Автономная некоммерческая образовательная организация высшего образования
“Сколковский институт науки и технологий”

На правах рукописи

Волхонский Денис Алексеевич

ГЛУБОКОЕ ГЕНЕРАТИВНОЕ ОБУЧЕНИЕ ДЛЯ МОДЕЛИРОВАНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ИЗОБРАЖЕНИЙ

РЕЗЮМЕ
диссертации на соискание ученой степени
кандидата компьютерных наук

Научный руководитель:
доктор физико-математических наук
Бурнаев Евгений Владимирович

Москва — 2022

Диссертационная работа выполнена в Сколковском институте науки и технологий.

Научный руководитель: Бурнаев Евгений Владимирович, д.ф.-м.н., профессор, руководитель Исследовательского центра в сфере искусственного интеллекта по направлению оптимизации управленческих решений в целях снижения углеродного следа Сколковского института науки и технологий.

1. Тема диссертации

Для получения надежных результатов во многих задачах машинного обучения и анализа данных, включая задачи обработки последовательности изображений, таких как видеоданные и трехмерные воксельные данные, требуются обучающие выборки значительного объема. Однако не всегда возможно получить достаточное количество данных [27; 1]. В таких ситуациях одним из решений является обогащение обучающих выборок синтетическими данными, искусственно созданными генеративной моделью.

Генерация последовательностей изображений имеет ряд других применений. Например, синтез видеоконтента, что важно при создании мультфильмов [2] и производстве других видео. Ещё одним примером являются трёхмерные воксельные данные, которые также представляют собой последовательность изображений. Приложение для синтеза 3D воксельных текстур полезно в *цифровой физике* [23] при разработки новых нефтяных месторождений.

Во многих приложениях, таких как генерация пористых сред [23], аугментация данных магнитно-резонансной томографии [27], существует дополнительное требование, чтобы свойства генерируемых данных были аналогичны свойствам реальных данных. Так, при моделировании пористых сред важно иметь сходные свойства проницаемости и пористости созданных и реальных объектов. Для генерации видео важно сохранить распределение пикселей. Несмотря на то, что существуют методы генерации последовательности на основе изображений, они не позволяют сохранять такие свойства или не способны сохранять их с достаточным качеством [23; 28; 27]. Ещё одним ограничением существующих решений является то, что они позволяют генерировать последовательности изображений только с низким разрешением [31; 23]. Более того, некоторые генеративные методы для последовательностей видеокадров [5; 31; 22] требуют для обучения до 500 тензорных процессоров, что для большинства пользователей недоступно с вычислительной точки зрения.

Данная диссертация посвящена проблеме генерации последовательностей изображений с сохранением их свойств. Как указано выше, эта тема представляет значительный интерес для научного сообщества.

Цель данной работы — разработать генеративные модели для последовательности изображений, которые позволяют сохранять свойства объектов, генерировать объекты с высоким разрешением и являются реализуемыми с вычислительной точки зрения. Обозначенная цель состоит из следующих задач:

1. разработать способ генерации трехмерных воксельных данных из двумерных срезов с сохранением физических характеристик объектов;
2. разработать способ генерации трехмерных воксельных данных высокого разрешения, позволяющий сохранять физические характеристики объектов;
3. разработать способ генерации последовательностей видеокадров, сохраняющий распределение пикселей видео;

4. разработать метод удаления фона и артефактов изображений, сохраняющий семантически важную информацию.

2. Основные результаты

Научная новизна данной работы заключается в следующем:

1. Мы разработали новый метод генерации трехмерных воксельных данных из двумерного среза. Этот метод позволяет сохранить двумерный входной срез и физические характеристики объектов.
2. Мы разработали новый метод генерации трехмерных воксельных данных с высоким разрешением, при этом обучаясь только на данных с низким разрешением. Он превосходит существующие методы по качеству и позволяет сохранить физические характеристики объектов.
3. Мы разработали новый метод генерации кадров видео, который использует меньше вычислительных ресурсов, чем современные методы, при схожем качестве. Наш метод может быть обучен на 8 графических процессорах (GPU), в то время как существующие методы генерации видео требуют до 512 тензорных процессоров (TPU). Наш метод позволяет сохранять распределение пикселей видео.
4. Мы предложили новый метод удаления фона и артефактов с изображений, который позволяет сохранить семантически важную информацию на изображении. Он превосходит существующие методы очистки по качеству и позволяет закрашивать пропущенные части в семантически важной информации.

Теоретическая и практическая значимость

Теоретическая значимость заключается в новых разработанных методах:

- новый метод генерации трехмерных воксельных данных из двумерных срезов с сохранением физических характеристик объектов;
- новый метод генерации трехмерных воксельных данных высокого разрешения, позволяющий сохранять физические характеристики объектов;
- новый метод генерации последовательностей видеокадров, позволяющий сохранить распределение видеопикселей;
- новый метод удаления фона и артефактов изображений, который позволяет сохранить семантически важную информацию.

Эти методы могут быть использованы учеными и разработчиками при создании новых методов генеративного моделирования и решении новых прикладных задач обработки и анализа изображений.

Практическая значимость заключается в применении наших методов к генерации пористых сред для ускорения разработки нефтяных месторождений. Другими приложениями, продемонстрированными в этой работе, являются генерация видео и очистка растровых изображений. Потенциальное применение также включает в себя генерацию медицинских, геологических или сейсмических данных для их аугментации и анализа.

Основные положения, выносимые на защиту:

1. новый метод генерации трехмерных воксельных данных из двумерного среза;
2. новый метод генерации трехмерных воксельных данных высокого разрешения;
3. новый метод генерации видео с уменьшенными вычислительными требованиями;
4. новый метод удаления фона и артефактов с изображений.

Личный вклад

Автор данной диссертации получил все заявленные результаты. Во всех упомянутых случаях как текст, так и экспериментальные результаты, представленные в статье, являются результатом сотрудничества между авторами.

Результат 1 был разработан в «Generative adversarial networks for reconstruction of three-dimensional porous media from two-dimensional slices», автор разработал и внедрил алгоритм обучения и провел все эксперименты.

Результат 2 был разработан в «User-Controllable Multi-Texture Synthesis with Generative Adversarial Networks», автор разработал метод для трехмерных данных и провел все эксперименты на трёхмерных данных.

Результат 3 был разработан в «Latent Video Transformer», автор реализовал алгоритм трансформер и провёл часть экспериментов.

Результат 4 был разработан в «Deep Vectorization of Technical Drawings», автор разработал метод очистки изображений растрового сканирования и провёл часть экспериментов по очистке.

3. Публикации и апробация работы

Публикации повышенного уровня

1. *Denis Volkhonskiy, Oleg Sudakov, Ekaterina Muravleva, Denis Orlov, Boris Belozerov, Evgeny Burnaev, Dmitry Koroteev* Generative adversarial networks for reconstruction of three-dimensional porous media from two-dimensional slices. Physical Review E, Q2 Journal. Индексировано SCOPUS, Web of science.
2. *Egiazarian Vage*, Oleg Voynov*, Alexey Artemov, Denis Volkhonskiy, Aleksandr Safin, Maria Taktasheva, Denis Zorin, Evgeny Burnaev* Deep Vectorization of Technical Drawings. ECCV 2020, CORE A. Индексировано SCOPUS.

∗ — Equal Contribution

Публикации стандартного уровня

1. *Rakhimov Ruslan*, Denis Volkhonskiy*, Alexey Artemov, Denis Zorin, Evgeny Burnaev* Latent Video Transformer. VISAPP 2021, CORE B. Индексировано SCOPUS.

2. *Aibek Alanov*, Max Kochurov*, Denis Volkhonskiy, Daniil Yashkov, Evgeny Burnaev, Dmitry Vetrov* User-Controllable Multi-Texture Synthesis with Generative Adversarial Networks. VISAPP 2020, CORE B. Индексировано SCOPUS.

Доклады на конференциях и семинарах

1. Доклад «Latent Video Transformer» на конференции *VISAPP*, онлайн, 2021;
2. Доклад «Steganographic generative adversarial networks» на конференции *ICMV*, Амстердам, 2019;
3. Доклад «Reconstruction of 3D Porous Media from 2D Slices» на конференции *Multiscale methods and high performance scientific computing*, Москва, 2018;
4. Доклад «Inductive Venn-Abers Predictive Distribution» на конференции *COPA 2018*, Маастрихт, 2018;
5. Доклад «Inductive Conformal Martingales for Change-Point Detection» на конференции *COPA 2017*, Стокгольм, 2017.
6. Постер на конференции Skoltech & MIT Conference: Shaping the Future: Big Data, Biomedicine and Frontier Technologies, Москва, 2017

Патенты

1. *Denis Volkhonskiy, Oleg Sudakov, Dmitry Koroteev, Ekaterina Muravleva, Evgeny Burnaev, Leyla Ismailova, Denis Orlov* System for recovery of rock sample three-dimensional structure. RU2718409C1

Автор также внес свой вклад в следующие публикации

1. *Denis Volkhonskiy, Ivan Nazarov, and Evgeny Burnaev* Steganographic generative adversarial networks. ICMV 2019, CORE C. Indexed by SCOPUS.

2. *Denis Orlov, Mohammad Ebadi, Ekaterina Muravleva, Denis Volkhonskiy, Andrei Erofeev, Evgeny Savenkov, Vladislav Balashov, Boris Belozerov, Vladislav Krutko, Ivan Yakimchuk, Nikolay Evseev, Dmitry Koroteev* Different methods of permeability calculation in digital twins of tight sandstones. Journal of Natural Gas Science and Engineering, Q1. Indexed by SCOPUS.

3. *Denis Volkhonskiy, Evgeny Burnaev, Ilia Nouretdinov, Alexander Gammerman, and Vladimir Vovk* Inductive conformal martingales for change-point detection. In Conformal and Probabilistic Prediction and Applications. PMLR 2017. Indexed by SCOPUS.

4. *Ilia Nouretdinov, Denis Volkhonskiy, Pitt Lim, Paolo Toccaceli, and Alexander Gammerman* Inductive venn-abers predictive distribution. PMLR 2018. Indexed by SCOPUS.

4. Содержание работы

Тема диссертации раскрывается в следующих главах, в каждой главе кратко излагается соответствующая статья.

В 4.1 мы предложили новый метод генерации трехмерных воксельных данных из двумерных срезов, который позволяет сохранить физические свойства. В 4.2 мы предложили новый метод генерации трехмерных воксельных данных высокого разрешения, который позволяет сохранить физические свойства. В 4.3 мы предложили новый метод генерации видео, который использует гораздо меньше вычислительных ресурсов, чем современные методы, при схожем качестве. В 4.4 мы предложили новый метод очистки изображений, который позволяет сохранить семантически важную информацию.

4.1. Генеративные состязательные сети для реконструкции трехмерных пористых сред из двумерных срезов

Во многих отраслях наук о земле возникает проблема изучения горных пород на микроуровне. Однако, как правило, для изучения требуются значительные размеры репрезентативных выборок данных, что не всегда достижимо. Получение новых образцов горных пород является сложной задачей: требуется дорогостоящая и времязатратная разработка месторождений горных пород с последующим процессом оцифровки. Таким образом, проблема генерации образцов со сходными свойствами становится актуальной. Синтетические образцы могут быть использованы вместе с реальными образцами для цифрового исследования горных пород.

Одним из наиболее перспективных методов генерации данных являются генеративные состязательные сети (GANs) [13]. GANs выучивают сложные распределения вероятностей непосредственно из выборок. В первой статье об использовании GANs в контексте 3D-генерации пористых сред [23] рассматривалась задача генерации синтетических изображений. Но никакая дополнительная информация (например, двумерные срезы) не использовалась на этапе генерации.

В нашей работе мы предлагаем новую архитектуру глубокой нейронной сети на основе GANs, которая может эффективно генерировать 3D-структуры, учитывая срезы исходного изображения в качестве входных данных. Мы добились этого, внедрив модуль автокодировщика в архитектуру глубокой нейронной сети.

Описание модели

Цель нашего метода — сгенерировать трёхмерную пористую среду $\hat{x}$ размером $h \times w \times d$ из двумерного входного среза s размером $h \times w$. Мы добавляем требование, чтобы центральный срез в $\hat{x}$ был близок к s с точки зрения евклидова расстояния.

Наша модель состоит из следующих нейронных сетей:

- *Кодировщик* $E_\tau(s)$ с параметрами τ. Он переводит входной двумерный срез s в векторное представление h;
- *Генератор* $G_\theta(h, z)$ с параметрами θ. Он переводит входной случайный вектор z и закодированный срез h в трёхмерное изображение x;
- *Дискриминатор* $D_\phi(x)$ с параметрами ϕ. Он предсказывает класс входного трёхмерного изображения x: реальный или сгенерированный. Это стандартный дискриминатор архитектуры GANs.

Чтобы получить центральный срез из трёхмерного изображения, мы вводим маску **M**. Это функция, которая принимает трёхмерное изображение в качестве входных данных и возвращает его центральный срез. Все три нейронные сети обучаются с использованием комбинации функции потерь в евклидовом пространстве (1) и состязательной функции потерь (2). Наша модель представлена на рис. 1.

$$L(s) = \| \, s - \mathbf{M} \odot G_\theta(E_\tau(s), z) \, \|_2^2 \to \min_{\tau,\theta} \tag{1}$$

$$L(D_\phi, G_\theta) = \mathbb{E}_{x \sim p_{data}(x)}[\log D_\phi(x)] + \mathbb{E}_{z \sim p_{noise}(z)}[\log(1 - D_\phi(G_\theta(E_\tau(s), z)))] \to \min_\theta \max_\phi . \tag{2}$$

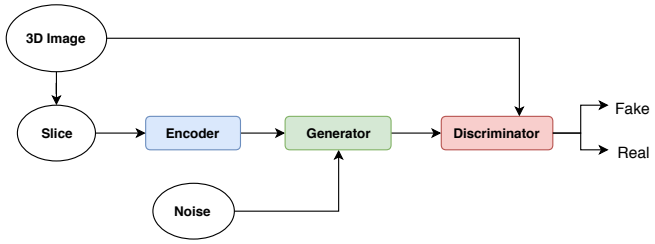
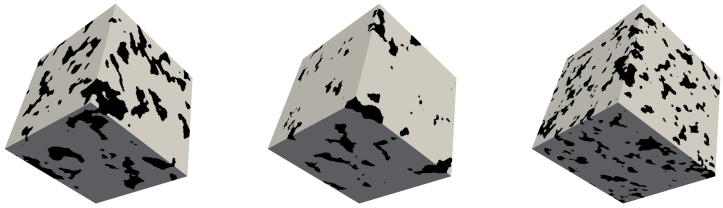


Рис. 1: Архитектура предлагаемого метода. В качестве входных данных выступает двумерный срез пористой структуры и вектор шума. В качестве выхода генератора получается трёхмерная пористая структура. Дискриминатор отвечает за состязательную функцию потерь.

Эмпирические результаты

Чтобы оценить наш метод, мы сравниваем наши сгенерированные трёхмерные образцы (рис. 2) с реальными трёхмерными образцами из набора данных. Для этой цели в нашей статье мы показываем, что распределение пористости, проницаемости, функционалов Минковского [20] и двухточечных корреляционных функций [3] реальных образцов близко к распределению этих показателей на сгенерированных образцах.

Мы также сравниваем наш метод с бейзлайном [23]. Мы используем дивергенцию Кулбека-Лейблера (KL) между реальным и бейзлайновым распределениями $KL(p_{real}, p_{baseline})$, а также



(a) (b) (c)

Рис. 2: Сгенерированные трёхмерные образцы трёх типов: Берея (a), Кеттон (b), Южно-русский песчаник (c)

Таблица 1: Сравнение дивергенции Кулбека-Лейблера между распределением пористости и проницаемости реальных, бейзлайна [23] и наших образцов.

Название образца	Характеристика	$KL(p_{real}, p_{baseline}) \downarrow$	$KL(p_{real}, p_{ours}) \downarrow$
Берея	пористость	0.4974	**0.2667**
Кеттон	пористость	0.5251	**0.1425**
Берея	проницаемость	1.0035	**0.2487**
Кеттон	проницаемость	0.2301	**0.0646**

между реальным и нашим распределениями $KL(p_{real}, p_{ours})$. Результаты сравнения дивергенции Кулбека-Лейблера представлены в таблице 1. Распределение как пористости, так и проницаемости наших образцов ближе к реальному распределению пористости, чем исходное.

Заключение

Мы предложили новый метод получения трехмерных пористых сред из двумерных центральных срезов. Мы показали, что наш метод превосходит бейзлайн по качеству и генерирует трёхмерные воксельные породы, сохраняющие свойства реальных пород. Потенциально такая модель может быть применена к сейсмическим и геологическим данным [24; 6]. Другим возможным применением является генерация текстур.

4.2. Управляемый пользователем синтез текстур с помощью генеративных состязательных нейронных сетей

Чтобы получить надежные результаты при анализе свойств сгенерированных последовательностей изображений, необходимо иметь возможность генерировать их с высоким разрешением. Однако существующие модели, при использовании для объектов с высоким разрешением, сталкиваются с двумя проблемами [23; 30]. Во-первых, они не могут поместиться в вычислительную память графических процессоров. Во-вторых, для обучения обычно недостаточно существующих данных с высоким разрешением. Таким образом, разработка генератив-

ных моделей для последовательностей изображений с высоким разрешением является важной проблемой.

В нашей работе мы разработали новую модель генерации последовательности изображений. Полученная нами модель может генерировать трёхмерные изображения в высоким разрешении, обучаясь только на изображениях с низким разрешением. Для достижения такого эффекта мы предложили новую архитектуру обусловленной генерации. Она позволила нам использовать обучающую выборку с последовательностями изображений разных классов для обучения одной модели, что привело к улучшению качества генерации.

4.2.1. Описание модели

Наша задача — создать метод для синтеза объектов разных классов. Он должен обусловлено генерировать последовательности изображений, обеспечивать полный охват набора данных и быть масштабируемым по отношению к размеру обучающей выборки. Мы используем сеть кодировщик $E_\varphi(x)$, которая отображает объекты в скрытое пространство с низкой размерностью. Генератор $G_\theta(z)$ используется для генерации выборок из скрытого пространства.

Мы используем три типа состязательной функции потерь:

- *Функция потерь генератора* (3) $\mathcal{L}_x$ для соответствия распределения сгенерированных образцов $G_\theta(z)$ и реконструкций $G_\theta(z_\varphi)$ распределению реальных объектов $p^*(x)$:

$$\mathcal{L}_x(\theta) = -\frac{1}{st}\sum_{i,j}^{s,t}\Big[\mathbb{E}_{p(z)}\log D_\psi^{ij}(G_\theta(z)) + \mathbb{E}_{q_\varphi(z)}\log D_\psi^{ij}(G_\theta(z_\varphi))\Big] \to \min_\theta. \tag{3}$$

- *Функция потерь для пар* (4) $\mathcal{L}_{xx}$ для соответствия распределения пар (x, x') распределению пар $(x, G_\theta(z_\varphi(x)))$, где x и x' являются объектами одного класса. Данная функция потерь позволяет сделать так, чтобы $G_\theta(z_\varphi(x))$ имел тот же класс, что и x. Детальная структура дискриминатора для соответствия пар представлена на рисунке 4:

$$\mathcal{L}_{xx}(\theta, \varphi) = -\frac{1}{pq}\sum_{i,j}^{p,q}\mathbb{E}_{p_{\theta,\varphi}(x,y)}\log D_\tau^{ij}(x, y) \to \min_{\theta,\varphi}. \tag{4}$$

- *Функция потерь кодировщика* (5) $\mathcal{L}_z$ для близости распределения $q_\varphi(z)$ с распределением $p(z)$.

$$\mathcal{L}_z(\varphi) = -\mathbb{E}_{q_\varphi(z)}\log D_\zeta(z) \to \min_\varphi. \tag{5}$$

Как для генератора G_θ, так и для кодировщика E_φ мы оптимизируем следующие функции потерь:

- функция потерь генератора G_θ

$$\mathcal{L}(\theta) = \alpha_1\mathcal{L}_x(\theta) + \alpha_2\mathcal{L}_{xx}(\theta, \varphi) \to \min_\theta; \tag{6}$$

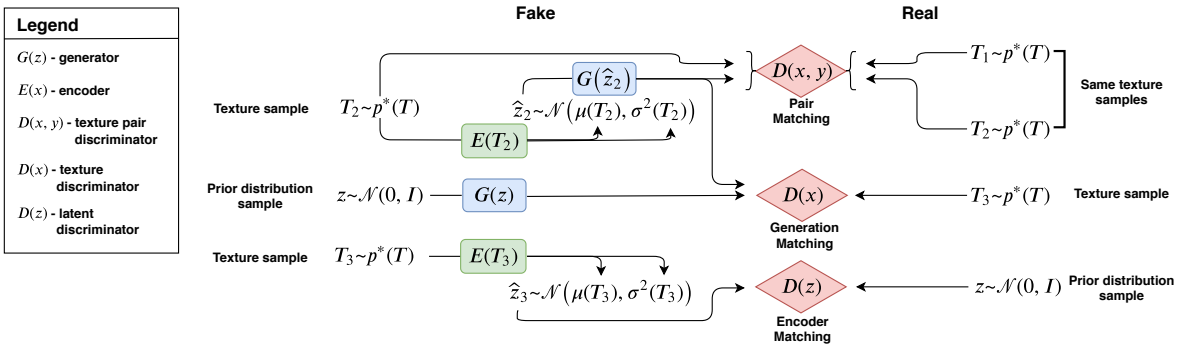


Рис. 3: Способ обучения предложенного метода генерации последовательности изображений.

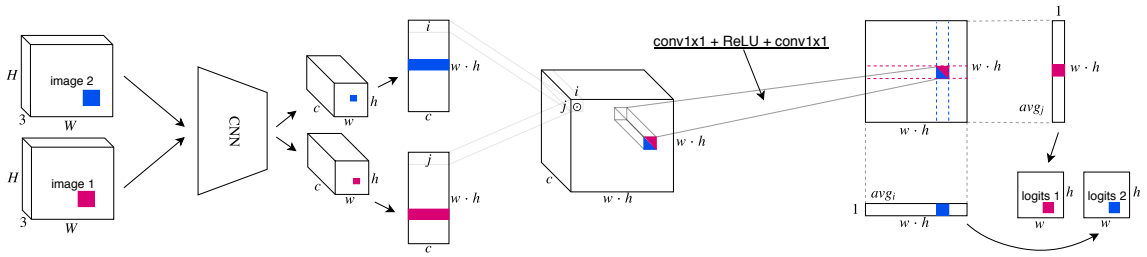


Рис. 4: Архитектура дискриминатора для соответствия пар $D_\tau(x, y)$.

- функция потерь кодировщика E_φ

$$\mathcal{L}(\varphi) = \beta_1 \mathcal{L}_z(\varphi) + \beta_2 \mathcal{L}_{xx}(\theta, \varphi) \to \min_\varphi . \tag{7}$$

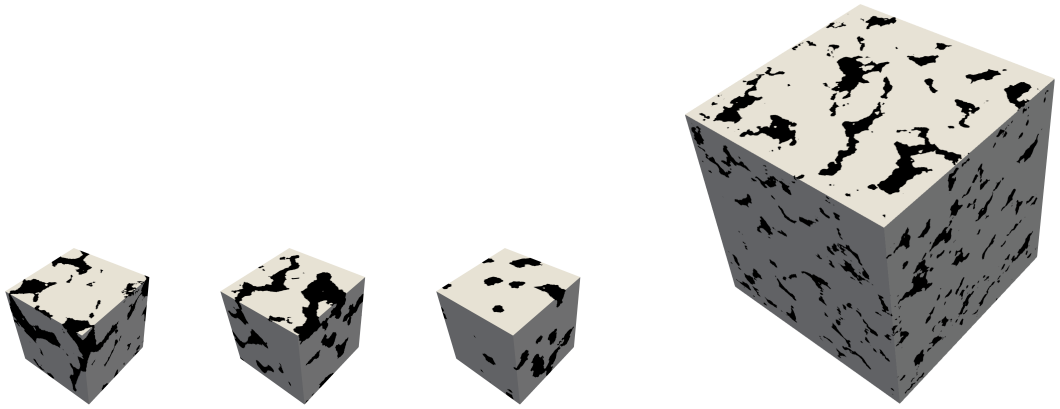
Полный метод обучения нашей модели представлен на рисунке 3.

4.2.2. Эмпирические результаты

В данном разделе мы демонстрируем применимость нашей модели к цифровой физике горных пород. Мы обучили нашу модель на трёхмерных структурах пористых сред[1] (см. рис. 5a) пяти различных типов: Кеттон, Берея, Доддингтон, Эстайладес и Бентхаймер. Каждый тип породы имеет начальный размер 1000^3 бинарных вокселей. В качестве бейзлайна мы использовали модель [23], которая представляет собой глубокие свёрточные генеративные состязательные нейронные сети с трёхмерными свёрточными слоями.

Для сравнения нашей модели с реальными образцами и образцами бейзлайна мы используем статистику проницаемости и два функционала Минковского [20]. Проницаемость — это мера способности пористого материала пропускать жидкости. Функционалы Минковского описывают морфологию и топологию трехмерных бинарных структур. В наших экспериментах мы использовали два функционала: площадь поверхности и эйлерову характеристику. Если

[1]Все образцы были взяты из Базы данных Imperial College



(a) (b) (c) (d)

Рис. 5: Образец Берея. Реальный (a, размер 150^3), сгенерированный нашей моделью (b, размер 150^3), сгенерированный бейзлайном (c, размер 150^3), сгенерированный нашей моделью (d, размер 428^3)

рассмотренные показатели на синтетических образцах близки к таковым на реальных образцах, это гарантирует, что синтетические образцы подходят для приложений цифровой физики горных пород.

Мы поставили эксперименты следующим образом. Мы обучили нашу модель на случайных вырезах размером 160^3 из больших образцов на всех типах пористых структур. Мы также обучили пять базовых моделей для каждого типа отдельно. Затем мы сгенерировали 500 синтетических образцов размером 160^3 каждого типа, используя нашу модель и модель бейзлайна. Мы также вырезали 500 случайных образцов размером 160^3 из реальных данных. В результате для каждого типа структуры мы получили три набора объектов: реальные, синтетические и базовые.

Визуальный результат синтеза представлен на рис. 5 для песчаника Берея. На рисунке представлены три образца: реальный (т.е. вырезанный из исходной большой выборки), сгенерированный нашей моделью и сгенерированный бейзлайном [23]. Поскольку наша модель полностью сверточная, мы можем увеличить размер сгенерированного объекта, расширив пространственные размеры скрытого вектора z при генерации без повторного обучения нейронной сети. Это достижимо, поскольку мы знаем распределение z и можем получить его из генератора случайных чисел. Мы демонстрируем синтезированную 3D пористую среду размером 428^3 на рисунке 5d. Затем мы выполнили следующие шаги:

1. Для каждого реального, синтетического и бейзлайнового объектов мы рассчитали три статистики: проницаемость, площадь поверхности и характеристику Эйлера.

2. Чтобы измерить расстояние между распределениями статистик для реальных, синтетических и бейзлайновых образцов, мы аппроксимировали эти распределения дискретными с помощью гистограмм с 50 интервалами.

3. В завершении каждой статистики мы посчитал дивергенцию Кулбека-Лейблера между распределениями статистик а) реальных и наших сгенерированных образцов; б) реальных и образцов, сгенерированных бейзлайном.

Сравнение дивергенции Кулбека-Лейблера для проницаемости представлено в таблице 2. Как видно из таблицы, наша модель превосходит бейзлайн по качеству для большинства типов пористых структур.

Таблица 2: Дивергенция Кулбека-Лейблера между реальным, нашим и бейзлайн распределениями проницаемости для размера 160^3. Стандартное отклонение было вычислено с использованием метода бутстреп с 1000 повторными выборками.

Образец	$KL(p_{real}, p_{ours})$	$KL(p_{real}, p_{baseline})$
Кеттон	5.06 ± 0.35	4.68 ± 0.56
Берея	0.49 ± 0.07	0.50 ± 0.12
Доддингтон	$\mathbf{0.42 \pm 0.10}$	3.41 ± 1.68
Эстайладес	$\mathbf{0.80 \pm 0.24}$	3.41 ± 0.46
Бентхаймер	$\mathbf{0.47 \pm 0.08}$	1.38 ± 0.49

Заключение

В данной главе мы предложили новую модель для синтеза трёхмерных воксельных данных. Мы показали его применимость к цифровой физике горных пород. Наша модель позволяет генерировать выборки с высоким разрешением при обучении только на выборках с низким разрешением. В большинстве случаев она превосходит бейзлайн, что доказывает полезность модели при решении реальных задач.

4.3. Скрытый Видео Трансформер

Предсказание и генерация видео являются важной проблемой для многих приложений: автономное вождение, обнаружение аномалий, генерация таймлапсов [25], анимация ландшафта [11] и т.д. Задача состоит в том, чтобы сгенерировать наиболее вероятные последующие кадры, учитывая несколько первых кадров.

Последние достижения в области генеративного обучения позволяют создавать реалистичные объекты с высоким качеством: изображения, текст и речь. Однако генерация видео попрежнему остается сложной задачей. Даже для коротких видеороликов (16 кадров) низкого разрешения нейронным сетям для параллельного обучения требуется до 512 тензорных процессоров (ТПУ) [22] для параллельного обучения. Несмотря на это, качество сгенерированного видео остается низким.

В этой работе мы предлагаем новый метод под названием Скрытый Видео Трансформер. Мы объединяем идею обучения представлений и рекуррентной генерации видео. Вместо то-

го, чтобы работать в пространстве пикселей, мы проводим процесс генерации в скрытом пространстве. Наша модель имеет тенденцию значительно снижать требования к вычислениям без существенного ухудшения качества.

Ключевым новшеством в нашей модели является использование дискретного скрытого пространства [29]. Это позволяет нам представлять каждый кадр в виде набора индексов. Благодаря дискретному представлению, мы можем использовать авторегрессионные генеративные модели и другие подходы из обработки естественного языка.

Метод

Рассмотрим видео X как последовательность T кадров $\{x_t\}_{t=1}^{T}$. Каждый кадр $x_t \in \mathbb{R}^{H\times W\times 3}$ имеет высоту H, ширину W и 3 канала RGB. Учитывая первые кадры T_0, цель состоит в том, чтобы сгенерировать оставшиеся кадры $T - T_0$.

Мы используем автокодировщик кадров для обучения компактного скрытого представления, чтобы мы могли перенести задачу моделирования видео из пиксельного пространства в дискретное скрытое пространство. Затем применяется рекуррентная модель используется для генерации новых кадров. Мы обучаем автокодировщик кадров для передачи отдельных изображений (кадров) в скрытое пространство. В качестве автокодировщика был выбран VQ-VAE [29] - вариационный автокодировщик с дискретным скрытым пространством.

Автокодировщик (см. Рис. 6) учится кодировать входное изображение $x \in \mathbb{R}^{H\times W\times 3}$, используя кодовую книгу $e \in \mathbb{R}^{K\times D}$, где K обозначает размер кодовой книги (т.е. скрытое пространство является категориальным с K категориями), а D представляет размер представления в кодовой книге.

Наш автокодировщик состоит из:

- *кодировщика*, который кодирует изображение в более компактное представление $E(x) = z_e(x) \in \mathbb{R}^{h\times w\times D}$;
- *внутренней части*, которая дискретизирует каждый пиксель путем сопоставления его с ближайшим представлением e_i из кодовой книги и выдает $z(x) \in [K]^{h\times w\times 1}$;
- *декодера* D, который принимает в качестве входных данных дискретные скрытые коды $z(x)$, сопоставляет индексы с соответствующими представлениями и декодирует результат отображения $z_q(x) \in \mathbb{R}^{h\times w\times D}$ обратно во входное пиксельное пространство.

Автокодировщик обучается со следующей оптимизационной задачей:

$$L = \|x - D(z_q(x))\|^2 + \|z_e(x) - \text{sg}[e]\|^2, \tag{8}$$

где sg$[\cdot]$ — это оператор остановки градиента (stop gradient), который возвращает свой аргумент во время прямого прохода и нулевые градиенты во время обратного прохода. Первое слагаемое — это функция потерь для восстановления, а второе слагаемое — это регуляризация,

позволяющая сделать коды менее волатильными. Мы используем экспоненциальное обновление скользящим средним по значениям из кодовой книги.

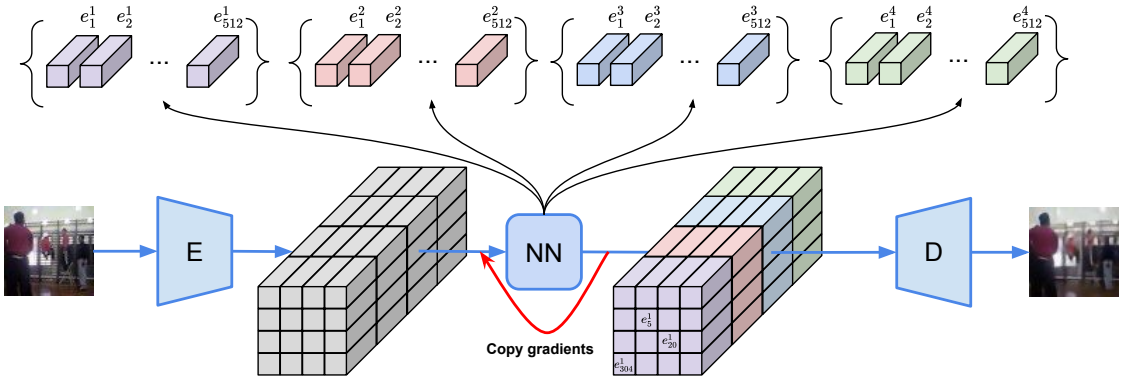


Рис. 6: Архитектура автокодировщика кадров. Входное изображение пропускается через кодировщик и разбивается по каналам на части $n_c = 4$. Затем мы сопоставляем пиксели в каждой части с соответствующими ближайшими вложениями в кодовой книге. Эти ближайшие вложения затем передаются в качестве входных данных в декодер.

Кодировщик кадров преобразует первые кадры T_0 в дискретное представление $Z_0 \in [K]^{T_0 \times h \times w \times n_c}$. Авторегрессионная модель используется для генерации новых кадров $T - T_0$, обусловленных Z_0. В качестве такой модели мы используем Видео Трансформер [31], авторегрессионную модель генерации видео, но применяем ее в скрытом пространстве в отличие от пространства пикселей в оригинальной статье.

Далее следует описание архитектуры видео трансформера. Мы называем скрытое представление видео как *скрытое видео*, а отдельные его элементы — *скрытые кадры* и *пиксели*.

Модель принимает в качестве входных данных тензор $Z \in [K]^{T \times h \times w \times n_c}$ и запускает процесс генерации на первых T_0 заданных скрытых кадрах, т.е. $Z_{:T_0,:,:,:} = Z_0$.

Процесс генерации происходит фрагмент за фрагментом, пиксель за пикселем внутри одного фрагмента, канал за каналом для одного пикселя:

$$p(Z) = \prod_{i=0}^{Thw-1} \prod_{k=0}^{n_c-1} p\left(Z^k_{\pi(i)} | Z_{\pi(<i)}, Z^{<k}_{\pi(i)}\right) \tag{9}$$

Модель состоит из кодировщика и декодера. Чтобы сгенерировать новое значение пикселя внутри среза $Z_{(a,b,c)}$, сначала кодировщик выводит представление уже сгенерированных срезов $Z_{<(a,b,c)}$. Это представление поступает в декодер, который смешивает его с представлением уже сгенерированных пикселей внутри текущего среза $Z_{(a,b,c)}$. Этот порядок авторегрессии сохраняется путем заполнения входного скрытого видео внутри кодировщика и маскировки, используемой в свертках и механизме внимания внутри декодера. После генерации нового значения пикселя мы заменяем соответствующее заполнение сгенерированным выводом и повторяем процесс генерации рекурсивно. Процесс генерации в случае пространственно-временного ($s_t > 0, s_h > 0, s_w > 0$) случая можно увидеть на рис. 7.

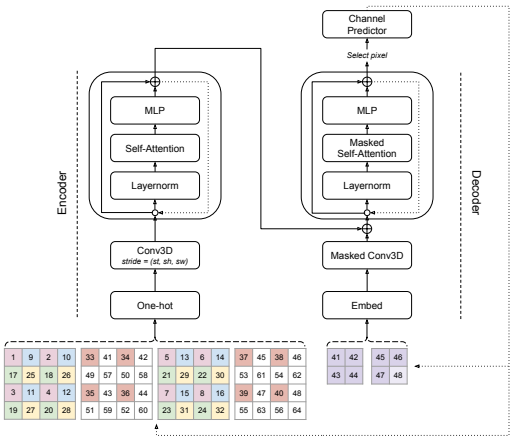


Рис. 7: Видео трансформер, адаптированный к скрытым кодам. Числа представляют порядок генерации. Пиксели окрашены, если они уже сгенерированы. Пиксели белого цвета заполняются нулем. Пиксели с одинаковым цветом принадлежат одному и тому же скрытому кадру. Пример представляет собой генерацию последнего пикселя фрагмента $Z_{(1,0,1)}$ для скрытого видео размером $(t, h, w) = (4, 4, 4)$ и $(s_t, s_h, s_w) = (2, 2, 2)$.

Наконец, когда процесс генерации завершен, декодер скрытых кадров принимает в качестве входных данных $Z \in [K]^{T \times h \times w \times n_c}$ (теперь все значения действительны), сопоставляет его с уже выученными представлениями $Z_q \in \mathbb{R}^{T \times h \times w \times D}$ и декодирует его обратно кадр за кадром в исходное пиксельное пространство $X \in \mathbb{R}^{T \times H \times W \times 3}$.

Эмпирические результаты

Мы моделируем видео длиной $T = 16$ и пространственным размером 64×64 аналогично предыдущим работам в этой области [5; 31]. Мы сравниваем качество генерации видео в наборе данных BAIR Robot Pushing [9] и Kinetics-600 [4].

В качестве основных показателей качества мы используем Fréchet Video Distance (FVD) [14] и количество бит на размерность bits/dim — отрицательная $\log_2$-вероятность, усредненная по всем сгенерированным (скрытым) пикселям и каналам.

Мы приводим результаты FVD и bits/dim для набора данных BAIR Robot Pushing в таблице 3 и для набора данных Kinetics-600 в таблице 4. Мы также предоставляем примеры генерации нашей моделью на рис. 8.

Мы достигаем сопоставимого качества по сравнению с другими методами для набора данных BAIR robot pushing. Наши результаты незначительно уступают другим работам на наборе данных Kinetics-600. Мы приходим к выводу, что это вызвано накоплением ошибок внутри модели трансформатора. Мы связываем это с высокой сложностью и разнообразием набора данных Kinetics-600.

Таблица 3: Сравнение различных методов прогнозирования видео на наборе данных BAIR Robot Pushing.

Метод	**bits/dim**(↓)	**FVD**(↓)
Baseline	-	320.90
VideoFlow [18]	1.87	-
SVP-FP [7]	-	315.5
CDNA [12]	-	296.5
LVT (ours, $n_c = 1$)	1.25	275.71 ± 5.41
SV2P [8]	-	262.5
LVT (ours, $n_c = 4$)	1.53	125.8 ± 2.9
SAVP [19]	-	116.4
DVD-GAN-FP [5]	-	109.8
TriVD-GAN-FP [22]	-	103.3
Axial Transformer [15]	1.29	-
Video Transformer [31]	1.35	94 ± 2

Таблица 4: Сравнение различных методов прогнозирования видео на наборе данных Kinetics-600.

Метод	**Bits/dim**(↓)	**FVD**(↓)
Baseline	-	271.00
LVT (наш метод)	2.14	224.73
Video Transformer [31]	1.19	170 ± 5
DVD-GAN-FP [5]	-	69.15 ± 1.16
TriVD-GAN-FP [22]	-	25.74 ± 0.66

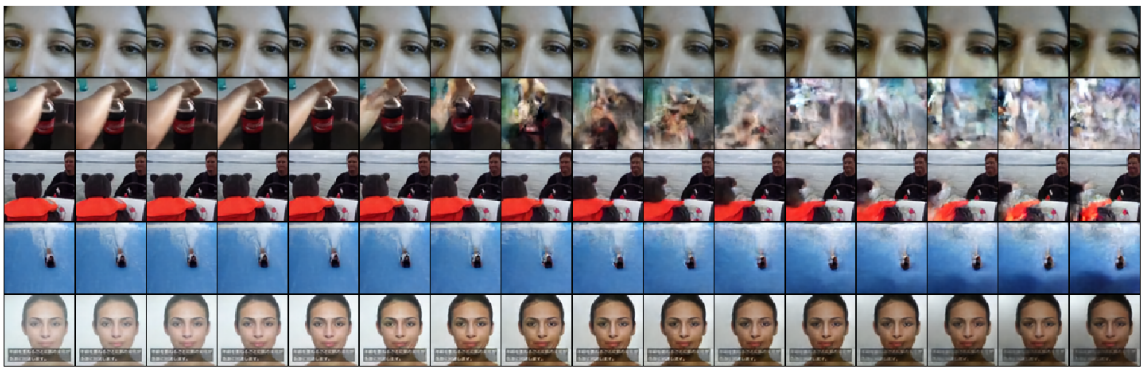


Рис. 8: Образцы из набора данных Kinetics-600. Каждая строка представляет собой одно видео, первые пять кадров которого являются реальными, а остальные сгенерированы.

Заключение

В данной главе мы рассмотрели проблему генерации видео. Цель состояла в том, чтобы, имея несколько первых кадров, предсказать продолжение видео. Современные методы генерации видео требуют до 512 ТПУ для параллельного обучения. Мы были сосредоточены на снижении вычислительных требований для обучения модели. Мы показали, что можно добиться сопоставимых результатов в прогнозировании видео, обучив модель с использованием обычной исследовательской установки - 8 графических процессоров V100. Мы продемонстрировали сопоставимые результаты на наборе данных BAIR Robot Pushing. В то же время в некоторых случаях мы наблюдаем визуальные артефакты в наборе данных Kinetics-600.

4.4. Глубокая векторизация технических изображений

Векторные изображения имеют ряд преимуществ перед растровыми изображениями. Они включают в себя масштабирование без потерь качества и малый размер представления. Кроме того, векторные изображения позволяют независимо модифицировать свои примитивы. Однако многие технические изображения доступны только в растровом формате. Таким образом, важно иметь возможность из растрового формата получать векторный.

В [10] был предложен новый метод векторизации. Он состоит из следующих шагов. Сначала исходное растровое изображение очищается от шума и артефактов. Это называется *очисткой*. Затем семантически важные части аппроксимируются примитивами. Наконец, количество примитивов, по возможности, сводится к минимуму.

В этом тезисе мы фокусируемся на *очистке* изображений. Целью данной работы является создание метода, который позволит преобразовать растровое изображение с артефактами и недостающими линиями в чистое растровое изображение с заполненными пропусками. Целью очистки является решение следующих проблем:

- очистить входное растровое изображение от артефактов;
- сделать фон белым, если его цвет отличается (например, серый или коричневый фон);
- обнаружить и закрасить пропущенные части в линиях, если они существуют.

4.4.1. Метод

Целью этапа очистки является преобразование необработанных входных данных в растровое изображение с четкой структурой линий путем устранения шума, заполнения недостающих частей линий и изменения всех фоновых / неокрашенных областей на белый цвет. Эту задачу можно рассматривать как семантическую сегментацию изображения в том смысле, что пикселям присваивается класс фона или переднего плана.

Наша сеть очистки имеет структуру U-NET [26; 21; 28] и состоит из двух частей: понижения и повышения размерности. Часть понижения размерности представляет собой последовательность свёрточных слоев с нелинейностями между ними. В части повышения размерности

	IoU,%	PSNR
MS [28]	49	15.7
Наша модель	**92**	**25.5**

Таблица 5: Количественная оценка метода очистки

используются транспонированные свертки, что позволяет увеличить размер входного изображения. После каждого свёрточного и транспонированного свёрточного слоя мы используем нормализацию батча [16] и ReLU слои. Отличительной особенностью нашей архитектуры являются пропускные соединения, которые соединяют слои из части понижения размерности со слоями части повышения размерности. Полная структура модели представлена на рис. 9.

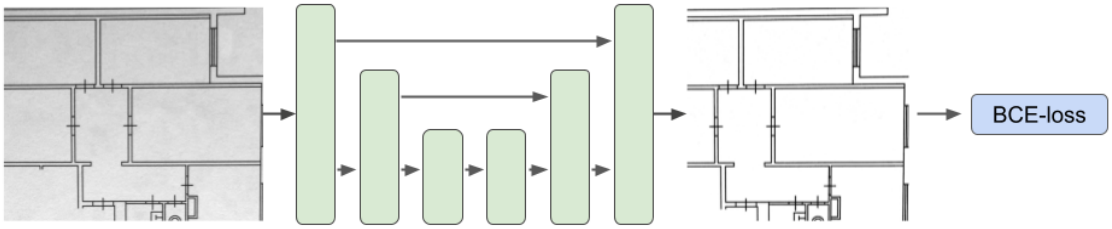


Рис. 9: Метод очистки. Входное растровое изображение обрабатывается с помощью нашей модели. На вход модель принимает изображение. В качестве функции потерь используется бинарная кросс-энтропия.

Чтобы обучить такую нейронную сеть, мы минимизируем бинарную кросс-энтропию, используя алгоритм Adam [17]. Для пикселя $x_{i,j}$ в местоположении i, j и истинного значения в том же местоположении $y_{i,j}$ потери для этого конкретного пикселя вычисляются как $l_{i,j} = y_{i,j} \cdot \log x_{i_j} + (1 - y_{i,j}) \cdot \log(1 - x_{i_j})$. Полная функция потерь для изображения с высотой H и шириной W является средним значением потерь всех пикселей:

$$L = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} l_{i,j}.$$

Эмпирические результаты

Мы сравниваем нашу сеть для очистки с общедоступной предварительно обученной моделью Mastering Sketching (MS, [28]). Мы показываем количественные результаты этой оценки в таблице 5 и качественные результаты на рисунке 10.

Наша сеть для очистки сохраняет прямые и повторяющиеся линии, обычно встречающиеся на техническом чертеже. В то же время MS создает волнистые штрихи и имеет тенденцию соединять повторяющиеся прямые линии, тем самым нарушая структуру чертежа.

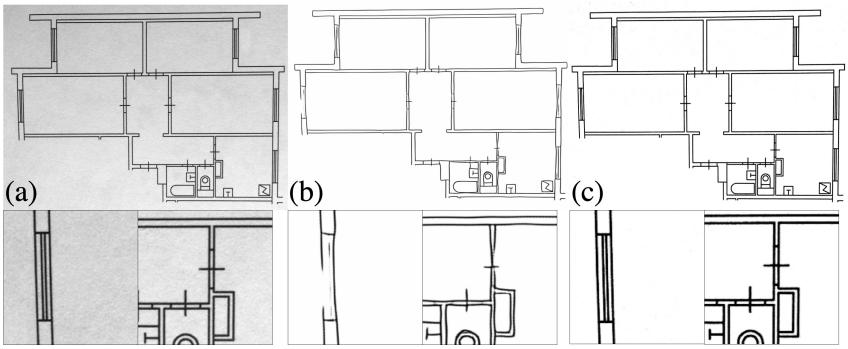


Рис. 10: Примеры очистки: (a) входное изображение, (b) результат модели MS [28], (c) результат нашей модели.

Заключение

В данной главе мы предлагаем метод очистки растровых изображений от артефактов и удаления пропущенных частей в примитивах. Этот метод является частью метода векторизации в [10]. Метод очистки позволяет дополнительно повысить качество векторизации и уменьшить количество конечных векторизованных примитивов.

5. Заключение

В заключительном разделе мы приводим основные результаты данной работы.

1. Мы разработали новый метод генерации трехмерных воксельных данных из двумерного среза. Мы применили этот метод для получения пористых сред. С помощью численных экспериментов с пористыми средами мы продемонстрировали применимость предложенной модели. Мы показали, что наш метод позволяет сохранить физические характеристики, такие как пористость, проницаемость и двухточечную корреляционную функцию, лучше, чем существующие методы. Наша модель генерирует трёхмерные воксельные данные хорошего визуального качества.

2. Мы разработали метод генерации трехмерных воксельных данных, позволяющий генерировать выборки высокого разрешения, обучаясь только на изображениях с низким разрешением. С помощью численных экспериментов на пористых средах мы продемонстрировали его способность сохранять физические характеристики лучше, чем существующие методы.

3. Мы разработали новый метод генерации видео, который позволяет снизить вычислительные требования к обучению современных моделей с 512 тензорных процессоров до 8 графических процессоров при схожем качестве генерации.

4. Мы разработали генеративную модель для очистки изображений. Данная модель позволяет удалять фон и артефакты с изображения, сохраняя при этом семантически важную

информацию. Наша модель превосходит существующие методы и закрашивает недостающие участки в изображении. Данный метод был разработан как часть метода векторизации растровых изображений.

References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[2] Jihye Back. Fine-tuning stylegan2 for cartoon face generation. *arXiv preprint arXiv:2106.12445*, 2021.

[3] Y. Bazaikin, V. Baikov, I. Taimanov, and A. Yakovlev. Numerical analysis of topological characteristics of three-dimensional geological models of oil and gas fields. *Matematicheskoe Modelirovanie*, 25(10):19--31, 2013.

[4] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[5] A Clark, J Donahue, and K Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

[6] Guillaume Coiffier, Philippe Renard, and Sylvain Lefebvre. 3d geological image synthesis from 2d examples using generative adversarial networks. *Frontiers in Water*, 2:30, 2020.

[7] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *arXiv preprint arXiv:1802.07687*, 2018.

[8] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *arXiv preprint arXiv:1802.07687*, 2018.

[9] Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *arXiv preprint arXiv:1710.05268*, 2017.

[10] Vage Egiazarian*, Oleg Voynov*, Alexey Artemov, Denis Volkhonskiy, Aleksandr Safin, Maria Taktasheva, Denis Zorin, and Evgeny Burnaev. Deep vectorization of technical drawings. In *ECCV 2020: European Conference on Computer Vision*, pages 582--598. Springer, 2020.

[11] Yuki Endo, Yoshihiro Kanamori, and Shigeru Kuriyama. Animating landscape: self-supervised learning of decoupled motion and appearance for single-image video synthesis. *arXiv preprint arXiv:1910.07192*, 2019.

[12] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pages 64--72, 2016.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672--2680, 2014.

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626--6637, 2017.

[15] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[17] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A flow-based generative model for video. *arXiv preprint arXiv:1903.01434*, 2(5), 2019.

[19] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[20] D. Legland, K. Kiêu, and M.-F. Devaux. Computation of minkowski measures on 2d and 3d binary images. *Image Analysis & Stereology*, 26(2):83--92, 2011.

[21] Chengze Li, Xueting Liu, and Tien-Tsin Wong. Deep extraction of manga structural lines. *ACM Transactions on Graphics (TOG)*, 36(4):117, 2017.

[22] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *arXiv preprint arXiv:2003.04035*, 2020.

[23] L. Mosser, O. Dubrule, and M. Blunt. Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Physical Review E*, 96(4):043309, 2017.

[24] Lukas Mosser, Olivier Dubrule, and Martin J Blunt. Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. *Mathematical Geosciences*, 52(1):53--79, 2020.

[25] Seonghyeon Nam, Chongyang Ma, Menglei Chai, William Brendel, Ning Xu, and Seon Joo Kim. End-to-end time-lapse video synthesis from a single outdoor image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1409--1418, 2019.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234--241. Springer, 2015.

[27] Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific reports*, 9(1):1--9, 2019.

[28] Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. Mastering sketching: adversarial augmentation for structured prediction. *ACM Transactions on Graphics (TOG)*, 37(1):11, 2018.

[29] Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306--6315, 2017.

[30] Denis Volkhonskiy, Ekaterina Muravleva, Oleg Sudakov, Denis Orlov, Evgeny Burnaev, Dmitry Koroteev, Boris Belozerov, and Vladislav Krutko. Generative adversarial networks for reconstruction of three-dimensional porous media from two-dimensional slices. *Phys. Rev. E*, 105:025304, Feb 2022.

[31] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *arXiv preprint arXiv:1906.02634*, 2019.