent directions aligned with the change of properties, then controllable generation from scratch can be easily achieved by moving latent vectors along the latent directions. Some works [136, 137, 143] simply interpolate linear direction between the data with and without desired properties as the directions to control the properties while others [141, 157, 159] also discover linear directions that control the properties but in an end-to-end manner guided by a discriminator that predicts the properties. Härkönen et al. [160] leverage PCA to decompose the latent space learned by the model to factorize controllable latent directions. Shen and Zhou [144] discover latent directions that control semantically meaningful properties of data by decomposing generator parameters without the need to access semantic labels as in previous methods. Parthasarathy et al. [164] leverage a GANs-based model and linearly interpolates in a metrics space which correlates with the properties instead of directly in the latent space.

The second way to find $f(z)$ is to leverage regularization over the latent space and manipulate the latent vectors to achieve controllable generation from scratch by following the bias introduced to the model. Johnson et al. [135] propose a new framework that learns a structured latent-variable model for controllable latent variables in a structured representation. Ma et al. [140] aim to generate semantically valid graphs and formulate semantic constraints as regularization terms to generative models. Guo et al. [146, 154], Yang et al. [161], Li et al. [150] and Zhou et al. [153] introduce inductive biases over the structure of the studied data and propose a structured latent-variable generative model that learns controllable latent variables in an unsupervised learning manner. Pati and Lerch [149], Du et al. [152] and Xu et al. [162] leverage a disentangled VAEs model that regularizes the latent variables for controlling properties of data.

The third way to find $f(z)$ is the optimization-based method, in which a heuristic function is learned to estimate properties c of a given z. With the learned heuristic functions, optimization-based methods can be applied to search z' with desired properties. For instance, Jin et al. [2] and Zang and Wang [122] optimize the latent vectors learned by the generative models for generating data with desired properties.

Data space optimization directly optimizes over the data space to obtain data x' with desired properties instead of learning a continuous latent space, as illustrated in Figure 2(d), unlike latent space optimization. Even though data space optimization suffers from the high-dimensional and usually discrete searching problem, it does not require the mapping to continuous space which sometimes causes loss of information: $x' = f(x)$.

Kang et al. [130, 169] and Nigam et al. [166] design a genetic algorithm with a neural network (often as a property evaluator) to search data with desired properties. Xie et al. [33] devise an MCMC-based approach for controllable generation from scratch that samples data maximizing the distribution of desired properties. Fu et al. [168] formulate data in a differentiable format which allows direct optimization over the data space. Song et al. [237] borrow a score-based generative diffusion model to generate data conditioning on the target properties. The intuition is that at the last step of the reverse diffusion process, the data is sampled from $p_0(x_0|c^*)$. This only requires to compute $p_t(c^*|x_t)$ at all steps that can be modeled as the property predictor using neural networks. Additionally, classifier-guidance generative diffusion is another approach that guides the model to generate data with desired properties by training a classifier and supervising the model with its gradient [238]. For instance, Kim et al. [239] proposed Guided-TTS that achieved text-to-speech generation while controlling the target speaker. For classifier guidance, Guided-TTS utilizes an unconditional diffusion model trained on untranscribed speech, combined with a phoneme classifier from a large-scale speech recognition dataset. Kawar et al. [240] incorporated the time-dependent classifier's gradient into the diffusion process to encourage the generated image to be recognized as the target class. Note that in general classifier-guidance generative diffusion is not efficient as classifier-free diffusion guidance as it needs to compute gradient at each time step.

3.1.3 Peri-generation Control. Peri-generation control, as shown in Figure 2 (3), learns the distribution of data and its relation to desired properties c^* by supervised learning during the training process. In general, the distribution of data can be learned towards target properties in two ways: (a) *goal-oriented control* initializes the generator with either random noise or an initialized sample, and optimizes the goal calculated by the target properties of generated data in an iterative manner; (b) *distribution-guided control* trains the generator learned from the data with the desired properties. As a result, the distribution of data with desired properties is preserved by the generator during the training process. Since distribution-guided control generates data from the distribution by the observations, it usually cannot generate out-of-sample data as goal-oriented control does.

Since the control signal is provided in the training process, moreover, distribution-guided control usually requires a more complicated training strategy such as transfer learning [241].

Goal-oriented control optimizes the goal regarding target properties and data, as shown in Figure 2(e), which is either designed based on domain knowledge [131] or estimated from observations [242]. Define $R(x, c)$ as the reward function, where x is the data and c is corresponding properties, then the goal-oriented control can be formularized as:

$$x = \arg\max_x R(x, c); \quad s.t. \quad c = c^*, \tag{9}$$

where c^* is the target properties supposed to be preserved by the generated data. In implementation, $R(x, c)$ can be in different forms depending on the application to various domains and different generation techniques. For instance, Liu et al. [131] devise an energy-based model that regards the reward function as the negative energy of molecules that incorporates the properties as part of the energy terms. Dathathri et al. [170] take the likelihood of the desired attribute as the reward function and introduce a plug-and-play method on a pre-trained language model that trains an additional property classifier to guide the generation process. Yang et al. [176] deploy a distributed MCTS to the molecule generation problem that searches for molecules with desired properties where the reward score is provided by a chemistry simulator that evaluates the properties of the generated molecules. Samanta et al. [177] manage to learn a property-optimized decoder for the

VAEs-based model with a gradient-based algorithm to generate molecules with desired/optimal properties. In addition, the reward function in Equation (9) can also serve as the reward optimized by reinforcement learning-based techniques. For example, You et al. [1] utilize a reinforcement learning approach to formulate the molecule generation problem which optimizes domain-specific rewards via policy gradient. Hu et al. [172] expand the posterior regularization approach [243] that imposes knowledge constraints on posterior distributions of probabilistic models with the inverse reinforcement learning idea to allow the model to learn the constraints jointly with optimizing the objective. De Cao and Kipf [3] formulate a GANs-based model under a reinforcement learning framework that utilizes the discriminator to guide the process to generate molecules with desired properties. Guo et al. [7] also formulate a GANs model under the reinforcement learning framework but additionally incorporate the high-level features learned by the discriminator into the generator to allow the control signal to be smooth over the long text rather than a single scalar value. Sanchez-Lengeling et al. [244], Putin et al. [173] and De Cao and Kipf [3] formulate the molecule generation problem under the reinforcement learning framework with a GANs model which utilizes the reward or discriminator network to provide feedback to generate molecules with desired properties. Tambwekar et al. [174] formulate text generation under the reinforcement learning framework and introduce dense/intermediate reward rather than sparse reward at the end. Shi et al. [175] focus on the problem of molecule generation with a reinforcement learning framework that provides rewards for desired properties.

Distribution-guided control lies in the fact that data generation models learn the distribution of the data $p(x)$ and bias the input data distribution towards the target distribution $p(x|c^*)$ of desired properties (i.e., $f(x|c^*)$ in Figure 2(f)). Thus, the model can generate data under the target distribution $p(x|c)$ with desired properties. For instance, Gebauer et al. [179] employ an auto-regressive model for 3D molecule generation and biases the distributions of the target properties for controllable generation from scratch. To be specific, this model pre-selects 3D molecules with the target properties in desired ranges and fine-tunes the generative model with the subset of the whole training set to generate molecules with the target properties similar to the selected ones. Segler et al. [180] first pre-train an auto-regressive generative model with a large dataset and then fine-tune the model with a small subset of molecules known to be active against given biological targets. Yang et al. [10] focus on the problem to generate high-quality audios embedding secret bits stream. Specifically, it proposes an auto-regressive model to generate audio based on the encoded bits stream as conditions.

3.2 Controllable Transformation from Source Data

Depending on whether the desired values of the properties are prefixed or steerable by the users, as presented in Figure 5, controllable transformation from source data can be classified into two categories: (1) *fixed transformation* only allows the source data to be transformed into the data with predefined value (or value change) of properties that are prefixed during model training. For instance, one may train an image translator for daytime-to-nighttime transformation, so that it can only transfer daytime scenery into its nighttime view. By contrast, (2) *steerable transformation* enables users to input desired values (or value changes) of properties for guiding data transformation. For example, using steerable scenery-image transformation models, the model may take time (e.g., 1:25 pm) as the input from the user and then generate the scenery at that specified time based on the source scenery image.

Steerable transformation may stay more interpretable, given that the control signal is explicitly fed into the model along with the source data during the generation process. By contrast, the fixed transformation handles the control signal in an implicit way, in that it takes only the source data as the input and instead learns the control signal during the training process. Nevertheless, the more

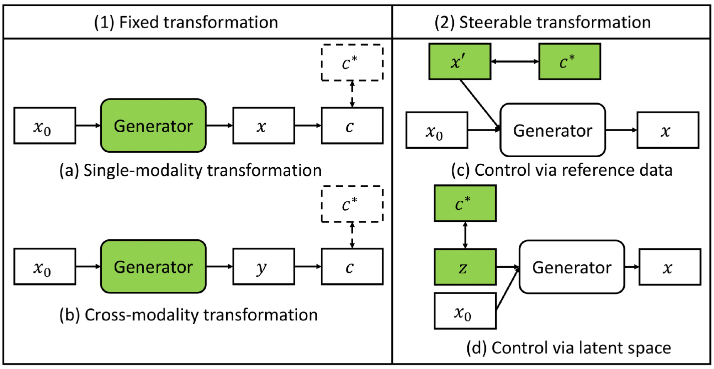

(1) Fixed transformation
(2) Steerable transformation
x_0
Generator
x
c
c^*
(a) Single-modality transformation
y
(b) Cross-modality transformation
x'
(c) Control via reference data
z
(d) Control via latent space

Fig. 5. Architecture of control via controllable transformation from source data. Green modules are the location of the control signal.

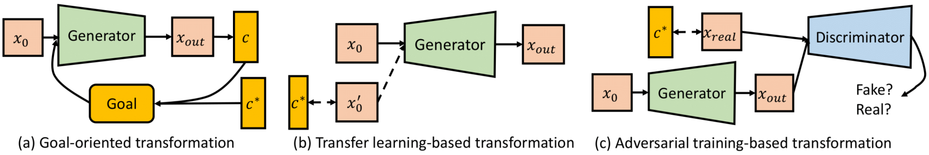

x_0
Generator
x_{out}
c
Goal
c^*
(a) Goal-oriented transformation
x_0'
(b) Transfer learning-based transformation
x_{real}
Discriminator
Fake?
Real?
(c) Adversarial training-based transformation

Fig. 6. Single-modality transformation for controllable transformation from source data.

flexible controlling ability is normally achieved at the cost of a more complex model structure to enable the properties to interact with the generator.

3.2.1 Fixed Transformation. Approaches for fixed transformation, as shown in Figure 5 (1), can be divided into two categories based on domains of the source data and the target data: (a) *single-modality transformation* conducts property-controllable data transformation within the same modality, such as image-to-image, text-to-text, graph-to-graph, and so on; and (b) *cross-modality transformation* translates the data from source modality to the data in another modality while controlling its properties. For example, one may transform text to audio while controlling the emotion expressed in the generated audio.

Single-modality transformation instead translates the source data to target data in the same modality while changing properties of generated data in a prefixed way, as shown in Figure 5(a) and Figure 6. For example, one may transform an image of human pose into another one with a predefined pose. As shown in Figure 6, single-modality transformation includes three types of techniques: 1) goal-oriented transformation; 2) transfer learning-based transformation and 3) adversarial training-based transformation. As shown in Figure 6(a), goal-oriented transformation usually borrows the framework of reinforcement learning to generate data guided by the reward constructed via target properties [168, 184, 188] with the same objective as Equation (10). For instance, Deep Reaction Optimizer optimizes certain chemical reactions by combining reinforcement learning with the domain knowledge of chemistry [188]. GraphDF was designed to enforce optimized molecules to have desired properties by fine-tuning the model to optimize the score of properties under a reinforcement learning framework [184]. In addition to optimizing the reward with the reinforcement learning-based approach, other goal-oriented methods also have been explored for various tasks. The differentiable scaffolding tree controls the properties of generated data by optimizing the composite objective under a multi-objective optimization framework [168]: $\log \hat{x}^{(t+1)} = \arg\max_{x \in \mathcal{N}(\hat{x}^{(t)})} F(x)$, where $\hat{x}^{(t)}$ is the generated molecule at the t-th iteration. $F(x)$

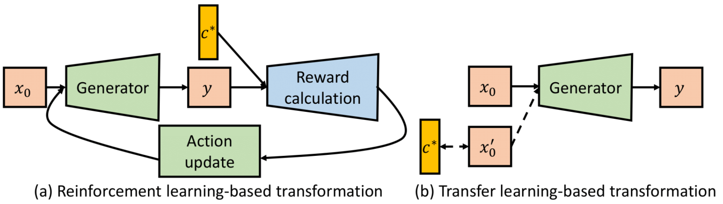

x_0
Generator
c^*
y
Reward calculation
Action update
(a) Reinforcement learning-based transformation
x_0
Generator
y
c^*
x_0'
(b) Transfer learning-based transformation

Fig. 7. Cross-modality transformation for controllable transformation from source data.

is the set of constraints related to target properties. $\mathcal{N}(\hat{x}^{(t)})$ is the neighborhood set of molecules $\hat{x}^{(t)}$ [168]. Instead, Prabhumoye et al. [181] designed a style classifier to guide the style-specific generator to modify the style of the given text.

As presented in Figure 6(b), transfer learning-based transformation first trains the model in a large corpus of the unlabeled dataset, and then fine-tunes the model using the small-sized dataset x_0' with labeled properties c^* [186]. Adversarial training-based transformation, as the name suggests and shown in Figure 6(c), borrows the adversarial loss to enforce the model to learn the distribution of the data with desired properties [182, 183, 185].

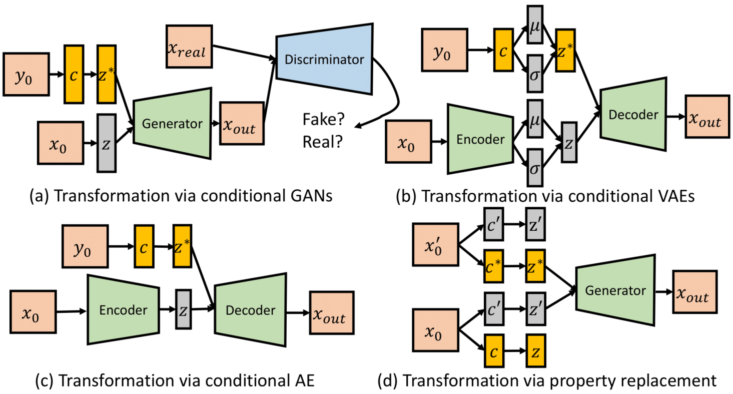
Cross-modality transformation translates the source data x_0 to the target data y while controlling its properties, where x_0 and y are across different modalities. As shown in Figure 5(b) and Figure 7, cross-modality transformation learns the distribution of the data with desired properties and generates from the source data directly without entailing information of properties during the generation process. The techniques used in this area roughly are aligned with those in peri-generation control discussed previously, except that it needs the source data as the input. In general, as shown in Figure 7, two techniques are borrowed to learn the distribution of data with desired properties to achieve cross-domain transformation: (1) reinforcement learning and (2) transfer learning. Reinforcement learning iteratively optimizes the reward calculated based on the target properties c and other properties identified from the generated data y:

$$y = \arg\max_{y} R(y, c, x_0) \quad s.t. \quad c \in c^*, \tag{10}$$

In this process, as shown in Figure 7(a), the identifier that evaluates the generated data and calculates the rewards can be pre-trained [190]. As illustrated by Figure 7(b), transfer learning-based cross-modality transformation first trains the model in a large general dataset x_0. Then the model is fine-tuned by a small-sized dataset x_0' with desired properties c [189]. As an example, Tits et al. [189] leveraged fine-tuning on a pre-trained deep learning-based **Text-to-Speech (TTS)** model to synthesize speech with specific emotions using a small-sized emotional dataset. Song et al. [207] generated the talking face video given an arbitrary speech clip and an arbitrary face image by encoding both image and speech via a recurrent adversarial network. In this work, the recurrent network first will be pre-trained for several epochs and then fine-tuned by adding discriminators to distinguish audio, image and lip reading.

3.2.2 Steerable Transformation. According to the source of the control signal, as shown in Figure 5 (2), steerable transformation techniques can be classified into two scenarios: (a) *control via reference data* usually requires reference data in addition to the source data as the input to provide the control guidance; (b) *control via latent space* controls the target properties via manipulating the latent variables corresponding to the properties of interest.

The first scenario suffers from the challenge of obtaining the control signal as the reference data may come from different data modalities. Nevertheless, the second scenario, control via latent



(a) Transformation via conditional GANs (b) Transformation via conditional VAEs

(c) Transformation via conditional AE (d) Transformation via property replacement

Fig. 8. Control via reference data for controllable transformation from source data.

space, usually requires learning the relationship between the latent space and the properties of generated data, which can be challenging too.

Control via reference data, as shown in Figure 8, requires both reference data and source data with the aim of altering the source data into the target data with some specific properties the same as those of the reference data. For instance, the style or emotion of synthesized speech via text-to-speech (TTS) techniques can be controlled by adapting the style of a reference speech as the input to the generator [190, 191, 196, 202, 205]. One can also transform the pose of a portrait into another pose provided by a reference portrait [111]. Techniques for control via reference data usually have an encoder-decoder structure, in which target properties are extracted and encoded from the reference data while the data is generated conditional on the property representation along with the embedding of source data. These techniques, however, differ in how the target properties are fed into the generator. As shown in Figure 8(a), (b) and (c), some works extract and encode target properties c from the reference data y_0 and feed its representation z^* along with the embedding of source data into the generator. In this way the data is generated by conditioning on c. This strategy involves three basic types of techniques: (1) adversarial learning (i.e., Figure 8(a)); (2) variational inference (i.e., Figure 8(b)) and (3) conditional AE-based transformation (i.e., Figure 8 (c)). The objective of adversarial learning-based models (i.e., Figure 8(a)) can be formularized as:

$$\mathcal{L} = \mathbb{E}_{x \sim p(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z, z^*|y_0)))], \tag{11}$$

where z is the representation of the source data, z^* is the representation of target properties from the reference data y_0. Under the framework of adversarial learning, Chen et al. [194] generate talking face video first by learning audio and landmarks representations. CIAGAN anonymizes the face of the image based on the identity that is encoded from reference data via CNN [195]. In this case, the objective function is aligned with Equation (11). In addition to models based on adversarial learning, the framework based on the variational inference employed in control via reference data (i.e., Figure 8(b)) can be formularized as:

$$\mathcal{L} = -\mathbb{E}_{z \sim p(z|x), z^* \sim p(z^*|y_0)}[\log p_\theta(x|z, z^*)] + KL[q_\phi(z, z^*|x_0, y_0)||p(z, z^*|y_0)], \tag{12}$$

where the notation follows in the same role as in Equation (11). Specifically, Chen et al. [192] control the syntax of generated text by conditioning on the representation of a sentential exemplar under the framework of conditional VAE. Hsu et al. [191] learn continuous attribute space of categorical observed labels, conditional on which the speech is generated from text. The learning objective of both models follows Equation (12).

As illustrated in Figure 8(c), the third type of control via reference data-based technique is conditional AE, which mainly serves the controllable text-to-speech (TTS) transformation [196, 200–202, 204, 205, 245]. In this case the encoder and decoder can be an autoregressive model that handles sequential data such as text, audio and speech. In general, the framework of conditional AE aims to minimize the objective as follows: $\mathcal{L} = |x_0 - G(z, z^*|y_0)|$, where $G(z, z^*|y_0)$ corresponds to the decoder of the AE given the input of representations z and z^*. The L_1 norm implemented above can be replaced with other forms of reconstruction loss such as MSE depending on different tasks. Under the framework of conditional AE, Valle et al. [200] generate speech in a variety of styles conditioning on rhythm and continuous pitch contours from an audio signal or music score. Kurihara et al. [201] proposed a model that controls prosodic features using phonetic and prosodic symbols as input for TTS transformation. Inoue et al. [204] pre-trained emotional expression models using speech uttered by a particular person, and applied it to another person to generate emotional speech with the person's voice quality. StyleTagging-TTS extracts style embedding from reference speech via a reference encoder [202] while Cai et al. [205] average the style tokens' weights for all reference audios to synthesize speech for this kind of audio. Li et al. [196] instead designed emotion classifier connected to the reference encoder to learn discriminative emotion embedding. UFC-BERT consists of modules of textual control, visual control, and preservation control to handle class labels and natural language descriptions, style transfer and preserving given image blocks, respectively [245].

By contrast, as shown in Figure 8(d), other works approach controllable transformation from source data in an interpretable manner by decomposing both source data and reference data into properties of interest, c and c^*, and other non-relevant properties c'. Then the representation of c, z, is replaced with z^*, the representation of c^*. The generator takes z^* and the representation z' of non-relevant properties c' as the input to generate target data [193]. Therefore, the objective of property replacement-based techniques can be concluded as follows:

$$\begin{aligned}\mathcal{L} = &-\mathbb{E}_{z, z'_{x_0} \sim p(z, z'_{x_0}|x_0)}[\log p_\theta(x_0|z, z'_{x_0})] - \mathbb{E}_{z^*, z'_{x'_0} \sim p(z^*, z'_{x'_0}|x'_0)}[\log p_\theta(x'_0|z^*, z'_{x'_0})] \\ &- \mathbb{E}_{z'_{x_0} \sim p(z'_{x_0}|x_0), z^* \sim p(z^*|x'_0)}[\log p_\theta(x|z'_{x_0}, z^*)],\end{aligned} \tag{13}$$

where x_0 is the source data and x'_0 is the reference data. x_0 and x'_0 are usually of the same type. z'_{x_0} and $z'_{x'_0}$ correspond to the representation that is not relevant to the properties of interest of x_0 and x'_0, respectively. Minimizing the learning objective above learns latents of target properties and other properties of interest by the first two terms, then the data is generated based on the last term conditional on those latents. Yang et al. [199] transfer the style of the text by switching the style code of two texts and generating target text along with the content vector z'. Li et al. [211] transfer attributes of text by removing markers z of the original attribute, then generating a new sentence conditioned on the remaining words and the target attribute z^*.

Control via latent space. This type of method first maps the properties of interest either learned from data or provided externally into the latent space, and then controls the target properties of generated data via manipulating the latent variables corresponding to these target properties. Techniques of control via latent space usually have an encoder-decoder framework, where the information of (external) properties is embedded into the latent space. As shown in Figure 9, according to how the properties serve as the input of the model, control via latent space-based techniques can be classified into two categories: (1) learned property embedding (Figure 9(a)) learns the properties from the source data via the encoder of the model [11, 212, 215–217, 219, 220, 222, 225, 226] and (2) external property embedding (Figure 9(b)) encodes the properties provided externally into the latent space from which new data is generated [213, 214, 218, 221].

For learned property embedding-based techniques, the properties of interest are embedded into the latent space and usually disentangled from the representation of other information of the data:

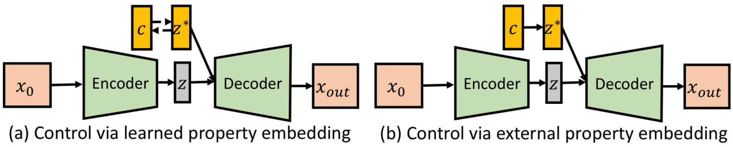

c
z^*
x_0
Encoder
z
Decoder
x_{out}
(a) Control via learned property embedding
c
z^*
x_0
Encoder
z
Decoder
x_{out}
(b) Control via external property embedding

Fig. 9. Control via latent space for controllable transformation from source data.

$\mathcal{L} = |x_0 - G(z, z^*|x_0)|$, where z^* is the embedding of properties extracted from the source data x_0. Again, the L_1 norm can be replaced with other forms of distance measures if needed. For instance, FastSpeech controls the voice speed and prosody of generated speech by tuning the parameter of the length regulator as a part of the encoder [222]. Differentiable Digital Signal Processing library integrates audio with classic signal processing elements, in which the fundamental frequency is learned from the source audio and embedded into the latent space [226]. EMOVIE achieves TTS transformation while controlling the emotion via the emotion embedding supervised by emotion labels [225]. Fabbro et al. [212] control TTS systems by decomposing the spectrogram into control variables, such as amplitude envelope, harmonic distribution, and filter coefficients. SCTKG generates essays from given topics while controlling sentiment for each sentence by injecting the sentiment information into the generator [220]. InterFaceGAN edits the face of the source image by interpolating disentangled latent semantics learned by GANs [219]. Habib et al. [11] control the prosody of the generated speech by manipulating the embedding of prosody learned by the TTS model. Disentangled anatomy arithmetic disentangles images into spatial anatomy (tensor) factors and accompanying imaging (vector) representations, and controls synthesized images by linearly combining the representation of corresponding properties [217]. GANs inversion inverts a given image back into the latent space of a pretrained GANs, then the image can be reconstructed from the latent space and the target properties can be controlled by intervening the style code on the latent space [216]. Low-rank subspaces enable more precise control of image generation via GANs by relating the latent space to the image region with the Jacobian matrix and then using low-rank factorization to discover steerable latent subspaces [215].

In addition to extracting the properties from the source data during the training process, external property embedding employs the external control signal, such as different property codes, and embeds the property information into the latent space [213, 214, 218, 221]: $\mathcal{L} = |x_0 - G(z, z^*|c)|$, where z^* is the embedding of the external properties c. To distinguish the external property embedding-based techniques from the control via reference data-based techniques (Figure 8), control via reference data-based techniques first extract these properties from the reference data whereas external property embedding-based methods encode only the properties of interest. Specifically, the prototype-then-edit model generates a sentence by sampling a random example from the training set and then editing it using an edit vector encoding the type of edit to be performed [221]. StyleFlow borrows conditional continuous normalizing flows in the GANs latent space conditioning on attribute features [214]. Attribute-controlled editing on images is achieved by the conditional exploration of entangled latent spaces [214]. Chang et al. [218] can produce a set of candidate topics by predicting the centers of word clusters, of which a user can select a subset to guide the generation of the text. John et al. [213] explicitly build a classifier on the style space that predicts the style label, and combine the style representation with the source text representation to generate text with the desired styles.

4 APPLICATIONS AND BENCHMARK DATASETS

Controllable deep data generation has been gaining increasing attention of the community from various domains, enabling a broad range of applications including applications in molecule synthesis and optimization, protein design, image editing, text style transfer, and emotional speech generation. In addition, more and more datasets are becoming available to the community to train robust and expressive models for controllable deep data generation to advance the development of this domain. This section provides details of different techniques used for each application domain as well as the representative datasets used in this domain. More detailed introduction of benchmark datasets can be found in Appendix Section C.

4.1 Molecular Synthesis and Optimization

Molecules can be represented by several different data modalities, from sequence of characters [246], graphs [3] to point clouds [179], and so on. Designing molecules with desired properties, such as molecular weight, ClogP values, drug-likeness (QED), and the like, naturally aligns with the goal of controllable deep data generation. Molecular design has been existing as a tough problem because of the complex molecular structure, vast chemical space, and its relationship with biological properties. This field can be divided into two types of problems: (1) molecular synthesis, which corresponds to the *de novo* molecular design that requires designed molecule to preserve desired properties; and (2) molecular optimization, which aims to optimize the structure of an existing molecule towards the direction that specific properties are improved. The conventional methods to approach the tasks usually rely on the expert-derived heuristics [247] and suffer from the daunting computational complexity [248]. The rapid advances in the development of deep learning models have been powering molecular design to generate molecules with desired biological and chemical profiles.

Molecular synthesis refers to the *de novo* generation of novel chemical structures that satisfy a set of property constraints [249, 250]. Deep learning-based approaches have been used by researchers to develop novel and expressive representation of molecular structures, coupled with the ability to reveal relationships with properties [251, 252]. Then the molecular structure can be generated from the representation using either one-shot [253] or autoregressive [254] strategies. This process often coordinates with techniques for goal-oriented control, such as reinforcement learning [1, 255], or multi-objective optimization to control the properties of generated molecules [34, 116]. Datasets that have been used for molecular optimization include QM9 [3, 131, 140, 152, 177], ZINC250K [1, 2, 110, 131, 140, 152, 173, 175–177] and ChEMBL [110, 116, 180, 255].

Molecular optimization can be viewed as a transformation problem that maps from the source molecule to the target molecule that has high structural similarity with the original one but better properties [98]. This task has been surmounted by extracting the signature of the target molecule and encoding its discrete chemical structure into the continuous latent space, which allows optimization in the molecular space. The control of properties of generated molecules requires single-modality fixed transformation with various strategies such as goal-oriented transformation, including reinforcement learning [256, 257] and multi-objective optimization [258], adversarial training-based transformation against target properties [182], and others [259, 260]. Datasets that have been borrowed for molecular optimization are aligned highly with those for molecular synthesis, including QM9 [122, 128, 184], ZINC250K [2, 122, 128, 168, 182, 184, 185, 256, 257, 259, 260], ChEMBL [33, 178, 256, 257] and MOSES [184].

4.2 Protein Design

Protein performs diverse physiological functions in a cell mediated by its three-dimensional structure composed of the sequence of amino acids [261]. Protein design is a fundamental challenge in biology because of the costs both financially and timewise and has vast applications in domains such as drug discovery. The goal of protein design is to arrange the sequence of amino acids that fold to a specific protein structure. Unlike text or graph, protein design requires two-fold information, sequence design and structure design. While the fact that the protein sequence solely determines the protein structure and the big success of AlphaFold2 in predicting protein structures from protein sequences [262], many works have been developed to design new sequences of protein and utilize AlphaFold2 to produce 3D structures. Sequences can be determined by the probability of 20 amino acids on each residue on the sequences [263] or generated via property mapping-based frameworks such as VAEs and GANs [264]. For some specific proteins, structures could be very diverse so that sequence-structure co-design is needed via property concatenation-based deep generative models conditioning on the full structure of the protein [30]. Datasets regarding the protein structure that are available for protein design include Protein Data Bank [30, 263], UniProt [264], UniRef50 [264], UniParc [264] and Pfam [30].

4.3 Image Editing

By reason of the rapid growth of the internet and digital capture devices, a huge volume of images has become available for public use that make the training of deep models possible. Meanwhile, the advances of deep learning provide powerful tools such as CNN to process image data. The image editing task aims to generate a new image from a source image by editing the contents of the source image under certain guidance while keeping other properties unchanged [265]. The input and output of the model for image editing are the images represented by the pixel matrix with multiple color channels. The popular properties to be controlled under this scenario include but are not limited to facial expression, age and illumination of human portraits, artistic style, angle, shape, and color of the object, and so on. The most popular framework to manage image editing is property concatenation or mapping-based techniques under the adversarial training of GANs. For instance, GAN-Control embeds the control signal in the latent space and achieves the controllability via modifying the latent variables [125]. Song et al. [207] generate the talking face using the input audio as the control signal. CPCGAN transforms the pose of a portrait into another pose provided by a reference portrait [111]. In addition to adversarial training-based approaches, frameworks based on variational inference [99, 119, 157, 158] or flow [6, 126, 137] are also borrowed for image editing. Datasets that have been employed for image editing cover a wide range of types of image, including digit image such as MNIST [6, 99, 119, 127, 134] and The **Street View House Numbers (SVHN)** [99, 151], and human character, such as CelebA [6, 126, 126, 136, 137, 160], **Flickr-Faces-HQ (FFHQ)** [125, 160], Helen [136], **Labeled Faces in the Wild (LFW)** [119] and Sprites [134], shape images such as dSprites [157, 158], 3D shapes [158] and ShapeNet [111], fashion image such as DeepFashion [172], animals such as Caltech-UCSD Birds 200 [119] and general scenes such as ImageNet [127, 137, 151, 159] and LSUN [137, 151].

4.4 Text Style Transfer

Deep learning has enabled the rapid development of designing novel **natural language process (NLP)** systems of controlling over various stylistic attributes of generated texts. Specifically, text style transfer intends to transfer a text with one stylistic attribute (e.g., politeness, emotion, humor, etc.) to another text with another stylistic attribute [92]. Text style transfer can borrow the control signal via reference data-based techniques under the scenario of steerable transformation. For instance, Yang et al. [199] proposed an unsupervised model for text style transfer that uses the language model to evaluate generated sentences. The control signal comes from reference sentences with specific styles. In addition, control via latent space-based models are also used. [213] employed an encoder-decoder structure to disentangle the latent variables of style and contents in language models. The style of generated texts can be controlled by feeding latent variables of target styles into the generator. CTRL instead is a conditional transformer language model trained to condition on control codes that govern style [113]. Text style transfer has been conducted on a few datasets including Yelp [162, 186, 199, 211, 213, 221], Amazon [113, 162, 186, 211, 213], Wikipedia [113], Rotten-Tomatoes [102], Project Gutenberg [113], OpenWebText [113], CAPTIONS [186, 211], EMNLP2017 WMT News [113, 181] and One Billion Word [221].

4.5 Emotional Speech Generation

Text to speech (TTS), also known as speech generation, aims to synthesize natural speech from text [266]. Further, the naturalness of speech generation relies in its human-like, natural-sounding voice with desired emotional expression generated by the model [190]. Recent advances of emotional speech generation are overwhelmingly contributed by the development of deep learning techniques, especially those designed for controllable transformation from source data, which can learn expressive feature representations to characterize the emotional attributes of the speech [267]. Control via reference data-based models is studied broadly to approach this task. For example, Liu et al. [190] proposed i-ETTS which borrows the reference audio to extract the emotion embedding. Then the speech is generated under the framework of reinforcement learning to ensure that it preserves the desired emotions. GST-Tacotron2 also borrows the reference audio sequence to provide control signal and feed the embedding of both emotion and text to the WaveNet vocoder to predict the mel-spectrogram used to synthesize a speech waveform [208]. Bian et al. [203] designed a model that inputs multiple reference audios that independently disentangle and control specific styles. By contrast, control via latent space-based models are also employed in this application domain. Inoue et al. [204] proposed a deep neural network-based model to control the emotion expressiveness of synthesized speech by directly feeding the emotional embedding to an emotion additive model. Datasets that have been utilized for emotional speech generation cover a broad range of languages including English such as LJSpeech [189, 209], LibriSpeech corpus [209], LibriTTS [209], CMU-ARCTIC [209], The **Interactive Emotional Dyadic Motion Capture (IEMOCAP)** [205, 209], Catherine Byers [11], The RECOLA Database [205] and Emotional Speech Database [190, 225], Chinese such as **Baidu Speech Translation Corpus (BSTC)** [203] and Chinese professional actress [196], Japanese such as Japanese emotional speech database [107] and French such as SIWIS French Speech Synthesis Database [209].

Together with the applications introduced above, more applications of controllable deep data generation along with representative works are summarized in Appendix Section D.

5 OPPORTUNITIES AND CHALLENGES

All in all, controllable deep data generation is a fast-growing domain as it has a wide range of applications across various domains with numerous models constantly developed. In this section, we highlight a few promising future opportunities in this domain, followed by potential challenges that remain to be studied further.

5.1 Opportunities

We highlight some promising future opportunities on controllable deep data generation as follows.

- **Control multiple properties simultaneously**. As far as we know, most of the existing works designed for controllable deep data generation only control one property at a time. Though technically challenging, simultaneously controlling multiple properties are commonly desired in real-world applications. For instance, chemists may want to generate QACs that have a high MIC value while preserving a small molecular weight. The task of simultaneously controlling multiple properties can be formalized naturally under a multi-objective optimization framework to enforce the generated data to satisfy multiple property constraints. A few works have been done in this direction. Jin et al. [116] perform multi-objective optimization on the latent space using the gradient from the property predictors in their testing process. Wang et al. [178] approach the multi-objective optimization task for controllable molecule generation by combining conditional transformer and reinforcement learning through knowledge distillation. Some more challenging tasks, however, such as constraining the property within a range during multi-objective optimization, still are under-explored.

- **Improve model interpretability**. The interpretability of the model for controllable deep data generation is critical for the community to understand data. For instance, chemists may want not only to generate QACs with the desired value of MIC but also to know which substructure of generated QACs contributes to the corresponding MIC value. Jin et al. [116] achieved the interpretability by learning the property rationals (i.e., substructure of the molecule) first and generating molecules with the specific property via assembling its rationals. In addition to interpreting properties of generated data, how to synthesize these data also attracts the attention of the community. For example, chemists may not be satisfied simply by generating molecules that are supposed to preserve desired properties but also want to know how to synthesize these molecules in the lab. Although some studies have been done regarding chemical reaction prediction [268–270], few works are combined with the controllable deep data generation to infer the concrete synthesis process of chemicals.
- **Develop semi-supervised or unsupervised learning-based techniques**. Existing datasets available for controllable deep data generation usually are trained by supervised learning. These properties, however, may not satisfy all needs of the community to generate data with properties of interest. For example, although one might generate QACs with specific MIC values, they cannot control other properties not labelled in the dataset. The property annotations, moreover, usually require significant time and labour costs, which are expensive for the community to obtain. As a result, semi-supervised learning-based techniques serve as a powerful tool to utilize sparsely annotated data. Unsupervised learning-based methods can also be designed to capture automatically properties of interest from unlabeled data via some well-designed regularization [139, 142, 149].
- **Entangle with hard constraint on data validity**. Data in some domains require strong validity constraint on their structure or properties. For instance, in molecular graph, the number of chemical bonds around an atom cannot exceed the numerical valency of the atom. Also, the combination of words in an sentence should follow specific grammar. Techniques for controllable deep data generation may learn from traditional methods to involve constraints of the domain knowledge when generating data with desired properties. Furthermore, these domain-specific constraints may be helpful in guiding the generation process.
- **Generate out-of-sample data**. Current deep learning techniques for controllable deep data generation are highly data-driven, basically learn the distribution of the observed data and tend to generate similar data. The novelty of generated data, however, is highly desired but limited to the distribution of the observed data and hence resistant to exploring unseen subspace that may also have the potential of the desired value of properties. As a result, the extrapolation ability to generate out-of-sample data is critical when designing the model for controllable deep data generation. This task has been approached by some existing works via interpolating the latent space that controls target properties towards a pre-defined direction [151, 157, 159]. These methods, however, need to identify the pathway on the latent space where the properties are controlled and thus require extra effort and is difficult to obtain if the pathway is non-linear.

5.2 Open Challenges

Despite numerous methods proposed for efficient and effective generation design, several open challenges still remain on controllable deep data generation.

- **Lack of theoretical ground**. Existing frameworks for controllable deep data generation, e.g., conditional GANs and conditional VAEs, learn and sample from the distribution of the observational data while neglecting domain-specific theories that potentially can improve the model performance. By contrast, those domain-specific theories have been borrowed widely by traditional models to guide the optimization process of the model [12]. For instance, conventional models for protein design borrow the energy functions such as those for nucleic acids and their interactions with protein to examine large combinatorial collections of protein structure candidates [12]. In this way, generated proteins theoretically can be guaranteed to have low energy. As a result, how to contain and model those domain-specific theories into deep learning-based frameworks to improve the performance of controllable deep data generation remains to be explored further.
- **Lack of interpretability**. Many works have performed well in optimizing the well-designed objective, while ignoring the interpretation of the design process. Specifically, some tasks conduct optimization on molecular structure only by searching structures that may cause specific molecular properties while putting less effort on the understanding of active groups that correspond to these properties. In addition, although generated molecules might own valid structures, chemists might be more interested in how to synthesize them in the lab. The synthesis pathways of these optimized molecules, however, still remain unknown or extremely hard to synthesize and lead to unrealistic applications.
- **Difficulty in capturing and optimizing correlated properties**. Capturing and considering the correlation among properties during controllable deep data generation can deepen our understanding of data. To date, most of works only consider controlling independent properties of generated data, even though correlated properties are ubiquitous for data in the real world. For instance, given two correlated properties such as molecular weights and solubility, chemists may still want to increase the solubility of the generated molecules while keeping the molecular weights unchanged. Optimizing correlated properties is challenging, since this process may lead to the conflicting property constraints of the multi-objective optimization framework and thus call for developing more advanced optimization techniques from the community.
- **Paucity of labelled dataset**. The property annotations of the datasets used to train the model for controllable deep data generation usually require enormous financial, time and human costs. For instance, the property of MIC of QACs that have been optimized to generate novel QACs needs to be measured manually in the lab [40]. The expensive annotation, moreover, may lead to other problems such as the small sample size of the available dataset (e.g., only 462 QACs are processed and contained in the QAC dataset), which is hard for the supervised training of techniques for controllable deep data generation such as those for goal-orientation transformation. As a result, a semi-supervised or unsupervised approach for controllable deep data generation is needed to learn and control the properties of small data.
- **Difficulty in automatic validation of properties of generated data**. Automatic validation of properties of generated data is critical yet challenging for some domains. For instance, the task of music generation often relies on manual validation, which requires a human evaluator to score and rank the generated data based on some pre-determined rules. Validation of the bio-activities of generated molecules instead requires conducting lab experiments. These downstream evaluations hence suffer from the high cost of time and money, as well as the noise introduced by the subjective evaluation or the batch effect of lab experiments.

6 CONCLUSION

This survey has provided a comprehensive review of existing techniques developed for controllable deep data generation. The first consideration is potential challenges and preliminaries. Then, the problem of controllable deep data generation receives a formal definition, followed by a novel taxonomy of techniques designed for controllable data generation. The evaluation metrics in this specific domain, moreover, are summarized in terms both of data quality and property controllability. After that, a systemic review and comparison of techniques involved in the taxonomy are conducted, followed by their exciting applications and benchmark datasets borrowed in the domain of controllable deep data generation. Finally, the survey concludes with the highlight of promising future directions and five potential challenges in this domain. We believe this survey will pave the way for the future study of controllable deep data generation in many disciplines.

REFERENCES

[1] Jiaxuan You, Bowen Liu, et al. 2018. Graph convolutional policy network for goal-directed molecular graph generation. *Conference on Neural Information Processing Systems* 31 (2018).

[2] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*. PMLR, 2323–2332.

[3] Nicola De Cao and Thomas Kipf. 2018. MolGAN: An implicit generative model for small molecular graphs. *International Conference on Machine Learning 2018 Workshop on Theoretical Foundations and Applications of Deep Generative Models* (2018).

[4] Karol Gregor, Ivo Danihelka, et al. 2015. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*. PMLR, 1462–1471.

[5] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[6] Rui Liu, Yu Liu, et al. 2019. Conditional adversarial generative flow for controllable image synthesis. In *Conference on Computer Vision and Pattern Recognition*. 7992–8001.

[7] Jiaxian Guo, Sidi Lu, et al. 2018. Long text generation via adversarial training with leaked information. In *AAAI Conference on Artificial Intelligence*, Vol. 32.

[8] Pradyumna Tambwekar, Murtaza Dhuliawala, et al. 2019. Controllable neural story plot generation via reward shaping. In *International Joint Conferences on Artificial Intelligence*.

[9] Marc Schröder. 2001. Emotional speech synthesis: A review. In *European Conference on Speech Communication and Technology*. Citeseer.

[10] Zhongliang Yang, Xingjian Du, et al. 2018. AAG-Stega: Automatic audio generation-based steganography. *arXiv preprint arXiv:1809.03463* (2018).

[11] Raza Habib, Soroosh Mariooryad, et al. 2020. Semi-supervised generative modeling for controllable speech synthesis. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJeqeCEtvH

[12] Shaun M. Lippow and Bruce Tidor. 2007. Progress in computational protein design. *Current Opinion in Biotechnology* 18, 4 (2007), 305–311.

[13] Ioannis Vourkas and Georgios Ch. Sirakoulis. 2016. Emerging memristor-based logic circuit design approaches: A review. *IEEE Circuits and Systems Magazine* 16, 3 (2016), 15–30.

[14] Hai Yang and Michael G. H. Bell. 1998. Models and algorithms for road network design: A review and some new developments. *Transport Reviews* 18, 3 (1998), 257–278.

[15] Simone Fischer-Hübner. 2001. *IT-security and Privacy: Design and Use of Privacy-enhancing Security Mechanisms.* Springer.

[16] Jan H. Jensen. 2019. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science* 10, 12 (2019), 3567–3572.

[17] Pavel G. Polishchuk, Timur I. Madzhidov, and Alexandre Varnek. 2013. Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of Computer-aided Molecular Design* 27, 8 (2013), 675–679.

[18] John P. Uyemura. 1999. *CMOS Logic Circuit Design*. Springer Science & Business Media.

[19] Qiao Chen, Xiaoping Wang, et al. 2016. A logic circuit design for perfecting memristor-based material implication. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36, 2 (2016), 279–284.

[20] Po-Ssu Huang, Scott E. Boyken, and David Baker. 2016. The coming of age of de novo protein design. *Nature* 537, 7620 (2016), 320–327.

[21] Reza Zanjirani Farahani, Elnaz Miandoabchi, et al. 2013. A review of urban transportation network design problems. *European Journal of Operational Research* (2013).

[22] Xuan Kan, Hejie Cui, et al. 2022. FBNETGEN: Task-aware GNN-based fMRI analysis via functional brain network generation. In *Medical Imaging with Deep Learning*. https://openreview.net/forum?id=oWFphg2IKon

[23] Chen Ling, Carl Yang, and Liang Zhao. 2021. Deep generation of heterogeneous networks. In *International Conference on Data Mining*. IEEE, 379–388.

[24] Ajay Shrestha and Ausif Mahmood. 2019. Review of deep learning algorithms and architectures. *IEEE Access* 7 (2019), 53040–53065.

[25] Samira Pouyanfar, Saad Sadiq, et al. 2018. A survey on deep learning: Algorithms, techniques, and applications. *Comput. Surveys* 51, 5 (2018), 1–36.

[26] J. Pamina and Beschi Raja. 2019. Survey on deep learning algorithms. *International Journal of Emerging Technology and Innovative Engineering* 5, 1 (2019).

[27] Shiyu Wang, Xiaojie Guo, Xuanyang Lin, Bo Pan, Yuanqi Du, Yinkai Wang, Yanfang Ye, Ashley Ann Petersen, Austin Leitgeb, Saleh AlKhalifa, Kevin Minbiole, Bill Wuest, Amarda Shehu, and Liang Zhao. 2022. Multi-objective Deep Data Generation with Correlated Property Control. (2022). arXiv:cs.LG/2210.01796

[28] Zheng Zhang and Liang Zhao. 2021. Representation learning on spatial networks. *Advances in Neural Information Processing Systems* 34 (2021), 2303–2318.

[29] Alex T. Muller, Jan A. Hiss, and Gisbert Schneider. 2018. Recurrent neural network model for constructive peptide design. *Journal of Chemical Information and Modeling* 58, 2 (2018), 472–479.

[30] John Ingraham, Vikas Garg, et al. 2019. Generative models for graph-based protein design. *Conference on Neural Information Processing Systems* 32 (2019).

[31] Namrata Anand, Raphael Eguchi, et al. 2022. Protein sequence design with a learned potential. *Nature Communications* 13, 1 (2022), 1–11.

[32] MD Zakir Hossain, Ferdous Sohel, et al. 2019. A comprehensive survey of deep learning for image captioning. *Comput. Surveys* 51, 6 (2019), 1–36.

[33] Yutong Xie, Chence Shi, et al. 2021. MARS: Markov molecular sampling for multi-objective drug discovery. In *International Conference on Learning Representations*. https://openreview.net/forum?id=kHSu4ebxFXY

[34] George Lambrinidis and Anna Tsantili-Kakoulidou. 2021. Multi-objective optimization methods in novel drug design. *Expert Opinion on Drug Discovery* 16, 6 (2021), 647–658.

[35] William L. Croft, Jörg-Rüdiger Sack, and Wei Shi. 2022. Differentially private facial obfuscation via generative adversarial networks. *Future Generation Computer Systems* 129 (2022), 358–379.

[36] Mao Li, Jiancheng Lv, et al. 2020. An abstract painting generation method based on deep generative model. *Neural Processing Letters* 52, 2 (2020), 949–960.

[37] Adeel Mufti, Biagio Antonelli, and Julius Monello. 2020. Conditional GANs for painting generation. In *International Conference on Machine Vision*. SPIE.

[38] Hao Hao Tan, Yin-Jyun Luo, and Dorien Herremans. 2020. Generative modelling for controllable audio synthesis of piano performance. In *International Conference on Machine Learning Workshop on Machine Learning for Music Discovery Workshop (ML4MD), Extended Abstract*.

[39] Noé Tits, Fengna Wang, et al. 2019. Visualization and interpretation of latent spaces for controlling expressive speech synthesis through audio analysis. In *Proc. Interspeech 2019*. 4475–4479.

[40] Sylvie Buffet-Bataillon, Bernard Branger, et al. 2011. Effect of higher minimum inhibitory concentrations of quaternary ammonium compounds in clinical E. coli isolates on antibiotic susceptibilities and clinical outcomes. *Journal of Hospital Infection* 79, 2 (2011), 141–146.

[41] Chaorui Deng, Ning Ding, et al. 2020. Length-controllable image captioning. In *European Conference on Computer Vision*. Springer, 712–729.

[42] Longteng Guo, Jing Liu, et al. 2019. MSCap: Multi-style image captioning with unpaired stylized text. In *Conference on Computer Vision and Pattern Recognition*. 4204–4213.

[43] Xiaojie Guo, Yuanqi Du, and Liang Zhao. 2021. Property controllable variational autoencoder via invertible mutual dependence. In *International Conference on Learning Representations*. https://openreview.net/forum?id=tYxG_OMs9WE

[44] Li-Wei Chen and Alexander Rudnicky. 2022. Fine-grained style control in Transformer-based text-to-speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 7907–7911.

[45] Zhiting Hu, Zichao Yang, et al. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*. PMLR, 1587–1596.

[46] Johannes Tolls, John van Dijk, et al. 2002. Aqueous solubility- molecular size relationships: A mechanistic case study using C10-to C19-alkanes. *The Journal of Physical Chemistry A* 106, 11 (2002), 2760–2765.

[47] Zhiting Hu and Li Erran Li. 2021. A causal lens for controllable text generation. In *Conference on Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). https://openreview.net/forum?id=kAm9By0R5ME

[48] Yuanqi Du, Xiaojie Guo, et al. 2022. Interpretable molecular graph generation via monotonic constraints. In *SIAM International Conference on Data Mining*. SIAM, 73–81.

[49] Jianzhi Zhang. 2000. Protein-length distributions for the three domains of life. *Trends in Genetics* 16, 3 (2000), 107–109.

[50] Djuke Veldhuis. 2011. Tree-like giant is largest molecule ever made. *New Scientist* 209, 2795 (2011), 17. DOI:http://dx.doi.org/10.1016/S0262-4079(11)60083-3

[51] Peter Kirkpatrick and Clare Ellis. 2004. Chemical space. *Nature* 432, 7019 (2004), 823–824.

[52] Greg Landrum et al. 2013. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum* (2013).

[53] Shi Dong, Ping Wang, and Khushnood Abbas. 2021. A survey on deep learning and its applications. *Computer Science Review* 40 (2021), 100379.

[54] Md. Zahangir Alom, Tarek M. Taha, et al. 2019. A state-of-the-art survey on deep learning theory and architectures. *Electronics* 8, 3 (2019), 292.

[55] Li Deng. 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing* 3 (2014).

[56] William Grant Hatcher and Wei Yu. 2018. A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access* 6 (2018), 24411–24432.

[57] Weibo Liu, Zidong Wang, et al. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234 (2017), 11–26.

[58] Shaveta Dargan, Munish Kumar, et al. 2020. A survey of deep learning and its applications: A new paradigm to machine learning. *Archives of Computational Methods in Engineering* 27, 4 (2020), 1071–1092.

[59] Maithra Raghu and Eric Schmidt. 2020. A survey of deep learning for scientific discovery. *arXiv preprint arXiv:2003.11755* (2020).

[60] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85–117.

[61] Qingchen Zhang, Laurence T. Yang, et al. 2018. A survey on deep learning for big data. *Information Fusion* 42 (2018), 146–157.

[62] Judith Butepage, Michael J. Black, et al. 2017. Deep representation learning for human motion prediction and classification. In *Conference on Computer Vision and Pattern Recognition.* 6158–6166.

[63] Zhibin Lv, Feifei Cui, et al. 2021. Anticancer peptides prediction with deep representation learning features. *Briefings in Bioinformatics* 22, 5 (2021), bbab008.

[64] Haiman Tian, Yudong Tao, et al. 2019. Multimodal deep representation learning for video classification. *International World Wide Web Conference* 22, 3 (2019), 1325–1341.

[65] Yang Guo, Zhenyu Wu, and Yang Ji. 2017. A hybrid deep representation learning model for time series classification and prediction. In *International Conference on Big Data Computing and Communications.* IEEE, 226–231.

[66] Yuning You, Tianlong Chen, et al. 2020. When does self-supervision help graph convolutional networks?. In *International Conference on Machine Learning.* PMLR.

[67] Linus Ericsson, Henry Gouk, et al. 2022. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine* 39, 3 (2022), 42–62.

[68] Guoqiang Zhong, Li-Na Wang, et al. 2016. An overview on data representation learning: From traditional feature learning to recent deep learning. *The Journal of Finance and Data Science* 2, 4 (2016), 265–278.

[69] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2013).

[70] Daokun Zhang, Jie Yin, et al. 2018. Network representation learning: A survey. *IEEE Transactions on Big Data* 6, 1 (2018), 3–28.

[71] Bentian Li and Dechang Pi. 2020. Network representation learning: A systematic literature review. *Neural Computing and Applications* 32, 21 (2020), 16647–16679.

[72] Fenxiao Chen, Yun-Cheng Wang, et al. 2020. Graph representation learning: A survey. *APSIPA Transactions on Signal and Information Processing* 9 (2020).

[73] Karl Ridgeway. 2016. A survey of inductive biases for factorial representation-learning. *CoRR* abs/1612.05299 (2016). arXiv:1612.05299 http://arxiv.org/abs/1612.05299

[74] Zhao Zhang, Yan Zhang, et al. 2021. A survey on concept factorization: From shallow to deep representation learning. *Information Processing & Management* 58, 3 (2021), 102534.

[75] Achraf Oussidi and Azeddine Elhassouny. 2018. Deep generative models: Survey. In *International Conference on Intelligent Systems and Computer Vision.* IEEE, 1–8.

[76] Lyle Regenwetter, Amin Heyrani Nobari, and Faez Ahmed. 2022. Deep generative models in engineering design: A review. *Journal of Mechanical Design* 144, 7 (2022), 071704.

[77] Peter Eigenschink, Stefan Vamosi, et al. 2021. Deep generative models for synthetic data. *Comput. Surveys* (2021).

[78] G. M. Harshvardhan, Mahendra Kumar Gourisaria, et al. 2020. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review* 38 (2020), 100285.

[79] Xiaojie Guo and Liang Zhao. 2020. A systematic survey on deep generative models for graph generation. *arXiv preprint arXiv:2007.06686* (2020).

[80] Yanqiao Zhu, Yuanqi Du, et al. 2022. A survey on deep graph generation: Methods and applications. *arXiv preprint arXiv:2203.06714* (2022).

[81] Faezeh Faez, Yassaman Ommi, et al. 2021. Deep graph generators: A survey. *IEEE Access* 9 (2021), 106675–106702.

[82] Zhihe Lu, Zhihang Li, et al. 2017. Recent progress of face image synthesis. In *Asian Conference on Pattern Recognition*. IEEE, 7–12.

[83] Sanbi Luo. 2021. A survey on multimodal deep learning for image synthesis: Applications, methods, datasets, evaluation metrics, and results comparison. In *International Conference on Innovation in Artificial Intelligence*. 108–120.

[84] Xiaodong He and Li Deng. 2017. Deep learning for image-to-text generation: A technical overview. *IEEE Signal Processing Magazine* 34, 6 (2017), 109–116.

[85] Gustavo H. de Rosa and Joao P. Papa. 2021. A survey on text generation using generative adversarial networks. *Pattern Recognition* (2021).

[86] Yin-Ping Cho, Fu-Rong Yang, et al. 2021. A survey on recent deep learning-driven singing voice synthesis systems. In *International Conference on Artificial Intelligence and Virtual Reality*. IEEE, 319–323.

[87] Xu Tan, Tao Qin, et al. 2021. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561* (2021).

[88] Zhaoxi Mu, Xinyu Yang, and Yizhuo Dong. 2021. Review of end-to-end speech synthesis technology based on deep learning. *arXiv preprint arXiv:2104.09995* (2021).

[89] D. Pederson. 1984. A historical review of circuit simulation. *IEEE Transactions on Circuits and Systems* 31, 1 (1984), 103–111.

[90] Valérie Guihaire and Jin-Kao Hao. 2008. Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice* 42, 10 (2008), 1251–1273.

[91] Martina Toshevska and Sonja Gievska. 2021. A review of text style transfer using deep learning. *TAI* (2021), 669–684.

[92] Di Jin, Zhijing Jin, et al. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics* 48, 1 (2022), 155–205.

[93] Yuanqi Du, Tianfan Fu, et al. 2022. MolGenSurvey: A systematic survey in machine learning models for molecule design. *arXiv preprint arXiv:2203.14500* (2022).

[94] Jintae Kim, Sera Park, et al. 2021. Comprehensive survey of recent drug discovery using deep learning. *International Journal of Molecular Sciences* 22, 18 (2021), 9983.

[95] Douglas B. Kell, Soumitra Samanta, and Neil Swainston. 2020. Deep learning and generative methods in cheminformatics and chemical biology: Navigating small molecule space intelligently. *Biochemical Journal* 477, 23 (2020), 4559–4580.

[96] Wenhao Gao, Tianfan Fu, et al. 2022. Sample efficiency matters: A benchmark for practical molecular optimization. *arXiv preprint arXiv:2206.12411* (2022).

[97] K. J. Box and J. E. A. Comer. 2008. Using measured pKa, LogP and solubility to investigate supersaturation and predict BCS class. *Current Drug Metabolism* 9, 9 (2008), 869–878.

[98] Xiaojie Guo, Shiyu Wang, and Liang Zhao. 2022. Graph neural networks: Graph transformation. In *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer, 251–275.

[99] Durk P. Kingma, Shakir Mohamed, et al. 2014. Semi-supervised learning with deep generative models. In *Conference on Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/file/d523773c6b194f37b938d340d5d02232-Paper.pdf

[100] Brian Cheung, Jesse A. Livezey, et al. 2014. Discovering hidden factors of variation in deep networks. *arXiv preprint arXiv:1412.6583* (2014).

[101] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*. PMLR, 2642–2651.

[102] Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*. 94–104.

[103] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. 2017. Real-valued (medical) time series generation with recurrent conditional GANs. *arXiv preprint arXiv:1706.02633* (2017).

[104] Navaneeth Bodla, Gang Hua, and Rama Chellappa. 2018. Semi-supervised FusedGAN for conditional image generation. In *European Conference on Computer Vision*.

[105] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. 2022. Equivariant diffusion for molecule generation in 3D. In *International Conference on Machine Learning*. PMLR, 8867–8887.

[106] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).

[107] Gustav Eje Henter, Jaime Lorenzo-Trueba, et al. 2018. Deep encoder-decoder models for unsupervised learning of controllable speech synthesis. *arXiv preprint arXiv:1807.11470* (2018).

[108] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. 2022. Human motion diffusion model. *arXiv preprint arXiv:2209.14916* (2022).

[109] Daniel Rothchild, Alex Tamkin, Julie Yu, Ujval Misra, and Joseph Gonzalez. 2021. C5T5: Controllable generation of organic molecules with transformers. *arXiv preprint arXiv:2108.10307* (2021).

[110] Yujia Li, Oriol Vinyals, et al. 2018. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324* (2018).

[111] Ximing Yang, Yuan Wu, et al. 2021. CPCGAN: A controllable 3D point cloud generative adversarial network with semantic label generating. In *AAAI Conference on Artificial Intelligence*, Vol. 35. 3154–3162.

[112] Lei Xu, Maria Skoularidou, et al. 2019. Modeling tabular data using conditional GAN. *Conference on Neural Information Processing Systems* 32 (2019).

[113] Nitish Shirish Keskar, Bryan McCann, et al. 2019. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* (2019).

[114] Shrimai Prabhumoye, Alan W. Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *International Conference on Computational Linguistics*. COLING, Barcelona, Spain (Online), 1–14. DOI: http://dx.doi.org/10.18653/v1/2020.coling-main.1

[115] Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020. MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models. *arXiv preprint arXiv:2010.00840* (2020).

[116] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2020. Multi-objective molecule generation using interpretable substructures. In *International Conference on Machine Learning*. PMLR, 4849–4859.

[117] Fania Mokhayeri, Kaveh Kamali, and Eric Granger. 2020. Cross-Domain face synthesis using a controllable GAN. In *Winter Conference on Applications of Computer Vision*. 252–260.

[118] Rongjie Huang, Max W. Y. Lam, et al. 2022. FastDiff: A fast conditional diffusion model for high-quality speech synthesis. In *International Joint Conferences on Artificial Intelligence*.

[119] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Conference on Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf

[120] Huajie Shao, Jun Wang, et al. 2021. Controllable and diverse text generation in e-commerce. In *International World Wide Web Conference*. 2392–2401.

[121] Sean Vasquez and Mike Lewis. 2019. MelNet: A generative model for audio in the frequency domain. *arXiv preprint arXiv:1906.01083* (2019).

[122] Chengxi Zang and Fei Wang. 2020. MoFlow: An invertible flow model for generating molecular graphs. In *Special Interest Group on Knowledge Discovery and Data Mining*. 617–626.

[123] Jaeuk Moon, Seungwon Jung, et al. 2020. Conditional tabular GAN-based two-stage data generation scheme for short-term load forecasting. *IEEE Access* 8 (2020), 205327–205339.

[124] Ke Chen, Cheng-i Wang, et al. 2020. Music SketchNet: Controllable music generation via factorized representations of pitch and rhythm. In *International Society for Music Information Retrieval*.

[125] Alon Shoshan, Nadav Bhonker, et al. 2021. GAN-control: Explicitly controllable GANs. In *International Conference on Computer Vision*. 14083–14093.

[126] You Lu and Bert Huang. 2020. Structured output learning with conditional generative flows. In *AAAI Conference on Artificial Intelligence*, Vol. 34. 5005–5012.

[127] Lynton Ardizzone, Carsten Lüth, et al. 2019. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392* (2019).

[128] Rafael Gómez-Bombarelli, Jennifer N. Wei, et al. 2018. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science* 4, 2 (2018), 268–276.

[129] Jooyoung Choi, Sungwon Kim, et al. 2021. ILVR: Conditioning method for denoising diffusion probabilistic models. *International Conference on Computer Vision* (2021).

[130] Yanfei Kang, Rob J. Hyndman, et al. 2018. *Efficient Generation of Time Series with Diverse and Controllable Characteristics*. Technical Report. Monash University, Department of Econometrics and Business Statistics.

[131] Meng Liu, Keqiang Yan, et al. 2021. GraphEBM: Molecular graph generation with energy-based models. *arXiv preprint arXiv:2102.00546* (2021).

[132] Yu Deng, Jiaolong Yang, et al. 2020. Disentangled and controllable face image generation via 3D imitative-contrastive learning. In *Conference on Computer Vision and Pattern Recognition*. 5154–5163.

[133] Noseong Park, Mahmoud Mohammadi, et al. 2018. Data synthesis based on generative adversarial networks. *Proc. VLDB Endow.* 11, 10 (Jun. 2018), 1071–1083. DOI: http://dx.doi.org/10.14778/3231751.3231757

[134] Michael F. Mathieu, Junbo Jake Zhao, et al. 2016. Disentangling factors of variation in deep representation using adversarial training. In *Conference on Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg,

I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/ef0917ea498b1665ad6c701057155abe-Paper.pdf

[135] Matthew J. Johnson, David K. Duvenaud, et al. 2016. Composing graphical models with neural networks for structured representations and fast inference. In *Conference on Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/7d6044e95a16761171b130dcb476a43e-Paper.pdf

[136] Paul Upchurch, Jacob Gardner, et al. 2017. Deep feature interpolation for image content changes. In *Conference on Computer Vision and Pattern Recognition*. 7064–7073.

[137] Durk P. Kingma and Prafulla Dhariwal. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Conference on Neural Information Processing Systems*, S. Bengio, H. Wallach, et al. (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf

[138] Ramiro Camino, Christian Hammerschmidt, and Radu State. 2018. Generating multi-categorical samples with generative adversarial networks. *arXiv preprint arXiv:1807.01202* (2018).

[139] Ziyu Wang, Dingsu Wang, et al. 2020. Learning interpretable representation for controllable polyphonic music generation. In *International Society for Music Information Retrieval 2020*, Julie Cumming, Jin Ha Lee, Brian McFee, Markus Schedl, Johanna Devaney, Cory McKay, Eva Zangerle, and Timothy de Reuse (Eds.). 662–669. http://archives.ismir.net/ismir2020/paper/000094.pdf

[140] Tengfei Ma, Jie Chen, and Cao Xiao. 2018. Constrained generation of semantically valid graphs via regularizing variational autoencoders. *Conference on Neural Information Processing Systems* 31 (2018).

[141] Lore Goetschalckx, Alex Andonian, et al. 2019. GANalyze: Toward visual definitions of cognitive image properties. In *International Conference on Computer Vision*.

[142] Thomas Möllenhoff and Daniel Cremers. 2019. Flat metric minimization with applications in generative modeling. In *International Conference on Machine Learning*. PMLR, 4626–4635.

[143] Kaushalya Madhawa, Katushiko Ishiguro, et al. 2019. GraphNVP: An invertible flow model for generating molecular graphs. *arXiv preprint arXiv:1905.11600* (2019).

[144] Yujun Shen and Bolei Zhou. 2021. Closed-form factorization of latent semantics in GANs. In *Conference on Computer Vision and Pattern Recognition*. 1532–1540.

[145] Yuanqi Du, Xian Liu, et al. 2022. ChemSpacE: Toward steerable and interpretable chemical space exploration. In *International Conference on Learning Representations 2022 Machine Learning for Drug Discovery*.

[146] Xiaojie Guo, Yuanqi Du, and Liang Zhao. 2021. Deep generative models for spatial networks. In *Special Interest Group on Knowledge Discovery and Data Mining*. 505–515.

[147] Yuanqi Du, Xiaojie Guo, et al. 2022. Disentangled spatiotemporal graph generative models. *AAAI Conference on Artificial Intelligence* 36, 6 (Jun. 2022), 6541–6549. DOI: http://dx.doi.org/10.1609/aaai.v36i6.20607

[148] Xiaojie Guo, Yuanqi Du, et al. 2021. Generating tertiary protein structures via interpretable graph variational autoencoders. *Bioinformatics Advances* 1, 1 (2021), vbab036.

[149] Ashis Pati and Alexander Lerch. 2021. Is disentanglement enough? On latent representations for controllable music generation. In *International Society for Music Information Retrieval*. Online.

[150] Shidi Li, Miaomiao Liu, and Christian Walder. 2022. EditVAE: Unsupervised parts-aware controllable 3D point cloud shape generation. *AAAI Conference on Artificial Intelligence* 36, 2 (Jun. 2022), 1386–1394. DOI: http://dx.doi.org/10.1609/aaai.v36i2.20027

[151] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1511.06434

[152] Yuanqi Du, Xiaojie Guo, et al. 2020. Interpretable molecule generation via disentanglement learning. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 1–8.

[153] Hang Zhou, Yu Liu, et al. 2019. Talking face generation by adversarially disentangled audio-visual representation. In *AAAI Conference on Artificial Intelligence*, Vol. 33. 9299–9306.

[154] Xiaojie Guo, Liang Zhao, et al. 2020. Interpretable deep graph generation with node-edge co-disentanglement. In *Special Interest Group on Knowledge Discovery and Data Mining*.

[155] Yuanqi Du, Yinkai Wang, et al. 2021. Deep latent-variable models for controllable molecule generation. In *International Conference on Bioinformatics and Biomedicine*. IEEE, 372–375.

[156] Yuanqi Du, Xiaojie Guo, et al. 2022. Small molecule generation via disentangled representation learning. *Bioinformatics (Oxford, England)* (2022), btac296.

[157] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. 2020. Controlling generative models with continuous factors of variations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1laeJrKDB

[158] Francesco Locatello, Michael Tschannen, et al. 2020. Disentangling factors of variations using few labels. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SygagpEKwB

[159] Ali Jahanian, Lucy Chai, and Phillip Isola. 2020. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HylsTT4FvB

[160] Erik Härkönen, Aaron Hertzmann, et al. 2020. GANSpace: Discovering interpretable GAN controls. *Conference on Neural Information Processing Systems* 33 (2020), 9841–9850.

[161] Jie Yang, Kaichun Mo, et al. 2020. DSM-Net: Disentangled structured mesh net for controllable generation of fine geometry. *arXiv preprint arXiv:2008.05440* 2, 3 (2020).

[162] Peng Xu, Jackie Chi Kit Cheung, and Yanshuai Cao. 2020. On variational learning of controllable representations for text without supervision. In *International Conference on Machine Learning*. PMLR, 10534–10543.

[163] Hao Hao Tan and Dorien Herremans. 2020. Music FaderNets: Controllable music generation based on high-level features via low-level feature modelling. In *International Society for Music Information Retrieval*.

[164] Dhasarathy Parthasarathy, Karl Bäckstrom, et al. 2020. Controlled time series generation for automotive software-in-the-loop testing using GANs. In *IEEE International Conference on Artificial Intelligence Testing*. IEEE, 39–46.

[165] Shiyu Wang, Xiaojie Guo, and Liang Zhao. 2022. Deep generative model for periodic graphs. *arXiv preprint arXiv:2201.11932* (2022).

[166] AkshatKumar Nigam, Pascal Friederich, et al. 2020. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1lmyRNFvr

[167] Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2021. A distributional approach to controlled text generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=jWkw45-9AbL

[168] Tianfan Fu, Wenhao Gao, et al. 2022. Differentiable scaffolding tree for molecule optimization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=w_drCosT76

[169] Yanfei Kang, Rob J. Hyndman, and Feng Li. 2020. GRATIS: GeneRAting time series with diverse and controllable characteristics. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 13, 4 (2020), 354–376.

[170] Sumanth Dathathri, Andrea Madotto, et al. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1edEyBKDS

[171] Lantao Yu, Weinan Zhang, et al. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *AAAI Conference on Artificial Intelligence*, Vol. 31.

[172] Zhiting Hu, Zichao Yang, et al. 2018. Deep generative models with learnable knowledge constraints. *Conference on Neural Information Processing Systems* 31 (2018).

[173] Evgeny Putin, Arip Asadulaev, et al. 2018. Reinforced adversarial neural computer for de novo molecular design. *Journal of Chemical Information and Modeling* 58, 6 (2018), 1194–1204.

[174] Pradyumna Tambwekar, Murtaza Dhuliawala, et al. 2019. Controllable neural story plot generation via reinforcement learning. In *International Joint Conferences on Artificial Intelligence*.

[175] Chence Shi, Minkai Xu, et al. 2020. GraphAF: A flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=S1esMkHYPr

[176] Xiufeng Yang, Tanuj Aasawat, and Kazuki Yoshizoe. 2021. Practical massively parallel Monte-Carlo tree search applied to molecular design. In *International Conference on Learning Representations*. https://openreview.net/forum?id=6k7VdojAIK

[177] Bidisha Samanta, Abir De, et al. 2020. NeVAE: A deep generative model for molecular graphs. *Journal of Machine Learning Research* (2020).

[178] Jike Wang, Chang-Yu Hsieh, et al. 2021. Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. *Nature Machine Intelligence* 3, 10 (2021), 914–922.

[179] Niklas Gebauer, Michael Gastegger, and Kristof Schütt. 2019. Symmetry-adapted generation of 3D point sets for the targeted discovery of molecules. *Conference on Neural Information Processing Systems* 32 (2019).

[180] Marwin H. S. Segler, Thierry Kogej, et al. 2018. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science* 4, 1 (2018), 120–131.

[181] Shrimai Prabhumoye, Yulia Tsvetkov, et al. 2018. Style transfer through back-translation. In *Proc. ACL*.

[182] Wengong Jin, Kevin Yang, et al. 2019. Learning multimodal graph-to-graph translation for molecule optimization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1xJAsA5F7

[183] Rewa Sood, Binit Topiwala, et al. 2018. An application of generative adversarial networks for super resolution medical imaging. In *International Conference on Machine Learning and Applications*. IEEE, 326–331.

[184] Youzhi Luo, Keqiang Yan, and Shuiwang Ji. 2021. GraphDF: A discrete flow model for molecular graph generation. In *International Conference on Machine Learning*. PMLR, 7192–7203.

[185] Tianfan Fu, Cao Xiao, and Jimeng Sun. 2020. Core: Automatic molecule optimization using copy & refine strategy. In *AAAI Conference on Artificial Intelligence*, Vol. 34. 638–645.

[186] Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. Transforming delete, retrieve, generate approach for controlled text style transfer. In *Conference on Empirical Methods in Natural Language Processing*.

[187] Håkon Hukkelås, Rudolf Mester, and Frank Lindseth. 2019. DeepPrivacy: A generative adversarial network for face anonymization. In *International symposium on visual computing*. Springer, 565–578.

[188] Zhenpeng Zhou, Xiaocheng Li, and Richard N. Zare. 2017. Optimizing chemical reactions with deep reinforcement learning. *ACS Central Science* 3, 12 (2017), 1337–1344.

[189] Noé Tits, Kevin El Haddad, and Thierry Dutoit. 2019. Exploring transfer learning for low resource emotional TTS. In *Proceedings of SAI Intelligent Systems Conference*. Springer, 52–60.

[190] Rui Liu, Berrak Sisman, and Haizhou Li. 2021. Reinforcement learning for emotional text-to-speech synthesis with improved emotion discriminability. In *Interspeech*.

[191] Wei-Ning Hsu, Yu Zhang, et al. 2019. Hierarchical generative modeling for controllable speech synthesis. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rygkk305YQ

[192] Mingda Chen, Qingming Tang, et al. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proc. of ACL*.

[193] Dong Liang, Rui Wang, et al. 2019. PCGAN: Partition-controlled human image generation. In *AAAI Conference on Artificial Intelligence*, Vol. 33. 8698–8705.

[194] Lele Chen, Ross K. Maddox, et al. 2019. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Conference on Computer Vision and Pattern Recognition*. 7832–7841.

[195] Maxim Maximov, Ismail Elezi, and Laura Leal-Taixé. 2020. CIAGAN: Conditional identity anonymization generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition*. 5447–5456.

[196] Tao Li, Shan Yang, et al. 2021. Controllable emotion transfer for end-to-end speech synthesis. In *International Symposium on Chinese Spoken Language Processing*. IEEE, 1–5.

[197] Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Conference of the European Chapter of the Association for Computational Linguistics*.

[198] Albert Pumarola, Stefan Popov, et al. 2020. C-flow: Conditional generative flow models for images and 3D point clouds. In *Conference on Computer Vision and Pattern Recognition*. 7949–7958.

[199] Zichao Yang, Zhiting Hu, et al. 2018. Unsupervised text style transfer using language models as discriminators. *Conference on Neural Information Processing Systems* 31 (2018).

[200] Rafael Valle, Jason Li, et al. 2020. Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 6189–6193.

[201] Kiyoshi Kurihara, Nobumasa Seiyama, and Tadashi Kumano. 2021. Prosodic features control by symbols as input of sequence-to-sequence acoustic modeling for neural TTS. *IEICE Transactions on Information and Systems* 104, 2 (2021), 302–311.

[202] Minchan Kim, Sung Jun Cheon, et al. 2021. Expressive text-to-speech using style tag. In *Interspeech*.

[203] Yanyao Bian, Changbin Chen, et al. 2019. Multi-reference tacotron by intercross training for style disentangling, transfer and control in speech synthesis. *arXiv preprint arXiv:1904.02373* (2019).

[204] Katsuki Inoue, Sunao Hara, et al. 2021. Model architectures to extrapolate emotional expressions in DNN-based text-to-speech. *Speech Communication* 126 (2021), 35–43.

[205] Xiong Cai, Dongyang Dai, et al. 2021. Emotion controllable speech synthesis using emotion-unlabeled dataset with the assistance of cross-domain speech emotion recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 5734–5738.

[206] Ben Krause, Akhilesh Deepak Gotmare, et al. 2020. GeDi: Generative discriminator guided sequence generation. In *Conference on Empirical Methods in Natural Language Processing*.

[207] Yang Song, Jingwen Zhu, et al. 2019. Talking face generation by conditional recurrent adversarial network. In *International Joint Conferences on Artificial Intelligence*. 919–925.

[208] Ohsung Kwon, Inseon Jang, et al. 2019. An effective style token weight control technique for end-to-end emotional speech synthesis. *IEEE Signal Processing Letters* 26, 9 (2019), 1383–1387.

[209] Noé Tits. 2019. A methodology for controlling the emotional expressiveness in synthetic speech-a deep learning approach. In *International Conference on Affective Computing and Intelligent Interaction Workshops and Demos*. IEEE, 1–5.

[210] Aghilas Sini, Sébastien Le Maguer, et al. 2020. Introducing prosodic speaker identity for a better expressive speech synthesis control. In *10th International Conference on Speech Prosody 2020*. ISCA, 935–939.

[211] Juncen Li, Robin Jia, et al. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

[212] Giorgio Fabbro, Vladimir Golkov, et al. 2020. Speech synthesis and control using differentiable DSP. *arXiv preprint arXiv:2010.15084* (2020).

[213] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 424–434. DOI: http://dx.doi.org/10.18653/v1/P19-1041

[214] Rameen Abdal, Peihao Zhu, et al. 2021. StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *TOG* 40, 3 (2021), 1–21.

[215] Jiapeng Zhu, Ruili Feng, et al. 2021. Low-rank subspaces in GANs. *Conference on Neural Information Processing Systems* 34 (2021).

[216] Weihao Xia, Yulun Zhang, et al. 2022. GAN inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 01 (Jun. 2022), 1–17. DOI: http://dx.doi.org/10.1109/TPAMI.2022.3181070

[217] Spyridon Thermos, Xiao Liu, et al. 2021. Controllable cardiac synthesis via disentangled anatomy arithmetic. In *International Conference on Medical Image Computing and Computer Assisted Intervention*. Springer, 160–170.

[218] Haw-Shiuan Chang, Jiaming Yuan, Mohit Iyyer, and Andrew McCallum. 2021. Changing the mind of transformers for topically-controllable language generation. In *Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2601–2611. DOI: http://dx.doi.org/10.18653/v1/2021.eacl-main.223

[219] Yujun Shen, Jinjin Gu, et al. 2020. Interpreting the latent space of GANs for semantic face editing. In *Conference on Computer Vision and Pattern Recognition*. 9243–9252.

[220] Lin Qiao, Jianhao Yan, et al. 2020. A sentiment-controllable topic-to-essay generator with topic knowledge graph. In *Findings of Conference on Empirical Methods in Natural Language Processing*.

[221] Kelvin Guu, Tatsunori B. Hashimoto, et al. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics* 6 (2018), 437–450.

[222] Yi Ren, Yangjun Ruan, et al. 2019. FastSpeech: Fast, robust and controllable text to speech. *Conference on Neural Information Processing Systems* 32 (2019).

[223] Xuan Luo, Shinnosuke Takamichi, et al. 2021. Controllable text-to-speech synthesis using prosodic features and emotion soft-label. *Sython.org*.

[224] Tuomo Raitio, Ramya Rasipuram, and Dan Castellani. 2020. Controllable neural text-to-speech synthesis using intuitive prosodic features. In *INTERSPEECH*.

[225] Chenye Cui, Yi Ren, et al. 2021. EMOVIE: A Mandarin emotion speech dataset with a simple emotional text-to-speech model. In *Proc. Interspeech 2021*. 2766–2770. DOI: http://dx.doi.org/10.21437/Interspeech.2021-1148

[226] Jesse Engel, Lamtharn (Hanoi) Hantrakul, et al. 2020. DDSP: Differentiable digital signal processing. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1x1ma4tDr

[227] Danica J. Sutherland, Hsiao-Yu Tung, et al. 2017. Generative models and model criticism via optimized maximum mean discrepancy. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HJWHIKqgl

[228] Liming Zhang, Liang Zhao, et al. 2021. TG-GAN: Continuous-time temporal graph deep generative models with time-validity constraints. In *International World Wide Web Conference*. 2104–2116.

[229] Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Conference on Neural Information Processing Systems* 32 (2019).

[230] Yu Cheng, Yongshun Gong, et al. 2021. Molecular design in drug discovery: A comprehensive review of deep generative models. *Briefings in Bioinformatics* 22, 6 (2021), bbab344.

[231] Hejie Cui, Jiaying Lu, et al. 2022. How can graph neural networks help document retrieval: A case study on cord19 with concept map generation. In *European Conference on Information Retrieval*. Springer, 75–83.

[232] Reda Yacouby and Dustin Axman. 2020. Probabilistic extension of precision, recall, and F1 score for more thorough evaluation of classification models. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*. 79–91.

[233] Chris Kedzie and Kathleen McKeown. 2020. Controllable meaning representation to text generation: Linearization and data augmentation strategies. In *Conference on Empirical Methods in Natural Language Processing*. 5160–5185.

[234] Andrew Rosenberg and Bhuvana Ramabhadran. 2017. Bias and statistical significance in evaluating speech synthesis with mean opinion scores. In *Interspeech*. 3976–3980.

[235] Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).

[236] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. 2020. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science* 11, 2 (2020), 577–586.

[237] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020).

[238] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 8780–8794.

[239] Heeseung Kim, Sungwon Kim, and Sungroh Yoon. 2022. Guided-TTS: A diffusion model for text-to-speech via classifier guidance. In *International Conference on Machine Learning*. PMLR, 11119–11133.

[240] Bahjat Kawar, Roy Ganz, and Michael Elad. 2022. Enhancing diffusion-based image synthesis with robust classifier guidance. *arXiv preprint arXiv:2208.08664* (2022).

[241] Fuzhen Zhuang, Zhiyuan Qi, et al. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.

[242] Kai Arulkumaran, Marc Peter Deisenroth, et al. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38.

[243] Kuzman Ganchev, Joao Graça, et al. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research* 11 (2010), 2001–2049.

[244] Benjamin Sanchez-Lengeling, Carlos Outeiral, et al. 2017. Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). *Chemrxiv.org*.

[245] Zhu Zhang, Jianxin Ma, et al. 2021. UFC-BERT: Unifying multi-modal controls for conditional image synthesis. *Conference on Neural Information Processing Systems* 34 (2021).

[246] Eric Anderson, Gilman D. Veith, and David Weininger. 1987. *SMILES, a Line Notation and Computerized Interpreter for Chemical Structures*. US Environmental Protection Agency, Environmental Research Laboratory.

[247] W. Patrick Walters and Regina Barzilay. 2020. Applications of deep learning in molecule generation and molecular property prediction. *Accounts of Chemical Research* 54, 2 (2020), 263–270.

[248] Daniel C. Elton, Zois Boukouvalas, et al. 2019. Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering* 4, 4 (2019), 828–849.

[249] Joshua Meyers, Benedek Fabian, and Nathan Brown. 2021. De novo molecular design and generative models. *Drug Discovery Today* 26, 11 (2021), 2707–2715.

[250] Ryan-Rhys Griffiths, Leo Klarner, et al. GAUCHE: A library for Gaussian processes in chemistry. In *International Conference on Machine Learning 2022 2nd AI for Science Workshop*.

[251] Zhaoping Xiong, Dingyan Wang, et al. 2019. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry* 63, 16 (2019), 8749–8760.

[252] Wan Xiang Shen, Xian Zeng, et al. 2021. Out-of-the-box deep learning prediction of pharmaceutical properties by broadly learned knowledge-based molecular representations. *Nature Machine Intelligence* 3, 4 (2021), 334–343.

[253] Martin Simonovsky and Nikos Komodakis. 2018. GraphVAE: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*. Springer, 412–422.

[254] Jiaxuan You, Rex Ying, et al. 2018. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning*. PMLR, 5708–5717.

[255] Daniel Neil, Marwin Segler, et al. 2018. Exploring deep recurrent models with reinforcement learning for molecule design. In *(ICLR'18)*. Workshop track.

[256] Sungsoo Ahn, Junsu Kim, et al. 2020. Guiding deep molecular optimization with genetic exploration. *Conference on Neural Information Processing Systems* 33 (2020), 12008–12021.

[257] Zhenpeng Zhou, Steven Kearnes, et al. 2019. Optimization of molecules via deep reinforcement learning. *Scientific Reports* 9, 1 (2019), 1–10.

[258] Robin Winter, Floriane Montanari, et al. 2019. Efficient multi-objective molecular optimization in a continuous latent space. *Chemical Science* 10, 34 (2019), 8016–8024.

[259] Tristan Aumentado-Armstrong. 2018. Latent molecular optimization for targeted therapeutic design. *arXiv preprint arXiv:1809.02032* (2018).

[260] Tianfan Fu, Cao Xiao, et al. 2021. Mimosa: Multi-constraint molecule sampling for molecule optimization. In *AAAI Conference on Artificial Intelligence*, Vol. 35. 125–133.

[261] Christine A. Orengo, Annabel E. Todd, and Janet M. Thornton. 1999. From protein structure to function. *Current Opinion in Structural Biology* 9, 3 (1999), 374–382.

[262] John Jumper, Richard Evans, et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 7873 (2021), 583–589.

[263] Jingxue Wang, Huali Cao, et al. 2018. Computational protein design with deep learning neural networks. *Scientific Reports* 8, 1 (2018), 1–9.

[264] Wenze Ding, Kenta Nakai, and Haipeng Gong. 2022. Protein design via deep learning. *Briefings in Bioinformatics* 23, 3 (2022), bbac102.

[265] Fangneng Zhan, Yingchen Yu, et al. 2021. Multimodal image synthesis and editing: A survey. *arXiv preprint arXiv:2112.13592* (2021).

[266] Paul Taylor. 2009. *Text-to-Speech Synthesis*. Cambridge University Press.

[267] Yishuang Ning, Sheng He, et al. 2019. A review of deep learning based speech synthesis. *Applied Sciences* 9, 19 (2019), 4050.

[268] David Fooshee, Aaron Mood, et al. 2018. Deep learning for chemical reaction prediction. *Molecular Systems Design & Engineering* 3, 3 (2018), 442–452.

[269] Philippe Schwaller, Alain C. Vaucher, et al. 2021. Prediction of chemical reaction yields using deep learning. *Machine Learning: Science and Technology* 2, 1 (2021), 015016.

[270] Tânia F. G. G. Cova and Alberto A. C. C. Pais. 2019. Deep learning for deep chemistry: Optimizing the prediction of chemical patterns. *Frontiers in Chemistry* 7 (2019), 809.

Received 12 November 2022; revised 6 September 2023; accepted 31 January 2024