

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет  
«Высшая школа экономики»

*На правах рукописи*

**Полыковский Даниил Александрович**

## **Генеративные модели для задачи поиска лекарств**

Резюме  
диссертации на соискание учёной степени  
кандидата компьютерных наук

Москва — 2021

**Диссертационная работа выполнена в** федеральном государственном автономном образовательном учреждении высшего образования «Национальный исследовательский университет «Высшая школа экономики».

**Научный руководитель:** Ветров Дмитрий Петрович, к.ф.-м.н., Национальный исследовательский университет «Высшая школа экономики».

# 1 Тема диссертации

В течение последних нескольких лет машинное обучение начало использоваться на разных этапах решения задачи поиска лекарств: поиске новых биологических мишеней, поиске активных молекулярных структур, прогнозировании биологических и физико-химических свойств [1, 2, 3, 4]. На данный момент несколько работ продемонстрировали активные молекулы, созданные с использованием искусственного интеллекта: такие молекулы были протестированы *in vitro* и *in vivo* [5, 6, 7, 8]. В данной диссертации рассмотрены три задачи, встречающиеся при поиске лекарств генеративными моделями: условная генерация, обучение распределению и оптимизация молекулярных свойств.

В задаче *обучения распределению* мы стремимся создать новые молекулярные структуры из того же распределения, что и обучающая выборка. Полученные выборки молекулярных структур можно использовать для предобучения предсказательных моделей без учителя или ранжирования сгенерированных молекул в соответствии с некоторой функцией качества (виртуальный скрининг). В задаче *условной генерации* модели порождают молекулы с заданными свойствами — такой подход позволяет сузить химическое пространство до желаемого класса молекулярных структур. Целью задачи *оптимизации молекулярных свойств* является обнаружение молекул с максимально возможным значением функции качества. Например, такая функция качества может быть предсказательной моделью активности против заданной биологической мишени.

Для каждой из обозначенных выше задач мы предлагаем новые модели машинного обучения. В первой работе [9] мы демонстрируем, что генеративные поатомные графовые генераторы работают хуже строковых моделей на задаче обучения распределению. Мы предлагаем новую модель иерархической генерации графов, которая работает значительно лучше существующих поатомных графовых моделей. Во второй работе мы изучаем задачу условной генерации [5] и применяем состязательные автокодировщики для создания новых молекулярных структур с желаемыми свойствами. С помощью предложенной модели мы смогли создать молекулярную структуру, которая позже показала селективную микромолярную *in vitro* активность против выбранной биологической мишени. В третьей работе мы оптимизируем молекулярные свойства при помощи байесовской оптимизации; оптимизация ведется

в пространстве латентных кодов вариационных автокодировщиков, обученных с детерминированным декодированием [10].

## Актуальность работы

Вычислительные подходы широко используются для предсказания молекулярных свойств [11] и исследования химического пространства с помощью высокопроизводительного скрининга, комбинаторных библиотек и эволюционных алгоритмов [12, 13, 14, 15]. В отличие от традиционных подходов к задаче поиска лекарств, когда молекулярные структуры создаются вручную, генеративные модели предлагают автоматизированный подход, при котором опыт медицинского химика необходим только на заключительных этапах — для проверки качества новых молекулярных структур. Такой подход не зависит от предвзятости эксперта и может учитывать явные или неявные ограничения при поиске. Хотя специалист может создать молекулярные структуры, имеющие определенные точки связывания и заданную форму, наши подходы также могут использовать высокоточные прогностические модели, проводить быструю оценку новизны и патентной чистоты. Такой мощный инструмент может предложить эффективные гипотезы за несколько недель с минимальным контролем со стороны человека [16].

В данной работе процесс поиска лекарств сформулирован как задача оптимизации функции качества  $f(x)$  молекулярной структуры  $x$ . В качестве примера функции  $f$  может быть выбрана модель прогнозирования активности или энергия связывания молекулы в белке, оцененная при помощи докинга или молекулярной динамики. Построение целевой функции — интересная и сложная задача, требующая экспертизы в предметной области. В данной работе мы используем стандартные используемые для сравнения функции качества. Реальные целевые функции, такие как те, что использовались в наших недавних статьях [16, 6], анализируют сгенерированные структуры с точки зрения активности, новизны, синтетической доступности и других тщательно отобранных критериев.

Первая проблема при решении задачи оптимизации — как представить молекулярную структуру. Наиболее распространены графовые и строковые представления. В графовом представлении вершины обозначают атомы, а ребра — связи. Строковое представление можно получить, записав символы атомов молекулы в порядке поис-

ка в глубину, используя специальные маркеры для циклических связей и ветвления. Такое представление называется системой упрощенного представления молекул в строке ввода (simplified molecular input line entry system, SMILES) [17, 18]. Многие предыдущие работы по обработке естественного языка можно применять напрямую к SMILES, что является преимуществом строкового представления. Например, посимвольная языковая модель может использоваться для порождения новых SMILES. Также существуют работы, которые используют контекстно-свободную грамматику [19], обратную польскую нотацию [20] или подструктурные представления, такие как деревья сочленений [2]. Модели на основе графов менее изучены, но это — быстро развивающаяся тема. Каждое представление может быть аннотировано трехмерными координатами атомов, молекулярными свойствами или отпечатками (fingerprints). Отпечатки — это двоичные векторы, описывающие наличие или отсутствие заданных подструктур. Например, наиболее популярные Моргановские отпечатки [21] кодируют подструктуры при помощи хеширования. Отпечатки можно использовать при обучении предсказательных моделей. Также их часто используют для задания меры сходства между молекулами. Например, похожесть можно оценивать при помощи коэффициента Танимото (количество битов, которые имеют значение 1 для обеих молекул, деленное на количество битов, которые имеют значение 1 по крайней мере для одной молекулы).

Существует несколько подходов к оптимизации  $f(x)$  — с использованием обучения с подкреплением, генетических алгоритмов или байесовской оптимизации. Оптимизацию можно вести в домене молекул, используя генетические алгоритмы или байесовскую оптимизацию на графах. В последнем случае графовые ядра и другие меры близости могут использоваться для обучения модели гауссовских процессов [22]. Гомез-Бомбарелли и др. [1] предложили обучить вариационный автокодировщик на молекулярных структурах и вести оптимизацию в домене скрытых кодов. Авторы использовали байесовский подход к оптимизации латентных кодов, но в последующих работах были использованы и другие подходы.

Помимо оптимизации молекулярных свойств, машинное обучение используется для обучения распределению. По обучающему набору молекулярных структур, взятых из неизвестного распределения, модели учатся создавать новые структуры из того же распределения. В статье [23] мы предложили набор данных и метрик для сравнения сгенерированных структур с точки зрения различных показателей, в том

числе: уникальности, новизны, разнообразия, сходства с обучающими объектами. Мы также реализовали несколько базовых моделей и сравнили их на основе этих показателей. Модели обучения распределению применяются для создания виртуальных библиотек, так как они позволяют автоматически учитывать неявные ограничения из обучающих данных. Сгенерированные структуры, которые можно использовать для ранжирования или поиска. Например, для оптимизации новой функции  $f(x)$  можно вычислить целевую функцию на структурах виртуальной библиотеки. Такой подход экономит время на переобучении моделей и позволяет быстро находить высокоскоринговые соединения.

За последние несколько лет мы внедрили несколько новых моделей и интегрировали их в автоматизированную платформу для поиска лекарств под названием Chemistry42. Chemistry42 поддерживает разработку лекарств как на основе лигандов, так и на основе структуры белка, создавая высокоскоринговые структуры в течение недели. В своей диссертации я описываю некоторые использованные в этой платформе модели и иллюстрирую их применение на стандартных наборах данных.

**Целью** этой работы является разработка новых генеративных моделей молекулярных структур для задач условной генерации, оптимизации молекулярных свойств и обучения распределению.

## 2 Основные результаты и выводы

**Вклад.** Основными результатами этой работы являются три генеративные модели и их приложения к задаче поиска лекарств.

1. Мы проанализировали поатомные графовые генеративные модели и предложили процедуру иерархической генерации графов, а также — фрагментно-ориентированное упорядочение атомов. Мы получили лучшие результаты по сравнению с другими поатомными графовыми генеративными моделями в задачах оптимизации молекулярных свойств и обучения распределению. Мы также продемонстрировали что скрытые коды нижнего уровня влияют на локальные свойства структуры, а скрытые коды верхнего уровня определяют глобальную структуру.

2. Предложена модель запутанного условного состязательного автокодировщика (ЕСААЕ), которая расширяет состязательный автокодировщик, обученный с учителем, и успешно поддерживает множественные бинарные и непрерывные условия. Показано, что предложенная модель может генерировать молекулярные структуры для условий за пределами исходного диапазона обучения и генерировать структуры с микромолярной активностью.
3. Для оптимизации молекулярных свойств мы изучили метод байесовской оптимизации скрытых кодов вариационных автокодировщиков и предложили детерминированное декодирование, чтобы избежать проблем связанных со стохастическим декодированием. Мы предложили подход к обучению, основанный на релаксированной целевой функции обучения, и доказали сходимость такого метода к оптимуму исходной задачи оптимизации. Мы также предложили применять апостериорные распределения с ограниченным носителем, чтобы гарантировать существование набора параметров вариационного автокодировщика, при котором достигается автокодирование без потерь.

**Теоретическая и практическая значимость.** Предложенные модели открывают путь для дальнейшего развития глубинного обучения в задаче поиска лекарств. Эти модели могут ускорить открытие новых лекарств и значительно снизить затраты на первоначальных этапах, что особенно важно во время глобальной пандемии. Для условного моделирования мы предложили новый алгоритм, позволяющий генерировать селективные молекулы с микромолярной активностью против выбранного белка. Мы также проанализировали проблему оптимизации молекулярных свойств и предложили новый подход к обучению вариационных автокодировщиков с детерминированным декодированием. Также мы улучшили качество моделей обучения распределению для поатомных графовых генеративных моделей с помощью иерархической генерации — мы получили 3,5-кратное улучшение основной метрики обучения распределению (Fréchet ChemNet Distance).

#### **Результаты, выносимые на защиту.**

1. Иерархическая графовая генеративная модель для молекулярных структур и ее применение к задаче обучения распределению и оптимизации молекулярных свойств

2. Модель запутанного условного состязательного автокодировщика для условной генерации молекулярных структур
3. Метод обучения вариационных автокодировщиков с детерминированным декодированием и применение этого метода для задачи оптимизации молекулярных свойств

**Личный вклад в результаты, выносимые на защиту.** Во второй и третьей статьях метод был предложен и реализован автором, все эксперименты проводились автором, текст написан автором; другие авторы контролировали исследование и помогали с экспертизой в предметной области. В первой статье автор разработал дизайн экспериментов, контролировал исследование и написал текст.

## Публикации и апробация работы

Автор диссертации является основным автором в трех основных статьях по теме диссертации.

### Публикации повышенного уровня.

1. **Даниил Польшковский**, Александр Жебрак, Дмитрий Ветров, Ян Иваненков, Владимир Аладинский, Полина Мамошина, Маринэ Боздаганян, Александр Алипер, Александр Жаворонков, Артур Кадурин. Запутанный состязательный автокодировщик для de novo поиска лекарств (Entangled Conditional Adversarial Autoencoder for de Novo Drug Discovery). Молекулярная фармацевтика (Molecular pharmaceuticals), 15(10):4398–4405, 2018. Q1 журнал, индексируется SCOPUS.
2. **Даниил Польшковский** и Дмитрий Ветров. Детерминированное декодирование для дискретных данных в вариационных автокодировщиках (Deterministic Decoding for Discrete Data in Variational Autoencoders). Материалы двадцать третьей международной конференции по искусственному интеллекту и статистике, том 108 трудов исследований в области машинного обучения, страницы 3046–3056 (Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, volume 108 of Proceedings of Machine Learning Research, pages 3046–3056). Конференция уровня А.



3. *Максим Кузнецов и Даниил Польшковский*. MolGrow: Графовый нормализационный поток для иерархической генерации молекулярных структур (MolGrow: A graph normalizing flow for hierarchical molecular generation). Конференция ассоциации по развитию искусственного интеллекта 2021 (Association for the Advancement of Artificial Intelligence Conference 2021). Конференция уровня A\*.

### Доклады на конференциях

1. Конференция по системам обработки нейронной информации (Neural information processing systems conference), 2 декабря 2018. Доклад на Экспо (Expo Tutorial). Тема: "Генеративные модели для задачи поиска лекарств".
2. Конференция по системам обработки нейронной информации (Neural information processing systems conference), 2 декабря 2018. Экспо мастерская (Expo Workshop). Тема: "Машинное обучение для задачи поиска лекарств и разработки биомаркеров".
3. Отменяя старение (Undoing Aging), 30 марта 2019. Тема: "Глубинный генеративный подход к транскриптомному анализу старения"
4. Международная конференция по машинному обучению (International conference on machine learning), 9 июня 2019. Доклад на Экспо (Expo Tutorial). Тема: "Генеративные модели для задачи поиска лекарств".

**Объем и структура работы.** Диссертация содержит введение, содержание публикаций и заключение. Полный объем диссертации 67 страниц.

## 3 Содержание работы

### 3.1 MolGrow: графовый нормализационный поток для иерархической генерации молекулярных структур

В предыдущем разделе мы предложили модель на основе VAE, в которой использовались строковые представления молекулярных структур. В недавних работах [24] было продемонстрировано, что графовое представление полезно для оптимизации молекулярных свойств. Кроме того, Мадхава и др. [25] продемонстрировали, что модели на основе нормализационных потоков дают хорошие результаты в задаче опти-

мизации молекулярных свойств. В предыдущих статьях не изучались проблемы обучения распределению и оптимизации молекулярных свойств одновременно. Несмотря на то, что модели обычно предварительно обучаются распределению, предыдущие поатомные графовые генеративные модели работают значительно хуже самых простых строковых генеративных моделей. В этом разделе мы предлагаем новую поатомную графовую генеративную модель, которая позволяет значительно сузить разрыв между строковыми и графовыми моделями.

Предыдущие работы по генерации графов порождали элементы графа либо последовательно [26, 27] или одновременно [25, 28]. Мы предложили новый подход к генерации: начиная с графа, состоящего из одной вершины, мы итеративно разбиваем каждую вершину на две и повторяем эту процедуру до тех пор, пока не получим граф заданного размера. Мы определили набор обратимых преобразований для разделения и слияния вершин, добавления и отделения шума. Мы также заметили, что стандартный порядок атомов (поиск в ширину) приводит к созданию нежелательных макроциклов в сгенерированных структурах. Мы предложили новое фрагментно-ориентированный порядок атомов. В таком порядке мы сначала разбиваем молекулярную структуру на набор осмысленных фрагментов, а затем выравниваем эти фрагменты с помощью слияния и расщепления вершин.

Мы представляем граф матрицей атрибутов вершин  $V \in \mathbb{R}^{N \times d_v}$  и тензором атрибутов ребер  $E \in \mathbb{R}^{N \times N \times d_e}$ , где  $d_v$  и  $d_e$  — размеры элемента. Для входных данных  $V_i$  определяет тип и заряд атома,  $E_{i,j}$  определяет тип ребра. Поскольку молекулярные графы неориентированы, мы сохраняем ограничение симметрии на всех промежуточных слоях:  $E_{i,j,k} = E_{j,i,k}$ . Окончательный граф имеет  $N = 2^L$  вершин, где  $L$  — количество слоев, разделяющих вершины.

Мы используем операции слияния и разделения вершин, чтобы контролировать размер графа. Эти операции обратны друг другу, и обе работают путем перестановки признаков вершин и ребер. Рассмотрим граф  $(V^k, E^k)$  с  $N_k$  вершинами. Операция слияния вершин объединяет вершины с номерами  $2i$  и  $2i + 1$  в одну вершину, конкатенируя их признаки и признаки ребер между ними. Мы также конкатенируем

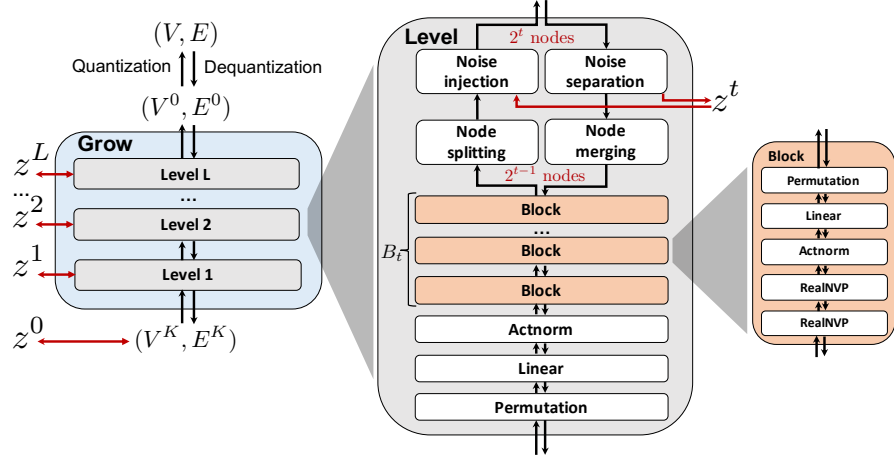


Рис. 1: Архитектура MolGrow. **Слева:** Полная архитектура объединяет несколько *уровней* для генерации скрытых кодов  $z^L, \dots, z^0$  из графа  $(V, E)$  и наоборот. **Посередине:** Каждый *уровень* отделяет шум, объединяет пары вершин, применяет несколько *блоков* и линейные преобразования; **Справа:** Каждый *блок* применяет три поканальных преобразования и два слоя RealNVP.

признаки ребер, соединяющих объединенные вершины:

$$\underbrace{V_i^{k+1}}_{2d_v+d_e} = \text{cat}\left(\underbrace{V_{2i}^k}_{d_v}, \underbrace{V_{2i+1}^k}_{d_v}, \underbrace{E_{2i,2i+1}^k}_{d_e}\right), \quad (1)$$

$$\underbrace{E_{i,j}^{k+1}}_{4d_e} = \text{cat}\left(\underbrace{E_{2i,2j}^k}_{d_e}, \underbrace{E_{2i,2j+1}^k}_{d_e}, \underbrace{E_{2i+1,2j}^k}_{d_e}, \underbrace{E_{2i+1,2j+1}^k}_{d_e}\right). \quad (2)$$

Расщепление вершин — это обратная операция по отношению к слиянию вершин: она разделяет признаки на исходные компоненты. MolGrow производит скрытый вектор для каждого *уровня*. Мы получаем скрытые коды уровня, разделяя половину признаков вершин и ребер до того как объединить вершины. Мы накладываем гауссовское априорное распределение на скрытые коды. Во время генерации мы семплируем скрытый код из априорного распределения и объединяем его с признаками вершин и ребер. Как показывают эксперименты, скрытые коды на разных уровнях по-разному влияют на генерируемую структуру. Скрытые коды, соответствующие более мелким промежуточным графам (верхний уровень) влияют на глобальную структуру, в то время как скрытые коды нижнего *уровня* определяют локальную структуру. Модель проиллюстрирована на рисунке 1.

В экспериментах мы сравниваем нашу модель с современными графовыми и строковыми генеративными моделями. В таблице 1 показано, что MolGrow превосходит

Таблица 1: Оптимизация молекулярных свойств: функция липофильности со штрафной поправкой (penalized logP) и количественная оценка подобию лекарствам (QED). Результаты для базовых моделей взяты из [29, 25].

Method	Penalized logP			QED		
	1st	2nd	3rd	1st	2nd	3rd
ZINC250k	4.52	4.30	4.23	<b>0.948</b>	<b>0.948</b>	<b>0.948</b>
Графовые модели						
GCPN [24]	7.98	7.85	7.80	<b>0.948</b>	0.947	0.946
MolecularRNN [27]	8.63	6.08	4.73	0.844	0.796	0.736
GraphNVP [25]	-	-	-	0.833	0.723	0.706
GraphAF [29]	12.23	11.29	11.05	<b>0.948</b>	<b>0.948</b>	<b>0.948</b>
MoFlow [28]	-	-	-	<b>0.948</b>	<b>0.948</b>	<b>0.948</b>
Предложенная модель						
MolGrow (генетика)	<b>14.01 ± 0.364</b>	<b>13.95 ± 0.424</b>	<b>13.92 ± 0.422</b>	<b>0.9484 ± 0.0</b>	<b>0.9484 ± 0.0</b>	<b>0.9484 ± 0.0</b>
MolGrow (генетика, только верхние уровни)	11.66 ± 0.31	11.65 ± 0.319	11.63 ± 0.306	<b>0.9484 ± 0.0</b>	<b>0.9484 ± 0.0</b>	<b>0.9484 ± 0.0</b>
MolGrow (генетика, только нижние уровни)	10.29 ± 3.32	10.29 ± 3.33	10.28 ± 3.32	<b>0.9484 ± 0.0</b>	<b>0.9484 ± 0.0</b>	<b>0.9484 ± 0.0</b>
MolGrow (направляемая предиктором оптимизация)	5.2 ± 0.347	4.94 ± 0.262	4.84 ± 0.22	<b>0.9484 ± 0.0</b>	0.9483 ± 0.0	0.9483 ± 0.0
MolGrow (REINFORCE)	4.81 ± 0.285	4.47 ± 0.145	4.39 ± 0.126	0.9468 ± 0.001	0.9459 ± 0.001	0.9455 ± 0.001
Модели на SMILES и фрагментах						
DD-VAE [10]	5.86	5.77	5.64	-	-	-
Grammar VAE [3]	2.94	2.88	2.80	-	-	-
SD-VAE [30]	4.04	3.50	2.96	-	-	-
JT-VAE [2]	5.30	4.93	4.49	<b>0.948</b>	0.947	0.947

лучшие графовые и строковые генераторы по двум наиболее часто используемым целевым функциям — липофильность со штрафной поправкой (penalized logP) и количественная оценка подобию лекарствам (QED). Мы также значительно превзошли лучший поатомный графовый генератор в задаче обучения распределению (Таблица 2) с точки зрения расстояния Fr échet ChemNet (FCD).

### 3.2 Запутанный условный состязательный автокодировщик для задачи поиска лекарств

В этой секции мы переходим к рассмотрению задачи условной генерации, где модель должна создавать новые молекулярные структуры с заданными свойствами. В ранних работах по генеративной химии [32, 33] предлагалось генерировать отпечатки молекулярных структур и извлекать наиболее похожие молекулы из базы молекулярных структур. Это не требует большого количества данных, а найденные структуры сразу доступны для покупки. Тем не менее, такие модели не позволяют генерировать новые молекулярные структуры. Обучение непосредственно SMILES требует большего количества данных даже для создания семантически правильных

Таблица 2: Метрики обучения распределению на данных MOSES.

Метод	FCD/Test ( $\downarrow$ )	Frag/Test ( $\uparrow$ )	Уникальность@10k ( $\uparrow$ )	Новизна ( $\uparrow$ )
Графовые модели				
MolecularRNN [27]	23.13	0.56	98.6%	99.9%
GraphVAE [31]	49.39	0.0	5%	<b>100%</b>
GraphNVP [25]	29.95	0.62	99.7 %	99.9%
GraphAF (BFS) [29]	21.84	0.651	97%	99.9%
MoFlow [28]	28.05	0.685	100%	99.99%
Предложенная модель				
MolGrow (фрагментно-ориентированный порядок)	<b>6.284 <math>\pm</math> 0.986</b>	0.929 $\pm$ 0.025	99.28 $\pm$ 0.62%	99.26 $\pm$ 0.12%
MolGrow (BFS)	9.962 $\pm$ 0.795	0.932 $\pm$ 0.01	<b>100 <math>\pm</math> 0.0%</b>	99.37 $\pm$ 0.08%
MolGrow (BFS на фрагментах)	16.15 $\pm$ 1.026	0.868 $\pm$ 0.018	<b>100 <math>\pm</math> 0.0%</b>	<b>100 <math>\pm</math> 0.0%</b>
MolGrow (случайная перестановка)	40.17 $\pm$ 4.709	0.051 $\pm$ 0.034	58.96 $\pm$ 38.11%	<b>100 <math>\pm</math> 0.0%</b>
MolGrow (слой GAT вместо CAGE)	6.523 $\pm$ 0.302	<b>0.941 <math>\pm</math> 0.013</b>	99.36 $\pm$ 0.3%	99.32 $\pm$ 0.05%
MolGrow (без позиционных представлений)	6.771 $\pm$ 0.555	0.937 $\pm$ 0.006	99.49 $\pm$ 0.19%	99.41 $\pm$ 0.06%
Модели на SMIELS и фрагментах				
CharRNN (из MOSES)	<b>0.073</b>	<b>0.9998</b>	99.73%	84.19%
VAE (из MOSES)	0.099	0.9994	99.84%	69.49%
JTN-VAE (из MOSES)	0.422	0.9962	<b>100%</b>	<b>91.53%</b>

строк. В нашей работе мы предложили объединить генератор отпечатков с условной генеративной моделью над SMILES. Мы подошли к такой задаче с точки зрения условной генерации: модель выучивает распределение  $p(x | y)$  над молекулярными структурами  $x$  с заданными свойствами  $y$  (отпечатками).

В статье [5] мы предложили условную модель на основе состязательного автокодировщика. Состязательные автокодировщики (AAE) [34] — это генеративные модели, которые моделируют распределение данных  $p_d(x)$  путем обучения регуляризованного автокодировщика. Регуляризатор вынуждает распределение скрытого кода  $q(z) = \int Q_E(z | x)p_d(x)dx$ , совпадать с априорным распределением  $p(z)$ . Распределения  $Q_E(z | x)$  и  $P_G(x | z)$  параметризуются нейронными сетями  $E$  и  $G$  соответственно:  $z = E(x)$  и  $x = G(z)$ , то есть кодирование и декодирование детерминированы.

Регуляризация латентного пространства реализуется процедурой состязательного обучения [35] при помощи дискриминатора  $D(z)$ . Дискриминатор обучается различать выборки из скрытого распределения  $q(z)$  и априорного распределения  $p(z)$ . Кодировщик  $E$  обучается изменять скрытый код так, чтобы дискриминатор не смог отличить скрытое распределение от априорного. Это приводит к минимаксной задаче

оптимизации  $\min_E \max_D \mathcal{L}_{\text{adv}}$ , где

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{x \sim p_d} \log D(E(x)) + \mathbb{E}_{z \sim p(z)} \log(1 - D(z)) \quad (3)$$

Состязательное обучение, в совокупности со штрафом за реконструкцию задает следующую задачу оптимизации:

$$\min_{E, G} \max_D \mathbb{E}_{x \sim p_d} \log D(E(x)) + \mathbb{E}_{z \sim p(z)} \log(1 - D(z)) - \mathbb{E}_{x \sim p_d} \log p(x | G(E(x))). \quad (4)$$

Предложенная модель, названная "Запутанный условный состязательный автокодировщик"(ЕСААЕ), вводит условное априорное распределение  $p_\theta(z | y) \sim \mathcal{N}(\mu_\theta(y), \Sigma_\theta(y))$  и использует условный дискриминатор для обучения модели. Мы используем трюк репараметризации  $\bar{z} = g_\theta(z, y) = \Sigma_\theta^{-1/2}(y)(z - \mu_\theta(y))$  для упрощения целевой функции:

$$\begin{aligned} \min_{E, G, \theta} \max_D \mathbb{E}_{(x, y) \sim p_d} \log D(g_\theta(E(x), y), y) + \mathbb{E}_{y \sim p(y)} \mathbb{E}_{\bar{z} \sim p(\bar{z})} \log(1 - D(\bar{z}, y)) \\ - \mathbb{E}_{(x, y) \sim p_d} \log p(x | G(E(x), y)). \end{aligned} \quad (5)$$

Мы также предлагаем дополнительный регуляризатор, улучшающий обучение. В приведенной выше задаче оптимизации цель дискриминатора может быть интерпретирована как обеспечение независимости репараметризованного скрытого кода и условия. Мы предложили технику, названную предиктивным распутыванием: она использует отдельную прогностическую модель для предсказания  $y$  из  $z$  и изменяет скрытые коды так, чтобы обмануть предсказательную модель. Аддитивный регуляризатор имеет вид

$$\min_E \max_q \lambda \mathbb{E}_{(x, y) \sim p_d} \log q(y | g_\theta(E(x), y)). \quad (6)$$

В экспериментах мы изучали различные условия генерации и сравнивали различные модификации ААЕ, включая модель с фиксированной сетью, задающей априорное распределение:  $p_\theta(z | y) = \mathcal{N}(0, I)$ . В таблице 3 мы сравнили генеративные модели, обусловленные на отпечатки MACCS — 166-битные вектора, каждый бит которых отвечает за наличие или отсутствие определенной подструктуры. В этой таблице "Нет" соответствует состязательному автокодировщику, обученному с учителем (SAAE), "Предиктивный" соответствует использованию прогнозирующего распутывания и случаю когда условие не передается на вход дискриминатору. "Совместный" соответствует подаче условия на дискриминатор. "Комбинированный" соответствует предсказательному распутыванию и подаче условия на вход дискриминатору.

В таблице 4 мы обусловили модель на три непрерывных свойства: липофильность, оценку синтетической доступности и энергию связывания с белком MCL1. Из таблицы видно, что удаление любого компонента модели приводит к игнорированию части условий. В статье мы также показали, что помощи предложенной модели можно создать новые молекулы с большей энергией связи, чем наилучшая молекула из обучающих данных. Мы также провели аналогичный эксперимент, обучив модель, обусловленную на дозу, необходимую для 50%-ого ингибирования (IC50) для белка JAK3 и сгенерировали молекулярную структуру, показавшую микромолярную активность против выбранного белка *in vitro*.

Таблица 3: Качество моделей, обученных с помощью различных методов распутывания; бинарные отпечатки в качестве условия. Обратите внимание на большой разрыв между моделью без распутывания (соответствующей [34]) и другими моделями. Первые четыре модели используют безусловную модель априорного распределения.

Распутывание	Танимото, %	Хэмминг	Точное удовлетворение условия, %	Оставшаяся вз. инф-я
Нет	80.0	10.49	4.4	2.75
Предсказательное	86.2	7.13	11.4	0.64
Совместное	88.7	5.78	17.4	1.56
Комбинированное	91.8	4.18	27.8	0.32
ЕСААЕ, без предск.	93.5	3.31	40.9	2.51
ЕСААЕ	<b>93.6</b>	<b>3.28</b>	<b>41.3</b>	1.30

Таблица 4: Качество моделей обученных на частично размеченной выборке. Коэффициент корреляции Пирсона  $r$  между указанным и полученным свойствами.

Распутывание	logP, $r$	SA, $r$	$E$ , $r$
—	$0.311 \pm 0.01$	$0.0522 \pm 0.009$	$0.02 \pm 0.04$
Предсказательное	$0.687 \pm 0.006$	$0.0893 \pm 0.008$	$0.063 \pm 0.05$
Совместное	$0.595 \pm 0.007$	$0.0838 \pm 0.008$	$0.109 \pm 0.04$
Комбинированное	$0.677 \pm 0.007$	$0.0896 \pm 0.007$	$0.116 \pm 0.04$
Запутанное	<b><math>0.804 \pm 0.005</math></b>	<b><math>0.593 \pm 0.007</math></b>	<b><math>0.406 \pm 0.04</math></b>

### 3.3 Детерминированное декодирование в вариационных автокодировщиках для дискретных данных

В то время как обучение распределению и условное моделирование полезны для быстрого исследования химического пространства, конечная цель задачи поиска лекарств — найти одну или несколько “идеальных” молекул. Задача оптимизации молекулярных свойств обычно используется для таких приложений как поиск первичных хитов и хит-оптимизация, если вычислительный подход может надежно оценивать качество молекулярных структур. Цель модели — обнаружить молекулу с максимальной мерой качества.

В этом разделе мы описываем проблему оптимизации молекулярных свойств в генеративной химии и рассматриваем часто используемый подход к решению этой задачи при помощи вариационных автокодировщиков (VAE) и байесовской оптимизации [1]. Мы обсуждаем проблему, связанную со стохастическим декодированием в VAE и предлагаем детерминированное декодирование в качестве решения [10].

Задача оптимизации молекулярных свойств является одной из важных задач вычислительной химии. Цель такой задачи — имея некоторую целевую функцию  $f(x)$ , которая оценивает качество молекулярной структуры, найти соединение  $x_*$  с наивысшим качеством:  $x_* \in \arg \max_{x \in \mathcal{X}} f(x)$ . В зависимости от задачи,  $f(x)$  может направлять оптимизацию в сторону структур с желаемыми физико-химическими свойствами или биологической активностью [36]. Также можно ограничить область поиска  $\mathcal{X}$  соединениями, структурно похожими на данную структуру. Примером задачи с ограничениями является хит-оптимизация, когда молекула-хит оптимизируется в сторону более высокой активности и лучших физико-химических свойств. Вычисление  $f(x)$  обычно дорогостоящее и трудоемкое, поскольку для практических задач  $f(x)$  часто требует сложного моделирования или синтеза и *in vitro* тестирования.

Основываясь на работах по обучению представлений при помощи вариационных автокодировщиков (VAE) [37], Гомез-Бомбарелли и др. [1] предложили адаптировать VAE для задач оптимизации молекулярных свойств. Их подход заключается в том, чтобы сначала обучить VAE на большом наборе молекулярных структур, а затем оптимизировать молекулярные свойства в пространстве скрытых кодов VAE. Авторы вычисляют скрытые коды всех обучающих примеров и обучают регрессионную модель на основе гауссовских процессов [38] для предсказания целевой функции. Ис-



пользуя такую модель, авторы находят скрытый код, соответствующий структуре с наибольшим ожидаемым улучшением. Полученную при помощи декодировщика молекулярную структуру добавляют к обучающей выборке и повторяют процедуру обучения несколько раз. Рассмотренный подход является частным случаем байесовской оптимизации.

В упомянутой выше статье используются нейросетевые кодировщик и декодировщик, обученные на SMILES. Такие сложные декодеры в VAE имеют тенденцию игнорировать скрытые коды, поскольку они могут моделировать целевое распределение самостоятельно, без помощи скрытых кодов. Однако для того, чтобы байесовская оптимизация была успешной, скрытые коды должны нести полезную информацию о кодируемых объектах — чем легче предсказать целевые свойства на основе скрытых кодов, тем легче искать оптимальные структуры. Чтобы декодировщик не игнорировал скрытые коды, мы рассматриваем детерминированное декодирование, где каждый скрытый код соответствует одной молекулярной структуре. Простой способ превратить стохастический декодер в детерминированный — заменить все операции семплирования на жадный выбор наиболее вероятного токена на каждой итерации генерации. Такая схема семплирования смещена и может уменьшить разнообразие генерируемых структур, поскольку режимы работы модели во время обучения и семплирования различаются. Мы предлагаем обучать модель с детерминированным декодированием напрямую.

Рассмотрим нижнюю границу на обоснованность вариационного автокодировщика с рекуррентным декодировщиком:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{x \sim p(x)} \left[ \mathbb{E}_{z \sim q_\phi(z|x)} \sum_{i=1}^{|x|} \log \pi_{x,i,x_i}^\theta(z) - \mathcal{KL}(q_\phi(z|x) \| p(z)) \right], \quad (7)$$

где  $p(x)$  — распределение данных,  $q_\phi$  — кодирующее (апостериорное) распределение, а  $\pi_{x,i,s}^\theta(z)$  — декодирующее распределение  $p_\theta(x_i = s | z, x_1, \dots, x_{i-1})$ . В детерминированных декодировщиках, декодируемая последовательность  $\tilde{x}_\theta(z)$  вычисляется как

$$\tilde{x}_i = \arg \max_s p_\theta(s | z, x_1, \dots, x_{i-1}) = \arg \max_s \pi_{x,i,s}^\theta(z) \quad (8)$$

Распределение декодируемых последовательностей и нижняя граница на обоснованность для детерминированного декодирования вычисляются по следующим форму-

лам:

$$p(x | \tilde{x}_\theta(z)) = \begin{cases} 1, & \tilde{x}_\theta(z) = x \\ 0, & \text{иначе} \end{cases} \quad (9)$$

$$\mathcal{L}_*(\theta, \phi) = \mathbb{E}_{x \sim p(x)} [\mathbb{E}_{z \sim q_\phi(z|x)} \log p(x | \tilde{x}_\theta(z)) - \mathcal{KL}(q_\phi(z | x) \| p(z))] \quad (10)$$

Мы предлагаем оптимизировать  $\mathcal{L}_*$ , аппроксимируя  $\arg \max$  гладкой функцией и постепенно понижая температуру:

$$\mathbb{I} \left[ i = \arg \max_j r_j \right] = \prod_{j \neq i} \mathbb{I} [r_i > r_j] \approx \prod_{j \neq i} \sigma_\tau(r_i - r_j), \quad (11)$$

$$\sigma_\tau(x) = \frac{1}{1 + \exp(-x/\tau) \left( \frac{1}{\tau} - 1 \right)} \xrightarrow{\tau \rightarrow 0} \mathbb{I} [x > 0] \quad (12)$$

$$\mathcal{L}_\tau(\theta, \phi) = \mathbb{E}_{x \sim p(x)} \left[ \mathbb{E}_{z \sim q_\phi(z|x)} \sum_{i=1}^{|x|} \sum_{s \neq x_i} \log \sigma_\tau(\pi_{x,i,x_i}^\theta(z) - \pi_{x,i,s}^\theta(z)) - \mathcal{KL}(q_\phi(z | x) \| p(z)) \right] \quad (13)$$

Мы столкнулись с двумя проблемами: как обеспечить конечность целевой функции ( $\mathcal{L}_* > -\infty$ ) хотя бы для каких-то значений параметров  $(\theta, \phi)$  и как оптимизация  $\mathcal{L}_\tau$  связана с оптимизацией  $\mathcal{L}_*$ .

Мы показываем, что невозможно получить конечное значение  $\mathcal{L}_*$ , если апостериорное распределение  $q(x | z)$  имеет полный носитель: если кодировщик сопоставляет любому объекту любой скрытый код с ненулевой вероятностью, то декодировщик должен декодировать любой объект из любого скрытого кода. Следовательно, детерминированный декодировщик всегда будет иметь ненулевую долю ошибок реконструкции и, соответственно, бесконечно малый  $\mathcal{L}_*$ . Чтобы решить эту проблему, мы предложили апостериорные распределения с ограниченным носителем — смещенные и отмасштабированные факторизованные ядра:

$$q_\phi(z | x) = \prod_{i=1}^d \frac{1}{\sigma_i^\phi(x)} K \left( \frac{z_i - \mu_i^\phi(x)}{\sigma_i^\phi(x)} \right). \quad (14)$$

Мы вывели аналитическую формулу для дивергенции Кульбака-Лейблера для различных ядер и двух априорных распределений: стандартного гауссовского и равномерного. Апостериорные распределения с ограниченным носителем совместно с достаточно гибким кодировщиком и декодировщиком гарантируют, что для некоторых  $(\theta, \phi)$ ,  $\mathcal{L}_*$  конечно.

Чтобы связать  $\mathcal{L}_\tau$  с  $\mathcal{L}_*$ , мы доказали следующую теорему.

**Теорема 1.** Пусть  $\Omega = \{(\theta, \phi) \mid \mathcal{L}_*(\theta, \phi) > -\infty\}$ . Пусть  $\Delta(\tilde{x}_\theta, \phi)$  — доля ошибок реконструкции на уровне последовательностей для заданной пары кодировщик-декодировщик, а  $\Delta(\phi)$  — доля ошибок реконструкции на уровне последовательностей для оптимального декодировщика (производящего последовательность с максимальной апостериорной вероятностью декодирования). Пусть  $\Omega \neq \emptyset$ , длина последовательностей ограничена ( $\exists L : |x| \leq L, \forall x \in \chi$ );  $\Theta$  и  $\Phi$  — компактные множества возможных значений параметров. Пусть  $q_\phi(z \mid x)$  равномерно непрерывна по полной вариации для любых  $\phi$  и  $x$ :

$$\begin{aligned} \forall \epsilon > 0, \exists \delta = \delta(\epsilon, x, \phi) > 0 : \\ \|\phi - \phi'\| < \delta \Rightarrow \int |q_\phi(z \mid x) - q_{\phi'}(z \mid x)| dz < \epsilon. \end{aligned} \quad (15)$$

Пусть  $\tau_n, \phi_n, \theta_n$  — такие последовательности что:

$$\lim_{n \rightarrow \infty} \tau_n = 0, \quad \tau_n \in (0, 1), \quad (16)$$

$$(\theta_n, \phi_n) \in \operatorname{Argmax}_{\theta \in \Theta, \phi \in \Phi} \mathcal{L}_{\tau_n}(\theta, \phi), \quad (17)$$

последовательность  $\{\phi_n\}$  сходится к  $\tilde{\phi}$ , и для любого  $\phi$  такого что  $\Delta(\phi) = 0$  существует  $\theta$  такая что  $\Delta(\tilde{x}_\theta, \phi) = 0$ . Пусть  $\tilde{\theta}$  выбрано таким образом что:

$$\tilde{\theta} \in \operatorname{Argmax}_{\theta \in \Theta} \mathcal{L}_*(\theta, \tilde{\phi}). \quad (18)$$

Тогда, доля ошибок реконструкции на уровне последовательностей убывает асимптотически как

$$\Delta(\tilde{x}_{\theta_n}, \phi_n) = \mathcal{O}\left(\frac{1}{\log(1/\tau_n)}\right), \quad (19)$$

Параметры  $(\tilde{\theta}, \tilde{\phi})$  решают задачи оптимизации для  $\mathcal{L}_*$ :

$$\mathcal{L}_*(\tilde{\theta}, \tilde{\phi}) = \sup_{\theta \in \Theta, \phi \in \Phi} \mathcal{L}_*(\theta, \phi). \quad (20)$$

Эта теорема показывает, что если мы оптимизируем модель и постепенно понижаем температуру  $\tau$ , мы получим оптимальный кодировщик. Если мы затем дообучим декодировщик, мы получим оптимальную пару кодировщик-декодировщик. В экспериментах мы обучили модель на SMILES и оптимизировали  $\mathcal{L}_\tau$  постепенно уменьшая  $\tau$ . Мы рассмотрели задачу обучения распределению и задачи оптимизации молекулярных свойств. При обучении распределению предлагаемая методика обучения и апостериорные распределения с ограниченным носителем улучшают Fréchet ChemNet Distance (FCD) и сходство с ближайшим соседом (SNN) в наборе данных

MOSES (Таблица 5). При оптимизации молекулярных свойств мы оптимизировали часто используемую функцию липофильности со штрафной поправкой [3]. В единых условиях мы получили молекулы лучшего качества, чем стандартный VAE и другие базовые модели. Модель также показала лучшее качество предсказания целевого свойства на основе скрытых кодов по сравнению с обычным VAE (Таблица 6).

Таблица 5: Обучение распределению с детерминированным декодированием на наборе данных MOSES для различной точности реконструкции. Показаны метрики FCD/Test (меньше — лучше) и SNN/Test (больше — лучше). Среднее значение  $\pm$  std на трех независимых запусках. G = гауссовское апостериорное распределение, T = трикватратное апостериорное распределение.

Метод	FCD/Test ( $\downarrow$ )			SNN/Test ( $\uparrow$ )		
	70%	80%	90%	70%	80%	90%
VAE (G)	0.205 $\pm$ 0.005	0.344 $\pm$ 0.003	0.772 $\pm$ 0.007	0.550 $\pm$ 0.001	0.525 $\pm$ 0.001	0.488 $\pm$ 0.001
VAE (T)	0.207 $\pm$ 0.004	0.335 $\pm$ 0.005	0.753 $\pm$ 0.019	0.550 $\pm$ 0.001	0.526 $\pm$ 0.001	0.490 $\pm$ 0.000
DD-VAE (G)	0.198 $\pm$ 0.012	0.312 $\pm$ 0.011	0.711 $\pm$ 0.020	<b>0.555 <math>\pm</math> 0.001</b>	0.531 $\pm$ 0.001	0.494 $\pm$ 0.001
DD-VAE (T)	<b>0.194 <math>\pm</math> 0.001</b>	<b>0.311 <math>\pm</math> 0.010</b>	<b>0.690 <math>\pm</math> 0.010</b>	<b>0.555 <math>\pm</math> 0.000</b>	<b>0.532 <math>\pm</math> 0.001</b>	<b>0.495 <math>\pm</math> 0.001</b>

Таблица 6: Точность восстановления (на уровне последовательностей) и валидность. Модели обучены на наборе данных ZINC; Качество предсказания разреженными гауссовскими процессами: логарифм правдоподобия (LL) и среднеквадратичная ошибка (RMSE); Значение целевой функции трех лучших молекул, найденных с помощью байесовской оптимизации. G = гауссовское апостериорное распределение, T = трикватратное апостериорное распределение.

Метод	Точность реконструкции	Валидность	LL	RMSE	топ-1	топ-2	топ-3
CVAE	44.6%	0.7%	-1.812 $\pm$ 0.004	1.504 $\pm$ 0.006	1.98	1.42	1.19
GVAE	53.7%	7.2%	-1.739 $\pm$ 0.004	1.404 $\pm$ 0.006	2.94	2.89	2.80
SD-VAE	76.2%	43.5%	-1.697 $\pm$ 0.015	1.366 $\pm$ 0.023	4.04	3.50	2.96
JT-VAE	76.7%	100.0%	-1.658 $\pm$ 0.023	1.290 $\pm$ 0.026	5.30	4.93	4.49
VAE (G)	87.01%	78.32%	-1.558 $\pm$ 0.019	1.273 $\pm$ 0.050	5.76	5.74	<b>5.67</b>
VAE (T)	90.3%	73.52%	-1.562 $\pm$ 0.022	1.265 $\pm$ 0.051	5.41	5.38	5.35
DD-VAE (G)	89.39%	63.07%	-1.481 $\pm$ 0.020	1.199 $\pm$ 0.050	5.13	4.84	4.80
DD-VAE (T)	89.89%	61.38%	<b>-1.470 <math>\pm</math> 0.022</b>	<b>1.186 <math>\pm</math> 0.053</b>	<b>5.86</b>	<b>5.77</b>	5.64

## 4 Заключение

В заключительном разделе мы подытоживаем основные результаты работы.

1. Мы обнаружили, что современные поатомные графовые генераторы показывают плохие результаты в задаче обучения распределению — значительно хуже, чем аналогичные строковые и подструктурные генераторы. Мы предложили новую модель генерации молекулярных графов, которая создает молекулярные структуры иерархически. Начиная с одной вершины, она итеративно увеличивает размер графа, разделяя каждую вершину на две. Мы получили хорошие результаты одновременного в задачах оптимизации молекулярных свойств и обучении распределению.
2. Мы изучили условные генеративные модели для задачи поиска лекарств и предложили новую модель — Запутанный условный состязательный автокодировщик. Эта модель может обрабатывать несколько условий и экстраполировать за пределы диапазона условий в обучающих данных. Эта статья одной из первых подтвердила возможность сгенерировать новую молекулярную структуру и продемонстрировать ее активность *in vitro* против искомой биологической мишени.
3. Мы изучили широко используемую комбинацию вариационных автокодировщиков и байесовской оптимизации и обнаружили потенциальные проблемы с использованием детерминированного декодирования во время генерации и вероятностного декодирования во время обучения. Мы построили детерминированную процедуру декодирования и предложили интуитивно понятную схему обучения с использованием релаксированной целевой функции. В экспериментах мы продемонстрировали улучшение основных метрик качества на задачах оптимизации молекулярных свойств.
4. Мы доказали теорему, связывающую релаксированную целевую функцию и исходную целевую функцию. Мы также обнаружили, что декодирование без потерь невозможно с апостериорными распределениями, имеющими полный носитель. Мы предложили использовать апостериорные распределения с ограниченным носителем для улучшения модели.

## Список литературы

- [1] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a Data-Driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, February 2018.
- [2] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2323–2332, Stockholmsmässan, Stockholm Sweden, 2018. PMLR.
- [3] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1945–1954. JMLR. org, 2017.
- [4] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.
- [5] Daniil Polykovskiy, Alexander Zhebrak, Dmitry Vetrov, Yan Ivanenkov, Vladimir Aladinskiy, Polina Mamoshina, Marine Bozdaganyan, Alexander Aliper, Alex Zhavoronkov, and Artur Kadurin. Entangled conditional adversarial autoencoder for de novo drug discovery. *Molecular pharmaceutics*, 15(10):4398–4405, 2018.
- [6] Alex Zhavoronkov, Bogdan Zagribelnyy, Alexander Zhebrak, Vladimir Aladinskiy, Victor Terentiev, Quentin Vanhaelen, Dmitry S Bezrukov, Daniil Polykovskiy, Rim Shayakhmetov, Andrey Filimonov, et al. Potential non-covalent sars-cov-2 3c-like protease inhibitors designed using generative deep learning approaches and reviewed by human medicinal chemist in virtual reality. 2020.
- [7] Daniel Merk, Lukas Friedrich, Francesca Grisoni, and Gisbert Schneider. De novo design of bioactive small molecules by artificial intelligence. *Molecular informatics*, 37(1-2):1700153, 2018.

- [8] Daniel Merk, Francesca Grisoni, Lukas Friedrich, and Gisbert Schneider. Tuning artificial intelligence on the de novo design of natural-product-inspired retinoid x receptor modulators. *Communications Chemistry*, 1(1):68, 2018.
- [9] Maxim Kuznetsov and Daniil Polykovskiy. Molgrow: A graph normalizing flow for hierarchical molecular generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, Feb. 2021.
- [10] Daniil Polykovskiy and Dmitry Vetrov. Deterministic decoding for discrete data in variational autoencoders. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3046–3056, Online, 26–28 Aug 2020. PMLR. URL <http://proceedings.mlr.press/v108/polykovskiy20a.html>.
- [11] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.*, 9(2):513–530, January 2018.
- [12] Stefano Curtarolo, Gus L W Hart, Marco Buongiorno Nardelli, Natalio Mingo, Stefano Sanvito, and Ohad Levy. The high-throughput highway to computational materials design. *Nat. Mater.*, 12(3):191–201, March 2013.
- [13] Xiangqian Hu, David N Beratan, and Weitao Yang. Emergent strategies for inverse molecular design. *Sci. China B*, 52(11):1769–1776, November 2009.
- [14] Tu C Le and David A Winkler. Discovery and optimization of materials using evolutionary approaches. *Chem. Rev.*, 116(10):6107–6132, May 2016.
- [15] Edward O Pyzer-Knapp, Changwon Suh, Rafael Gómez-Bombarelli, Jorge Aguilera-Iparraguirre, and Alán Aspuru-Guzik. What is High-Throughput virtual screening? a perspective from organic materials discovery. *Annu. Rev. Mater. Res.*, 45(1):195–216, 2015.
- [16] Alex Zhavoronkov, Yan A Ivanenkov, Alex Aliper, Mark S Veselov, Vladimir A Aladinskiy, Anastasiya V Aladinskaya, Victor A Terentiev, Daniil A Polykovskiy, Maksim D Kuznetsov, Arip Asadulaev, et al. Deep learning enables rapid

- identification of potent ddr1 kinase inhibitors. *Nature biotechnology*, 37(9):1038–1040, 2019.
- [17] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. 17:1–14, 1970.
- [18] David Weininger, Arthur Weininger, and Joseph L Weininger. Smiles. 2. algorithm for generation of unique smiles notation. *Journal of chemical information and computer sciences*, 29(2):97–101, 1989.
- [19] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. Selfies: a robust representation of semantically constrained graphs with an example application in chemistry. *arXiv preprint arXiv:1905.13741*, 2019.
- [20] Noel O’Boyle and Andrew Dalke. DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures. *ChemRxiv*, 2018.
- [21] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, May 2010.
- [22] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *The Journal of Machine Learning Research*, 11:1201–1242, 2010.
- [23] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11:1931, 2020.
- [24] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in neural information processing systems*, pages 6410–6421, 2018.
- [25] Kaushalya Madhawa, Katushiko Ishiguro, Kosuke Nakago, and Motoki Abe. GraphNVP: An invertible flow model for generating molecular graphs. *arXiv preprint arXiv:1905.11600*, 2019.
- [26] Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In Jennifer



- Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5708–5717, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL <http://proceedings.mlr.press/v80/you18a.html>.
- [27] Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. Molecularrrnn: Generating realistic molecular graphs with optimized properties. *arXiv preprint arXiv:1905.13372*, 2019.
- [28] Chengxi Zang and Fei Wang. Moflow: An invertible flow model for generating molecular graphs. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Aug 2020. doi: 10.1145/3394486.3403104. URL <http://dx.doi.org/10.1145/3394486.3403104>.
- [29] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *International Conference on Learning Representations*, 2020.
- [30] Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. Syntax-directed variational autoencoder for structured data. *arXiv preprint arXiv:1802.08786*, 2018.
- [31] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 412–422, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01418-6.
- [32] Artur Kadurin, Alexander Aliper, Andrey Kazennov, Polina Mamoshina, Quentin Vanhaelen, Kuzma Khrabrov, and Alex Zhavoronkov. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, 8(7):10883–10890, 2016. ISSN 1949-2553. doi: <https://doi.org/10.18632/oncotarget.14073>. URL <https://www.oncotarget.com/article/14073/>.
- [33] Artur Kadurin, Sergey Nikolenko, Kuzma Khrabrov, Alex Aliper, and Alex Zhavoronkov. druGAN: An advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Mol. Pharm.*, 14(9):3098–3104, September 2017.

- [34] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. 2016.
- [35] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. pages 2672–2680, 2014.
- [36] Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.
- [37] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*, 2013.
- [38] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.